SCUOLA DI DOTTORATO
Università degli Studi di Milano-Bicocca
PhD program in Computer Science

PhD program in Computer Science

Cycle XXXVIII

Generative AI for Scalable Multivariate Traffic Scenario Modelling in Urban Mobility Systems

Michele Carbonera
836648

Supervisor: Prof. Enza Messina

Co-Tutor: Prof. Michele Ciavotta

Tutor: Prof. Daniela Micucci

Coordinator: Prof. Gianluca Della Vedova

ACADEMIC YEAR 2025/2026

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA

ABSTRACT

The growing complexity of urban mobility systems requires tools capable of representing uncertainty directly from data. In this context, the generation of realistic traffic scenarios has become a key component for a wide range of applications, including simulation, forecasting, and optimization under uncertainty.

However, traditional statistical approaches for scenario generation struggle to scale to real, large-scale road networks: capturing spatial dependencies and high dimensionality leads to significant computational growth that limits their applicability.

This thesis addresses these limitations by proposing a set of models based on Generative AI architectures, such as Variational Autoencoders and Generative Adversarial Networks, each characterized by its architectural design and its ability to capture complex structures in the data. The proposed variants include extensions with Graph Convolutional Networks to model spatial correlations between adjacent road segments, as well as Transformer layers to more effectively represent long-range dependencies.

To assess the realism of the generated scenarios, the thesis introduces an evaluation framework that combines complementary statistical metrics, low-dimensional visualizations, and computational efficiency measures, providing a multi-level assessment of generative performance.

Experiments conducted on real traffic data highlight that the proposed models reproduce the main statistical properties observed in the data while scaling reliably to large-scale networks, a setting in which traditional statistical approaches are no longer applicable.

Overall, these results highlight the potential of VAE-based generative models as a scalable solution for producing realistic traffic scenarios that can support simulation and optimization processes in transportation systems.

SOMMARIO

La crescente complessità dei sistemi di mobilità urbana richiede strumenti in grado di rappresentare l'incertezza basandosi sui dati raccolti. In questo contesto, la generazione di scenari di traffico realistici è diventata un elemento chiave per numerose applicazioni, tra cui simulazione, previsione e ottimizzazione in condizioni di incertezza. Tuttavia, gli approcci statistici tradizionali per la generazione di scenari faticano a scalare su reti stradali reali e di grandi dimensioni: la necessità di catturare dipendenze spaziali e alta dimensionalità comporta una crescita computazionale significativa che ne limita l'applicabilità.

Questa tesi affronta tali limitazioni proponendo una serie di modelli basati su tecniche di Generative AI, come Variational Autoencoders e Generative Adversarial Networks, ciascuno caratterizzato dalla propria architettura e dalla capacità di catturare strutture complesse nei dati. Le varianti proposte includono estensioni con Graph Convolutional Networks per modellare in modo esplicito le correlazioni spaziali tra segmenti stradali adiacenti, oltre a livelli basati su Transformer per rappresentare in modo più efficace le dipendenze a lungo raggio.
Per valutare il realismo degli scenari generati, la tesi introduce un framework di valutazione che combina metriche statistiche complementari, visualizzazioni a bassa dimensionalità e integra anche misure di efficienza computazionale, fornendo una valutazione multilivello delle prestazioni generative.
Gli esperimenti condotti su dati di traffico reali mostrano che i modelli proposti riproducono le principali proprietà statistiche osservate nei dati, mantenendo al contempo un'elevata scalabilità su reti di grandi dimensioni, un contesto in cui gli approcci statistici tradizionali diventano computazionalmente inapplicabili.
Nel complesso, i risultati evidenziano il potenziale dei modelli generativi per la generazione di scenari di traffico realistici e ad ampia scala.

RÉSUMÉ

La complexité croissante des systèmes de mobilité urbaine exige des outils capables de représenter l'incertitude à partir des données collectées. Dans ce contexte, la génération de scénarios de trafic réalistes est devenue un élément clé pour de nombreuses applications, notamment la simulation, la prévision et l'optimisation en conditions d'incertitude. Cependant, les approches statistiques traditionnelles pour la génération de scénarios peinent à passer à l'échelle sur des réseaux routiers réels et de grande dimension : la nécessité de capturer des dépendances spatiales et une forte dimensionnalité entraîne une croissance computationnelle significative qui en limite l'applicabilité.
Cette thèse aborde ces limitations en proposant une série de modèles fondés sur des techniques de Generative AI, telles que les Variational Autoencoders et les Generative Adversarial Networks, chacun caractérisé par sa propre architecture et par sa capacité à capturer des structures complexes dans les données. Les variantes proposées incluent des extensions basées sur des Graph Convolutional Networks afin de modéliser explicitement les corrélations spatiales entre segments routiers adjacents, ainsi que des couches de type Transformer permettant de représenter plus efficacement les dépendances à longue portée.
Pour évaluer le réalisme des scénarios générés, la thèse introduit un cadre

d'évaluation combinant des métriques statistiques complémentaires, des visualisations en basse dimension et des mesures d'efficacité computationnelle, offrant ainsi une analyse multiniveau des performances génératives.
Les expériences menées sur des données de trafic réelles montrent que les modèles proposés reproduisent les principales propriétés statistiques observées dans les données, tout en maintenant une excellente capacité de passage à l'échelle sur des réseaux de grande dimension, un contexte dans lequel les approches statistiques traditionnelles deviennent computationnellement inapplicables.
Dans l'ensemble, les résultats mettent en évidence le potentiel des modèles génératifs pour la production de scénarios de trafic réalistes et à grande échelle.

PUBLICATIONS

Publications related to this PhD thesis:

[1] Michele Carbonera, Michele Ciavotta, and Enza Messina. « Driving into Uncertainty: An Adversarial Generative Approach for Multivariate Scenario Generation. » In: *2023 IEEE International Conference on Big Data (BigData)*. IEEE. 2023, pp. 2578–2587.

[2] Michele Carbonera, Michele Ciavotta, and Enza Messina. « Variational Autoencoders and Generative Adversarial Networks for Multivariate Scenario Generation. » In: *Data Science for Transportation* 6.3 (2024), p. 23.

[3] Michele Carbonera, Michele Ciavotta, and Enza Messina. « AI Generative Models for Realistic Urban Mobility Scenario Generation. » In: *Book of Abstracts*. 2025, p. 46.

[4] Michele Carbonera, Michele Ciavotta, and Enza Messina. « Generative AI for traffic scenarios: a GCN-VAE model. » In: *Book of Short Papers 2025 Conference of the 12th Scientific Meeting of the Statistics for the Evaluation and Quality of Services Group of the Italian Statistical Society (SVQS)*. 2025, pp. 1288–1294.

[5] Elisabetta Fersini, Simone Mottadelli, Michele Carbonera, and Enza Messina. « Deep attributed graph embeddings. » In: *International Conference on Modeling Decisions for Artificial Intelligence*. Springer International Publishing Cham. 2022, pp. 181–192.

Additional works submitted to journals:

[1] Michele Carbonera, Michele Ciavotta, and Enza Messina. « Generative AI for realistic urban traffic scenario generation: β-GCN-VAE and β-Transformer-VAE models. » In: *Quality & Quantity* (2026).

ACKNOWLEDGEMENTS

This thesis represents not a final destination, but rather an organic formalization and a starting point for future directions of research; nevertheless, it constitutes a fundamental milestone in my academic path and calls for a sincere acknowledgment of the people who accompanied me throughout my doctoral studies.

First, I would like to express my deepest gratitude to my supervisor, Professor Enza Messina, for her profound expertise and the insightful guidance she provided during every stage of my research. Alongside her, I wish to thank my co-supervisor, Professor Michele Ciavotta, for our constructive exchanges and for his fundamental contribution. My sincere thanks also go to my tutor, Professor Daniela Micucci, whose support was essential over these three years.

In addition to the professors at University of Milano-Bicocca, my gratitude extends to Professor Giuseppe Valenzise of Université Paris Saclay-CentraleSupélec, where I had the opportunity of conducting my period of research abroad. Working in such a stimulating environment was a formative experience that deeply motivated me.

Finally, my most heartfelt thanks go to my girlfriend, Martina, and to my family, who encouraged me in every choice: my mother Daniela, my father Graziano, Francesco, Maria Luisa, and Chiara. They never failed to provide their presence and unwavering support, even from a distance. Special thanks also go to Salem, whose paws kept me company and brought me comfort throughout this journey.

CONTENTS

LIST OF TABLES

INTRODUCTION AND URBAN CONTEXT

Chapter 1 outlines the growing complexity of managing transportation networks in urban areas, where congestion, environmental impacts, and systemic vulnerabilities increasingly challenge traditional planning and operational paradigms. Recognizing urban mobility as a high-dimensional and inherently stochastic system, the chapter motivates the central theme of the thesis: the design of advanced scenario generation methods capable of capturing and reproducing the complex dynamics of traffic road networks, preserving the spatial correlation inherent in real-world traffic data.

1 INTRODUCTION

Global urbanization has proven to be a central challenge of the XXI century. By 2025, more than 55% of the world's population lives in urban areas, and projections indicate a continued rise in the coming decades [181]. The smart city paradigm is emerging as a key approach to address this demographic and infrastructural pressure, integrating digital technologies, smart IOT sensors, big data, and artificial intelligence to optimize resources, enhance quality of life, and promote both environmental and economic sustainability.
In this context, urban mobility stands as a critical aspect: while vehicular transportation is indispensable for fostering accessibility and economic growth, it simultaneously imposes severe negative impacts. Vehicular transport entails considerable drawbacks: chronic traffic gridlock, environmental and noise pollution, road accidents, and elevated energy consumption.
Statistical reports such as the INRIX 2025 Global Traffic Scorecard [145] and recent studies on urban trends [138] highlight a significant resurgence of post-pandemic congestion across the continent. According to the INRIX 2025 Global Traffic Scorecard, in major European metropolitan areas such as London, Paris, and Milan, drivers lose an average of 100 to 150 hours per year due to traffic congestion. These dynamics not only erode productivity, with significant social costs related to health and economic well-being [4], but also fundamentally hinder the decarbonization targets set by the European Green Deal and jeopardize overall urban livability, as traffic levels in many EU cities have recently surpassed 2019 benchmarks [138].

The smart city paradigm addresses these challenges through data-driven approaches, including real-time traffic management systems (Intelligent Transportation Systems – ITS), adaptive traffic signals, Mobility-As-A-Service (MaaS) platforms, Vehicle-To-Everything communication (V2X), and, prospectively, autonomous vehicles. The overarching goal is to shift from a reactive and static management paradigm to a proactive and predictive framework capable of anticipating flows, preventing bottlenecks, and optimizing the allocation of road capacity.
Achieving this shift requires analytical tools capable of explicitly accounting for the inherent uncertainty of traffic dynamics. This uncertainty must be represented through multiple plausible realizations of traffic conditions, which serve as the basis for evaluating

and designing management strategies under variable and potentially adverse circumstances. To this end, scenario generation techniques have been developed to produce realistic and statistically coherent representations of traffic conditions.

This thesis, therefore, focuses on the problem of traffic scenario generation, which constitutes a fundamental building block for uncertainty-aware approaches.
The goal is to generate synthetic scenarios that faithfully reproduce the spatial correlations and variability observed in real-world data, thereby enabling decision-making frameworks to operate on reliable and representative scenario sets.
Effective scenario generation is not limited to replicating frequent or probable conditions; it must also generate low-probability events that can heavily influence the robustness and resilience of solutions.
Overlooking the events represented in the tails of data distributions may lead to overly optimistic policies that fail catastrophically in practice, whereas including them ensures that solutions are resilient across the full spectrum of possibilities.
Moreover, in transportation networks, scenarios must faithfully capture spatial correlations, not only between adjacent roads, where congestions propagate locally, but also between distant links within the broader road network, reflecting systemic inter-dependencies.

State-of-the-art scenario generation methods largely rely on statistical techniques to model the distributions, correlations, and variability observed in real-world data. Although effective in small-scale settings, these approaches struggle to scale to complex large road networks, where graph structure, interactions between adjacent segments, and long-range dependencies demand more flexible, powerful, and scalable models.
The fundamental challenge lies in the 'curse of dimensionality' and the inability of traditional parametric methods to capture the high-dimensional, non-linear dependencies inherent in urban traffic flows.
This thesis addresses these challenges by introducing novel generative models specifically designed for high-fidelity generation of traffic speed scenarios over large-scale, real-world road networks.
The proposed approach is validated on a widely recognized public benchmark, showing significant improvements over classical statistical baselines.

Results demonstrate the framework's effectiveness in terms of statistical fidelity, while maintaining superior scalability to large-scale road-network topologies compared to traditional methods.

1.1 THESIS CONTRIBUTIONS

In this PhD thesis, novel Generative AI models are proposed to synthesize realistic traffic speed scenarios based on real-world road networks. To evaluate and compare the faithfulness of the proposed models, a multi-metric evaluation framework is introduced.

The primary contributions of this work can be summarized as follows:

- **A novel Generative-AI-based framework for modelling spatial correlations in large-scale traffic networks**.
  The thesis proposes new generative models capable of capturing spatial correlations across road segments and long-range dependency patterns, thereby addressing the computational limitations of traditional statistical approaches.

- **An integrated evaluation framework for assessing the statistical fidelity of generated multivariate scenarios**.
  The thesis introduces and applies an evaluation methodology based on advanced statistical metrics to quantify the realism, variability, and spatial consistency of the generated scenarios.

- **Extensive empirical validation on real-world road-network datasets**.
  The proposed methods are evaluated on public benchmarks, highlighting improved scalability compared to classical statistical baselines and demonstrating their ability to reproduce realistic statistical patterns observed in real-world traffic data.

These contributions collectively advance the state of the art in generative modelling for transportation systems, offering improved tools for simulation, forecasting, and policy evaluation for smart-city applications.

1.2 ORGANIZATION OF THE THESIS

This thesis is structured in three main parts to guide the reader from the contextual foundations of the research to the methodological developments and finally to the experimental settings and results.

PART I – THEORETICAL FOUNDATIONS AND STATE OF THE ART

Part I provides the background required to understand the methodological developments presented in the following parts.
Chapter 2 reviews the fundamentals of artificial intelligence, with a focus on artificial neural networks, their architectures, and training paradigms.
Chapter 3 surveys the state of the art in scenario generation, examining both classical statistical approaches and modern Generative AI methods, such as GAN and VAE architectures, introduced in this chapter. Particular attention is devoted to methods applied to traffic data, highlighting their strengths, limitations, and open challenges that motivate the contributions of this thesis.

PART II – PROBLEM FORMULATION AND METHODOLOGICAL DESIGN

Part II formalizes the research challenges and introduces the proposed methodological framework.
Chapter 4 defines the problem of traffic scenario generation, detailing how road networks and traffic speeds are represented, and discussing the approximation of high-dimensional joint distributions required for realistic synthetic data generation.
Chapter 5 presents the methodological contributions of the thesis. It begins with baseline naive VAE architectures and progressively introduces more advanced configurations, including graph-based components, Copula-driven latent spaces, dual-latent vector designs, β-VAE regularization, and Transformer-enhanced modelling.

PART III – EVALUATION FRAMEWORK, EXPERIMENTAL SETTINGS AND RESULTS

Part III evaluates the proposed methods through a multifaceted assessment.
Chapter 6 introduces the evaluation framework, describing the statistical metrics employed and presenting the dataset used as benchmark.
Chapter 7 reports the experimental results, analyzing the statistical fidelity of the generated scenarios and their computational scalability. The chapter discusses the comparative performance of the proposed models and reflects on their practical implications for urban mobility applications.

Chapter 8 concludes the thesis by summarizing the main findings, discussing their significance, and outlining promising directions for

future research.

In appendices, an additional case study (Appendix A) and a set of extended results complementing the analyses of Chapter 7 (Appendix B) are presented.

Part I

THEORETICAL FOUNDATIONS AND STATE OF THE ART

In the following chapters, we explore the theoretical foundations of artificial intelligence, starting from the elementary components, proceeding to generative neural networks and discussing an overview of the state of the art of generative methods for traffic scenario generation.

Chapter 2 surveys the evolution of neural networks, from early models to modern deep learning architectures. It presents their fundamental building blocks: training procedures, regularization techniques, and key aspects of neural network architecture design, including activation functions, neural network layers, and hyperparameter optimization methods.

Chapter 3 focuses on generative modelling and scenario generation techniques. It discusses methods for generating synthetic data that capture real world patterns. The chapter provides an overview of scenario generation techniques and, after introducing Generative AI methods, compares traditional statistical approaches with innovative generative models for scenario generation.

These chapters provide theoretical and methodological foundations to the thesis, linking established concepts with contemporary tools and illustrating the application of machine learning to complex systems such as traffic and urban environments.

2

ARTIFICIAL INTELLIGENCE BACKGROUND

2.1 ARTIFICIAL NEURAL NETWORKS FOUNDATIONS

Machine Learning (ML) is a branch of Artificial Intelligence (AI) that focuses on algorithms capable of improving their performance through experience, learning directly from data. Rather than relying on explicitly programmed rules, ML models learn patterns within datasets and progressively enhance their performance as they are exposed to new information. By learning from data, ML techniques have transformed multiple fields, from natural language processing and medical diagnostics to autonomous driving, reshaping both research and real-world practice. Among ML techniques, Artificial Neural Networks (ANNs) have emerged as central tools [165]. Inspired by the functioning of the human brain, ANNs consist of layers of interconnected computational units that can approximate complex, non-linear mappings from inputs to outputs. Although they are not intended to replicate the full biological complexity of the brain, this abstraction of brain-like information processing has proven remarkably versatile: it delivers state-of-the-art performance on high-dimensional tasks such as classification, prediction, and pattern recognition. Thanks to their expressiveness and adaptability, ANNs have become a foundational element of modern machine learning.

EARLY THEORETICAL FOUNDATIONS The study of artificial neurons began in 1943 with Warren McCulloch and Walter Pitts [122], who proposed a mathematical model of a neuron based on binary logic. Their framework showed that neural circuits, with sufficient connectivity, could compute any logical function. These models were static and lacked learning mechanisms.
In 1949, Donald Hebb introduced a biologically plausible learning rule [72]: synaptic strength increases when two neurons activate simultaneously. Known as Hebbian learning and summarized as "neurons that fire together wire together", this principle provided the first hypothesis for activity-dependent plasticity and inspired unsupervised learning methods, including clustering, autoencoders, and generative models.

EMERGENCE OF LEARNING ALGORITHMS The transition to trainable networks came in 1958 with Frank Rosenblatt's Perceptron [160], a single-layer model that adjusted weights to minimize classification

errors on labeled data using a threshold activation. It converged for linearly separable problems but failed on non-linear ones, such as the XOR problem. In 1960, Widrow and Hoff developed ADALINE [199], which used a linear activation and the Least Mean Squares (LMS) algorithm, an early form of gradient descent, improving convergence but retaining the linear separability constraint. In 1969, Marvin Minsky and Seymour Papert demonstrated in Perceptrons [127] that single-layer networks cannot solve non-linearly separable problems. This analysis triggered widespread skepticism and the first AI winter (1974-1980), during which funding and interest in neural networks sharply declined.

THE RISE OF DEEP ARCHITECTURES These limitations shifted attention toward multilayer architectures, capable of learning hierarchical representations. Training such models, however, was hindered by vanishing or exploding gradients. The key advance arrived in 1986 when Rumelhart, Hinton, and Williams formalized backpropagation [161], enabling efficient gradient computation through multiple layers of Multi-Layer Perceptrons (MLPs). Building on earlier precursors, including Werbos's work in 1974 [198] and Linnainmaa's reverse-mode differentiation in 1976 [115], Rumelhart et al. demonstrated the practical feasibility of deep supervised learning. Despite this breakthrough, the adoption of deep architectures remained limited throughout the late 1980s and 1990s. This era is often referred to as the "second AI winter," a period of widespread disillusionment in the field of artificial intelligence following overhyped expectations from earlier decades. It led to drastic reductions in academic interest and commercial investment, much like the first AI winter in the 1970s, as progress stalled and promised breakthroughs failed to materialize in practical applications. Key contributing factors included severe computational constraints (computers of the time lacked the processing power for training large multilayer networks), the scarcity of sufficiently large and diverse datasets needed to effectively train these models, and the common problem of overfitting, where a model becomes overly tailored to the specific details and noise in its training data, performing well on that data but failing to generalize accurately to new, unseen examples, essentially "memorizing" rather than "learning" patterns. During this downturn, simpler "shallow" methods, such as support vector machines, dominated the machine learning landscape because they required fewer resources and delivered more reliable results on the smaller datasets available at the time.

THE MODERN DEEP LEARNING REVOLUTION Training deep networks became feasible after 2006 through a convergence of innovations:

- Unsupervised pretraining for weight initialization [78]

- Rectified Linear Units (ReLU) to reduce vanishing gradients [132]
- Large labeled datasets, such as ImageNet (2009) [42]
- GPU-accelerated computation

In 2012, AlexNet [104] advanced the state of the art on ImageNet, cutting error rates in half and launching the deep learning revolution. Later innovations like residual connections [71], batch normalization [91], and attention mechanisms [185] enabled the training of networks containing hundreds of layers and billions of parameters.
Empirical scaling laws [76, 81] later formalized performance gains with increased model size, dataset scale, and computational resources, though efficiency concerns have driven research into pruning, quantization, and lightweight architectures.
The expressive capacity and hierarchical structure of ANNs enable them to approximate complex probability distributions in high-dimensional settings. The sections that follow introduce the essential components and training procedures of neural networks, and subsequently present generative techniques that leverage these elements to model joint distributions and produce realistic synthetic samples.

2.2 ARCHITECTURES OF NEURAL NETWORKS

The architecture of an artificial neural network specifies the arrangement of computational units, known as neurons, into layers and their interconnections. Combined with trainable parameters (weights and biases) and non-linear activation functions, this architecture shapes the network's expressive power, that is, the class of functions it can approximate, as well as its ability to capture complex patterns in data.
An artificial neural network processes input data through a series of successive transformations: raw inputs are progressively transformed into increasingly abstract, higher-level representations, enabling tasks such as classification, regression, or generation. The effectiveness of a network depends critically on its architecture, which governs information flow, computational efficiency, and generalization to unseen data. The principal constituents of an artificial neural network include:

- **Neurons (or computational units)**: the basic processing elements that perform weighted summation of inputs, add a bias, and apply a non-linear activation function.
- **Layers**: input, hidden, and output layers. Hidden layers enable hierarchical feature learning.

- **Activation functions**: introduce non-linearity (e.g., ReLU, Sigmoid, TanH), allowing the network to model complex relationships.
- **Connection patterns**: the arrangement of interconnections between layers (e.g., fully connected, convolutional, recurrent, attention-based), which determines information flow, inductive biases, and computational efficiency.

Subsequent sections will explore major architectural families, highlighting how design choices address specific data modalities (e.g., images, sequences) and challenges (e.g., vanishing gradients, computational scalability).

2.2.1 *The Artificial Neuron*

The artificial neuron, also known as the Perceptron Unit, forms the foundational building block of neural networks. The concept traces back to the logical model proposed by McCulloch and Pitts in 1943 [122], which described neurons as binary threshold units capable of computing logical functions. This was later extended by Rosenblatt in 1958 [160] with the introduction of trainable weights and a learning algorithm, making it a practical computational unit. A modern artificial neuron (depicted in Figure 1) computes a weighted sum of its inputs, adds a bias term, and applies a non-linear activation function:

$$z = w^\top x + b \qquad a = \sigma(z) \tag{1}$$

where z is the pre-activation (also called net input or logit), $x \in \mathbb{R}^{d_{in}}$ is the input vector, $w \in \mathbb{R}^{d_{in}}$ the weight vector, $b \in \mathbb{R}$ the bias, $\sigma(\cdot)$ a non-linear activation function (e.g., sigmoid, ReLU, or tanh), and a is the resulting activation (output of the neuron).
The bias b increases the model's flexibility by allowing the activation function to shift, enabling decision boundaries that do not pass through the origin. Crucially, the non-linearity $\sigma(\cdot)$ is essential: without it, a multi-layer network would collapse to a single linear transformation, severely limiting its expressive power. This underpins the universal approximation theorems, which prove that multi-layer networks with non-linear activations can approximate any continuous function on compact subsets of $\mathbb{R}^n$ to arbitrary accuracy [35, 83]. In fully connected layers, the aggregation performed by each individual neuron is a weighted sum of its inputs, corresponding to a dot product between the input vector and the neuron's weight vector. Modern extensions, such as Graph Neural Networks and Attention Mechanisms [33, 185], generalize this aggregation process by incorporating topology-aware or input-dependent operations, thereby enabling the modelling of irregular data structures. This enhanced flexibility is es-

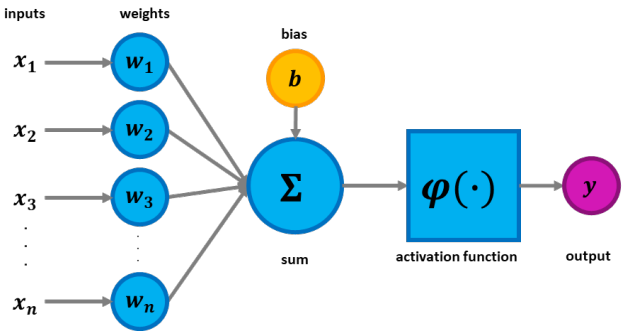


Figure 1 – Diagram of a single artificial neuron. Inputs $X = x_1, \ldots, x_n$ are multiplied by weights $W = w_1, \ldots, w_n$, summed with bias b to form $z = W^\top X + b$, and passed through a non-linear activation $\sigma(\cdot)$ to produce output $a = \sigma(z)$.

sential for domains involving irregular and context-specific relationships, such as road networks.

2.2.2 *Activation Functions*

Activation functions are essential for introducing non-linearity into neural networks, thereby enabling them to approximate complex, non-linear functions that would otherwise be impossible with linear layers alone [35, 83]. In the absence of non-linear activations, even multi-layer architectures reduce to simple linear transformations, limiting their expressive power.
Historically, early neural networks relied on smooth, saturating activation functions such as the logistic sigmoid (Equation 2) and the hyperbolic tangent (Equation 3).

$$\sigma(z) = \frac{1}{1 + e^{-z}} \qquad (2) \qquad\qquad \mathrm{TanH}(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \qquad (3)$$

These functions were favored due to their differentiability and bounded outputs: the sigmoid constrains values to $[0, 1]$ (well-suited for probability-like interpretations), whereas TanH produces zero-centered outputs in $(-1, 1)$. However, both are prone to saturation in their tails, leading to vanishing gradients that severely impeded the training of deep networks [59].
This limitation was largely overcome with the introduction of the rectified linear unit (ReLU), $\sigma(z) = \max(0, z)$ [132], and its variants, such as Leaky ReLU (LReLU) [120] and Parametric ReLU (PReLU) [70]. ReLU activations maintain a constant gradient for positive inputs, facilitating efficient computation and enabling the scaling of networks to unprecedented depths. As a result, they have become the standard choice for hidden layers in feed-forward and convolutional architectures.
In generative architectures, particularly those modelling bounded,

real-valued data is normalized to the interval $[-1, 1]$, and TanH activation function is often preferred, as illustrated in Table 1. Its range constraint and zero-centered outputs make it especially suitable for reconstruction tasks, and empirical studies consistently report enhanced stability compared to unbounded activations such as ReLU [77, 99, 108]. This practice has become standard across a wide spectrum of generative models, from Variational AutoEncoders to Generative Adversarial Network-based frameworks, in which compatibility with normalized data distributions plays a critical role. More recent architectures, as reported in Table 1, adopt TanH at the output stage, even when employing ReLU, Swish [153], GELU [75], and Mish [129] in hidden layers, underscoring its role as the canonical activation for bounded generative tasks. As illustrated in Figure 2, the different activation functions highlight the contrast between bounded and unbounded behaviors, with TanH constraining outputs to $[-1, 1]$.

Model	Hidden Layers	Output Layer
Kingma et al. VAE (2014) [99]	TanH	TanH
Rezende et al. VAE (2014) [155]	TanH	TanH
Higgins et al. β-VAE (2017) [77]	TanH	TanH
Karras et al. StyleGAN (2019) [95]	ReLU / LReLU	TanH
Vahdat et al. NVAE (2020) [182]	ReLU / Swish	TanH
Diffusion Models (2020) [79]	GELU / Swish	TanH

Table 1 – Adoption of activation functions in generative AI models.

Smooth activations such as Swish, GELU, and Mish offer marginal improvements primarily on unbounded data (images, text) processed by very large transformers. In the bounded, real-valued setting considered, this activation functions provide no measurable advantage and increase computational cost.

2.2.3 *Neural Network Layers*

Artificial Neural Networks (ANNs) consist of neurons that are interconnected and organized into successive layers. When arranged in this way, these computational units operate collectively to form a hierarchical processing architecture in which each layer extracts progressively more complex and abstract representations from the preceding one. Architectures with limited hierarchical depth are typically referred to as shallow networks, whereas stacking many such layers yields a deep neural network (DNN). Increasing depth enhances the network's ability to capture intricate patterns, model nonlinear relationships, and learn high-level features from high-dimensional data, a property that underpins the success of modern deep learning. Build-

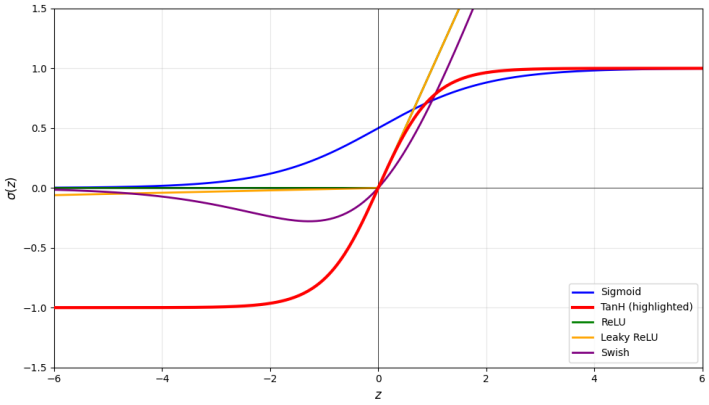


Figure 2 – Standard activation functions in deep learning. The TanH function (highlighted) is systematically adopted throughout all generative architectures proposed in this thesis, reflecting common practice in the VAE literature when input data are scaled to $[-1, 1]$.

ing on this hierarchical view, in this section, we examine the main classes of layers that form the backbone of state-of-the-art neural architectures.

Fully Connected Layers

Fully Connected (FC) layers connect every neuron in one layer to every neuron in the subsequent layer, realizing the transformation

$$a^{(l)} = \sigma\left(W^{(l)} a^{(l-1)} + b^{(l)}\right) \tag{4}$$

where $a(l) \in \mathbb{R}^{n_l}$ denotes the activation vector of layer l, $a^{(l-1)} \in \mathbb{R}^{n_{l-1}}$ is the activation vector (or input features) coming from the previous layer, $W^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$ is the weight matrix, $b^{(l)} \in \mathbb{R}^{n_l}$ is the bias vector, and $\sigma(\cdot)$ is an element-wise non-linear activation function.

As result of this connectivity, fully connected layers exhibit considerable expressive power: multilayer networks built from such layers with non-linear activations are known to be universal function approximators [83]. This property makes them particularly crucial for learning complex patterns and performing highly non-linear mappings between inputs and outputs.

Figure 3 illustrates a typical fully connected neural network (also known as a multilayer perceptron), highlighting how every neuron in one layer connects to every neuron in the next, enabling the rich representational capacity discussed above. FC layers are therefore the architecture of choice in the final stages of deep neural networks (e.g., for classification or regression heads), where they integrate high-level features extracted by preceding specialized layers (such as convolutions, transformers, etc.) into the desired output space [60]. Their

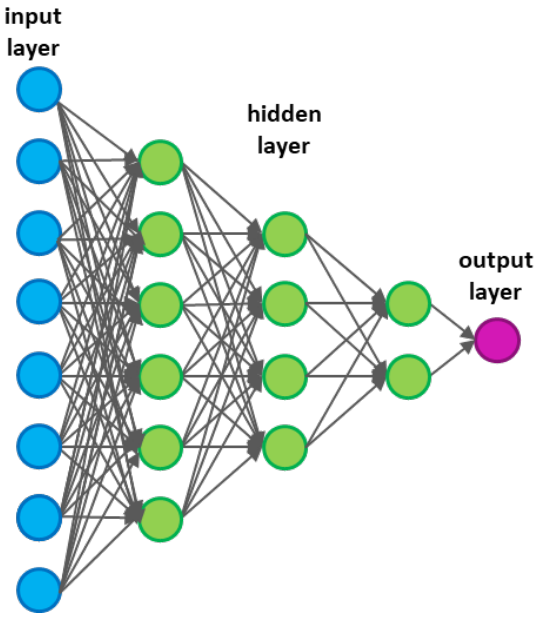


Figure 3 – Example of a fully connected neural network. Every neuron in one layer is connected to every neuron in the subsequent layer.

flexibility and proven theoretical effectiveness have firmly established fully connected layers as a core building block of modern neural architectures.

Convolutional Layers

Convolutional layers are the building blocks of Convolutional Neural Networks (CNNs) (Figure 4) and provide the foundational inductive bias for processing spatially structured data, most notably images. A CNN processes an input through a sequence of convolutional layers (extracting local patterns via learned filters), non-linear activations, and pooling layers (reducing spatial dimensions while preserving important features), progressively building hierarchical representations from low-level edges to high-level semantic concepts, before final classification via fully connected layers. This approach demonstrated remarkable effectiveness on the ImageNet benchmark with the introduction of AlexNet [104]. Rather than establishing fully connected mappings between all input and output units (as in fully connected layers), CNNs impose two principled structural priors: spatial locality (via local receptive fields, restricting each output neuron to a small contiguous patch of the input) and translation equivariance (via weight sharing, applying the same filter across all spatial positions so that shifting the input shifts the feature map correspondingly) [109].

These priors are realized by learning a collection of compact filters (kernels) that are systematically convolved with the input feature map. Formally, given an input feature map $A^{(l-1)} \in \mathbb{R}^{H \times W \times C_{in}}$ at

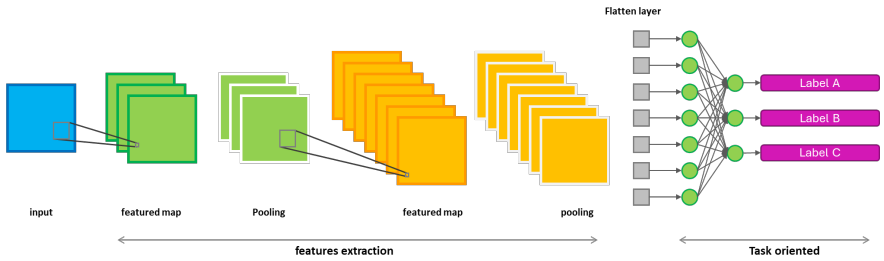


Figure 4 – Architecture of a convolutional neural network (CNN). The input image is processed through successive convolutional and pooling layers to extract hierarchical features, followed by fully connected layers for the classification task.

layer $l-1$, the output activation $A^{(l)}_{i,j,k}$ of the k^{th} output channel at spatial position (i,j) is computed as

$$A^{(l)}_{i,j,k} = \sigma\left(b^{(l)}_k + \sum_{c=1}^{C_{in}} \sum_{u=0}^{K_h-1} \sum_{v=0}^{K_w-1} W^{(l)}_{u,v,k,c}\, A^{(l-1)}_{i\cdot s_h+u,j\cdot s_w+v,c}\right) \quad (5)$$

where $W^{(l)} \in \mathbb{R}^{K_h\times K_w\times C_{out}\times C_{in}}$ is the learnable kernel tensor, (s_h, s_w) denote the vertical and horizontal strides, $b^{(l)}_k$ is the bias term, and $\sigma(\cdot)$ is the element-wise non-linear activation function. Figure 5 illustrates the scalar case with a 3×3 kernel, stride 2, and no padding.

The stride controls both the density of the computed responses and the spatial downsampling factor: a stride of 1 preserves input resolution (up to border effects), while strides greater than 1 reduce the output spatial dimensions. Optional zero-padding can be applied to preserve or deliberately expand the spatial support.

Crucially, the same filter parameters are shared across all spatial locations, yielding a reduction in the number of trainable parameters compared to fully connected layers and enabling the network to scale gracefully to high-resolution inputs. Applying multiple filters in parallel produces a multi-channel output tensor, where each channel specializes in detecting distinct local pattens (edges, textures, corners, etc.) at every position. As depth increases, the hierarchical composition of these elementary features gives rise to increasingly abstract and semantically rich representations [211].

Graph Neural Networks

While convolutional layers exploit translational invariance and local regularity in Euclidean data (e.g., images, audio, time series), many real-world domains are modeled as graphs, including molecular structures, social networks, citation graphs, knowledge bases, and transportation systems. In these domains the topology is irregular, permutation-invariant, and cannot be embedded into a fixed grid without distorting relational information.

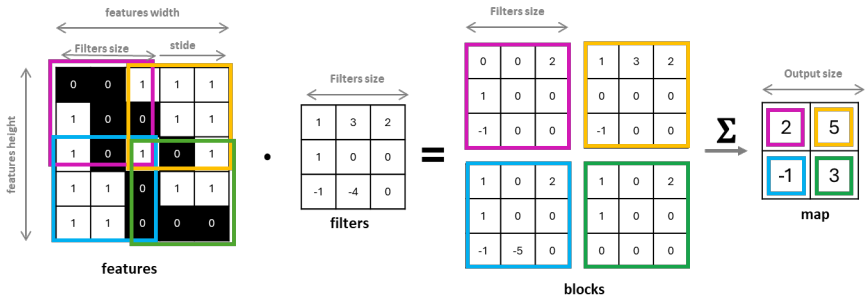


Figure 5 – Illustration of a 2D convolution with a 3×3 filter and stride 2 on a 5×5 input feature map (no padding). Colored regions highlight the receptive fields contributing to each output position.

Graph Neural Networks (GNNs) (Figure 6) generalize the convolution paradigm to non-Euclidean domains via iterative message-passing: each node refines its representation by aggregating information from its local neighborhood, followed by a learned transformation and non-linear activation. A generic message-passing layer at iteration l is formulated as

$$h_i^{(l)} = \sigma\Big(W^{(l)} \cdot \mathrm{AGG}\{h_j^{(l-1)} : j \in \mathcal{N}(i) \cup \{i\}\} + b^{(l)}\Big) \tag{6}$$

where $h_i^{(l)}$ is the embedding of node i at layer l, $\mathcal{N}(i)$ its set of neighbors, $\mathrm{AGG}\{\cdot\}$ a permutation-invariant aggregation operator (sum, mean, max, or attention-weighted), $W^{(l)}, b^{(l)}$ learnable parameters, and σ a non-linear activation function. The message-passing

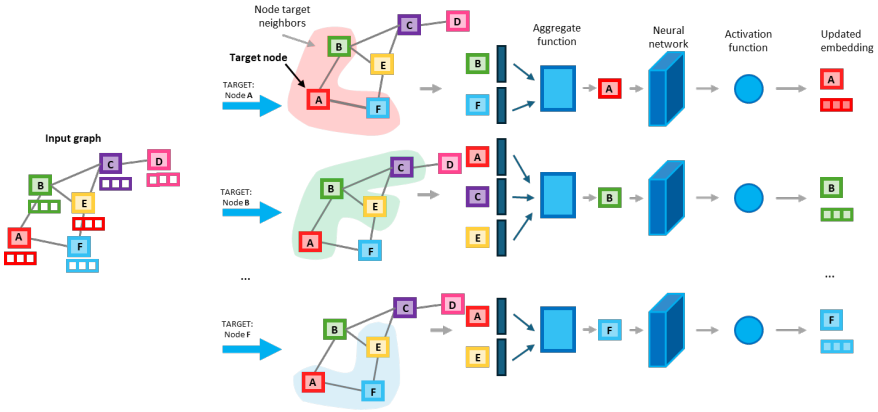


Figure 6 – Architecture of a Graph Neural Network (GNN) based on message passing. The target node aggregates information from its neighbors, including itself via a self-loop. Each neighbor is processed through a neural transformation, and the resulting representations are aggregated to compute the updated embedding of the central node.

paradigm (described in Equation 6) defines the canonical formulation of Graph Neural Networks (GNNs) [164, 201, 219]. GNN

has become the universally accepted umbrella for neural architectures that iteratively propagate and aggregate information over graph-structured data. Within this family, the Graph Convolutional Network (GCN) introduced by Kipf and Welling [100] offers an elegant and computationally efficient formulation:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{7}$$

where A is the adjacency matrix of the graph, I the identity matrix (introducing self-loops), $\tilde{A} = A + I$ the adjacency matrix with added self-loops, $\tilde{D}$ its diagonal degree matrix, $H^{(l)}$ the matrix of node embeddings at layer l, and $W^{(l)}$ the learnable weight matrix.
This model achieves linear complexity in the number of edges and has become a foundational building block in countless GNN architectures due to its simplicity, stability, and strong empirical performance across diverse tasks.
The GCN's degree-normalized mean aggregation effectively smooths node features in a manner analogous to low-pass filtering in the graph spectral domain [201]. While this isotropic weighting is well-suited for domains with relatively homogeneous edge semantics, it can be further enhanced in structured domains where edge types encode distinct relational priors.

Modern GNNs build upon the GCN foundation in several complementary directions:

- learnable per-edge attention (GAT [186], transformer-based GNNs),
- expressive aggregators with increased discriminative power (GraphSAGE [66], GIN [203], PNA [33]),
- explicit edge-feature conditioning or relation-specific transformations.

Early spectral methods [22, 39], while historically significant and rigorously grounded in graph signal processing [136, 137, 172], imposed scalability challenges due to eigen-decomposition or localization constraints. The GCN can be interpreted as a first-order, scalable approximation of these spectral models [201], striking a practical balance between theoretical grounding and computational efficiency. Methods proposed in this thesis (Chapter 5) adopt GCN layers, to capture correlations induced by the graph structure of the data and to preserve computational efficiency.

2.3 TRAINING AND OPTIMIZATION OF DEEP NEURAL NETWORKS

The training of an ANN consists of an iterative process during which the model progressively refines its internal parameters to better capture the statistical regularities present in the training data. At

the heart of this process lies an objective function (known as loss function), which quantifies the discrepancy between the network's current predictions and the desired behavior on the training examples.
Training aims to learn parameters that minimize the loss on observed data while enabling the model to generalize effectively to unseen examples, whether the task involves classification, regression, representation learning, or the generation of new realistic samples.
The training process is guided by two essential components:

- **a differentiable objective function (the loss)**, which measures the deviation of the current model predictions from the target behavior and supplies the critical information needed to steer parameter updates. Depending on the specific problem, this objective may take the form of reconstruction error, classification loss, perceptual metrics, negative log-likelihood, or distributional divergence measures;
- **an iterative optimization method** that exploits the gradient of the loss with respect to the parameters to perform updates that consistently decrease the value of the objective.

The training of deep neural networks follows a defined iterative process. Each epoch represents a full pass through the training dataset, divided into multiple iterations. In each iteration, a small random subset of samples, a mini-batch, is selected and fed through the network: inputs are processed layer by layer to compute predictions and evaluate the loss function. Gradients of the loss with respect to the parameters are then computed via backpropagation and used to update the weights via an optimizer. Yet the success of the entire process critically depends on numerous additional choices and techniques, detailed in the following sections, such as optimization strategies, regularization methods, and monitoring practices to prevent overfitting.
Finally, beyond the parameters directly optimized by gradient descent, deep learning models depend on a large set of hyperparameters (e.g., network depth, width of layers). Finding effective configurations for these is itself an optimization problem, typically addressed through different strategies.
The principles and techniques described in this section provide the groundwork for the training methods applied in the thesis' experimental chapters.

2.3.1 *Loss Functions*

Loss function provides a differentiable scalar measure of discrepancy that drives the optimization of a neural network's parameters θ. In traditional supervised learning, where each training example consists of an input-target pair (x_i, y_i), the loss quantifies the error between the model prediction $f_\theta(x_i)$ and the ground-truth label y_i.

Common example of loss function include Mean Squared Error (MSE) for regression tasks

$$\mathcal{L}_{\text{MSE}}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \|f_\theta(x_i) - y_i\|_2^2 \tag{8}$$

and the binary cross-entropy loss for binary classification problems

$$\mathcal{L}_{\text{BCE}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \left[y_i \log f_\theta(x_i) + (1 - y_i) \log\big(1 - f_\theta(x_i)\big) \right] \tag{9}$$

However, many contemporary deep learning paradigms, particularly generative models, operate in settings where explicit ground-truth label y_i are either absent or conceptually inappropriate. In these cases, the learning objective can no longer be expressed as a simple point-wise error. Instead, training reduces distributional matching: the model is required to induce a distribution p_θ over high-dimensional outputs (e.g., multivariate traffic speed trajectories across an entire road network) that faithfully approximates the unknown true data-generating distribution p_{data}.

To formalize this goal, the loss function is replaced by (or augmented with) a statistical divergence or distance between probability measures. The most commonly encountered families are:

- f**-divergences** [3, 34], including the Kullback–Leibler (KL) divergence, Jensen–Shannon divergence, and reverse KL, all of fundamental importance in maximum-likelihood and variational inference;
- **Integral Probability Metrics (IPMs)** [131], with the 1-Wasserstein distance (Earth Mover's distance) as the prototypical example, widely adopted for its favorable geometric properties and training stability.

2.3.2 *Regularization Techniques*

Despite the primary goal of empirical risk minimization, deep neural networks remain highly susceptible to *overfitting*, particularly when model capacity significantly exceeds the intrinsic complexity of the underlying data distribution. In the context of generative methods, overfitting typically manifests as the production of synthetic samples that closely replicate the most common patterns observed during training, rather than learning and reproducing the full underlying data distribution.
Classical regularization techniques constrain the parameter space by augmenting the objective with penalty terms, the most prevalent being weight decay (L_2 regularization), commonly implemented via

decoupled weight decay in modern optimizers such as Adam [98].
In addition to explicit regularization techniques, many contemporary generative models benefit from various forms of implicit regularization that arise from the structure of the learning objective and from architectural design choices. In Variational Autoencoders (VAEs) and related frameworks, the divergence between approximate posteriors and the prior distribution imposes a principled constraint on the effective capacity of the latent space.

In adversarial frameworks such as Generative Adversarial Networks (GANs), the generator is indirectly regularized through the dynamic feedback signal originating from the discriminator, which continuously shapes the generated distribution toward statistical plausibility. Architectural mechanisms such as normalization layers and dropout further reinforce this effect by reducing co-adaptation and facilitating optimization.
The following paragraphs briefly describe the most commonly employed regularization techniques.

Weight Decay

Weight decay, also known as L_2 regularization, consists in augmenting the objective function with a quadratic penalty on the model parameters:

$$\mathcal{L}_{\text{total}}(\theta) = \mathcal{L}_{\text{data}}(\theta) + \frac{\lambda}{2}\|\theta\|_2^2 \tag{10}$$

where $\lambda > 0$ is a hyperparameter that controls the strength of the regularization, $\mathcal{L}_{\text{data}}(\theta)$ is the empirical loss computed on the training data (e.g., cross-entropy for classification or mean squared error for regression), and θ denotes the vector of all trainable parameters of the model [105]. The quadratic penalty $\frac{\lambda}{2}\|\theta\|_2^2$ term encourages smaller weights, leading to smoother learned functions and helping to prevent overfitting. Weight decay is one of the most widely adopted regularization techniques in deep learning and is routinely applied in virtually all neural network training procedures, including those employing adaptive optimizers such as Adam (presented in Section 2.3.3).

Normalization Layers

Normalization layers mitigate the problem of internal covariate shift by standardizing the distribution of layer activations during training. Batch Normalization [91] computes mean and variance across the mini-batch and remains a standard choice in many convolutional architectures. Layer Normalization [7] normalizes each training example independently, and has become prevalent in recurrent networks and Transformer-based models due to its

robustness to sequence length variations and mini-batch size. Both techniques significantly accelerate training, allow the use of higher learning rates, and provide a moderate regularization effect.

Dropout

Dropout [178] is a simple yet effective stochastic regularization method. During training, each neuron is randomly dropped (i.e., temporarily deactivated) with probability p (typically between 0.1 and 0.5), preventing complex co-adaptations among units. This procedure can be interpreted as implicitly training an exponential ensemble of thinner subnetworks. During inference, all neurons are kept active (i.e., no units are dropped), and their outputs are scaled by a factor $1/(1-p)$ to preserve the expected activation magnitude. Dropout and its structured variants continue to be widely adopted across a broad range of tasks, including supervised learning, unsupervised representation learning, and generative modelling frameworks.

2.3.3 *Gradient-Based Optimization*

Training deep neural networks requires minimizing a high-dimensional, non-convex objective function through iterative first-order optimization algorithms. Given the scale of modern architectures and the availability of highly parallel computing resources (like GPU/TPU accelerators), methods with linear computational complexity in the number of parameters have become the standard approach [17].

The most widely used algorithm in contemporary deep learning, particularly in generative modelling, is Adam (Adaptive Moment Estimation) [98]. Adam combines the advantages of momentum-based acceleration with per-parameter adaptive learning rates derived from exponentially decaying averages of past gradients and their squares. This dual mechanism provides fast initial progress, stable convergence, and robustness to noisy or sparse gradients, making it especially well-suited to the training of variational autoencoders, generative adversarial networks, and hybrid architectures.

Stochastic Gradient Descent with momentum (SGD + momentum) [146] remains a widely adopted baseline for large-scale supervised classification tasks, but has been largely superseded by Adam and its variants in unsupervised and generative modelling contexts, where rapid and stable optimization of complex multimodal objectives is crucial.

Variants of Adam include AdamW [116], which decouples weight decay from the adaptive learning rate component, and more recent large-batch optimizers such as LAMB [207]. Throughout the deep generative modelling literature of the past decade, including the works on Variational Autoencoders [99] and Generative Adversarial

Networks [61] that form the foundation of this dissertation, Adam in its original formulation has consistently been the optimizer of choice.

2.3.4 *Forward and Backward Propagation*

The training of modern neural architectures relies on the automatic differentiation of a scalar loss function with respect to the model parameters. This process is decomposed into two successive phases executed iteratively: forward propagation and backward propagation.
During the **forward pass**, an input sample x is fed through the network following the network's computational graph: each layer sequentially applies an affine transformation followed by a activation function, eventually producing the model prediction $\hat{y} = f_\theta(x)$. The loss $\mathcal{L}(\hat{y}, y)$ is then computed by comparing predictions $\hat{y}$ to the ground-truth targets y.
In the **backward pass (backpropagation)**, the gradient of the loss with respect to every trainable parameter is efficiently calculated by applying the chain rule in reverse topological order across the computational graph [161].
Backpropagation efficiently reuses the intermediate activations computed during the forward pass, enabling linear time and memory complexity with respect to the number of parameters. When combined with first-order optimization algorithms, it has constituted the foundational training paradigm for virtually all deep neural network architectures, including feed-forward networks, convolutional networks, recurrent networks, graph neural networks, and Transformers. Modern deep learning frameworks implement fully symbolic or automatic differentiation systems that transparently execute both phases, relieving practitioners from manual gradient derivation while preserving numerical precision and computational efficiency.

2.3.5 *Hyperparameter Optimization and Training Practices*

In neural networks, hyperparameters (e.g., learning rate, batch size, architectural dimensions, regularization coefficients, and task-specific terms such as the β parameter in β-VAE) are not learned via gradient descent and must be specified prior to training. In deep generative models, where training is particularly sensitive to configuration choices, their selection is typically performed through a combination of manual search guided by validation performance and automated optimization methods [13].
A widely adopted strategy in this context is Bayesian Optimization, which models the performance surface as a probabilistic function

and sequentially selects hyperparameter configurations to evaluate based on an acquisition criterion. This approach balances exploration of uncertain regions with exploitation of promising candidates, yielding efficient search even in high-dimensional, expensive-to-evaluate settings [174]. As a result, Bayesian optimization has become a technique for tuning deep generative models, where training runs are computationally costly and sensitive to hyperparameter choices.

Through the composition of multiple layers of non-linear transformations, deep neural networks have acquired the ability to extract increasingly abstract and structured representations of complex data [60]. This representational power has been a key driver of the remarkable progress achieved by deep learning across a wide range of domains over the past decade. At the same time, it has enabled the emergence of a new class of models capable of producing novel and realistic data samples.

3

STATE OF THE ART

3.1 MODELLING FOR SCENARIO GENERATION UNDER UNCERTAINTY

Modern urban mobility systems exhibit complex, highly stochastic dynamics that cannot be adequately captured by traditional deterministic models. Traffic conditions fluctuate due to a wide range of interacting factors, making uncertainty an intrinsic property of the system. As a result, data-driven approaches capable of learning and reproducing this variability have become increasingly central to contemporary transportation analytics.

A widely adopted strategy for representing uncertainty consists in discretizing it through a finite set of scenarios, each corresponding to a possible realization of the underlying stochastic parameters [85, 144]. Formally, a scenario is a discrete realization of the stochastic process [144, 158]. Scenario generation (SG) provides a framework for generating such representations, enabling the synthesis of data instances that reflect the variability, correlations, and structural dependencies observed in real systems. By replacing unknown random variables with a structured collection of representative samples, SG transforms complex stochastic phenomena into a form that can be analyzed, simulated, or used to train models [85, 170, 188].

A key distinction exists between scenario generation and classical forecasting. Forecasting aims to predict the most likely future state, whereas SG is designed to characterize the full distribution of plausible outcomes, including low-probability events that lie in the tails of the distribution [167]. Scenarios are therefore essential for tasks such as simulation, stress testing, and the evaluation of data-driven algorithms under diverse and potentially adverse conditions. Their usefulness depends critically on their statistical fidelity: a scenario set must adequately span the support of the uncertainty while preserving marginal distributions, multivariate correlations, and domain-specific structural patterns [43, 47, 73, 152, 171].

A broad spectrum of methodologies has been proposed to address this challenge. Early approaches relied on sampling techniques such as Monte Carlo and Latin Hypercube Sampling, which are simple and general but often require large sample sizes to achieve acceptable approximation quality in high-dimensional settings [74, 124, 139, 171, 179, 220]. More advanced statistical methods construct discrete scenario sets that systematically approximate continuous distributions

[85, 96], enabling the analysis of rare yet plausible conditions and providing a structured representation of uncertainty [9, 40, 144].

The increasing availability of large-scale data has recently positioned scenario generation as a rapidly evolving, interdisciplinary research area [31, 147].

In logistics and transportation, facility location and inventory management problems are optimized against ensembles of demand and disruption scenarios to ensure operational resilience [169]. Analogous applications have emerged also in healthcare, where scenarios are employed to evaluate epidemic spread and treatment strategies across diverse disease progression trajectories [51].

Building upon this framework, which has established the conceptual foundation for the proposed approach, the subsequent section provides a detailed overview of the main scenario generation methodologies. We will discuss classical statistical approaches (Section 3.2) before presenting advanced approaches based on generative artificial intelligence, which represent the core methodological focus and primary contribution of this research, outlining their foundation principles and architectures (Section 3.3), and their use in transportation applications (Section 3.4).

3.2 STATISTICAL APPROACHES FOR SCENARIO GENERATION

Statistical techniques offer a distribution-aware framework for generating scenarios that reflect the uncertainty inherent in stochastic systems. These methods rely on explicit probabilistic modelling or empirical estimation to generate scenarios that preserve key statistical properties, including marginal distributions, dependence structures, and tail behavior. They are particularly valuable in applications requiring interpretability, convergence guarantees, and controlled approximation error.

The following subsections examine the two complementary paradigms of sampling-based methods and forecasting-based approaches, presenting the main techniques and their mathematical formulations. However, since this thesis does not take into account temporal dependencies, this complete overview of the state-of-art methods will subsequently narrow its focus by identifying the most appropriate approaches, grounded in the multivariate statistical structure of the data.

- **Sampling-based methods** rely on probabilistic sampling techniques to generate synthetic scenarios that statistically replicate the distributional and dependence properties of the observed data.
- **Forecasting-based approaches**, in contrast, employ time-series predictive models to extrapolate patterns from historical data

and generate predictive scenarios. This paradigm is especially suitable when temporal dependencies, trends, seasonality, or exogenous drivers play a dominant role in the dynamics of the system.

These two classes of techniques are frequently combined in practice to exploit their respective strengths: sampling-based methods provide statistical fidelity and flexibility in uncertainty representation, while forecasting-based methods ensure temporal consistency and plausibility in projected paths.

3.2.1 *Sampling-based Methods*

Sampling-based methods generate scenarios by drawing instances from a learned probability distribution of stochastic variables. These methods aim to represent the uncertainty and variability inherent in complex systems by generating representative scenarios that preserve the key statistical properties of the data. This category encompasses several widely used techniques, including the Monte Carlo (MC) method [68], Markov Chain Monte Carlo (MCMC) [141], Latin Hypercube Sampling (LHS) [208], and Copula function sampling [10, 183].

Monte Carlo sampling

Monte Carlo sampling (MC) is a widely used statistical technique that generates scenarios by randomly sampling from the known probability density function (PDF), denoted as $f_X(x)$, of the stochastic variables under consideration [101, 126, 157].
Let $X = (X_1, X_2, \ldots, X_d) \in \mathbb{R}^d$ represent the vector of uncertain inputs with joint PDF $f_X(x)$. The MC method proceeds by drawing N independent and identically distributed (i.i.d.) samples $\{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$ such that $x^{(i)} \sim f_X(x)$ for $i = 1, \ldots, N$.
For any measurable function of interest $g : \mathbb{R}^d \to \mathbb{R}$ (e.g., system cost, reliability index, or performance metric), the expected value is approximated via the sample average:

$$\mathbb{E}[g(X)] \approx \hat{g}_N = \frac{1}{N} \sum_{i=1}^{N} g(x^{(i)}) \tag{11}$$

with the variance of the estimator given by

$$\mathrm{Var}(\hat{g}_N) = \frac{\mathrm{Var}(g(X))}{N} \tag{12}$$

By the central limit theorem [53], $\hat{g}_N$ converges to $\mathbb{E}[g(X)]$ at a rate of $\mathcal{O}(N^{-1/2})$ as $N \to \infty$, provided $\mathrm{Var}(g(X)) < \infty$. MC-based methods rely on repeated random samples to simulate the behavior of complex

systems under uncertainty, progressively refining the approximation of the true distribution as the sample size grows.
The Monte Carlo method necessitates the prior specification or estimation of the PDF characterizing each stochastic variable. Subsequently, independent realizations are generated in accordance with this estimated PDF, typically via inverse transform sampling, where $x_i = F_{X_i}^{-1}(u_i)$ and $u_i \sim \mathcal{U}(0,1)$ generating scenarios for stochastic optimization, simulation, or risk assessment.
A key advantage of the Monte Carlo approach lies in its methodological simplicity and wide-ranging application fields, as it demands minimal assumptions concerning the underlying system dynamics or the precise form of the probability distribution.
However, Monte Carlo sampling encounters significant challenges in practice. In cases where the stochastic variables do not conform to simple or well-characterized distributions, such as Gaussian, exponential, or uniform, accurately specifying the underlying PDF proves difficult. Non-parametric estimation via kernel density methods or histogram-based approaches may be employed, but these introduce estimation error and require substantial historical data. This uncertainty often necessitates the generation of a substantially large number of samples to adequately capture the full extent of variability and higher-order statistical characteristics (e.g., skewness, kurtosis) inherent in the data. As a result, Monte Carlo methods may become computationally intensive and inefficient, particularly in high-dimensional or complex problem domains where the convergence rate remains $\mathcal{O}(N^{-1/2})$ irrespective of dimensionality.
Moreover, the traditional MC approach treats samples as independent realizations and therefore does not inherently account for autocorrelation, defined as

$$\rho(\tau) = \frac{\mathrm{Cov}(X_t, X_{t-\tau})}{\sqrt{\mathrm{Var}(X_t)\mathrm{Var}(X_{t-\tau})}} \tag{13}$$

which is commonly present in time series and spatial data. This limitation implies that Monte Carlo sampling may inadequately capture temporal persistence or spatial coherence, which are critical in applications such as weather modelling, financial forecasting, renewable energy generation, and, as in our case, traffic scenario generation. Overall, while Monte Carlo sampling remains a fundamental and versatile tool in uncertainty quantification, its computational cost and limitations in handling dependent data motivate the development and adoption of more advanced sampling techniques.

Markov Chain Monte Carlo sampling

To address the inherent limitation of the traditional Monte Carlo method in capturing autocorrelation, advanced approaches based

on Markov Chain Monte Carlo (MCMC) [5] have been proposed. Unlike standard Monte Carlo sampling, MCMC methods generate samples that are not independent but form a Markov chain $\{x^{(0)}, x^{(1)}, \ldots, x^{(T)}\}$, where each sample depends on the previous one according to a transition kernel $p(x' \mid x^{(t-1)})$. This dependency structure allows MCMC to effectively model autocorrelation within the data and to better represent the joint probability distribution of multiple stochastic variables.
A widely used MCMC algorithm is the Metropolis-Hastings sampler[126], which constructs a reversible chain with stationary distribution $f_{\mathbf{X}}(x)$ by proposing a candidate $\mathbf{y} \sim q(\mathbf{y} \mid x^{(t-1)})$ and accepting it with probability

$$\alpha = \min\left(1, \frac{f_X(y)\, q(x^{(t-1)} \mid y)}{f_X(x^{(t-1)})\, q(y \mid x^{(t-1)})}\right) \tag{14}$$

By incorporating these dependencies, MCMC-based methods improve the accuracy and realism of the generated scenarios, making them particularly suitable for applications where temporal or spatial correlations are significant. Consequently, MCMC techniques have found widespread use in fields such as Bayesian inference, time series analysis, and spatial statistics, providing a powerful toolset for uncertainty quantification in complex stochastic systems.
However, despite their strengths, MCMC methods can be computationally intensive and require careful tuning of proposal distributions and step sizes to foster proper convergence, which ensures that the Markov chain adequately explores the entire support of the target distribution and that sample averages converge to their theoretical expectations. The dependency between samples, inherent to the Markov chain structure, often leads to high autocorrelation and reduced Effective Sample Size (ESS), approximated as

$$\mathrm{ESS} \approx \frac{T}{1 + 2\sum_{k=1}^{\infty} \rho(k)} \tag{15}$$

further increasing computational cost, especially in high-dimensional problems.

Latin Hypercube sampling

Another effective approach within this category is Latin Hypercube Sampling (LHS). LHS is a statistical method designed to generate representative and well-distributed samples of stochastic variables for uncertainty quantification, simulation, and global sensitivity analysis. Unlike MC and MCMC sampling, which may produce clustered or correlated samples, LHS divides the cumulative distribution function (CDF) $F_{X_i}(x_i)$ of each variable into N equally probable intervals $[(k-1)/N, k/N]$, $k = 1, \ldots, N$, and samples exactly once from each interval.

Formally, for variable i and stratum k, a uniform random variable $U_{i,k} \sim \mathcal{U}((k-1)/N, k/N)$ is drawn, and the corresponding sample is obtained via inverse transform:

$$x_{i,k} = F_{X_i}^{-1}(U_{i,k}) \tag{16}$$

This systematic sampling improves efficiency by capturing the variability of each input variable more effectively with fewer samples. LHS-based methods typically consist of two fundamental steps: (i) the stratified sampling step, during which a set of representative samples is generated to reflect the statistical characteristics of the underlying stochastic variables, and (ii) the permutation step, which rearranges the sampled values across variables to minimize spurious correlations. By optimizing the pairing of ranks, often using algorithms such as simulated annealing or genetic optimization, this step enhances the statistical independence of the input variables, thereby improving the quality and robustness of simulation or analysis results.

Compared to Monte Carlo sampling methods, the LHS-based approach requires fewer samples, avoids repetition, and guarantees that the sampling points cover the entire support of each marginal distribution. Moreover, LHS-based methods can represent arbitrary theoretical distributions and ensure full stratification across the input space.
However, classical LHS does not inherently capture the correlations between stochastic variables, which are often present in real-world problems. Therefore, to model such dependencies accurately, Copula-based methods have been introduced as a powerful tool to represent and capture the dependence structure between variables while preserving their marginal distributions.

Copula sampling

Based on Sklar's Theorem [173], Copula sampling [133] method is a powerful statistical tool used to generate synthetic scenarios that replicate dependence structures of multivariate random variables [10, 183]. Sklar's Theorem states that any multivariate joint distribution $F_X(x_1, \ldots, x_d)$ of a d-dimensional random vector $X = (X_1, X_2, \ldots, X_d)$ can be expressed in terms of its marginal distribution CDFs and a Copula function C that captures the dependence structure:

$$F_{X_1,\ldots,X_d}(x_1, \ldots, x_d) = C\left(F_{X_1}(x_1), F_{X_2}(x_2), \ldots, F_{X_d}(x_d)\right) \tag{17}$$

where $F_{X_i}(x_i)$ denotes the marginal Cumulative Distribution Function (CDF) of the variable X_i, and $C : [0,1]^d \to [0,1]$ is the copula function.

PHASE 1: MODELLING AND CALIBRATION PHASE This phase aims to accurately capture the empirical properties of the data. The joint distribution is decomposed into independent marginal distributions and a Copula function $C(u_1, \ldots, u_d)$, where $u_i = F_{X_i}(x_i) \in$.
The steps involved are:

- **Fitting Marginal Distributions**: Suitable marginal cumulative distributions (CDFs) F_i are fitted to each component of the multivariate data.
- **Copula Selection and Calibration**: a suitable Copula family (e.g., Gaussian, Student's t, Clayton, Gumbel, or Vine copulas [1, 49, 94]) is selected and its parameters calibrated. The selection process is driven by the empirical characteristics of the data, primarily the nature of the tail dependence. Goodness-of-fit tests or information criteria (like AIC) are often employed to determine the best-fitting family. Calibration involves estimating the parameters that maximize the likelihood of observing the historical data using methods such as Maximum Likelihood Estimation (MLE) or rank-based methods.

PHASE 2: SAMPLING (SCENARIO GENERATION) PHASE Once the model is calibrated, new scenarios are generated through a two-step procedure derived from the inverse of Sklar's Theorem:

(i) Sampling from the Copula: a sample vector $u = (u_1, u_2, \ldots, u_d)$ is drawn from the selected copula distribution, $u \sim C(u_1, u_2, \ldots, u_d)$. For a Gaussian Copula, this involves:
  - Generating a multivariate normal sample $z \sim \mathcal{N}(0, \Sigma)$, where Σ is the correlation matrix.
  - Applying the standard normal CDF Φ to each component: $u_i = \Phi(z_i), \quad i = 1, \ldots, d$

(ii) Applying the Inverse Marginal CDFs: Each component $u_i \in$ is transformed via the inverse marginal CDF (quantile function) $F^{-1}_{X_i}$ to obtain a sample in the original variable space: $x_i = F^{-1}_{X_i}(u_i), \quad i = 1, \ldots, d$

The resulting vector $x = (x_1, x_2, \ldots, x_d)$ follows the empirical joint distribution, combining realistic marginals with the observed dependence structure.

This two-step mechanism ensures robust and accurate reproduction of the full empirical joint distribution.
One major advantage of this approach is its flexibility in capturing complex dependencies, such as:

- Tail dependence (e.g., with t-Copulas or Archimedean Copulas),
- Asymmetric relationships between variables,
- Nonlinear correlations in high-dimensional settings.

Copula sampling provides a flexible and theoretically sound approach for multivariate scenario generation in several domains including finance, energy, and hydrology [50, 58, 140, 163], where accurate modelling of joint behavior is critical.
In the transportation domain, Cervellera et al. [26] propose a data-driven Copula-based model that generates correlated Origin-Destination (OD) demands, link flows, and simulated intersection speeds for urban networks, sampling input parameters (e.g., driver behavior, demand) from fitted Copulas trained on a traffic dataset.
Other examples in the transportation domain are [117] and [118]. In the first work, a Vine Copula has been used to generate multi-lane speeds and heavy vehicle flows, with emphasis on tail dependencies for risk-averse management, while the second adopts a spatial Copula to generate vehicle flows and traffic density.

Table 2 – Summary of sampling-based scenario generation methods.

Method	Description
Monte Carlo (MC)	**Advantages:** Simple, general-purpose; minimal assumptions; widely applicable. **Limitations:** Requires large number of samples; computationally expensive in high dimensions; ignores autocorrelation.
Markov Chain Monte Carlo (MCMC)	**Advantages:** Captures temporal/spatial dependencies; suitable for Bayesian inference. **Limitations:** Computationally intensive; requires careful tuning; reduced effective sample size.
Latin Hypercube Sampling (LHS)	**Advantages:** Ensures uniform coverage of input space; efficient with fewer samples. **Limitations:** Classical LHS does not capture correlations between variables.
Copula-based Sampling	**Advantages:** Explicitly models multivariate dependencies; flexible (tail dependence, asymmetry). **Limitations:** Fitting can be complex in high dimensions.

3.2.2 *Forecasting-based Methods*

Forecasting-based methods generate future scenarios by leveraging predictive models trained on historical data, without requiring explicit assumptions about the underlying multivariate probability density functions (PDFs). These approaches are particularly valuable when the statistical properties of the data are complex, non-stationary, or poorly characterized, as is often the case in renewable energy production, electricity market prices, or urban traffic flows. By learning patterns directly from observed time series, forecasting models can produce temporally coherent trajectories that reflect realistic dynamics, making them intuitive and widely adopted in operational planning. Among the most established techniques are linear stochastic processes such as Auto-Regressive Moving Average (ARMA) models and their extension, Auto-Regressive Integrated Moving Average (ARIMA), alongside a broad family of Artificial Neural Network (ANN)-based architectures, including Radial Basis Function Neural Networks (RBFNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Generative Adversarial Networks (GANs).

Auto-Regressive Moving Average

The Auto-Regressive Moving Average (ARMA) technique [21, 125] represents a fundamental component of classical time-series forecasting. It expresses the value of a stochastic process at time t as a linear combination of p past observations (autoregressive component) and q past white-noise error terms (moving-average component), plus an independent noise term:

$$X_t = \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} \quad (18)$$

where $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$. Scenario generation is performed by recursively applying the fitted model to historical residuals, yielding multiple plausible future paths.

The method has been successfully employed for wind speed forecasting [56, 90, 194], solar irradiance prediction, and day-ahead electricity price scenario construction [57]. Its primary advantages are analytical tractability, low computational cost once parameters are estimated, and the availability of well-established identification procedures (e.g., Box–Jenkins methodology). However, ARMA imposes strict assumptions: the time series must be weakly stationary (constant mean, variance, and autocovariance), and the innovations are assumed Gaussian. Real-world renewable generation and traffic data frequently exhibit trends, seasonality, heteroskedasticity, and heavy tails, violating these hypotheses.

Furthermore, univariate ARMA models are inherently limited in their

ability to capture cross-correlations among spatially or temporally interdependent variables. While multivariate extensions such as Vector Autoregressive Moving Average (VARMA) models [46] have been developed to address these dependencies, their practical application is often hindered by the curse of dimensionality, which makes parameter estimation computationally intractable as the number of variables increases. Moreover, the underlying Gaussian assumption in standard VARMA frameworks remains overly restrictive, failing to represent the non-linear and asymmetric characteristics typical of complex stochastic processes.

Auto-Regressive Integrated Moving Average

To overcome the stationarity limitation, ARIMA (Auto-Regressive Integrated Moving Average) [28, 44] introduces an integration order d that applies differencing $\Delta^d X_t = (1 - B)^d X_t$ (where B is the backshift operator) until the series becomes stationary. The general ARIMA(p, d, q) model thus combines differencing with an ARMA(p, q) structure on the transformed series.
Seasonal extensions, such as the SARIMA model [20], incorporate seasonal autoregressive and moving-average components to capture the periodic dependencies inherent in stochastic processes. This framework is particularly effective for modelling the complex intra-day and weekly seasonality typical of electricity demand and traffic congestion patterns.
ARIMA-based scenario generation is performed by forecasting the differenced series and applying cumulative integration to reconstruct the paths on the original scale. Although significantly more flexible than ARMA for trending data, ARIMA still assumes linearity and Gaussian innovations, limiting its ability to reproduce extreme events or asymmetric dependence structures.

Deep Neural Network-based Methods

Recently, Deep Neural Networks (DNN) have gained significant popularity for forecasting-based scenario generation. The Long Short-Term Memory (LSTM) network, introduced by Hochreiter and Schmidhuber [80], addresses the vanishing gradient problem inherent in standard Recurrent Neural Networks (RNNs). This is achieved through a specialized architecture featuring gating mechanisms, specifically forget, input, and output gates, that selectively retain or discard information over extended time horizons. LSTMs have demonstrated superior performance in multivariate traffic flow prediction, wind power ramp forecasting, and load profiling, effectively capturing nonlinear dynamics and long-range dependencies that traditional linear models fail to identify. Furthermore,

encoder-decoder architectures enable LSTMs to generate multi-step scenario sequences conditioned on historical observation windows.

Within this class of models, Generative Adversarial Networks (GANs) have also been adapted for time-series forecasting. In these applications, GANs are trained to learn temporal patterns from historical data and to generate plausible future trajectories rather than acting as general-purpose generative models.
Time-series GAN variants have been successfully employed in renewable energy forecasting, producing wind and solar power trajectories that capture intermittency, ramps, and spatial coherence across multiple sites [29]. In financial contexts, they replicate congestion waves and price spikes that traditional models tend to underestimate [54]. Despite their strong performance, GAN-based forecasting models require large training datasets, are sensitive to hyperparameter tuning, and may suffer from mode collapse or training instability, which can limit scenario diversity or convergence reliability.

Table 3 – Summary of forecasting-based scenario generation methods.

Method	Description
ARMA/ARIMA Forecasting	**Advantages:** Strong predictive capability; computationally efficient; well-established in time-series modelling. **Limitations:** Assumes stationarity and approximate Gaussianity; limited ability to capture multivariate or highly nonlinear dependencies.
DNN, LSTM, GAN Forecasting	**Advantages:** Capable of learning nonlinear dynamics; suitable for complex and high-dimensional time series. **Limitations:** Requires large and high-quality datasets; prone to overfitting; reduced interpretability.

3.2.3 *Focusing on Sampling-based Methods*

After surveying the broader spectrum of statistical approaches for scenarios generation, we narrow our focus on the presented sampling methods. This choice reflects the thesis's scope, which does not account for temporal dependencies inherent in forecasting-based sampling approaches.
A comparative analysis of sampling-based techniques highlights important differences in their ability to reproduce complex dependence structures in multivariate settings.

Classical Monte Carlo, Markov Chain Monte Carlo, and Latin Hypercube Sampling techniques provide valuable tools for uncertainty quantification, yet they exhibit intrinsic limitations when the objective is to reproduce highly complex dependence structures across multiple variables.
MC and LHS can incorporate specific types of dependence, but they are constrained in capturing more intricate relationships such as nonlinearities, asymmetries, and tail dependencies.
MCMC, while capable of sampling from a specified true joint distribution given an appropriate likelihood and prior, poses challenges in scenario generation contexts, mainly in terms of computational inefficiency, and the difficulty of specifying flexible dependence structures a priori without relying on restrictive parametric assumptions.
Copula-based methods, by contrast, model the dependence structure, enabling accurate reconstruction of nonlinear, asymmetric, and tail-dependent relationships. The intrinsic characteristics of copulas, capturing marginals and dependence structures, make them particularly suited for generating realistic multivariate scenarios in settings where correlations play a central role.
These reasons, resumed in Table 4, highlight Copulas as the most appropriate and theoretically grounded baseline for the comparison with the scenario generation approaches proposed in this thesis.

However, the recent emergence of modern Generative AI models has opened a new line of research aimed at overcoming the computational limitations that traditional statistical methods exhibit when applied to large, high-dimensional real-world datasets.
As modern applications increasingly involve massive data volumes and a growing number of correlated variables, neural architectures capable of generating synthetic instances (such as GANs, VAEs, and related generative frameworks) have gained significant attention. The next section introduces these generative models, first outlining their core architectures and then discussing how they can be leveraged for scenario generation.

Table 4 – Comparison of statistical sampling-based methods for high-dimensional multivariate scenario generation with correlated components.

Method	Strengths	Limitations	Suitability
Monte Carlo (MC)	Simple; general-purpose	Cannot reproduce correlations unless the full joint PDF is known; poor performance in high dimensions; unable to model nonlinear or tail dependence.	Low unsuitable for high-dimensional correlated multivariate scenarios.
Markov Chain Monte Carlo (MCMC)	Can sample from complex joint distributions when fully specified; useful in Bayesian settings.	Requires an explicit joint model; computationally expensive in high dimensions; dependence arises from the chain, not from the true multivariate structure.	Low impractical for flexible dependence modelling with many correlated components.
Latin Hypercube Sampling (LHS)	Ensures full stratification of each marginal; efficient coverage of high-dimensional input space.	Assumes independence unless correlations are imposed externally; correlation control is approximate; cannot capture nonlinear or tail dependence.	Moderate to Low limited ability to reproduce realistic multivariate dependence in large dimensions.
Copula-based Sampling	Explicit modelling of dependence structure; supports nonlinear, asymmetric, and tail dependence; scalable to many variables.	Requires estimation of marginals and copula parameters; calibration may be challenging in high dimensions.	High well-suited for high-dimensional multivariate scenarios with correlated components.

3.3 GENERATIVE AI ARCHITECTURES

Generative Artificial Intelligence (Gen AI) denotes deep neural networks architectures capable of synthesizing novel data instances (e.g., text, images, audio, video, 3D structures, source code, and scientific data) by learning an approximation of the underlying data distribution $p_{data}(x)$. Unlike discriminative models that estimate conditional probabilities $p(y|x)$, generative models aim to model the joint distribution $p(x)$.
The most prominent paradigms within generative modelling are:

- **Generative Adversarial Networks (GANs) and their variants**, which combine a generator that produces synthetic samples with a discriminator that assesses their realism relative to the true data [61].
- **Variational Autoencoders (VAEs) and their extensions**, which learn a latent probabilistic representation of the data and generate new samples by decoding latent variables drawn from a prior distribution [99, 155].
- **Autoregressive models**, which model the joint distribution of the data through a sequential factorization and generate samples one element at a time, each conditioned on previously generated elements [135, 151].
- **Diffusion models**, which synthesize data by learning to reverse a gradual noising process, transforming pure noise into coherent samples through iterative denoising steps [79, 175].

In this section we provide an in-depth overview of the first two paradigms, Generative Adversarial Networks and Variational Autoencoders, as they constitute the theoretical and architectural foundation for the models proposed in this thesis.

3.3.1 *Generative Adversarial Networks (GANs)*

Generative Adversarial Networks (GANs), introduced by Goodfellow et al. (2014) [61], represent one of the most influential architectures in deep generative modelling throughout the past decade. Rather than explicitly modelling the likelihood $p(x)$ or its lower bound (as in VAEs), GANs learn to generate data through an adversarial training.
The architecture is built upon the interplay of two adversarial neural networks:

- **Generator** (G_θ): a neural network that generates a synthetic sample $G(z)$ from a low-dimensional noise vector $z \sim p_z(z)$. Its objective is to synthesize instances that closely resemble those drawn from the true data distribution, thereby challenging the discriminator.

- **Discriminator** (D_ϕ): a neural network that receives either real samples $x \sim p_{data}(x)$ or synthetic samples $G(z)$, and outputs a scalar value representing the probability that the input originates from the true data distribution rather than from the generator. Its objective is to distinguish real data from generated data, thereby providing the adversarial feedback that guides the generator's improvement.

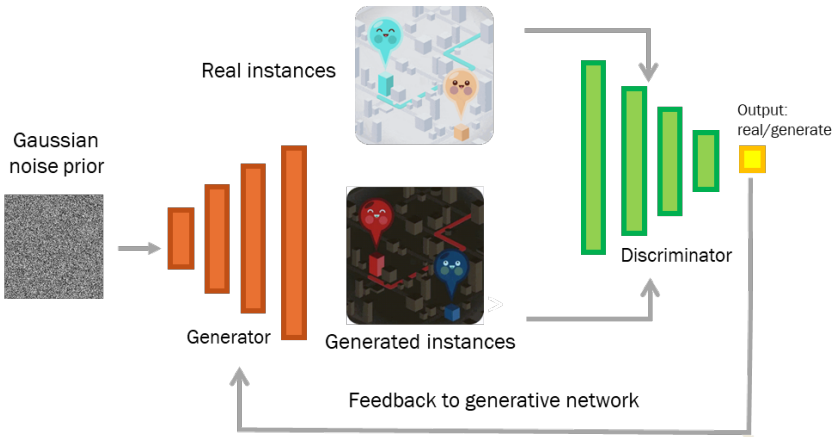


Figure 7 – Architecture of a generative adversarial network (GAN). The generator maps random noise vectors to synthetic samples, while the discriminator evaluates whether inputs are real or generated. The adversarial training process drives the generator to produce increasingly realistic outputs by minimizing its ability to be distinguished from real data.

The training of GANs is designed as a two-player minimax game from game theory, where the generator G_θ and the discriminator D_ϕ have adversarial objectives:

$$\mathcal{L}_{GAN} = \min_{G_\theta} \max_{D_\phi} V(D_\phi, G_\theta) \tag{19}$$

This adversarial training promotes mutual improvement: the generator learns to synthesize increasingly realistic samples to fool the discriminator, while the discriminator improves its ability to distinguish real data from synthetic samples. The training process proceeds by alternating updates: the discriminator is typically optimized for several steps (to maintain an adequate discriminative power relative to the current generator), followed by a single update to the generator. This asymmetric schedule mitigates training instability by preventing premature dominance of either network. At inference time, only the generator is retained for sample generation, while the discriminator is used only as an adversarial supervisor during training.

The adversarial objectives are formalized through the following function:

$$V(D_\phi, G_\theta) = \mathbb{E}_{x\sim p_{data}(x)}\left[\log D_\phi(x)\right] + \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1 - D_\phi(G_\theta(z))\right)\right] \tag{20}$$

The first term encourages the discriminator to assign high probability to real samples, whereas the second drives it to assign low probability to generated ones. From the generator's perspective, minimizing the second term drives $D_\phi(G_\theta(z))$ toward 1, enabling the generator to produce samples that are increasingly indistinguishable from real data.
Subsequent works have refined this objective and architecture:

- Conditional GAN models have been introduced to guide generation with labels or other information [128];
- architectural best practices for the use of Deep Convolutional networks (DCGAN) have been established [150];
- additional stabilization techniques have been developed, such as label smoothing and the addition of noise [162];
- Wasserstein GAN [6] uses Wasserstein Distance in the loss function and enforces Lipschitz continuity on the discriminator, leading to more stable training and meaningful loss values.

These evolutions underscore the minimax framework's flexibility and its central role in GAN success.

Among the most significant advancements in deep learning, Generative Adversarial Networks (GANs) have gained prominence for their ability to generate high-fidelity synthetic data, with applications in several domain, such as computer vision, energy systems [45], healthcare and, increasingly, transportation systems. A notable example is provided by Xiurong et al. (2024) [217], where a WGAN-based model is employed to generate wind power scenarios that accurately capture the intermittent nature of renewable resources.

3.3.2 *Variational Autoencoders (VAEs)*

Variational Autoencoders (VAEs), introduced by Kingma and Welling (2014) [99], represent a probabilistic extension of traditional AutoEncoders [18, 103], combining deep neural networks with variational Bayesian inference to enable scalable generative modelling and approximate posterior inference. At their foundation, VAEs build upon the autoencoder framework, which consists of an encoder network that compresses high-dimensional input data x into a lower-dimensional latent representation z, and a decoder network

that reconstructs the original data from this compressed code (Figure 8, left).

In autoencoders, the latent space is deterministic, with training focused solely on minimizing a reconstruction loss (e.g., mean squared error). VAEs extend this framework by introducing stochasticity into the latent space, modelling z as a probabilistic distribution, typically a multivariate Gaussian parameterized by mean μ and variance σ^2 vectors output by the encoder $q_\phi(z|x)$. Training optimizes the Evidence Lower Bound (ELBO) loss function on the marginal log-likelihood:

$$\mathcal{L}(\theta, \phi; x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z)) \tag{21}$$

where the first term enforces reconstruction fidelity through the decoder $p_\theta(x|z)$, and the second term regularizes the posterior to align with a simple prior $p(z)$ (e.g., $\mathcal{N}(0, I_d)$). To enable differentiable optimization via backpropagation, the reparameterization trick is employed, sampling $z = \mu + \sigma \odot \epsilon$ where $\epsilon \sim \mathcal{N}(0, I_d)$, transforming stochastic nodes into deterministic functions of parameters and noise. This formulation not only facilitates generation from the prior but also promotes a smooth, structured latent manifold, making VAEs particularly suited for applications where capturing complex spatiotemporal dependencies is essential, like traffic scenario synthesis in this thesis.

VAEs assume a simple prior $p(z) = \mathcal{N}(0, I_d)$, encouraging a smooth, interpolable latent manifold. Unlike GANs, VAEs provide both generative capabilities (sampling from the prior and decoding) and inference (encoding inputs to latents), making them ideal for semi-supervised learning and downstream tasks.

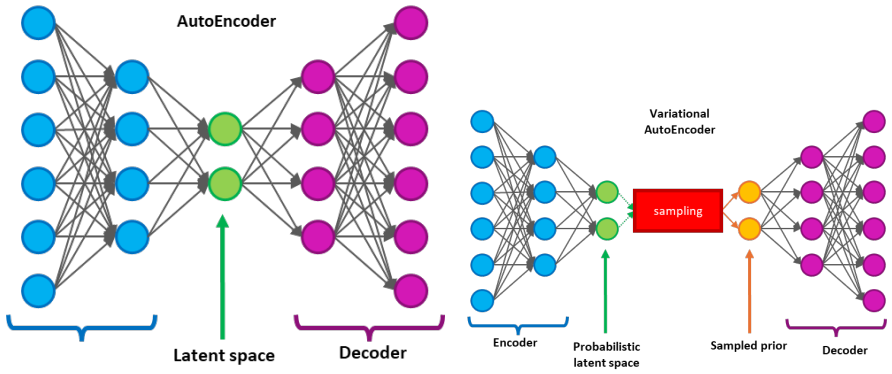


Figure 8 – Architectures of autoencoder (left) and variational autoencoder (right). In the VAE, the encoder approximates the posterior $q(z|x)$, while the decoder reconstructs from latent samples. The reparameterization trick enables backpropagation through stochastic nodes.

Variational Inference and Loss Functions

Several important extensions to the standard ELBO loss function have been proposed to address well-known limitations such as posterior collapse, blurry generations, and poor disentanglement:

- **β-VAE** [77] introduces a trade-off parameter β in the objective

$$\mathcal{L}_\beta = \mathbb{E}_{q_\phi}[\log p_\theta(x|z)] - \beta \, D_{KL}(q_\phi \| p) \tag{22}$$

  thereby strengthening the regularization effect and promoting more disentangled latent representations, although often at the expense of reconstruction quality.

- **Annealed KL weighting** [19] gradually changes its value during training. This strategy mitigates posterior collapse by allowing the model to first learn useful representations before enforcing strong alignment with the prior.

- **MMD-VAE** [218] replaces or augments the KL term with alternative divergence measures, most notably the Maximum Mean Discrepancy (MMD). These approaches are particularly effective in preventing collapse when the decoder is highly expressive or overparameterized. Likelihood-free variants (e.g., [202]) employ contrastive, score-matching, or adversarial objectives in place of explicit likelihood terms, making VAEs applicable to simulators and other settings where the reconstruction likelihood is intractable while still preserving the variational optimization framework.

These methodological variants significantly enhance the robustness of VAEs when applied to high-dimensional, structured data, alleviating issues related to mode-covering behavior, posterior collapse, and limited expressiveness in complex domains.
The evolution of VAEs has been marked by increasing sophistication in handling complex data modalities, scaling to deeper hierarchies, and integration with other paradigms.

By focusing on foundational and recent advancements, we can present an overview of key contributions to VAE architectures.

- **Kingma & Welling, "Auto-Encoding Variational Bayes" - (ICLR 2014) [99]** Seminal work introducing the VAE framework, ELBO derivation, and reparameterization trick. Demonstrated on MNIST and Frey faces, establishing VAEs as a scalable alternative to RBMs and wake-sleep algorithms.

- **Rezende et al., "Stochastic Backpropagation and Approximate Inference in Deep Generative Models" - (ICML 2014) [155]** Independent concurrent development emphasizing hierarchical latents and importance sampling for tighter bounds.

- **Higgins et al., "β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework" - (ICLR 2017)** [77] Introduced the β-weighted ELBO for disentangled representations, enabling interpretable factors (e.g., rotation, scale) in latent space.
- **Van den Oord et al., "Neural Discrete Representation Learning" - (NeurIPS2016)** [135] Replaced continuous latents with vector-quantized discrete codes, enabling autoregressive priors (e.g., PixelCNN) for sharper image generation. Basis for DALL·E and later multimodal models.
- **Razavi et al., "Generating Diverse High-Fidelity Images with VQ-VAE-2"** [154] Scaled VQ-VAE with hierarchical codes and large-scale training, achieving class-conditional ImageNet synthesis rivaling early GANs.
- **Vahdat & Kautz, "NVAE: A Deep Hierarchical Variational Autoencoder" - (NeurIPS 2020)** [182] Introduced normalizing flows in posteriors and residuals in hierarchies, yielding state-of-the-art likelihoods on CIFAR-10 (2.91 bpd) without autoregression.
- **Child, "Very Deep VAEs Generalize Autoregressive Models and Can Outperform Them on Images" - (ICLR 2021)** [30] Demonstrated that ultra-deep (up to 600 layers) hierarchical VAEs surpass PixelRNN/CNN in density estimation while maintaining fast sampling.
- **Wei et al., "Recent Advances in Variational Autoencoders With Representation Learning for Biomedical Informatics: A Survey" - (IEEE Access 2021)** [196] Comprehensive review of VAEs in healthcare, including anomaly detection in MRI and drug discovery, highlighting domain-specific adaptations like sparse priors.

VAE has emerged as a powerful and versatile model architecture across several application fields, including road network modelling [63, 216], molecular structures [36], social network behaviors [112], and climate synthesis [191]. In addiction, VAEs have been applied to structured and streaming signals, including speech enhancement via recurrent architectures [110], and anomaly detection in traffic networks and biomedical signals [92].
In energy production forecasting, VAEs have been used to predict photovoltaic solar power production, as seen in [38], and to forecast energy productivity, as proposed in [97]. Shifting to the agriculture domain, VAE-based models have been used for desertification forecasting [212] and to predict the rice crop yield [89]. An application to the medicine field is shown in [64] and [37], where VAEs have been used to forecast the spread of infectious diseases; conditional and disentangled variants support medical image synthesis for data

augmentation and pathology-specific generation [27].
In imaging, VAEs have been applied to compression, synthesis, restoration, lossy compression of 3D point clouds [149], and forensic analysis [8, 16, 69].

These VAE-based approaches demonstrate the architecture's adaptability to irregular domains, enabling principled trade-offs between compression, inference accuracy, and computational cost [148].

Recent developments in VAE architectures have led to their growing adoption in transportation domains, where they are increasingly employed to model complex urban mobility patterns. Applications include bicycle and pedestrian traffic forecasting [67], as well as road traffic prediction using real-world datasets [15], where VAEs learn dependencies from traffic flows. These contributions further underscore the versatility of VAEs and their growing relevance in forecasting domains.

Several studies have introduced VAE-based models for sample generation tasks, focusing on replicating the distributions of real data. Examples of applications where VAEs are used as generative models can be found in the field of imaging, such as [86, 189], audio [32], and text [195].

Since 2020, VAEs have evolved well beyond their original role as standalone generative models, emerging as essential building blocks within increasingly sophisticated hybrid architectures. A significant example is found in Latent Diffusion Models (LDMs), with Stable Diffusion [159] serving as one of the most influential realizations of this paradigm. In this architecture, a pretrained VAE compresses high-resolution images into a compact latent space, and the diffusion model generates samples efficiently within this reduced representation, combining structured encodings with high-fidelity synthesis. This synergy has positioned LDMs as a central approach in text-to-image generation.

3.4 GENAI MODELS FOR TRANSPORTATION

In the transportation research field, generative models for scenario generation are increasingly central for addressing diverse operational and analytical tasks, including safety validation of autonomous systems, large-scale traffic simulations, network optimization under uncertainty, and policy assessment in complex urban environments. These models generate traffic instances from learned data distributions, enabling the synthesis of realistic scenarios.
The generative models presented in this section operate at different

Table 5 – Operational scales in transportation scenario generation.

Scale	Data Generated	Application Context
Micro	Trajectories, vehicle actions	AV testing, intersection control
Micro–meso	Scene maps + agent paths	Simulation training data
Meso	Link speeds, flows, density	Signal optimization, monitoring
Meso–macro	Network flows, OD sub-matrices	Urban logistics assignment
Macro	OD matrices, modal capacity	Regional planning policy

levels of spatial aggregation.

Following the hierarchical framework established in transportation modelling literature [82], we categorize the generation of scenarios according to the following operational scales, distinguished by their spatial granularity:

- **Micro**: Trajectories and vehicle-level actions within localized scenes.
- **Micro–meso**: Full driving scenes with road geometry and multi-agent dynamics over short network segments.
- **Meso**: Link-level average speeds, flows, and densities across urban corridors or sensor networks.
- **Meso–macro**: Network-wide flow conservation, queue propagation, and aggregated demand in city-scale systems.
- **Macro**: Origin-destination matrices, modal demand, and infrastructure capacity at regional or multimodal level.

A summary of these scales, along with representative examples, is reported in Table 5.

A prominent example at the micro-scale is TrafficGen [52], an autoregressive generative model that sequentially generates vehicle placements and complete trajectories (positions and velocities over time). Trained on fragmented real-world driving data from the highD [102] and InterAction [213] datasets, it encodes the current traffic context and decodes initial states followed by long-term dynamics, producing diverse scenarios suitable for augmenting simulations and

training reinforcement learning agents for safer autonomous driving. Similarly, DriveSceneGen [180] introduces a two-stage pipeline using latent diffusion to generate entire driving scenes from scratch, including static lane maps and dynamic multi-agent trajectories, operating at the micro-to-meso scale. By learning the joint distribution of map elements and agent behaviors in a vectorized representation, it addresses data scarcity in autonomous driving datasets and produces high-fidelity scenes that reduce domain gaps between simulation and reality.
Among diffusion-based approaches, Versatile Behavior Diffusion (VBD) [88] generates future multi-agent trajectories and contextual actions (e.g., braking, overtaking) at the micro-scale, focusing on closed-loop, scene-consistent interactions conditioned on high-definition maps. It excels in synthesizing long-tail safety-critical scenarios via inference-time editing with game-theoretic guidance, outperforming baselines on the Waymo Sim Agents Benchmark.
ChatTraffic [214] proposes a generative model that synthetizes future traffic states (including segment-level flows and average speeds) at the meso scale. Using a generative model enhanced with Graph Convolutional Networks to capture spatial correlations, it translates natural language prompts (e.g., congestion due to lane closure) into realistic traffic evolutions under abnormal conditions, preserving joint dependencies over long horizons.

The deployment of GAN-based models has facilitated significant advancements in transportation management. GANs enable the synthesis of realistic traffic patterns that faithfully reflect multi-scale spatial dependencies and the underlying topological constraints of the road network [130, 192]: several recent studies have demonstrated the versatility of these architectures in transportation research.
Demetriou et al. [41], for instance, employed a GAN-based model to generate vehicle trajectories for the safety verification of autonomous vehicles, capturing subtle variations in driving behavior and trajectory curvature to simulate critical edge-case scenarios.
Similarly, Chao et al. [200] introduced a GAN framework tailored to spatio-temporal traffic flow generation, successfully reproducing congestion dynamics and flow transitions across urban networks.
Other works have adapted GANs to generate spatially coherent driving behaviors, such as lane-change maneuvers on highways [106].
Generative Adversarial Networks are also used in Precog [156] to generate interactive future trajectories of vehicles at the micro-scale. SocialGAN [65] is a GAN-based generative model that produces realistic pedestrian paths at the micro-scale. Other recent surveys [204] highlight GANs and VAEs for generating synthetic traffic flows and congestion patterns at the meso scale, enabling resilience testing

under rare events like extreme weather.

Hybrid designs combining adversarial training with topology-aware and physics-based loss function have been proposed in [130] to improve mode coverage and enforce physical constraints in the generated instances, such as flow conservation on links.
Furthermore, conditional variants, such as Conditional GANs (cGANs) and Auxiliary Classifier GANs (AC-GANs), allow for the controlled generation of scenarios based on specific network attributes, including link type and congestion levels, while preserving the joint spatial structure of the data [134, 192].
SceneDiffuser [87] integrates denoising diffusion models with scene graphs to generate physically plausible multi-agent trajectories and interactions at the micro-scale, conditioned on historical maps for multimodal outputs.
WcDT [205] advances this by generating global action distributions and long-horizon coherent trajectories at the micro-to-meso scale, offering a "world-centric" view that outperforms agent-centric baselines on real-world-datasets.
GraphMats [197] uses diffusion model with Graph Neural Networks to generate link flows and queue lengths at the meso-to-macro scale, enforcing flow conservation across the network.
PhysDiff [210] incorporates diffusion models with physics-based loss functions to generate multi-agent trajectories at the micro-scale. However, this is used to create realistic human motion trajectories at the micro scale, illustrating an application in human movement rather than traffic simulation itself.
These hybrid approaches are ideal for applications such as assessing multimodal resilience [119], which generates scenarios of demand and capacity across different transport modes at the macro scale.
Table 6 provides a compact overview of the main generative models for transportation presented in this section.

While these methods successfully capture diverse traffic dynamics at multiple scales, they do not specifically target the generation of correlated multivariate scenarios of link speeds, which are critical for decision-making under uncertainty in urban transportation systems.
These scenarios support a wide range of transportation analyses, enabling realistic representations of traffic dynamics, congestion patterns, and variability across the network.
By preserving statistical dependencies induced by topology, scenarios enable the design of solutions that are both resilient and robust against worst-case realizations.

The next part (Part II) formalizes the problem of traffic speed scenario generation and the proposed models designed to generate re-

alistic, statistically coherent, and physically consistent traffic speed scenario generation.

Model	Main Objective	Generated Output	Scale	Key Feature
TrafficGen [52]	Autoregressively complete realistic traffic scenes	Trajectories, behaviors	Micro	Sequential decoding
DriveSceneGen [180]	Generate full driving scenes from scratch	Map + agents	Micro–meso	Latent vectorized diffusion
VBD [88]	Closed-loop multi-agent synthesis with editing	Actions, trajectories	Micro	Game-theoretic guidance
ChatTraffic [214]	Text-to-traffic: future states from language prompts	Link flows, speeds	Meso	GCN-augmented diffusion
Precog [156]	Interactive future trajectory forecasting	Ego + neighbors	Micro	Social-context GAN
SocialGAN [65]	Socially compliant multi-agent path generation	Multi-agent paths	Micro	Interaction pooling
SceneDiffuser [87]	Physically plausible scenes via DDPM + scene graphs	Trajectories, physics	Micro	Scene-graph conditioning
TSDiT / WcDT [205]	Global traffic evolution with transformer diffusion	Long-horizon trajectories	Micro–meso	World-centric view
PhysDiff [210]	Physics-informed diffusion for valid trajectories	Kinematically feasible trajectories	Micro	Collision avoidance

Table 6 – Summary of main generative models for transportation.

Part II

PROBLEM FORMULATION AND METHODOLOGICAL DESIGN

This part introduces the research question and the modelling strategies designed to address it.

Chapter 4 formalizes the central problem addressed in this thesis by introducing the mathematical framework for the representation and analysis of traffic speed observations across a road network. The proposed formulation serves as the foundation for generating realistic traffic scenarios, enabling the modelling of complex spatial dependencies and variability inherent in urban mobility systems.

In **Chapter 5** a spectrum of generative architectures is proposed and comparatively analyzed: the baseline Naive-VAE, its adversarial extension (Naive-VAE+GAN), a Copula-VAE that relaxes the independence assumption in the latent space, a graph-based Naive-VAE that integrates topological structures, and advanced formulations such as the dual-latent VAE, the β-VAE with dynamic scheduling, and β-VAE with non-causal Transformer mechanisms. These models progressively enhance the generative process, achieving a principled balance between model structure and the realism of the generated data.

4

PROBLEM FORMULATION FOR TRAFFIC SCENARIO GENERATION

The objective of this work is to design generative models capable of synthesizing road traffic scenarios that preserve the spatial correlations observed in real road networks. By learning the joint statistical distribution of traffic speeds from empirical data, the proposed approaches aim to generate novel yet realistic traffic scenarios for use in simulation, planning, and decision-support in transportation systems.

MOTIVATION AND OBJECTIVES

modelling and generating realistic road traffic scenarios is a key challenge in the field of transportation systems. The goal is to generate coherent sets of speed values that could realistically occur across a road network. Accounting for statistical dependencies among road segments is of crucial importance in the modelling and generation of realistic traffic scenarios.

In this context, spatial correlations refer to the statistical dependencies between the speeds on different road segments: traffic conditions on one link tend to influence, and be influenced by, those on adjacent or connected links, due to network topology, flow dynamics, and routing patterns. Preserving these correlations is essential to ensure that generated scenarios reflect the joint behavior observed in real traffic systems. Such capability is essential for urban planning, where scenario simulation supports infrastructure design and evaluation; it is also crucial in traffic management issues, because it enables the optimization of control strategies; furthermore, it can also be fundamental in policy and innovation assessment, where the impact of extraordinary events, new regulations, or emerging technologies such as autonomous vehicles can be tested. By capturing the underlying joint distribution of speeds and their spatial dependencies, these models provide a foundation for decision-making aimed at improving mobility efficiency, reducing congestion, and mitigating environmental impacts.

In the following sections, we present a mathematical formalization of the problem of generating realistic multivariate data scenarios in a general setting. The goal is to define models that capture the essential statistical dependencies among variables, and that can later be specialized and applied to specific domains.

Subsequently, we will present how this general formalization can be

applied in the domain of road traffic scenario generation, where the variables correspond to traffic speed measurements across a road network. This approach allows us to leverage the theoretical framework while addressing the particular characteristics and constraints of urban traffic networks.

4.1 GENERAL PROBLEM FORMALIZATION

A complex system can be modelled through a multivariate random variable

$$X = (X^1, X^2, \ldots, X^n, \ldots, X^N) \quad n = 1, \ldots, N \tag{23}$$

whose components X^n represent the marginal variables of the considered multivariate. This formulation captures their joint variability and interdependence, which is essential when analysing systems where the behaviour of each component is correlated with the others. The behaviour of the system can be inferred from a finite set of T observations in which all N variables are recorded simultaneously.
Each observation corresponds to a realization of the multivariate random variable X and is represented as an N-dimensional vector

$$s_t = (x_t^1, x_t^2, \ldots, x_t^N) \in \mathbb{R}^N \quad t = 1, \ldots, T \tag{24}$$

where x_t^n denotes the observed value of the marginal variable X^n in the t^{th} observation. These realizations provide empirical evidence of the underlying joint distribution and form the basis for modelling dependencies, correlations, and variability patterns.
Each vector s_t, with $t = 1, \ldots, T$, captures the state of all marginal variables at the same timestamp. Collecting all observations, the dataset S can be written as

$$S = \{s_1, s_2, \ldots, s_t, \ldots, s_T\} \qquad s_t \in \mathbb{R}^N \tag{25}$$

Table 7 summarizes the notation introduced in this section.

In many practical settings, the true joint distribution of X is unknown and cannot be directly derived analytically. However, the empirical realizations collected in S make it possible to approximate the underlying distribution.

This enables the construction of models capable of generating new, realistic samples that preserve the statistical properties and dependencies observed in the data.

To achieve this, we model X using a generative neural network M, trained on the dataset S, which learns the joint distribution of the variables and can produce synthetic observations that reflect their variability and interdependence.

Symbol	Description
X	Multivariate random variable
N	Number of variables (marginals of X)
T	Number of observations (timestamps)
$n \in \{1, \dots, N\}$	Index of a marginal variable
X^n	n^{th} marginal variable of X
$t \in \{1, \dots, T\}$	Timestamp index
x_t^n	Measurement of variable n at timestamp t
s_t	Observation vector at timestamp t
S	Dataset (collection of all s_t)

Table 7 – Notation used throughout the General Problem Formalization

4.2 ROAD TRAFFIC SCENARIO REPRESENTATION

Building on this general framework, we specialize the problem to the road traffic domain, where the variables of interest represent vehicle speeds measured across the streets of an urban area. To describe the structure of the road system, we represent the physical infrastructure as a road network. Formally, a road network can be represented by a directed graph $G = (V, E)$, where V is the set of nodes that correspond to road intersections, and $E \subseteq V \times V$ is the set of directed edges. Each edge $e_{i,j} = (v_i, v_j) \in E$ denotes a road from intersection $v_i \in V$ to intersection $v_j \in V$.

Line Graph Representation

Since our objective is to model the multivariate joint distribution of traffic speeds and their spatial dependencies, the road network representation must align with the way the variables are defined. In our setting, each marginal variable X^n corresponds to the set of traffic speed measurements collected on road segment n.
As Graph Convolutional Networks (GCNs) propagate information through node-to-node message passing, they require features to be defined at the nodes. To avoid a mismatch between the required data structure and our problem formulation, it is therefore more appropriate to represent each road segment as a node in the graph. This choice places traffic speeds at the node level, aligning the data representation with the GCN framework.
A natural way to obtain such a representation is through the Line Graph construction.
Unlike the primal graph, where nodes represent intersections and

edges represent road segments, Line Graph [11, 12] reverses this perspective:

- each node corresponds to a road segment,
- an edge connects two nodes if the corresponding segments share a common intersection in the primal graph.

This representation shifts the analytical focus from intersections to segments, enabling the study of transitions between roads, such as congestion propagation, flow dynamics, and route optimization, at the level where traffic measurements are actually collected.
Recent works have demonstrated the effectiveness of this approach. Jepsen and Jensen [93] adopted this structure to model complex road networks, highlighting its potential for topological analysis and traffic prediction. Their framework was later formalized and extended by Zhang and Long [215], who proved its suitability for integration with graph-based machine learning techniques, particularly Graph Neural Networks (GNNs). Line Graph representation allows traffic-related attributes, such as average speed, road category, or congestion level, to be directly associated with nodes, making it possible to learn expressive node embeddings that capture both the intrinsic properties of each segment and its relational context within the network.

Undirected case

Formally, let the original road network be represented as a primal indirected graph $G = (V, E)$, where V denotes intersections and E denotes road segments. To construct the line graph $G^* = (V^*, E^*)$, we associate each node $v^* \in V^*$ with a road segment $e \in E$ of the primal graph. Two nodes v_i^* and v_j^* are connected in G^* if the corresponding segments e_i and e_j share an endpoint in G.

The construction relies on the vertex–edge incidence matrix of the primal graph, $B \in \{0,1\}^{|V|\times|E|}$, where $B_{ve} = 1$ if intersection $v \in V$ is incident to segment $e \in E$. This matrix encodes how segments connect to intersections in G, and this information determines the adjacency structure of the line graph G^*.

The adjacency matrix $A^* \in \{0,1\}^{|E|\times|E|}$ of the line graph G^* is defined as

$$A^* = B^\top B - 2I_{|E|} \tag{26}$$

where $(B^\top B)_{ij}$ denotes the number of vertices shared by edges e_i and e_j: it equals 1 if the two edges share exactly one vertex, and 0 otherwise. In the case $i = j$, the incidence vector of e_i contains exactly two entries equal to 1 (one for each endpoint of the edge), so

its dot product with itself is 2. Subtracting $2I_{|E|}$ removes these diagonal terms, thereby eliminating self–loops and yielding the proper adjacency matrix of the line graph.

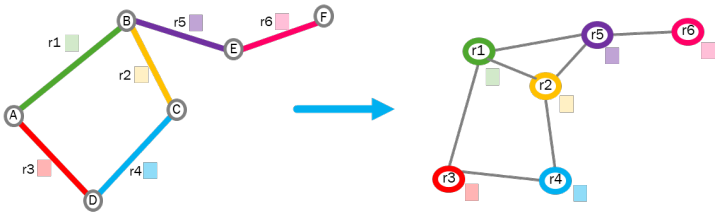


Figure 9 – Transformation from a primal graph to its dual representation. In the primal graph $G = (V, E)$, intersections are modeled as nodes and road segments as edges. In the line graph $G^* = (V^*, E^*)$, each road segment becomes a node, and an edge is created between two nodes if the corresponding segments share a common intersection in the primal graph. This structure enables segment-level analysis and supports node-based learning techniques.

Directed case

In the case of a directed primal graph, as in a real road network, each edge is represented by an ordered pair of vertices (v_i, v_j), indicating the permitted direction of travel from intersection v_i to intersection v_j. The resulting edge set $E \subseteq V \times V$ may contain both (v_i, v_j) and (v_j, v_i) to model bidirectional streets, or only one of them to represent one-way streets.
Each directed edge $e \in E$ is associated with a source node, denoted by e^-, and a target node, denoted by e^+.

For each edge $e \in E$, we define the incidence vectors

$$b^-_{\cdot,e} \in B^-, \qquad b^+_{\cdot,e} \in B^+ \tag{27}$$

For each vertex $v \in V$, we set

$$b^-_{v,e} = \begin{cases} 1 & \text{if } v = e^- \\ 0 & \text{otherwise} \end{cases} \qquad b^+_{v,e} = \begin{cases} 1 & \text{if } v = e^+ \\ 0 & \text{otherwise} \end{cases} \tag{28}$$

Following [2], the adjacency matrix of the directed line graph is:

$$A^* = (B^+)^\top B^- \tag{29}$$

which ensures that $A^*_{ij} = 1$ if and only if the target node of e_i coincides with the source node of e_j, i.e., if it is possible to traverse e_j immediately after e_i following the directionality of the primal graph.

Sub-Graph

Real-world road networks are inherently connected and often comprise a very large number of streets, making it impractical to process the entire network in traffic scenario generation. To focus the analysis while preserving essential spatial and topological properties, we extract a connected subgraph $G_N \subseteq G$ consisting of N adjacent road segments. Subgraphs are essential for reducing the dimensionality of the problem and for assessing the scalability of the proposed methods. The resulting primal graph preserves the structural coherence of the original network and captures realistic traffic dependencies.
In the line graph representation, this restriction is mirrored: the subgraph G_N induces a reduced line graph $G_N^* \subseteq G^*$, whose vertices correspond to the selected N segments and whose edges represent the admissible transitions between them. Both G_N and G_N^* thus describe exactly the same portion of the network, enabling consistent segment-level analysis in either domain.

4.3 TRAFFIC-SPECIFIC PROBLEM FORMALIZATION

Building on the general problem formalization introduced in Section 4.1, we specialize it to the context of road traffic. In our setting, each marginal variable corresponds to a road segment, and the multivariate observations represent the traffic speeds simultaneously recorded across the selected portion of the network.
Each node of the line graph G_N^* is therefore associated with the speed measurements collected on its corresponding segment.
Formally, let

$$X^n = (x_1^n, x_2^n, \dots, x_t^n, \dots, x_T^n) \in \mathbb{R}^T \qquad n = 1, \dots, N \tag{30}$$

denote the time series of traffic speed measurements recorded on road segment n at timestamps $t = 1, \dots, T$. At each timestamp t, the global traffic state over the selected portion of the road network is represented by the N-dimensional vector

$$s_t = (x_t^1, x_t^2, \dots, x_t^n, \dots, x_t^N) \in \mathbb{R}^N \qquad t = 1, \dots, T \tag{31}$$

where x_t^n denotes the speed recorded on segment n at timestamp t. Each vector s_t is therefore a realization of the multivariate random variable $X = (X^1, X^2, \dots, X^N)$.
Collecting all observations, the dataset is defined as

$$S = \{s_1, s_2, \dots, s_t, \dots, s_T\} \subset \mathbb{R}^{N \times T} \tag{32}$$

which provides an empirical realization of the unknown joint distribution of X.

Figure 10 provides a visual illustration of this data structure, showing both the road network topology and its corresponding multivariate representation.

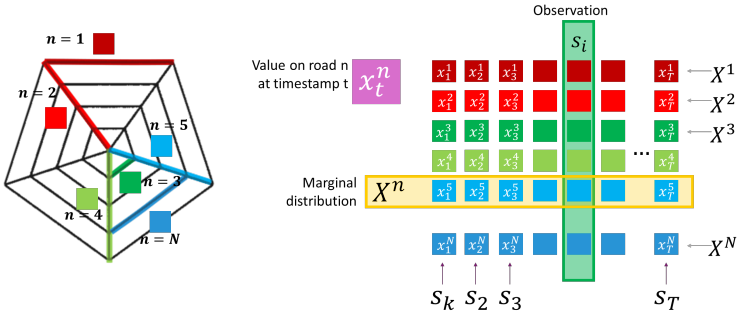


Figure 10 – Visual representation of multivariate traffic observations. Left: road network topology, where each segment is associated with a recorded speed value at a given timestamp, constituting an observation vector. Right: structured data representation, where each row X^n corresponds to a variable (i.e., a road segment), and each column S_t represents a full observation across all segments at timestamp t.

4.4 APPROXIMATING THE JOINT DISTRIBUTION FOR REALISTIC SCENARIO GENERATION

Urban traffic systems are inherently complex and influenced by a wide range of factors, including infrastructure design, driver behavior, temporal fluctuations, and external events such as accidents or weather conditions. These factors interact in non-linear and often unpredictable ways, making it impractical to assume a parametric form (e.g., Gaussian or exponential) for the underlying distribution of X. In this context, approximating the joint distribution becomes a necessary step to enable the generation of realistic and plausible synthetic traffic scenarios. This approach allows us to generate traffic conditions that could realistically occur, even if they were not explicitly recorded in the original dataset.

Generative neural networks, such as Generative Adversarial Networks (GANs), and Variational Autoencoders (VAEs), offer a powerful framework for this task. These models, once trained on dataset S, can generate synthetic traffic scenarios that emulate the behavior of traffic speeds across the road network.
The approximation of the joint distribution X is a foundational step toward building a flexible and scalable framework for traffic scenario generation. It enables the synthesis of realistic data that supports simulation, analysis, and decision-making in intelligent transportation systems.

Building on this formulation, the next section introduces the proposed modelling approaches, describing how they approximate the joint distribution X and leverage it to generate realistic traffic scenarios consistent with the observed spatial correlations.

5

PROPOSED MODELS FOR TRAFFIC SCENARIO GENERATION

The following chapter outlines the architecture developed to generate realistic synthetic road traffic scenarios. The proposed models are designed to learn the multivariate joint distribution X of traffic speed values across a road network from the dataset S, and to generate synthetic road traffic scenarios that preserve the spatial correlations intrinsic to the road network topology. This section outlines the guiding principles and design choices, before providing a detailed account of the architectures and training procedures.

5.1 NAIVE VARIATIONAL AUTOENCODER MODELS

The first model introduced within the proposed framework is a NaiveVAE designed to capture the underlying multivariate distribution (Figure 11).

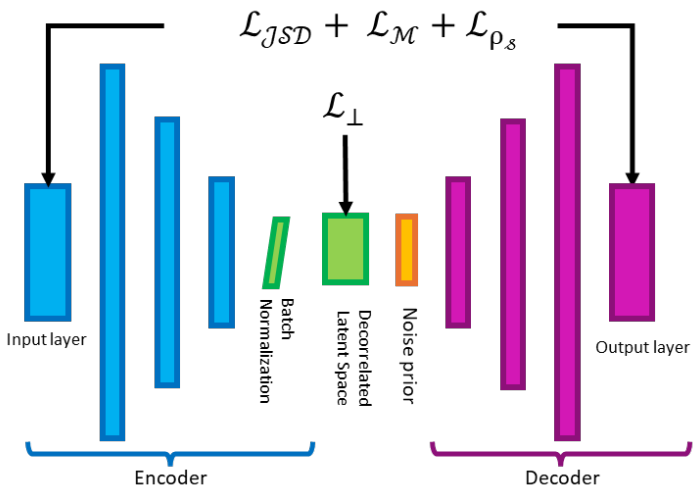


Figure 11 – NaiveVAE baseline architecture. The encoder compresses the input instances into a latent vector directly regularized towards an isotropic Gaussian prior. The decoder generates synthetic scenarios from latent samples drawn from gaussian noise.

Unlike a deterministic AutoEncoder, which maps each input to a fixed latent code, the VAE learns a probabilistic latent representation. This enables stochastic sampling and supports the generation of realistic samples that reflect the learned distribution. Fully connected neural networks are used for both encoder and decoder, and the hidden layers adopt the hyperbolic tangent (TanH) activation function

to produce smooth non-linear transformations and bounded outputs in $(-1, 1)$. Through a sequence of hidden layers with decreasing dimensionality, the encoder network compresses the input into a latent vector $z \in \mathbb{R}^d$.

Through the loss function, the model imposes a standard Gaussian distribution $\mathcal{N}(0, 1)$ on the components of the latent vector and encourages them to be mutually independent. Following the regularization of the latent space to match a Gaussian distribution, the decoder receives a prior sampled from $\mathcal{N}(0, I_d)$, where each component is drawn independently. The decoder employs a sequence of hidden layers whose dimensionality increases progressively, mirroring the compression performed by the encoder.

In the generative phase, the naiveVAE decoder is used as the generator. By feeding it with latent vectors sampled from the prior distribution (in this setting Gaussian Noise), it synthesizes new traffic speed scenarios.

To formalize the model architecture, let $x_{in} \in S$ be an input instance. The encoder E_Φ maps x_{in} into a latent vector $z = E_\Phi(x_{in}) \in \mathbb{R}^d$, constrained to lie in a normalized latent space with independent components drawn from $\mathcal{N}(0, I_d)$.

A latent vector z sampled from this Gaussian prior is passed to the decoder D_θ, which generates a new synthetic traffic scenario $x_{gen} = D_\theta(z)$. This process can be formally expressed as:

$$x_{in} \in S \xrightarrow{E_\Phi} z \sim \mathcal{N}(0, I_d) \xrightarrow{D_\theta} x_{gen} \tag{33}$$

The effectiveness of this architecture relies on the design of the loss function, which determines how the latent space is shaped and how faithfully the input multivariated distribution is captured. Accordingly, the proposed loss is constructed to fulfill two key objectives:

(i) Preserve the multivariate structure of the input data;

(ii) Enforce an isotropic latent space with independent components drawn from a standard Gaussian distribution $\mathcal{N}(0, I_d)$.

To simultaneously address these two objectives, the loss function must incorporate dedicated terms that operate not on individual instances, but across batches of instances. This enables the model to capture statistical relationships and enforce distributional constraints that emerge only at the population level patterns which cannot be inferred from isolated samples. This form of batch-wise optimization is essential for preserving the global structure of the multivariate distribution.

To implement this strategy, the proposed loss function $\mathcal{L}_{VAE}$ combines multiple components, each designed to promote distributional fidelity, latent regularity, and statistical independence. Its full formulation is given by:

$$\mathcal{L}_{VAE} = \alpha \cdot \mathcal{L}_{JSD} + \beta \cdot \mathcal{L}_M + \gamma \cdot \mathcal{L}_{\rho_s} + \delta \cdot \mathcal{L}_\perp \tag{34}$$

In our proposed formulation, each term of Equation 34 corresponds to a distinct statistical criterion: $\mathcal{L}_{JSD}$ employs the Jensen–Shannon Divergence to compare the shape of the input and reconstructed distributions, $\mathcal{L}_M$ penalizes deviations in the median of each variable, and $\mathcal{L}_{\rho_s}$ uses the Spearman rank correlation to preserve monotonic dependencies, thus addressing Goal (i). The term $\mathcal{L}_\perp$, on the other hand, promotes orthogonality among latent dimensions, thereby encouraging statistical independence and isotropy in the latent space, Goal (ii).
The scalar coefficients α, β, γ, and δ the respective loss terms and are optimized to balance their relative influence.

JENSEN–SHANNON DIVERGENCE LOSS The term $\mathcal{L}_{JSD}$ (presented in Equation 35) is included to force the model to generate a reconstructed joint distribution in which each constituent marginal distribution closely aligns with the corresponding input one. To quantify this discrepancy, we employ the Jensen–Shannon Divergence (JSD) [114], a metric quantifying the similarity between two distributions. We calculate JSD to assess the dissimilarity between the n^{th} input marginal variable (X^n_{in}) from the encoder and its corresponding generated counterpart (X^n_{gen}). We subsequently aggregate these measurements across all marginal variables ($n = 1, \dots, N$). The $\mathcal{L}_{JSD}$ is calculated as follows:

$$\mathcal{L}_{JSD} = \sum_{n=1}^{N} \sqrt{\frac{1}{2} \cdot KL(X^n_{gen}, m_n) + \frac{1}{2} \cdot KL(X^n_{in}, m_n)} \tag{35}$$

where $m_n = \frac{1}{2} \cdot (X^n_{in} + X^n_{gen})$ and KL represents the Kullback-Leibler divergence [107].

MEDIAN PRESERVATION LOSS The loss term $\mathcal{L}_M$ measures the aggregated distance between the medians of each input marginal variable ($\widetilde{X}^n_{in}$) and its generated counterpart ($\widetilde{X}^n_{gen}$).

$$\mathcal{L}_M = \sum_{n=1}^{N} |\widetilde{X}^n_{in} - \widetilde{X}^n_{gen}| \tag{36}$$

where $n = 1, \dots, N$ indexes the marginal variables.

SPEARMAN CORRELATION LOSS The loss term $\mathcal{L}_{\rho_s}$ is introduced to penalize the dissimilarity between the input and reconstructed variables concerning their correlation, as quantified by Spearman correlation. Similar to Pearson correlation [143], Spearman correlation (ρ_s) [166, 177] assesses the rank correlation of two random variables. Nonetheless, whereas Pearson correlation evaluates

linear relationships, Spearman correlation considers both linear and non-linear relationships, as well as monotonic relationships.

The Spearman correlation is calculated as follows:

$$\rho_s(X, Y) = \frac{cov(rg_X, rg_Y)}{\sigma_{rg_X}\sigma_{rg_Y}} \tag{37}$$

Here, rg_X and rg_Y represent the ranks associated with the random variables X and Y.

In our approach, we compute the Spearman correlation matrices for the input and generated batches, and define the loss term $\mathcal{L}_{\rho_s}$ as the aggregated distance between these two matrices:

$$\mathcal{L}_{\rho_s} = \sum_{n=1}^{N}\sum_{l=1}^{N} \left|\rho_s(X_{in}^n, X_{in}^l) - \rho_s(X_{gen}^n, X_{gen}^l)\right| \tag{38}$$

where, $\rho_s(X_{in}^n, X_{in}^l)$ represents the Spearman correlation calculated for the n^{th} and l^{th} input marginal variables while $\rho_s(X_{gen}^n, X_{gen}^l)$ is the Spearman correlation for the generated counterparts.

LATENT SPACE DECORRELATION LOSS Finally, to drive the training process to generate an uncorrelated latent space, the term $\mathcal{L}_\perp$ is incorporated into the loss function, drawing inspiration from [206]. By forcing the latent variables to be uncorrelated, the model avoids embedding dependencies in the latent space, requiring the decoder to learn these correlations in order to generate data consistent with the input distribution. Consequently, it becomes possible to generate realistic instances, i.e., instances that are consistent with the joint distribution used to train the VAE, by providing the decoder with uncorrelated Gaussian noise.

The $\mathcal{L}_\perp$ loss term is calculated as follows:

$$\mathcal{L}_\perp = \sum_{k=1}^{Z}\sum_{i=1}^{Z} \left| \frac{1}{M}\left(\sum_{j=1}^{M} z_j^k z_j^i\right) - \frac{1}{M^2}\left(\sum_{j=1}^{M} z_j^k \sum_{j=1}^{M} z_j^i\right) \right| \tag{39}$$

where Z represents the dimension of the latent space, M is the batch size, and z_j^k and z_j^i denote the values of the k^{th} and i^{th} latent space variables for the j^{th} batch instance, respectively.

Extending the NaiveVAE introduced in this section, the NaiveVAE+GAN model embeds the pretrained VAE decoder within a GAN architecture to introduce adversarial training. In this configuration, the generator enhances the synthesized scenarios by exploiting a discriminative signal that augments the statistical structure captured by the VAE.

5.2 GENERATIVE ADVERSARIAL NETWORK WITH NAIVE VAE-PRETRAINED

Building upon the baseline NaiveVAE introduced in 5.1, we propose a model that embeds the decoder of the NaiveVAE as the generator within a Generative Adversarial Network architecture (NaiveVAE+GAN)(Figure 12).

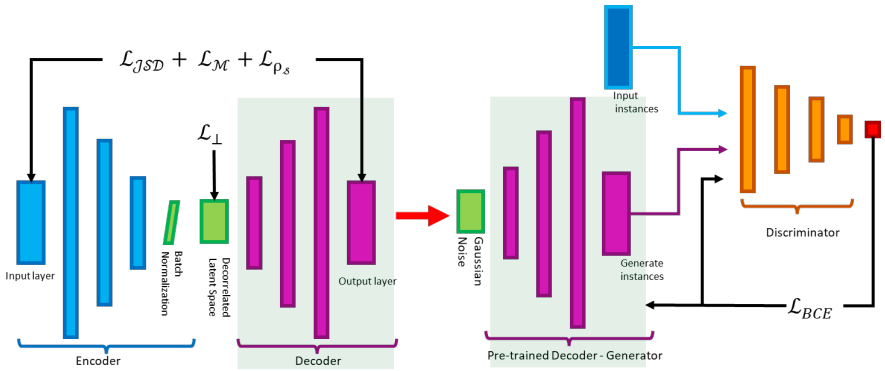


Figure 12 – Naive VAE+GAN architecture. The generator is the pretrained decoder of the NaiveVAE, which maps latent vectors sampled from $\mathcal{N}(0, I_d)$ into synthetic traffic scenarios. This initialization provides a stable starting point that already captures the statistical structure of the data. The discriminator, implemented as a fully connected neural network, is trained to distinguish between real and synthetic instances.

The proposed NaiveVAE+GAN model consists of two main components:

(i) the generator, initialized with the pretrained NaiveVAE decoder, fed with Gaussian noise vectors, to generate synthetic scenarios;

(ii) the discriminator, implemented as a fully connected neural network, which is trained to distinguish between real data and the synthetic instances produced by the generator.

This design leverages the ability of the NaiveVAE to learn to generate new instances directly from Gaussian noise. By reusing the pretrained NaiveVAE decoder as the GAN generator, the model benefits from a stable initialization that already captures the statistical structure of the data, while adversarial training further refines the generator to improve the realism of the synthesized scenarios.

In the proposed *NaiveVAE+GAN* architecture, the training process is structured in two phases: the In the first phase, the training involves a NaiveVAE–based model (as introduced in Section 5.1). In the second phase, the pretrained generator is embedded into a GAN framework. During each training iteration, the discriminator is alternately optimized on real and synthetic instances, while the generator

is updated to improve its ability to fool the discriminator. This adversarial interplay, formalized as a min-max optimization problem, progressively improves the realism of the generated scenarios.
Formally, let $x_{in} \in S$ denote a real instance, and $z \sim \mathcal{N}(0, I_d)$ be a latent vector of dimension d, whose components z_i are independently drawn from a standard normal distribution, the generator G_θ produces a synthetic instance $x_{gen} = G_\theta(z)$, while the discriminator D_ψ outputs the probability that a given instance is real.
The training objective of a GAN is based on the binary cross-entropy loss (Equation 9).

GANs are notoriously difficult to train: the adversarial dynamics often lead to instability, oscillatory behavior, or sensitivity to hyperparameter choices. As suggested by the results presented in Chapter 7, GAN-based models struggle to generate realistic traffic scenarios, failing to adequately capture the correlation structure among the components of the multivariate distribution. Motivated by these limitations, we introduce a Copula-based latent space that explicitly models structured correlations among the latent variables, thereby enabling a more faithful representation of latent-space dependencies.

5.3 NAIVE VAE WITH COPULA-BASED LATENT SPACE

The NaiveVAE architecture, presented in Section 5.1, is grounded on an isotropic Gaussian prior, which assumes independence among latent dimensions. In this section, we propose the Copula-naiveVAE model, which relaxes this assumption by explicitly modelling correlations among latent components. The core of this model is a Copula-based prior that provides correlated latent samples used as prior to the decoder (Figure 13).

Through a tailored loss function, the latent space of Copula-naiveVAE is constrained to encode the data into a compact, low-dimensional subspace, thereby fulfilling the same representational objective typically enforced in autoencoder architectures. However, through the use of a batch normalization layer in the model design, the distribution of each latent component is standardized to a Gaussian, while correlations across coordinates are preserved. As a result, each latent variable exhibits a Gaussian marginal distribution while preserving the underlying dependence structure.
Fitting a Copula to this correlated latent space enables the generation of new latent vectors that retain the underlying dependence structure. These correlated samples serve as a prior for the decoder, enabling the generation of scenarios that reproduce the realistic joint patterns observed in the data.
Formally, let $z = (z_1, \ldots, z_d)$ denote the latent representation learned by the encoder, where d is the dimension of the latent space. A Cop-

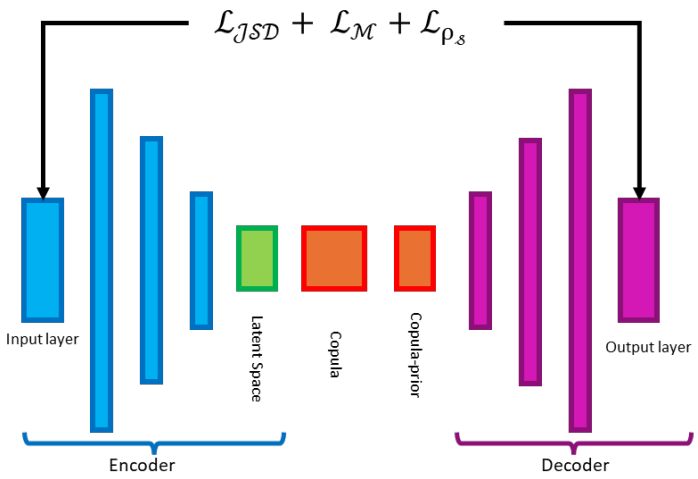


Figure 13 – Copula-naiveVAE architecture. The encoder maps input instances into a latent space with standardized Gaussian marginals, preserving correlations across latent dimensions. A Copula is then fitted on this latent space to capture the dependency structure among latent variables. The obtained prior is used to feed the decoder to generate new synthetic scenarios.

ula $\widehat{C}$ is fitted to the latent space distribution, and the corresponding marginal distribution functions $\widehat{F}_1, \ldots, \widehat{F}_d$ are estimated. This defines the multivariate prior from which new latent vectors z_{new} are drawn. Formally,

$$z_{new} \sim \widehat{C}(\widehat{F}_1, \ldots, \widehat{F}_d) \tag{40}$$

where $\widehat{F}_i$ denotes the empirical marginal distribution of the i^{th} latent component z_i, and $\widehat{C}$ is the Copula fitted to their joint distribution. The sampled latent vectors z_{new} are then used as input to the decoder to generate synthetic scenarios.

Among the various Copula models, we adopt the Gaussian Copula, which provides a natural extension of the standard VAE prior [190]. This choice ensures compatibility with the Gaussian marginals of the latent space and allows the dependence structure to be modeled through a correlation matrix.

By operating in the compact latent space learned by the encoder, the decoder focuses on the most salient features summarized in the prior. This low-dimensional representation enables the Gaussian Copula to effectively capture dependencies among latent variables, while mitigating the statistical and computational difficulties typically encountered in high-dimensional Copula modelling.
The Copula-naiveVAE adopts the same loss formulation as the Naive-

VAE, but without the independence term $\mathcal{L}_{\perp}$, which enforces isotropy in the latent space. The resulting objective is

$$\mathcal{L}_{CopulaNaiveVAE} = \alpha \cdot \mathcal{L}_{JSD} + \beta \cdot \mathcal{L}_{M} + \gamma \cdot \mathcal{L}_{\rho_s} \tag{41}$$

where $\mathcal{L}_{JSD}$ measures distributional similarity via Jensen–Shannon Divergence, $\mathcal{L}_{M}$ penalizes deviations in the marginal medians, and $\mathcal{L}_{\rho_s}$ preserves monotonic dependencies through Spearman rank correlation. The coefficients α, β and γ act as weighting factors for the loss components and are selected through hyperparameter optimization.

5.4 NAIVE VAE WITH GRAPH CONVOLUTIONAL NETWORK

Building upon the baseline NaiveVAE introduced in Section 5.1, we extend the architecture by incorporating a Graph Convolutional Network (GCN) layer in the encoder and a Graph Deconvolutional Network (GDN) layer in the decoder. This architectural design makes spatial correlations explicit, offering a principled mechanism for modelling spatial dependencies that variants without GCN layers would otherwise need to infer only implicitly.

In this architecture, the encoder is initialized with a GCN layer, which performs a neighborhood aggregation operation over the road network graph. This mechanism integrates information from adjacent nodes according to the underlying topological structure, thereby producing the first hidden representation of the input data:

$$H^{(1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}XW^{(0)}\right) \tag{42}$$

where $\tilde{A} = A + I$ is the adjacency matrix with self-loops, $\tilde{D}$ is its degree matrix, $W^{(0)}$ are trainable weights, and $\sigma(\cdot)$ is a non-linear activation function.
This formulation ensures that each node's representation is initialized by aggregating information from its directly connected neighbors (first-order neighbors), thus embedding spatial dependencies into the latent space from the first encoder layer.
The final layer of the decoder is a GDN layer that maps latent features back into the nodes of the road graph, thereby reconstructing spatially structured outputs. This operation can be regarded as the inverse counterpart of the aggregation performed by the GCN layer: whereas the encoder integrates information from neighboring nodes to obtain compact latent embeddings, the decoder diffuses the latent representations across the graph, thereby reconstructing traffic speeds at the level of road segments.
By embedding graph-based operations in both the encoder and the decoder, the proposed architecture achieves a dual benefit:

(i) the latent space captures spatial correlations induced by the road topology, and

(ii) the reconstruction process generates traffic scenarios that are statistically consistent and structurally faithful to the underlying network.

The model employs the same training objective as the NaiveVAE described in Section 5.1. Similarly, the latent vectors z are sampled from the same isotropic Gaussian prior $\mathcal{N}(0, I_d)$, consistent with the NaiveVAE formulation.

5.5 COPULA-NAIVEVAE WITH GRAPH CONVOLUTIONAL NETWORK

Following the Copula-based NaiveVAE presented in Section 5.3, which employs a Copula prior, the architecture is further enhanced through the integration of graph-based layers positioned at the encoder–decoder boundaries. As in the GCN-augmented NaiveVAE introduced in Section 5.4, the encoder is initialized with a GCN layer that performs a neighborhood aggregation operation over the road network graph, thereby enriching the latent representation with spatially informed features.
Symmetrically, the decoder concludes with a GDN layer, which diffuses latent features back across the graph topology to reconstruct traffic speeds at the level of individual road segments. This design ensures that both the encoding and decoding processes explicitly respect the connectivity of the underlying road network.

5.6 VARIATIONAL AUTOENCODER WITH DUAL LATENT VECTORS

Within the proposed framework for traffic scenario generation, we further examine an architecture grounded in the classical Variational Autoencoder (VAE) paradigm.

In this model, the encoder produces two distinct outputs: a mean vector μ and a variance vector σ^2, which together enable stochastic sampling through the reparameterization trick and allow the model to approximate the underlying data distribution while preserving differentiability of the training objective.

Formally, the pair (μ, σ^2) parameterizes a multivariate Gaussian distribution with diagonal covariance, thereby enforcing independence across latent dimensions:

$$q(z \mid x_{in}) = \mathcal{N}\big(z; \mu(x_{in}), \mathrm{diag}(\sigma^2(x_{in}))\big) \tag{43}$$

where x_{in} denotes a input instances, $\mu(x_{in})$ and $\sigma^2(x_{in})$ denote the mean and variance vectors produced by the encoder, and z is the

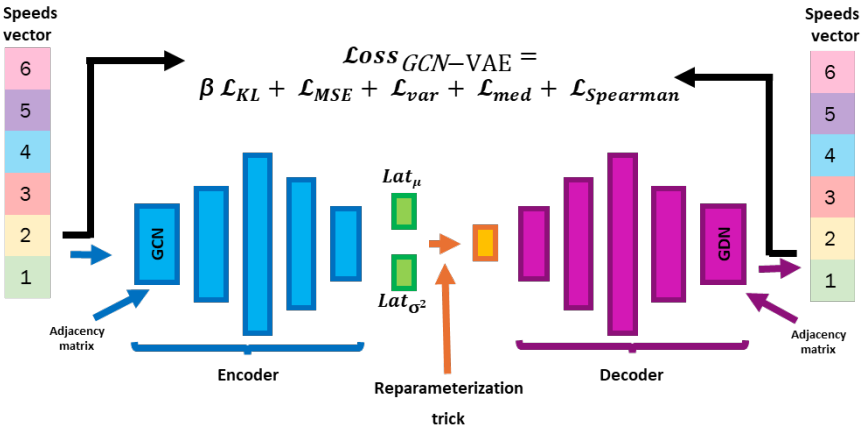


Figure 14 – Variational Autoencoder (VAE) with dual latent vectors and Graph Convolution layers. The encoder outputs a mean vector μ and a variance vector σ^2, which parameterize a diagonal Gaussian distribution in the latent space. Latent samples are drawn via the reparameterization trick and used to feed the decoded to generate synthetic traffic scenarios.

corresponding latent representation sampled from this approximate posterior.

As previously discussed in Section 3.3.2, during the training phase, VAE encodes the input traffic data into a diagonal Gaussian distribution parameterized by (μ, σ^2). Latent variables $z \in \mathbb{R}^d$ are then sampled from these distributions using the reparameterization trick, which ensures differentiability of the stochastic mapping.

In the generative phase, by contrast, the decoder is therefore fed with latent samples drawn directly from the prior distribution $p(z) = \mathcal{N}(0, I_d)$, enabling the synthesis of novel traffic scenarios that are consistent with the learned latent representation.
The training objective of this architecture is based on ELBO, which combines a reconstruction term with a regularization term.

In our implementation, the reconstruction loss is expressed as the mean squared error (MSE) between the observed data x_{in} and the generated x_{gen}, while the regularization is given by the Kullback–Leibler divergence between the approximate posterior and the isotropic Gaussian prior. Formally:

$$\mathcal{L}_{\text{ELBO}}(x_{in}) = \underbrace{\|x_{in} - x_{gen}\|^2}_{\text{MSE}} + \underbrace{D_{KL}(q(z|x_{in}) \,\|\, p(z))}_{\text{KL divergence}} \tag{44}$$

where x_{in} denotes an input traffic instance, x_{gen} is the corresponding generated instance produced by the decoder and z is the latent representation sampled from the approximate posterior $q(z|x_{in})$.

While the canonical ELBO objective provides a principled trade-off between reconstruction fidelity and latent regularization, it does not

explicitly enforce the preservation of higher-order statistical properties of the data. To address this limitation, we propose a custom loss function that augments the standard terms $\mathcal{L}_{\text{MSE}}$ and $\mathcal{L}_{\text{KL}}$ with additional components designed to align the statistical structure of generated samples with that of the input data. Formally, the total loss is defined as

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MSE}} + \mathcal{L}_{\text{KL}} + \lambda_{\text{var}}\,\mathcal{L}_{\text{var}} + \lambda_{\text{med}}\,\mathcal{L}_{\text{med}} + \lambda_{\text{Spearman}}\,\mathcal{L}_{\text{Spearman}} \tag{45}$$

where:

- $\mathcal{L}_{\text{MSE}}$: reconstruction error between input and generated instances;
- $\mathcal{L}_{\text{KL}}$: Kullback–Leibler divergence between approximate posterior and prior distribution;
- $\mathcal{L}_{\text{var}}$: variance matching term, aligning dispersion of input and generated data;
- $\mathcal{L}_{\text{med}}$: median matching term, enforcing robust alignment of central tendency;
- $\mathcal{L}_{\text{Spearman}}$: correlation matching term, preserving cross-variable dependencies.
- The coefficients λ_{var}, λ_{med}, and $\lambda_{\text{Spearman}}$ are hyperparameters tuned during optimization to balance the relative contribution of each term in the total loss.

VARIANCE PRESERVATION LOSS Realistic synthetic scenarios require preserving the variability inherent in each marginal variable. To ensure that the generated data reproduce this variability, we introduce a variance matching term:

$$\mathcal{L}_{\text{var}} = \sum_{n=1}^{N} \left(\text{var}(x_{in}^{n}) - \text{var}(x_{gen}^{n})\right)^2 \tag{46}$$

where $\text{var}(x_{in}^{n})$ and $\text{var}(x_{gen}^{n})$ denote the variances computed on the n^{th} marginal variable in the input and generated data, respectively, and N is the number of marginal variables.

This term penalizes discrepancies between the empirical variances of the input and generated data, enforcing consistency of the distributions. Although the MSE loss term encourages reconstructions that are close to the inputs at the instance level, it primarily enforces pointwise similarity and does not guarantee preservation of global variability. As a result, a model trained solely with MSE may converge to solutions that minimize reconstruction error while distorting the natural variability of the data, either collapsing the variance and producing overly homogeneous samples, or inflating

it and generating unrealistic levels of noise. Such distortions are particularly problematic in domains like traffic modelling, where variability in measurements is a crucial aspect.

By explicitly penalizing variance mismatches across marginal variables, the model gains direct control over the variability of the generated data. This encourages the generative process to reproduce not only the average behavior of the system but also its intrinsic variability, which is crucial for producing realistic and operationally meaningful scenarios. In this sense, variance matching complements the MSE by extending the training objective beyond instance-level fidelity to batch-levels, to preserve distributional properties.

MEDIAN PRESERVATION LOSS To preserve the variability of the data, it is important to ensure that the generated distribution remains centered around a representative value of the system. In many real-world datasets, the mean can be influenced by asymmetries or irregularities in the distribution, making it an unstable indicator of the system's typical operating condition. This may lead mean-based alignment to reproduce a biased notion of centrality.

To address this issue, we introduce a median matching loss term that enforces alignment between the empirical medians of the input and generated distributions:

$$\mathcal{L}_{\text{med}} = \sum_{n=1}^{N} \left(\text{median}(x_{in}^{n}) - \text{median}(x_{gen}^{n})\right)^2 \tag{47}$$

where $\text{median}(x_{in}^{n})$ and $\text{median}(x_{gen}^{n})$ denote the median computed on the n^{th} marginal variable in the input and generated data, respectively, and N is the number of marginal variables.

Incorporating this term ensures that the generated distribution remains anchored to a representative central value, preventing systematic shifts away from the relevant center. By penalizing discrepancies in the median, the model is encouraged to reproduce the typical position of the data distribution, leading to synthetic scenarios that are statistically coherent.

SPEARMAN CORRELATION PRESERVATION LOSS While variance and median matching ensure that the generated data reproduce the correct scale and central tendency of each marginal variable, such constraints do not account for the interdependencies that exist across variables. In many real-world domains, including traffic networks, marginal variables exhibit strong correlations, and their joint behavior often carries more information than their marginal distributions.

Preserving such dependency structures is therefore essential for generating realistic synthetic data.

To enforce this property, we introduce a correlation matching term based on Spearman's rank correlation:

$$\mathcal{L}_{\text{Spearman}} = \left\| \rho^{\text{in}} - \rho^{\text{out}} \right\|_F \tag{48}$$

where ρ^{in} and ρ^{out} denote the Spearman correlation matrices of the input and generated data, respectively, and $\| \cdot \|_F$ is the Frobenius norm.

This loss term penalizes discrepancies in the dependency structure and aligns the correlation matrices of the input and generated instances, promoting the preservation of the data's multivariate dependence structure. This aspect is crucial in traffic modelling, where dependencies are often nonlinear. Models that overlook these dependencies may produce scenarios that appear plausible when viewed at the level of individual components but unrealistic when evaluated over the full road network. Enforcing Spearman correlation matching ensures that the synthetic data respect these complex interconnections, yielding scenarios that are both statistically coherent and realistic.

In this proposed loss, the different terms jointly provide a comprehensive framework that balances instance-level accuracy, distributional robustness, and realism in the generated data.

Building on this foundation, our study examines two architectural design choices for the VAE framework:

- **Fully Connected (FC) architecture**: both encoder and decoder are implemented as fully connected networks, providing a baseline architecture that isolates the effect of the dual latent representation.
- **Graph Convolutional (GCN/GDN) architecture**: the encoder begins with a Graph Convolutional Network (GCN) layer, while the decoder concludes with a Graph Deconvolutional Network (GDN) layer, embedding the spatial correlations induced by the road network topology directly into the latent encoding and reconstruction processes (Figure 14).

These implementations allow us to assess the robustness of the proposed objective across architectures of different complexity and to evaluate the additional benefits of explicitly incorporating graph-based inductive biases into the generative process.

5.7 BETA-VARIATIONAL AUTOENCODER WITH DUAL LATENT VECTORS AND GCN

Building on the β-VAE architecture, introduced in [77], we extend the VAE–GCN model described in Section 5.6 by incorporating the parameter β into the loss function.

In the β-VAE paradigm, the KL-divergence term in the ELBO loss function is weighted by β, which controls the trade-off between reconstruction fidelity and latent regularization. Higher values of β encourage disentanglement and enhance the interpretability of the latent space, while lower values prioritize reconstruction accuracy, resulting in more entangled representations.
In our setting, the role of β becomes particularly important, as it controls both the regularization and the amount of variability preserved in the latent space. Rather than fixing β to a constant value, we adopt dynamic scheduling strategies, reflecting the intuition that the optimal balance between reconstruction and regularization evolves during training.

At the beginning of training, prioritizing reconstruction helps the model capture the dominant structure of the data, while stronger regularization in later stages promotes a more organized and interpretable latent space. We therefore consider two scheduling strategies:

- **Linear schedule**:
  $\beta(t)$ increases linearly from 0 to 1 during training. This schedule allows the model to initially prioritize reconstruction, gradually increasing the strength of the regularization as training progresses, reducing the risk of latent-space collapse (Figure 15 left).
- **Cosine schedule**:

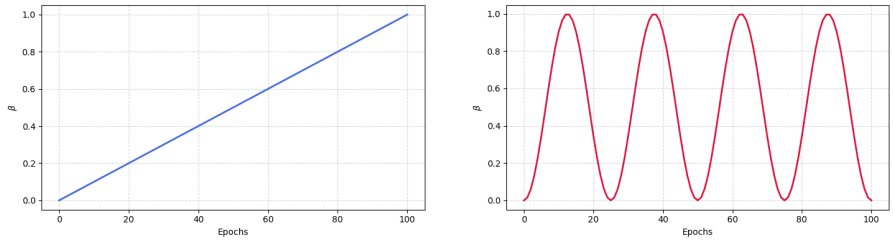
$$\beta(t) = \frac{1}{2}\left(1 - \cos\left(\pi\, t/T\right)\right) \tag{49}$$

  where t is the epoch index and T the total number of training epochs. This smooth annealing curve provides a more gradual increase of the KL penalty, reducing sudden variations in the optimization landscape and encouraging disentanglement while preserving training stability.
  Compared to the linear schedule, the cosine variant offers a smoother transition, which can be advantageous in avoiding oscillations or regularization. Moreover, its cyclical pattern can be extended to allow repeated phases of exploration and consolidation, further enhancing the robustness of the learned representations (Figure 15 right).

In the context of traffic scenario generation, this is particularly valuable: disentangled latent codes allow the model to capture

independent generative factors enhancing both interpretability and controllability of the synthetic scenarios.

The β-VAE with dual latent vectors thus provides a principled mechanism to reconcile reconstruction fidelity with the need for structured, disentangled latent spaces. By combining the stochastic richness of dual latent parameterization with the adaptive flexibility of dynamic β scheduling, this architecture offers a robust and interpretable generative framework well-suited to complex, high-dimensional domains such as traffic modelling.



(a) Linear schedule

(b) Periodic (cosine) schedule

Figure 15 – Left: linear β-scheduling over $T = 100$ epochs. Right: periodic (cosine) β-scheduling over $T = 100$ epochs with period $P = 25$, showing cyclical modulation between low- and high-regularization phases.

5.7.1 β-*VAE Linear with Transformer (T-VAE)*

The Transformer-based β-VAE (β-T-VAE) represents a further step in the exploration of generative models for traffic scenario synthesis.

It extends the β-VAE framework by integrating non-causal Transformer layers [176, 193] into both the encoder and decoder, with the primary aim of modelling long-range spatial dependencies among road segments. While Graph Convolutional Networks (GCNs) are adept at capturing local correlations constrained by the adjacency matrix of the road network, Transformers enable the modelling of global, non-local correlation by allowing every segment to attend to all others, without topological restrictions. This is crucial in urban traffic scenarios, where real-world phenomena can induce strong correlations between distant segments that are not captured with GCN, which takes into account the neighborhoods of a road.

Unlike classical Transformers [185], which are typically employed in sequential tasks such as natural language processing or time-series forecasting, the model introduced in this section adopts a non-causal Transformer architecture. In this setting, the attention mechanism is not restricted by causal masking, and positional encodings are

omitted. By removing positional information, the model avoids imposing temporal relationships on the input instances, preventing the Transformer from incorrectly interpreting road segments as elements of a time-ordered sequence and thereby preserving the inherently spatial nature of the problem.
This architectural choice reflects the structure of the data: each instance corresponds to an independent snapshot of traffic conditions over a road network, rather than an ordered temporal sequence.

The encoder takes as input an instance $x_{in} \in \mathbb{R}^N$ from the dataset. An initial linear projection maps x_{in} to an intermediate latent space of dimension d, yielding an embedding $H_0 \in \mathbb{R}^{N \times d}$.
This representation is then processed through non-causal Transformer blocks, each comprising:

- Multi-head self-attention with a fully dense mask,
- A feed-forward network with GELU activation,
- Residual connections and layer normalization.

The core of the self-attention mechanism is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{50}$$

where Q, K, V are the query, key, and value matrices derived from the input embeddings via learned projections, and d_k is the dimension of the keys.

Following the Transformer blocks, two linear projections compute the parameters of the latent distribution: $\mu(x), \log \sigma^2(X) \in \mathbb{R}^z$.
The latent sample is obtained via the reparameterization trick:

$$z = \mu(x_{in}) + \sigma(x_{in}) \odot \epsilon \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, 1_d) \tag{51}$$

The decoder mirrors the encoder's structure: the latent vector z is first projected to a space of dimension $N \times d$, creating an initial representation that is refined through a second series of non-causal Transformer blocks. A final linear layer maps the output representation to the generated scenario $x_{gen} \in \mathbb{R}^N$.
The β-T-VAE is trained using an extended ELBO objective function:

$$\mathcal{L} = \beta\, D_{KL} + \mathcal{L}_{MSE} + \lambda_{var}\, \mathcal{L}_{var} + \lambda_{med}\, \mathcal{L}_{med} + \lambda_{spearman}\, \mathcal{L}_{spearman} \tag{52}$$

The additional terms play a pivotal role in the urban traffic domain:

- $\mathcal{L}_{var}$: Maintains segment-wise variability, critical for replicating instability and shockwave phenomena;
- $\mathcal{L}_{med}$: Aligns the median values, ensuring robust matching of central traffic distribution tendencies;
- $\mathcal{L}_{Spearman}$: Captures non-linear, monotonic dependencies between segments, reflecting spatial congestion propagation.

The β parameter allows fine-tuned regularization of the latent space, promoting a more structured and interpretable representation that mitigates posterior collapse while enhancing disentanglement.

The adoption of non-causal Transformers is driven by three domain-specific requirements:

- **Non-Local Spatial Dependencies**: Distant segments in the road network may exhibit strong correlations due to alternative routes, spillback, or rerouting behaviors-interactions that GCNs, limited to local neighborhoods, systematically underfit.
- **Absence of Temporal Order**: Each scenario is an independent observation; positional embeddings would introduce unwarranted bias, whereas the Transformer's global attention naturally handles set-like inputs.
- **Scalability**: Global attention enables learning complex relationships without exponentially increasing GCN depth or relying on incomplete adjacency matrices, making the model suitable for large N (e.g., thousands of segments).

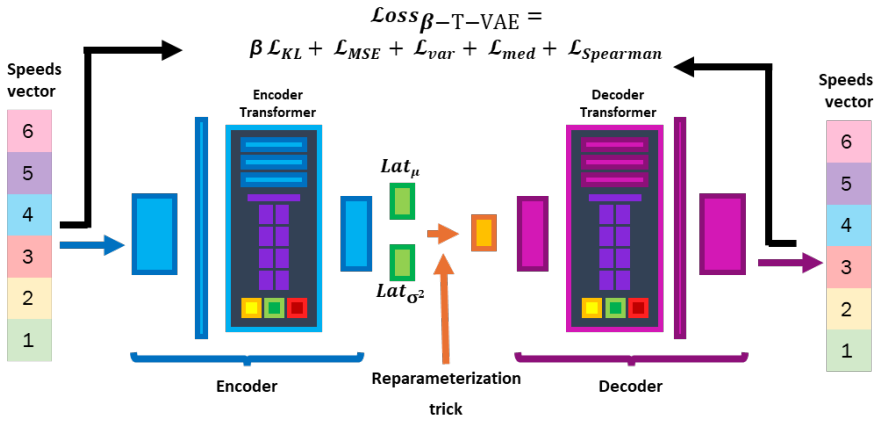


Figure 16 – Transformer-based Variational Autoencoder (β-T-VAE). The encoder, that integrates non-causal Transformer blocks, maps input traffic scenarios to a latent space by producing a mean vector μ and a variance vector σ^2, which parameterize a diagonal Gaussian distribution. Latent samples are drawn via the reparameterization trick and used to feed the decoder, which integrates non-causal Transformer blocks to generate synthetic traffic scenarios with long-range spatial dependencies.

As summarized in Table 8, the proposed architectures provide a comprehensive methodological framework for traffic scenario generation. By comparing models of increasing complexity, ranging from simple VAEs to Transformer-based formulations, we are able to assess the relative contribution of structural priors, disentanglement strategies, and attention mechanisms.

Model	Architecture	Latent Type	Loss Function	Key Features
nVAE-FC Noise	Fully Connected	Gaussian Noise	$L_{JSD} + L_{med} + L_{Ps} + L_{\perp}$	Decorrelated latent; independent prior
nVAE-FC Copula	Fully Connected	Gaussian Copula	$L_{JSD} + L_{med} + L_{Ps}$	Correlated latent prior
nVAE-GCN Noise	Graph Conv	Gaussian Noise	$L_{JSD} + L_{med} + L_{Ps} + L_{\perp}$	Graph topology, Decorrelated latent, independent prior
nVAE-GCN Copula	Graph Conv	Gaussian Copula	$L_{JSD} + L_{med} + L_{Ps}$	Graph topology, Correlated latent prior
nVAE+GAN	Fully Connected	Gaussian Noise	$L_{JSD} + L_{med} + L_{Ps} + L_{\perp}$ and L_{BCE} adversarial	GAN, Decorrelated latent, independent prior
VAE-FC	Fully Connected	Gaussian Noise	$ELBO(MSE + L_{KL}) + L_{var} + L_{med} + L_{spearman}$	VAE with μ and σ^2
VAE+GCN	Graph Conv	Gaussian Noise	$ELBO(MSE + L_{KL}) + L_{var} + L_{med} + L_{spearman}$	Graph-aware, VAE with μ and σ^2
β-lin-GCN-VAE	Graph Conv	Gaussian Noise	$ELBO(MSE + \beta L_{KL}) + L_{var} + L_{med} + L_{spearman}$	β-scheduler linear, Graph-aware, VAE with μ and σ^2
β-cos-GCN-VAE	Graph Conv	Gaussian Noise	$ELBO(MSE + \beta L_{KL}) + L_{var} + L_{med} + L_{spearman}$	β-scheduler cosine, Graph-aware, VAE with μ and σ^2
β-T-VAE	Fully Connected	Gaussian Noise	$ELBO(MSE + \beta L_{KL}) + L_{var} + L_{med} + L_{spearman}$	Transformer mechanism, VAE with μ and σ^2

Table 8 – Comparison of proposed models.

Part III

EVALUATION FRAMEWORK, EXPERIMENTAL SETTINGS AND RESULTS

Evaluating the quality of scenarios generated is an open challenge: no single metric can reliably quantify the similarity between two high-dimensional multivariate variables while jointly accounting for both marginal behaviors and correlation structures. In this part we introduce a set of complementary metrics designed to address this limitation and to provide a comprehensive assessment of the models previously proposed for traffic-speed scenario generation.

Chapter 6 introduces the proposed framework designed to evaluate generative models, comparing the quality and consistency of traffic scenarios generated to real world data. In this chapter we also present the real world dataset and experimental settings adopted throughout the evaluation.

Chapter 7 reports the quantitative and qualitative performance of the proposed architectures, assessing their scalability by progressively enlarging the portion of the road network considered.

6

EVALUATION FRAMEWORK AND EXPERIMENTAL VALIDATION

Evaluating the performance of generative models for traffic scenario generation requires metrics capable of quantifying how closely the generated data resemble real-world observations. Although well-established distance measures exist for comparing univariate distributions, assessing similarity between joint multivariate distributions remains fundamentally challenging, particularly in high-dimensional and spatially correlated settings such as road-network traffic speeds. In these contexts, faithfully capturing the full spectrum of spatial and topological dependencies is essential for any meaningful evaluation of scenario generation methods, although no single comprehensive metric exists that can adequately quantify the discrepancy between two multivariate distributions.
For this reason, to evaluate the closeness of the generated scenarios with respect to real data, we propose a comprehensive evaluation framework that integrates both quantitative and qualitative measures. This multi-faceted approach allows us to assess statistical fidelity, preservation of correlations, and the practical usability of the model when generating new scenarios.
In this chapter, we introduce the metrics framework that are applied in Chapter 7 to compare the scenarios generated by the models presented in Chapter 5.
In the second part of this chapter, we describe the real-world datasets employed in the experimental validation, along with the proposed sampling algorithm, which selects representative subsets of roads from the dataset's network and enables a systematic assessment of model scalability with respect to road-network size.

6.1 EVALUATION METRICS

The limitation in evaluating our proposed models arises from the need to simultaneously assess multiple complementary aspects, including joint dependencies, marginal distributions, geometric structure, and higher-order statistics.
To address this challenge, we propose a multi-metric evaluation framework that combines a set of complementary measures: statistical distributional distances (both multivariate and marginal) and qualitative visual analysis via non-linear dimensionality reduction.

The presented evaluation encompasses three perspectives:

- **Quantitative distributional metrics**: these yield scalar scores that measure how closely the synthetic data reproduce the statistical properties of the real data across multiple dimensions (joint and marginal distributions).
  - **Bhattacharyya distance**[14]: a multivariate metric based on Mahalanobis distance, used to measure the statistical separation and overlap between two distributions
  - **Wasserstein distance**[187]: employed to evaluate the geometric differences between the marginal distributions of individual variables.
- **Qualitative visual assessment**: complementary visual analysis through non-linear dimensionality reduction techniques, aimed at revealing structural discrepancies and similarities that may not be fully captured by quantitative distributional metrics alone.
  - **Uniform Manifold Approximation and Projection (UMAP)** [123]: an advanced non-linear dimensionality reduction tool for qualitatively assessing the retention of local neighborhoods, global geometry, and cluster distinctions in real traffic scenarios when projecting both real and synthetic data into a lower-dimensional (usually 2D) space.
- **Computational efficiency**: evaluation of model efficiency and scaling behaviour.
  - **Generation time**: generation time per synthetic scenario. Used to compare model efficiency at same complexity (e.g., same sub-split road networks) and to assess scaling behaviour as problem size grows (e.g., with increasing number of segments considered).

The following subsections provide the mathematical definition and details for each metric, together with the specific parameter choices adopted in this study.

6.1.1 *Bhattacharyya Distance (BD)*

The Bhattacharyya Distance (BD)[14] is a Mahalanobis-based [121] distance, designed for quantifying the similarity between two multivariate probability distributions.
While the classical Mahalanobis distance calculates the separation between a single point and a distribution's centroid, the Bhattacharyya distance generalizes this idea to compare two multivariate Gaussian distributions. This extension incorporates differences in both the distributions' first-order statistics (means) and second-order structures (covariances) into a single scalar metric.
Mathematically, given two multivariate Gaussian distributions $P \sim$

$\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $Q \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, the Bhattacharyya distance is defined as:

$$BD(P, Q) = \frac{1}{8}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + \frac{1}{2}\ln\left(\frac{\det(\boldsymbol{\Sigma})}{\sqrt{\det(\boldsymbol{\Sigma}_1)\det(\boldsymbol{\Sigma}_2)}}\right) \tag{53}$$

where $\boldsymbol{\Sigma} = \frac{\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2}{2}$.
The first term in this equation is structurally analogous to a Mahalanobis distance, measuring the standardized separation between the distribution means weighted by their pooled covariance $\boldsymbol{\Sigma}$. The second term acts as a penalty function that captures the divergence between the covariance structures of the two distributions.
Unlike Kullback–Leibler divergence, which is asymmetric and can diverge when distributions barely overlap, the Bhattacharyya Distance always yields a finite assessment of their similarity. Its values lie in $[0, +\infty)$: values near zero indicate strong overlap and high similarity in both means and covariances, whereas larger values signal increasing separation. This makes BD particularly suitable for assessing distributional alignment and fidelity.
Real-world traffic speed data exhibit non-Gaussian characteristics, such as pronounced skewness and heavy tails. In these cases, the parametric BD does not provide a reliable absolute measure of dissimilarity between the real distribution P and the generated one Q, because the Gaussian approximation introduces a systematic bias.

However, when the goal is to compare multiple generative models against the same real dataset, the BD can still be used in a relative manner with good practical validity. The bias introduced by the Gaussian assumption (and thus the mismatch between the true distributional shape and its Gaussian approximation) affects all generated distributions in a broadly comparable way. As a result, the Bhattacharyya distance remains suitable for ranking models relative to one another, even if it cannot serve as an absolute measure of fidelity.

In our evaluation framework, the Bhattacharyya distance serves as a relative metric for model comparison, focusing on the joint distribution of traffic speed features. Lower BD values indicate better alignment between the Gaussian approximations of real and generated data. To complement BD and better capture non-Gaussian aspects, we employ the Wasserstein distance. While BD is sensitive to means and covariances, Wasserstein operates directly on the empirical distributions and is more robust to higher-order discrepancies. Their combined use provides a more comprehensive and reliable ranking of generative models on real-world traffic data.

6.1.1.1 *Wasserstein Distance (WD)*

The Wasserstein Distance (WD) [187], also known in the literature as the Earth Mover's Distance (EMD), captures geometric differences between probability distributions. It quantifies the minimum "cost" (in terms of mass transport) required to transform one distribution P into another Q over a given metric space, interpreting probability mass as piles of mass that must be moved from one configuration to the other. Mathematically, the p-Wasserstein distance is defined as:

$$W_p(P,Q) = \left(\inf_{\gamma \in \Gamma(P,Q)} \int_{\mathbb{R}\times\mathbb{R}} |x-y|^p \, d\gamma(x,y) \right)^{1/p} \tag{54}$$

where $\Gamma(P,Q)$ is the set of all couplings (joint distributions) with marginals P and Q.
Differently from measures like Kullback-Leibler or Jensen-Shannon divergence, which can become infinite or undefined when distributions have non-overlapping supports, WD provides a finite, meaningful value even under such conditions.
In the univariate setting, which is relevant for assessing marginal distributions, the p-Wasserstein distance for p=1 (often simply called the 1-Wasserstein or Kantorovich-Rubinstein distance) is formally defined for two probability measures P and Q as:

$$W_1(P,Q) = \inf_{\gamma \in \Gamma(P,Q)} \mathbb{E}_{(x,y)\sim\gamma} [|x-y|] \tag{55}$$

In the context of this study, we apply the WD to evaluate the fidelity of generated traffic scenarios by computing it separately for each of the N marginal distributions (corresponding to individual road segments or monitored variables) and then averaging these values across all variables to obtain an aggregate score.
The WD serves as a complement to multivariate metrics within our evaluation framework. While the Mahalanobis-based Bhattacharyya Distance (BD) provides insights into the global joint-structure fidelity by incorporating the covariance matrix, the WD focuses on the individual marginals, ensuring that each univariate component faithfully reproduces the empirical shape, spread, location, and tail behavior of its real-world counterpart. This complementary role is crucial because, even if a model preserves overall correlations (as measured by BD), it may still distort individual distributions, for instance by shifting means or compressing tails, thereby undermining the realism of traffic scenarios.

6.1.2 *Uniform Manifold Approximation and Projection (UMAP)*

While numerical metrics provide essential objective benchmarks for evaluating synthetic data fidelity, they often fail to capture the subtle nuances inherent in the high-dimensional geometric and topological

structures of the underlying data. Scalar distances frequently overlook critical features such as local neighborhood preservation, global continuums, and intricate cluster formations. To address these limitations and enrich the assessment, this study incorporates a qualitative analysis leveraging Uniform Manifold Approximation and Projection (UMAP). As a state-of-the-art non-linear dimensionality reduction technique, UMAP exhibits a superior ability to preserve both local and global data structures compared to traditional alternatives such as t-SNE [184] or Principal Component Analysis (PCA) [84, 142]. Introduced by McInnes et al. (2018) [123], UMAP is built on the idea that complex data points are not scattered randomly but lie on a hidden geometric shape called manifold. UMAP operates under three intuitive principles:

- **Uniformity**: It assumes that data is spread evenly across this manifold. If some areas look denser than others, UMAP considers it a result of the manifold's geometry being stretched or compressed.
- **Locality**: UMAP assumes that the manifold is locally Riemannian, meaning that at a sufficiently small scale, the metric space can be approximated as locally flat (Euclidean). This allows the algorithm to define local distance measures that remain consistent even when the global geometry is non-linear and complex.
- **Connectivity**: The manifold is assumed to be locally connected, ensuring that the underlying topological space does not consist of disjoint, isolated points. This allows UMAP to construct a continuous representation where the relationships between samples are modeled as a coherent structure rather than a collection of independent observations.

Based on these assumptions, UMAP creates a probabilistic network (formally called a Fuzzy Simplicial Set) that connects nearby points. This network acts as a "skeleton" that captures the essential shape of the original data. Finally, UMAP projects this skeleton into a reduced-dimensional space, typically 2D or 3D for visualization, carefully arranging the points to keep the skeleton's structure as faithful as possible to the original.

Compared to t-SNE, which focuses almost exclusively on local neighborhoods and often suffers from computational intractability on large datasets, UMAP is significantly more scalable and better preserves global hierarchical structures. Furthermore, unlike PCA, which is restricted to linear projections of maximum variance, UMAP adeptly uncovers the non-linear manifold patterns and multimodal distributions characteristic of urban traffic data.

In this work, UMAP is employed as a qualitative method to assess the fidelity of the generated traffic scenarios. By co-projecting real and synthetic multivariate samples into a shared two-dimensional embedding space, we can intuitively assess the preservation of the

underlying manifold structure. A substantial overlap between the real and synthetic point clouds indicates that the generative model has successfully captured the data support, including low probability events.
Conversely, observable discrepancies, such as disjoint clusters, systematic shifts, or compressed subspaces, serve as visual indicators of potential shortcomings, such as a failure to model long-range spatial dependencies. This metric complements our quantitative framework by providing an overview of the data's topological structure. Ensuring these features are preserved is critical for generating realistic traffic scenario.
Integrating UMAP into our multi-metric framework enhances model interpretability and aligns with best practices in high-dimensional data analysis, where visual exploration often uncovers structural insights inaccessible through numerical metrics alone.

6.1.3 *Computational Generative Time (CGI)*

The computational efficiency of a generative model is a decisive factor in its practical viability, particularly within the high-stakes environment of large-scale urban mobility. Synthetic traffic samples produced using scenario generation methods are essential for a wide range of applications, including stochastic optimization, simulation studies, and the evaluation of traffic-management strategies. In time-sensitive settings, however, excessive generation latency can render high-fidelity models unsuitable for deployment.
Analyzing generative time provides a direct assessment of a model's structural complexity. When executed on the same hardware, the inference duration reflects the computational depth and the volume of operations inherent in the architecture. Evaluating this metric allows for the quantification of the computational resources required by a specific model to generate realistic scenarios.
Furthermore, analyzing generation time is essential for studying the scalability of models. As the considered road networks expand to include more segments, the dimensionality of the problem increases accordingly. A robust model must demonstrate the ability to handle this growth without suffering from an exponential surge in resource demands. By benchmarking the time required to generate a fixed number of samples across different network sizes, we can identify potential bottlenecks and assess whether the architecture can maintain a balance between feasibility and performance as the system scales.
To quantify these aspects, we evaluate the time required to generate 1,000 synthetic scenarios. This metric focuses exclusively on the inference phase, as real-world applications typically rely on trained models for repeated generation tasks. To ensure a fair comparison,

all models are evaluated on the same hardware. This standardized protocol allows us to explore how each model scales in relation to the dimensionality of the considered road network.

6.2 DATASET: CHENGDU URBAN ROAD NETWORK

To evaluate the models proposed in Chapter 5, we conducted experiments on the real-world large-scale Chengdu Urban Road Network dataset, presented in "Urban link travel speed dataset from a megacity road network" [55, 62]. This dataset has been collected over the road network of Chengdu, a major megacity in western China with a population of over 20 million inhabitants. The network is concentrated within the central city area, enclosed by the ring expressway, and comprises 1,902 nodes (intersections) and 5,943 directed links (road segments).
The travel speeds on each link are estimated from a massive collection of GPS trajectory data gathered from vehicles, primarily taxis (more than 12,000 vehicles) computed at 2-minute intervals. The released version of the dataset spans 45 days, from June 1, 2015, at 03:00 to July 15, 2015, at 23:00. In this release, data are organized into five representative daily time windows: 3:00–5:00, 8:00–10:00, 12:00–14:00, 17:00–19:00, and 21:00–23:00. These intervals capture diverse traffic conditions, including off-peak hours, morning and evening rushes, and nighttime flows, reflecting non-linear congestion patterns influenced by urban topology, human behavior, and external factors such as weather or points of interest (e.g., schools, hospitals, and commercial districts).
The Chengdu dataset has been extensively used in the literature for tasks related to traffic forecasting, anomaly detection, and uncertainty modelling due to its topological complexity and high-dimensional nature. It has served as a benchmark in studies on graph neural networks for speed prediction [113, 209]. The raw data, sourced from loop detectors and GPS probes, exhibit strong spatial correlations (speeds on adjacent links are interdependent), making it ideal for evaluating generative models that must preserve multivariate distributions. The dataset can indeed be modeled and formalized as a high-dimensional multivariate structure (similar to the representation introduced in Chapter 4). Its thousands of interdependent variables pose significant challenges for scenario generation, as models must capture not only marginal distributions but also higher-order dependencies arising from the road network's graph topology.

6.2.1 *Selection of Road Subsets via ICBS*

To evaluate the models' ability to scale with increasing network size, we selected subsets of the road network with several cardinali-

ties. Rather than random sampling, which could disrupt topological coherence and introduce bias by ignoring spatial dependencies, we propose Iterative Connected Breadth-First Sampling (ICBS), a Breadth-First Search (BFS)–based method.
ICBS ensures that the selected subsets form connected subgraphs that are topologically consistent and representative of real traffic dynamics, preserving correlations and variability essential for realistic scenario generation. The ICBS method proceeds as follows:

1. **Geographic Seed Selection**: identify the road segment whose midpoint is closest to the geographic centroid of the entire road network, computed as the average of all start and end node coordinates. This ensures that the sampling process begins from the most spatially central area of the city, typically corresponding to the urban core or central business district.
2. **Cumulative Iterative Expansion via BFS**: starting from the endpoints of the seed link, iteratively expand the selected subgraph using Breadth-First Search (BFS) on the directed graph representation of the network. At each step targeting cardinality k, add new edges connected to the current frontier until the desired number of additional links is reached.
3. **Fallback Mechanism**: if BFS cannot provide enough new edges (e.g., due to local network sparsity), supplement the selection with the geographically closest remaining links (sorted by distance of their midpoint from the network centroid). This guarantees that the target cardinality is always met while preserving spatial coherence as much as possible.
4. **Assignment of ICBS Levels**: assign to each selected link the smallest cardinality level k at which it was included. This produces nested, connected subnetworks of increasing size $N \in \{16, 32, 64, 96, 128, 192, 256, 512, 768, 1024\}$, all containing the geographic core and expanding outwards in a spatially contiguous manner.

This geographically anchored, BFS-driven sampling strategy guarantees that all selected subsets are strongly connected and spatially coherent, while the growth pattern naturally mimics the radial expansion of urban areas from the city centre outwards. As a result, dependencies between nearby links are preserved at every scale, and the resulting subnetworks remain highly representative of real traffic dynamics.
This approach is particularly suitable for scenario generation tasks, where preserving spatial structure and local correlations is more critical than mere topological centrality [111].

Figure 17 – Left: full Chengdu urban road network. Right: ICBS-selected road subset, with colors indicating the ICBS inclusion levels.

6.3 EXPERIMENTAL SETUP

To evaluate the proposed architectures and their statistical baselines, we designed a comprehensive experimental framework aimed at assessing the ability of each model to generate realistic traffic speed scenarios across urban road networks of increasing structural complexity. The evaluation focused on several aspects as the preservation of distributional fidelity, both joint and marginal, the influence of different β-scheduling strategies (linear and periodic), and overall generative performance and the scalability of the models, particularly in large-scale settings where traditional statistical approaches become impractical.
The comparison includes the proposed architectures and a state-of-the-art baseline method.
The baseline model consist of a Gaussian Copula model (GCop), which fits marginal distributions and a Gaussian dependence structure.

The **proposed models** compared comprise:
naive VAEs (nVAE-FC and nVAE-GCN), employing fully connected (FC) or graph-convolutional (GCN) architectures and relying on noise or copula priors [23]; a hybrid naive VAE GAN model (nVAE+GAN) that augments the VAE with an adversarial discriminator [23]; and dual-latent space VAEs (VAE-FC and VAE+GCN) with hand-crafted loss functions, as presented in [24, 25]; a β-GCN-VAE with linear β-scheduling, where the coefficient increases starting from 0 to 1 over the course of training (β-lin-GCN-VAE); a β-GCN-VAE with periodic (cosine) scheduling, in which β oscillates between 0 and 1 following a cosine annealing pattern (β-cos-GCN-VAE); and the β-Transformer-VAE (β-T-VAE) (Section 5.7.1), which integrates a non-causal Transformers with eight-head self-attention and adopts linear β-scheduling.

Models are trained on different data splits of the Chengdu dataset (see Section 6.2). Each data split comprises speed measurements from subsets of the road network, that are selected using the ICBS strategy (see Section 6.2.1). This progressive subsampling design allows for evaluation of models' performance across datasets of increasing spatial scale, enabling assessment of scalability and robustness to larger graph structures.

All speed values are preprocessed via min-max mapping to the range $[-1, 1]$. Specifically, the minimum recorded speed in the dataset (4.36 km/h) is mapped to -1, and the maximum (101.55 km/h) is mapped to 1. This scaling is essential because all proposed generative architectures employ the TanH activation function, which requires values in the interval $(-1, 1)$. Scaling the input data to match this bounded range ensures compatibility with the activation, promotes numerical stability during training, and facilitates more effective gradient flow and convergence, practices well-established in the Variational Autoencoder literature for continuous bounded data.

To ensure a fair comparison across all architectures, the hyperparameters of each evaluated model have been tuned using Bayesian Optimization (ByO, presented in Section 2.3.5), taking into account different ICBS dataset subportions. This procedure allows identifying the best hyperparameters configuration for each model with respect to each specific ICBS dataset subportion. ByO efficiently explores the search space by modelling the performance surface through a surrogate function. This strategy is particularly well-suited for generative models, as it produces more stable and reliable hyperparameter configurations than traditional grid or random search. Acquisition functions, designed to balance exploration and exploitation, lead the search toward the most promising hyperparameter configurations. The surrogate function guiding the optimization process is based on the Bhattacharyya distance. The choice to adopt the Bhattacharyya distance as a surrogate function is motivated by its sensitivity to both marginal and joint distributions, making it a robust measure of the underlying generative performance.

Figure 18 provides an overview of the hyperparameter search process conducted via Bayesian Optimization for the nVAE model on the 16-roads subportion of the Chengdu dataset. Each point corresponds to a distinct configuration evaluated during optimization.

The left panel displays the set of hyperparameter configurations explored during the optimization. Each point represents a trial of a hyperparameter configuration, and its position reflects the values assigned to the selected hyperparameters. This plot offers insight into the sampling strategy of the acquisition function, revealing how the optimizer balances exploration of the search space with exploitation of promising regions.

The right panel reports the corresponding values of the surrogate ob-

jective function for each configuration. Lower values indicate better alignment between the generated and empirical distributions. The color gradient highlights the topology of the surrogate surface, allowing visual identification of local minima and regions of high generative fidelity. Red stars denote the best-performing configurations selected by the optimizer, while black dots represent other sampled points.

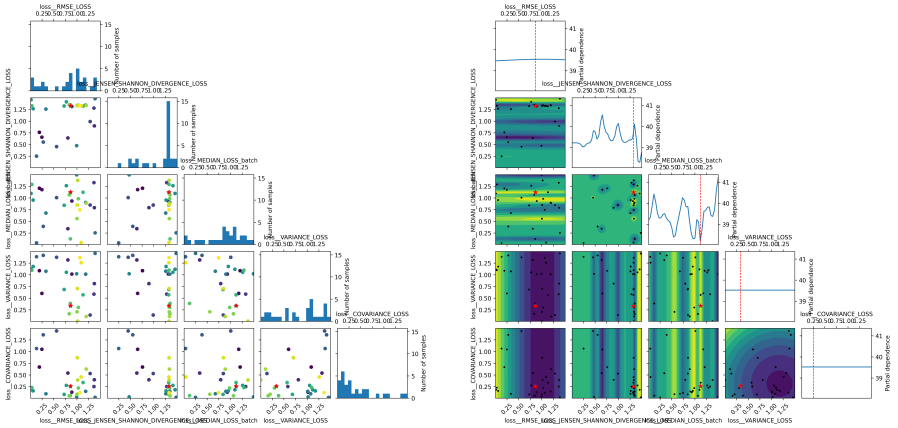


Figure 18 – Bayesian Optimization of the nVAE model on the 16-roads subportion of the Chengdu dataset. Left: hyperparameter configurations explored during the optimization process, where each point corresponds to a distinct trial sampled by the acquisition function. Right: surrogate objective values (based on Bhattacharyya distance) associated with each configuration, with lower values indicating better alignment between generated and empirical distributions. Red stars mark the best-performing configurations identified by the optimizer.

These plots illustrate the efficiency and selectivity of the Bayesian Optimization process, as well as the informativeness of the Bhattacharyya-based surrogate in guiding the search toward statistically coherent generative models.

To mitigate the inherent stochasticity in the training process, each experimental configuration was replicated across 30 independent runs, with different random seeds applied to model initialization, data shuffling, and sampling procedures. This approach yielded reliable mean and standard deviation values for all performance metrics, thereby ensuring greater robustness and reproducibility of the results against random fluctuations.
For a comprehensive assessment, we generated 1,000 synthetic traffic scenarios for each run of the model and data subset. Based on the presented evaluation framework, metrics such as Bhattacharyya Distance (BD), Wasserstein Distance (WD), and Computational Generative Time (CGT) were computed and averaged across these runs. Qualitative insights were further supported by UMAP visualizations

of the embedded distributions.

"Following the description of the experimental setup and dataset, Chapter 7 reports the quantitative, qualitative, and scalability results obtained across all models and ICBS dataset partitions

7

EXPERIMENTAL RESULTS

This chapter presents the experimental results of the proposed generative framework, following the evaluation methodology and performance metrics introduced in Section 6.1. The analysis is structured to assess the ability of the models to generate realistic urban traffic speed scenarios across multiple dimensions.
A quantitative assessment is conducted to measure how closely the synthetic data resemble the real world data distributions. This includes both joint and marginal distribution, enabling a detailed examination of distributional fidelity and the preservation of spatial dependencies. Complementary to this, a qualitative analysis based on low-dimensional embeddings provides visual insight into how well the generated samples reproduce the structure of real traffic patterns.
The chapter then investigates the computational efficiency of the models, with a particular focus on generation time. This aspect is crucial for applications requiring large-scale scenario sampling or real-time integration within transportation systems.
All experiments are performed on networks of increasing size (from 16 to 1024 road segments) to evaluate the scalability of the proposed architectures. This progressive enlargement of the spatial domain allows us to study how model performance evolves with dimensionality, highlighting strengths and limitations in both accuracy and efficiency.

All detailed numerical results, extended tables, and additional figures supporting the analyses presented in this chapter are reported in Appendix B.

7.1 EVALUATING JOINT AND MARGINAL DISTRIBUTIONS

To evaluate the ability of the proposed generative models to reproduce the statistical properties of real traffic speed data, we analyze both joint and marginal distributions. Following the framework introduced in Section 6.1, we use the Bhattacharyya Distance (BD) to assess multivariate similarity and the Wasserstein Distance (WD) to quantify discrepancies between real and generated distributions at the level of individual road segments. Together, these metrics provide a comprehensive view of distributional fidelity across the models introduced in Section 5. In addition to these quantitative measures, we analyze two-dimensional UMAP projections of both

real and generated samples. This provides a crucial qualitative assessment of each model's effectiveness in capturing the global structure of the data and preserving the intricate spatial correlations inherent in the dataset.

EVALUATION OF NAIVE VAE ARCHITECTURES

Initially, we conducted an analysis of the performance of naive VAE-based models across network sizes ranging from 16 to 128 roads, including fully connected VAEs with Gaussian noise priors, a GCN-based naive VAE variant, Copula-based fully connected naive VAEs, Copula-based GCN naive VAEs, comparing them with the Gaussian Copula-based model. The corresponding Wasserstein and Bhattacharyya distances are summarised in Tables 9 and 10.

nVAE-FC Noise

Naive Variational Autoencoders (nVAE-FC Noise) with a standard Gaussian prior struggle to learn and generate scenarios as data complexity increases, as confirmed quantitatively in Tables 10 and 9. Both the Bhattacharyya Distance (BD) and the Wasserstein Distance (WD) remain consistently high across all network sizes, indicating that simple isotropic priors fail to capture the intricate interdependencies of high-dimensional urban traffic data. This limitation becomes more pronounced as dimensionality grows: for instance, BD for the nVAE-FC Noise baseline increases from 2.217 at 16 roads to 6.918 at 128 roads, while WD rises from 3.487 to 16.512. These trends show that Gaussian-prior VAEs systematically underestimate long-range correlations and cannot reproduce heavy-tailed behaviours, a structural limitation particularly evident in the sharp growth of WD, which reflects poor alignment of marginal distributions and an inability to model skewed or tail-dominated traffic-speed patterns.

nVAE-FC Copula

A clear improvement emerges when replacing the Gaussian prior with a Copula-based prior (nVAE-FC Copula). As shown in both tables 10 and 9, BD and WD decrease consistently across all network sizes for Copula-based variants. For example, at 64 roads, BD drops from 6.107 (Gaussian prior) to 5.689 (Copula prior), and WD decreases from 18.932 to 17.842. This demonstrates that Copula priors provide a more expressive latent structure capable of modelling complex non-linear dependencies. This enhancement is especially pronounced in the marginal perspective, where the reduction in WD highlights a closer alignment with empirical histograms, including skewness and tail behaviour.

nVAE-GCN Noise

Introducing graph convolutional layers further refines performance by incorporating the spatial structure of the road network directly into the model architecture. However, when combined

with a Gaussian prior (nVAE-GCN Noise), GCNs offer only limited improvements: BD and WD decrease marginally compared to the naive VAE with Gaussian noise or Copula-based priors. For instance, at 64 roads BD decreases only from 6.107 (nVAE-FC Noise) and 5.689 (nVAE-FC Copula) to 5.325, while WD decreases from 18.932 (nVAE-FC Noise) and 17.842 (nVAE-FC Copula) to 17.316. These reductions confirm that, although GCNs help encode local spatial relationships, the latent space remains insufficiently expressive when constrained by an isotropic Gaussian prior.

nVAE-GCN Copula

Pairing GCNs with Copula priors yields additional gains, as shown by the nVAE-GCN Copula results (e.g., BD = 4.812 and WD = 16.982 at 64 roads). Yet, even in this configuration, improvements remain moderate. GCNs effectively capture local spatial correlations but their ability to propagate information across large networks is inherently limited. As the number of road segments increases, the receptive field of shallow GCN layers grows too slowly relative to network size, causing long-range dependencies to remain under-represented in the latent space.
This limitation becomes evident in the WD trend: although WD is low at 16 roads (3.023), it increases sharply to 15.216 at 128 roads. This rise reflects the fact that, in high-dimensional settings, GCNs struggle to maintain global consistency across distant parts of the network. As a result, marginal distributions drift away from the empirical ones, especially in the tails, leading to diminishing performances as dimensionality increases.

nVAE+GAN

The naive VAE+GAN hybrid performs substantially worse than all other models. Tables 10 and 9 show that both BD and WD remain extremely high and grow rapidly with dimensionality (e.g., BD reaches from 4.012 at 16 roads to 15.176 and WD from 5.984 at 16 roads to 16.984 at 128 roads). Although adversarial training can enhance sharpness and local realism, these results indicate unstable behaviour in terms of distributional fidelity. Without a structured prior or explicit mechanisms to preserve global statistical properties, the adversarial component alone is insufficient to capture the multivariate and marginal characteristics of real traffic-speed distributions.

GCop

Despite these architectural enhancements, Copula-based statistical models (GCop) remain the most effective option within the naive-VAE baselines for small to medium network sizes. Their explicit modelling of multivariate dependencies enables them to achieve the lowest BD and WD values across all settings; for instance, BD remains below 0.256 and WD below 0.205 even at 128 roads. However, as discussed in the next paragraph, this advantage comes at the cost of severe computational inefficiency, which ultimately restricts their

Table 9 – Wasserstein Distance (WD) for naive VAE baselines, nVAE+GAN, and GCop across network sizes 16–128 roads.

#roads	nVAE-FC Noise	nVAE-FC Copula	nVAE-GCN Noise
16	3.487 ± 0.174	3.213 ± 0.161	3.098 ± 0.155
32	6.224 ± 0.311	5.942 ± 0.297	5.716 ± 0.286
64	18.932 ± 0.947	17.842 ± 0.892	17.316 ± 0.866
96	15.481 ± 0.774	14.826 ± 0.741	14.312 ± 0.716
128	16.512 ± 0.826	15.893 ± 0.795	15.476 ± 0.774

#roads	nVAE-GCN Copula	nVAE+GAN	GCop
16	3.023 ± 0.151	5.984 ± 0.299	**0.193 ± 0.017**
32	5.611 ± 0.281	8.176 ± 0.409	**0.184 ± 0.011**
64	16.982 ± 0.849	18.304 ± 0.915	**0.224 ± 0.015**
96	14.102 ± 0.705	14.984 ± 0.749	**0.193 ± 0.011**
128	15.216 ± 0.760	16.984 ± 0.849	**0.205 ± 0.005**

applicability to small-scale networks.

To complement the quantitative evaluation, we analyze two-dimensional UMAP projections of real and generated samples across all naive baselines and network sizes. These visualisations, shown in Figure 19, provide a qualitative assessment of latent-space geometry and reveal how well each model preserves the global structure of the data manifold. Models relying on Gaussian priors tend to produce fragmented or overly compact embeddings, especially as dimensionality increases. In contrast, Copula-based variants yield more coherent and topologically consistent projections, with smoother transitions and better separation between latent clusters. The GCop model exhibits the most faithful geometric alignment with the real data, particularly in low-dimensional settings, while the VAE+GAN hybrid shows unstable patterns, confirming its poor distributional fidelity observed in BD and WD metrics.

EVALUATION OF VAE ARCHITECTURES

We extend the previous analysis to a family of progressively more complex VAE architectures: fully connected VAEs (VAE-FC), graph-aware variants (VAE-GCN), β-VAEs with linear and cosine β-scheduling, and a Transformer-based β-VAE (β-T-VAE). The evaluation spans network sizes from 16 up to 1024 roads. In the previous paragraph analysing the performance of the naive VAEs, experiments were limited to network sizes up to 128 roads. Beyond this scale, the divergence in BD and WD metrics already made the behaviour of

Table 10 – Bhattacharyya Distance (BD) for naive VAE baselines, nVAE+GAN, and GCop across network sizes 16–128 roads.

#roads	nVAE-FC Noise	nVAE-FC Copula	nVAE-GCN Noise
16	2.217 ± 0.105	1.703 ± 0.085	1.421 ± 0.071
32	3.312 ± 0.166	2.918 ± 0.146	2.637 ± 0.132
64	6.107 ± 0.305	5.689 ± 0.284	5.325 ± 0.266
96	6.423 ± 0.321	6.015 ± 0.301	5.623 ± 0.281
128	6.918 ± 0.346	6.123 ± 0.306	5.847 ± 0.292

#roads	nVAE-GCN Copula	nVAE+GAN	GCop
16	1.198 ± 0.060	4.012 ± 0.201	**0.081 ± 0.016**
32	1.823 ± 0.091	6.048 ± 0.302	**0.111 ± 0.011**
64	4.812 ± 0.241	9.093 ± 0.455	**0.163 ± 0.008**
96	5.107 ± 0.255	13.084 ± 0.654	**0.206 ± 0.016**
128	5.398 ± 0.270	15.176 ± 0.759	**0.256 ± 0.016**

these models clearly suboptimal, reducing the usefulness of further comparisons. For completeness, results for larger configurations are reported in Appendix B, but they offer limited additional insight into the scalability of naive architectures.

Fully connected VAEs (VAE-FCs) provide a useful baseline but suffer from an intrinsic inability to capture long-range, non-local, and spatial dependencies as dimensionality grows.

VAE-FCs

A comparison with the naive VAE baselines discussed in the previous paragraph clarifies the role of architectural complexity. Fully connected VAEs (VAE-FCs) consistently outperform all naive VAE variants across all network sizes, both in BD and WD. For instance, at 16 roads VAE-FC achieves BD = 0.823 and WD = 2.930, whereas the best naive VAE (nVAE-GCN-Copula) reaches BD = 1.198 and WD = 3.023, and all other naive variants perform substantially worse. The gap widens at larger scales: at 128 roads VAE-FC obtains BD = 4.630 and WD = 14.784, while naive VAE baselines remain significantly higher (e.g., BD = 5.398 and WD = 15.216 for nVAE-GCN-Copula).
Despite outperforming naive VAEs, VAE-FCs still exhibit clear structural limitations. Their latent space rapidly collapses toward overly smooth or isotropic representations, preventing the model from reproducing the heterogeneous and heavy-tailed behaviours of real traffic-speed distributions. This limitation is clearly reflected in the Bhattacharyya and Wasserstein distances (Tables 12 and 11): BD increases from 0.823 at 16 roads to 41.127 at 1024 roads, while WD rises from 2.930 at 16 roads to 31.842 at 1024 roads. These

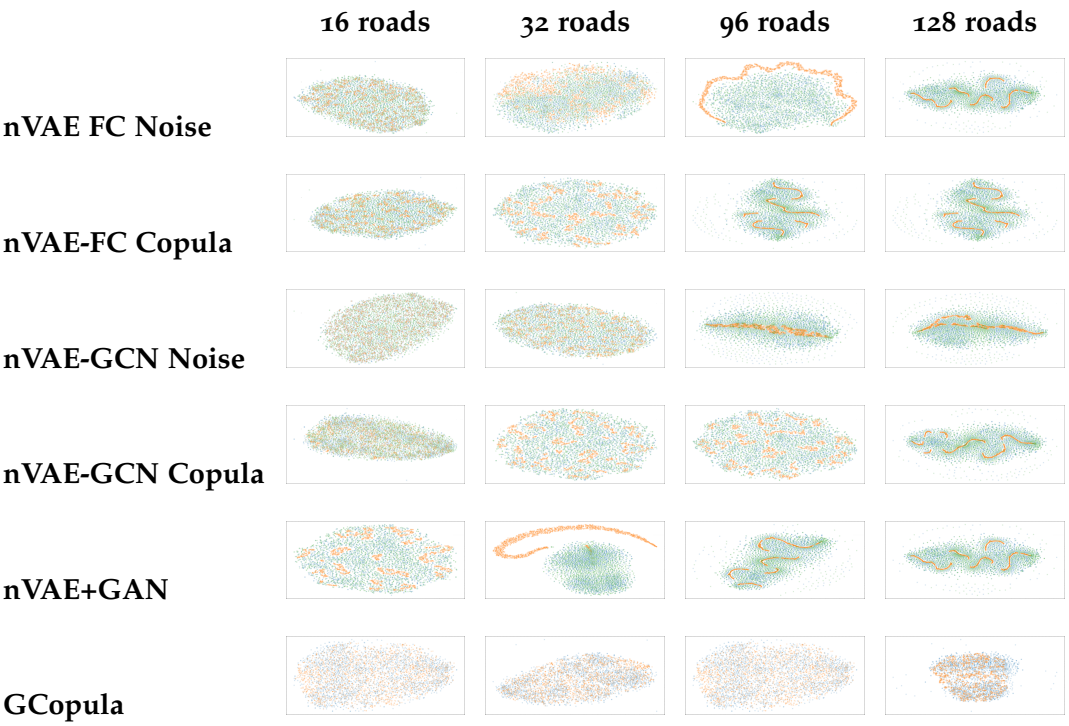


Figure 19 – UMAP comparison for naive VAE-based models across four road configurations (16, 32, 96, 128). The plot shows the real data in blue, the corresponding model-generated data in orange, and the copula-based generated data in green.

trends confirm that VAE-FCs fail to preserve joint dependencies and increasingly misalign marginal distributions, especially in the tails.

VAE-GCN

Incorporating graph convolutions (VAE-GCN) alleviates this limitation. By embedding the road-network adjacency structure directly into the encoder–decoder pipeline, GCNs enhance local spatial coherence and yield noticeable reductions in BD and WD for small-to-medium networks. For example, at 96 roads BD decreases from 4.044 (VAE-FC) to 2.455, and WD from 13.792 to 2.273. However, this improvement remains fundamentally local. As the number of road segments increases, GCNs struggle to propagate information across distant nodes, leading to a saturation effect in which global dependencies remain under-represented and performance reaches a plateau. This tendency is apparent when examining the quantitative results: while VAE-GCN performs well at 64–128 roads, its advantage diminishes rapidly at larger scales. At 256 roads BD rises to 9.084 and WD to 8.098, and at 512 roads the trend continues (BD = 12.276, WD = 12.004), approaching the degradation pattern of VAE-FC. By 1024 roads, the model no longer benefits from graph structure, with BD = 18.507 and WD = 19.782. This trend is also visible in the UMAP projections, where latent embeddings gradually lose

coherence and deviate from the geometric structure of the real data as dimensionality increases.

These considerations motivate the introduction of β-regularised architectures, which encourage richer latent representations and improve the modelling of both joint and marginal behaviours. We first examine the performance of β-GCN-VAE variants equipped with linear and cosine β-scheduling. Both approaches substantially enhance distributional fidelity compared to their non-regularised counterparts, as reflected in lower BD and WD values across all network sizes. At 256 roads, BD decreases from 9.084 (VAE-GCN) to 8.093 with cosine scheduling and to 4.903 with linear scheduling; WD decreases from 8.098 to 3.612 and 1.253 and respectively. Linear β-annealing generally provides steadier improvements in joint fidelity, particularly in larger networks, whereas cosine scheduling promotes smoother latent transitions that benefit marginal reconstruction in intermediate-scale configurations. These effects are also visible in the UMAP projections, where β-VAEs exhibit broader and more coherent coverage of the real data than standard VAE-GCN models.

beta-GCN-VAE

Despite these gains, graph-based architectures remain constrained by their inherently local receptive fields, which limits their ability to propagate information across distant segments in very large networks. This motivates the introduction of the Transformer-based β-VAE (β-T-VAE), which leverages multi-head self-attention to capture long-range spatial dependencies and complex non-linear interactions. The β-T-VAE consistently achieves the lowest BD and WD values across all tested configurations (Tables 12–11).
At 1024 roads, BD remains as low at 0.069 and WD at 0.664, far outperforming all previous presented models. Its UMAP embeddings show extensive overlap with real data, including regions associated with rare traffic events, highlighting its capacity to model both global structure and fine-grained variability. This ability to integrate information across the entire network allows the β-T-VAE to overcome the saturation effects observed in GCN-based models, ensuring scalability without sacrificing quality.
Overall, the β-T-VAE demonstrates that attention mechanisms are not only theoretically appealing but also practically decisive in capturing the multi-scale dependencies that define urban traffic systems. This conclusion is supported by the Wasserstein distance values reported in Table 11: while all other architectures experience a sharp degradation as dimensionality increases, the β-T-VAE maintains remarkably low WD across all network sizes.
For example, at 64 roads the β-T-VAE achieves WD = 0.523, compared to 1.684 for the β-cos-GCN-VAE, 1.242 for the β-lin-GCN-VAE,

beta-T-VAE

and more than 16 for the VAE-FC baseline. This advantage becomes even more evident at larger scales: at 256 roads WD remains as low as 0.508, while VAE-GCN rises to 8.098 and VAE-FC to 17.932. Even at 1024 roads, the most challenging road network configuration, the β-T-VAE preserves WD = 0.664, whereas VAE-GCN reaches 19.782 and VAE-FC exceeds 31.8.
Since the Wasserstein distance is highly sensitive to discrepancies in the marginal distributions, especially in the tails, these consistently low values indicate that the β-T-VAE accurately reproduces the speed distributions of individual road segments, including skewness and heavy-tailed behaviour. The stability of WD across increasing dimensionality therefore confirms that the attention mechanism effectively captures both fine-grained local and long-range spatial correlation.

When compared to Copula-based baselines, the advantages of β-regularised generative architectures become even more pronounced. While GCop performs competitively for small networks, achieving BD = 0.081 and WD = 0.193 at 16 roads, and remaining below BD = 0.26 and WD = 0.21 up to 128 roads, it becomes computationally intractable beyond 200 roads, with training times exceeding 1.5 hours and preventing its application at larger scales. This limitation stems from the exponential growth in complexity when fitting high-dimensional Copulas, which requires inversion of large correlation matrices and extensive sampling.

In contrast, β-GCN-VAEs and β-T-VAE maintain stable performance and millisecond-level generation times across all network sizes, as will be presented in detail in Section 7.2.
Their performance advantage becomes progressively more apparent with increasing dimensionality.
For example, at 256 roads GCop becomes computationally prohibitive and therefore is not reported, whereas the β-lin-GCN-VAE achieves BD = 4.903 and WD = 1.253, and the β-T-VAE reaches BD = 0.050 and WD = 0.508, values that are one to two orders of magnitude lower than those of VAE-FC (BD = 12.004, WD = 17.932) and VAE-GCN (BD = 9.084, WD = 8.098). The contrast becomes even sharper at 512 and 1024 roads: the β-T-VAE maintains BD = 0.054 and WD = 0.560 at 512 roads, and BD = 0.069 and WD = 0.664 at 1024 roads, while VAE-FC deteriorates to BD = 16.707 and WD = 20.384 at 512 roads, and BD = 41.127 and WD = 31.842 at 1024 roads. VAE-GCN follows a similar degradation pattern (BD = 12.276, WD = 12.004 at 512 roads; BD = 18.507, WD = 19.782 at 1024 roads). The comparison underscores a paradigm shift: whereas Copulas excel in small-scale fidelity but collapse under dimensionality, β-regularised VAEs, particularly the Transformer-based variant, bridge the gap between distributional accuracy and computational efficiency. Their

Table 11 – Wasserstein distance for VAE-FC, VAE-GCN, β-lin-GCN-VAE, β-cos-GCN-VAE, β-T-VAE, and GCop across selected network sizes.

#roads	VAE-FC	VAE-GCN	β-lin-GCN-VAE
16	2.930 ± 1.010	1.817 ± 0.219	0.613 ± 0.027
64	16.576 ± 0.196	2.461 ± 0.203	1.242 ± 0.104
96	13.792 ± 0.197	2.273 ± 0.940	0.843 ± 0.087
128	14.784 ± 0.146	3.022 ± 0.982	1.051 ± 0.325
256	17.932 ± 0.198	8.098 ± 1.386	1.253 ± 0.236
512	20.384 ± 0.243	12.004 ± 1.726	1.187 ± 0.218
1024	31.842 ± 0.354	19.782 ± 2.482	1.496 ± 0.264

#roads	β-cos-GCN-VAE	β-T-VAE	GCop
16	0.793 ± 0.039	**0.145 ± 0.005**	0.193 ± 0.017
64	1.684 ± 0.084	0.523 ± 0.002	**0.224 ± 0.015**
96	1.984 ± 0.099	0.571 ± 0.001	**0.193 ± 0.011**
128	2.176 ± 0.108	0.488 ± 0.001	**0.205 ± 0.005**
256	3.612 ± 0.203	**0.508 ± 0.002**	*
512	4.176 ± 0.209	**0.560 ± 0.009**	*
1024	5.612 ± 0.281	**0.664 ± 0.013**	*

ability to maintain BD below 0.07 and WD below 0.70 even at 1024 roads demonstrates a level of scalability unattainable for Copula-based models or traditional VAE architectures. This shift has direct implications for transportation systems, where the ability to generate thousands of realistic scenarios in real time can support adaptive routing and policy evaluation under uncertainty.

SCALING BEHAVIOUR OF BD AND WD

To conclude this comparative study, we examine how the Bhattacharyya Distance (BD) and the Wasserstein Distance (WD) evolve as the dimensionality of the road network increases from 16 to 1024 segments. The two scaling plots reported in Figure 21 provide a global view of the behaviour of all models across several orders of magnitude, highlighting structural differences that are not always evident when inspecting individual configurations.

The BD scaling curve reveals a clear stratification among model families. Naive VAEs, regardless of whether they employ fully connected or graph-based encoders, exhibit a sharp and monotone increase in BD as dimensionality grows. For instance, naiveVAE BD rises from 2.217–1.198 at 16 roads to 6.918–5.398 at 128 roads, confirming their limited capacity to preserve multivariate depen-

Table 12 – Bhattacharyya Distance (BD) for VAE-FC, VAE-GCN, β-lin-GCN-VAE, β-cos-GCN-VAE, β-T-VAE, and GCop across selected network sizes.

#roads	VAE-FC	VAE-GCN	β-lin-GCN-VAE
16	0.823 ± 0.345	0.380 ± 0.062	0.296 ± 0.017
64	4.477 ± 0.093	2.160 ± 0.119	1.957 ± 0.091
96	4.044 ± 0.112	2.455 ± 1.333	1.997 ± 0.251
128	4.630 ± 0.093	3.062 ± 1.137	2.855 ± 0.677
256	12.004 ± 0.198	9.084 ± 1.823	4.903 ± 0.177
512	16.707 ± 0.290	12.276 ± 2.107	5.884 ± 0.243
1024	41.127 ± 0.528	18.507 ± 2.790	7.581 ± 0.378

#roads	β-cos-GCN-VAE	β-T-VAE	GCop
16	0.453 ± 0.023	**0.010 ± 0.001**	0.081 ± 0.016
64	2.087 ± 0.104	**0.018 ± 0.004**	0.163 ± 0.008
96	2.362 ± 0.118	**0.059 ± 0.053**	0.206 ± 0.016
128	2.964 ± 0.294	**0.042 ± 0.017**	0.256 ± 0.016
256	8.093 ± 0.405	**0.050 ± 0.011**	*
512	9.701 ± 0.485	**0.054 ± 0.022**	*
1024	12.693 ± 0.635	**0.069 ± 0.028**	*

dencies in high-dimensional settings: long-range correlations are progressively lost, and the generated joint distributions diverge substantially from the empirical ones. The VAE-FC and VAE-GCN models offer moderate improvements, but their BD values still escalate sharply beyond 256 roads (e.g., BD = 12.004 for VAE-FC and BD = 9.084 for VAE-GCN at 256 roads), confirming that neither fully connected layers nor shallow graph convolutions are sufficient to capture the global structure of large-scale traffic networks.
Copula-based models (GCop) stand out in the low-dimensional regime, achieving the lowest BD values up to 192 roads. Their explicit modelling of dependency structures allows them to maintain high fidelity where dimensionality remains manageable. However, the BD curve for GCop abruptly terminates beyond this point, reflecting the computational intractability of fitting high-dimensional Copulas. This limitation highlights a fundamental trade-off: while Copulas excel in accuracy for small networks, their scalability is severely constrained.

The introduction of β-regularisation significantly alters this landscape. Both β-lin-GCN-VAE and β-cos-GCN-VAE maintain BD values below 16 even at 1024 roads, a substantial improvement over their non-regularised counterparts. Linear scheduling proves particularly effective in large networks, producing smoother and

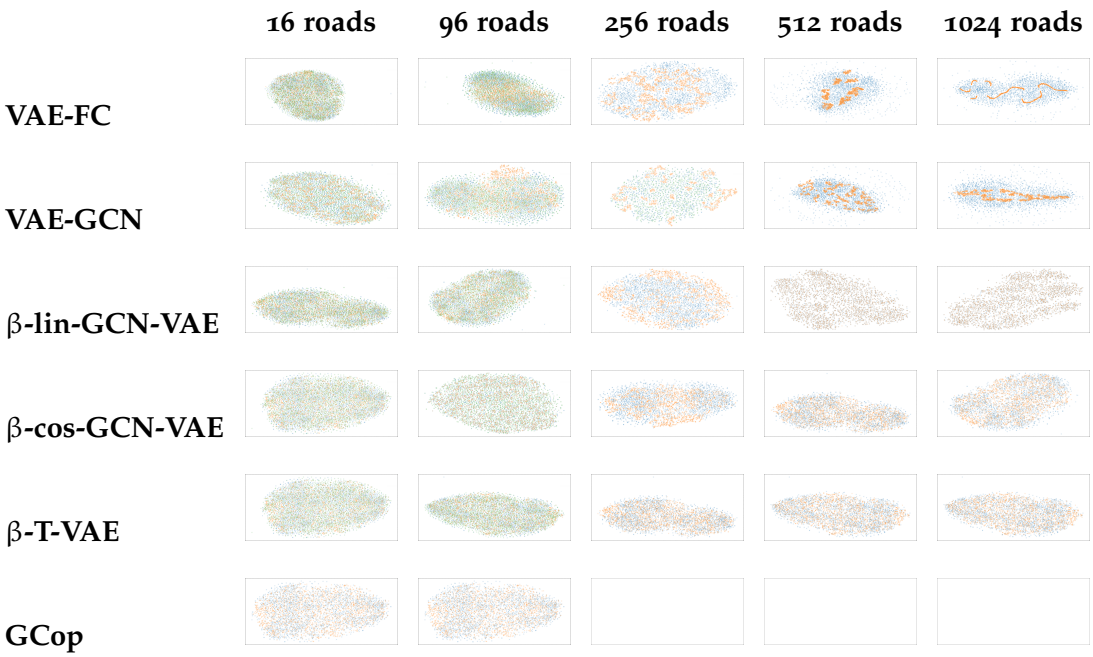


Figure 20 – UMAP comparison for VAE-based models and GCop across five road configurations (16, 96, 256, 512, 1024). The plot shows the real data in blue, the corresponding model-generated data in orange, and the copula-based generated data in green.

more stable BD growth, whereas cosine scheduling offers advantages in intermediate regimes by promoting smoother latent transitions. These results confirm that β-regularisation mitigates posterior collapse and enhances the expressiveness of the latent space, enabling the model to better capture multivariate structure.

The β-T-VAE consistently achieves the lowest BD across all scales, with values remaining below 0.07 even at 1024 roads. This remarkable stability highlights the effectiveness of multi-head self-attention in capturing long-range spatial dependencies and complex non-linear interactions. Unlike GCN-based models, which saturate as receptive fields remain local, the Transformer architecture maintains global awareness across the entire network, preventing the degradation observed in other models.

A complementary perspective emerges from the WD scaling plot, which focuses on marginal fidelity. Naive VAEs and GAN hybrids exhibit rapid WD escalation, with values exceeding 47 at 1024 roads. This indicates poor modelling of skewness, heavy tails, and rare-event behaviours. VAE-FC and VAE+GCN offer partial improvements, but their WD values still rise substantially with dimensionality, reflecting persistent limitations in capturing

marginal distributions.

GCop performs exceptionally well for small networks, maintaining WD values around 0.19-0.26 up to 192 roads. However, as with BD, the method becomes infeasible beyond this point. In contrast, β-GCN-VAEs maintain WD values below 1.5 across all scales, with linear scheduling providing the most stable performance. Cosine scheduling yields slightly higher WD values but offers smoother behaviour in intermediate configurations.

The β-T-VAE demonstrates superior scalability, with WD increasing modestly from 0.145 to 0.664 across the entire range. This stability indicates that the model not only preserves joint dependencies but also accurately reconstructs individual road-segment distributions, including skewness and tail behaviour. The ability to maintain fidelity in both joint and marginal perspectives is crucial for realistic traffic scenario generation.

Overall, the combined analysis of BD and WD scaling curves reinforces the central conclusion of this study: while traditional statistical models and naive VAEs struggle with high-dimensional traffic data, β-regularised architectures, and in particular the Transformer-based β-T-VAE, achieve a robust balance between distributional fidelity and computational scalability. These models remain accurate and efficient even in networks exceeding one thousand road segments.

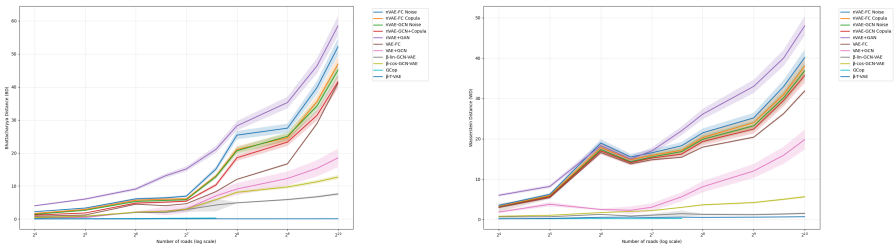


Figure 21 – Left: scaling of Bhattacharyya Distance (BD) across models and network sizes (log scale). Right: scaling of Wasserstein Distance (WD) across models and network sizes (log scale).

7.2 COMPUTATIONAL EFFICIENCY: GENERATION AND SCALABILITY

This section examines the computational efficiency of proposed generative models, focusing on their ability to produce large numbers of traffic speed scenarios. While distributional fidelity is essential for realistic modelling, efficient and scalable generation is equally critical for large-scale scenario analysis and optimization workflows.

Table 13 reports the average generation time for 1,000 scenarios, evaluated on a representative subset of network sizes between 16 and 1024 roads, highlighting substantial differences between deep generative architectures and the Copula-based GCop baseline.

Generative models exhibit consistently negligible generation times, remaining in the millisecond range even for the largest networks. The Transformer-based β-T-VAE achieves approximately 3-4 ms per scenario at 1024 roads, only slightly slower than the β-VAE variants (2.8–2.9 ms) and comparable to simpler architectures such as VAE+GCN and naiveVAEs with Gaussian noise priors (around 2.83 ms). These results demonstrate that the introduction of attention mechanisms does not compromise inference speed, thanks to efficient parallelization and amortised computation in the decoder. Similarly, the linear and cosine β-scheduling strategies yield nearly identical generation times, confirming that regularization primarily affects latent-space structure rather than computational cost.
Naive VAEs with Copula priors incur a modest overhead relative to their noise-prior counterparts, reflecting the additional latent-space dependency modelling. However, this overhead remains limited: generation times increase from approximately 1.15 ms to 1.19 ms at 16 roads, and from 2.83 ms to 36 ms at 1024 roads. Hybrid models such as nVAE+GAN also maintain competitive performance, with generation times around 2.84 ms at 1024 roads, demonstrating that adversarial components do not significantly hinder inference once training is complete.

In contrast, the GCop baseline exhibits prohibitively high generation times that grow rapidly with dimensionality. While generative models operate in the sub-millisecond to millisecond range, GCop requires 30.1 s at 16 roads, 612.7 s at 128 roads, and 2743 s at 192 roads, before becoming infeasible beyond 256 roads due to excessive training and sampling times. This quadratic-to-exponential scaling arises from the computational cost of fitting and sampling high-dimensional Copulas, which involves repeated inversion of large correlation matrices and evaluation of multivariate density functions. As a result, GCop becomes unusable for networks of even moderate size, despite its strong distributional fidelity in low-dimensional settings.
The practical implications of these findings are substantial. In applications involving optimization, simulation, or real-time decision making, generation speed becomes a critical requirement. For example, at 192 roads, GCop requires approximately 2743 s to generate 1000 scenarios, whereas the β-T-VAE produces 1000 scenario in 0.033 s.

To visually summarise these findings, Figure 22 reports the generation time across all models and network sizes, with shaded confidence intervals. The plot confirms the contrast between generative architectures and the Copula baseline: while all VAE-based models maintain sub-5 ms latencies even at 1024 roads, GCop exhibits exponential growth, surpassing 2700 s at 192 roads and becoming infeasible beyond this point. The Transformer-based β-T-VAE shows only a modest increase in generation time as dimensionality grows, remaining within practical bounds across the entire range. These trends confirm that deep models, particularly those equipped with attention mechanisms, offer a scalable and computationally efficient alternative to traditional statistical approaches, enabling real-time scenario generation in large-scale urban networks.

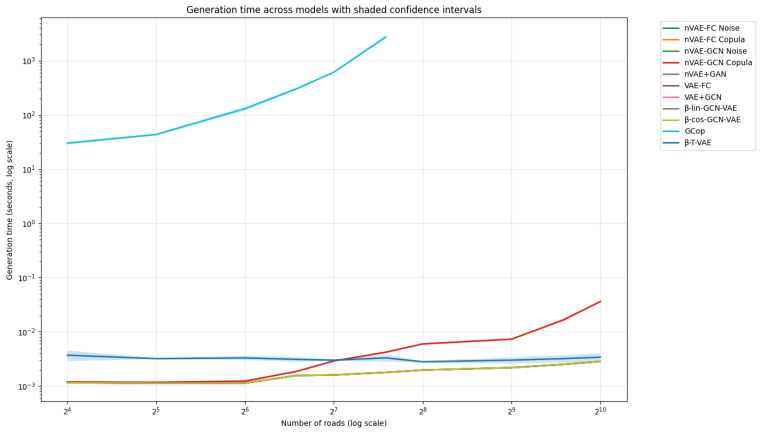


Figure 22 – generation time across models with shaded confidence intervals (log-log scale), showing the computational scalability of deep generative models compared to the exponential growth exhibited by GCop.

Table 13 – Generation time (seconds) for naive VAE variants, VAE baselines, and copula-based models across selected network sizes. The symbol '*' indicates no reported measurement due to excessive training time ($\gg$ 1 hour 30 minutes).

#roads	nVAE-FC Noise	nVAE-FC copula	nVAE-GCN noise
16	1.148e-3 ± 2.6e-5	1.191e-3 ± 2.7e-5	1.151e-3 ± 2.7e-5
96	1.560e-3 ± 2.8e-5	1.873e-3 ± 2.9e-5	1.563e-3 ± 2.9e-5
256	1.965e-3 ± 3.0e-5	5.968e-3 ± 3.1e-5	1.968e-3 ± 3.1e-5
512	2.178e-3 ± 3.2e-5	7.312e-3 ± 6.3e-5	2.181e-3 ± 3.3e-5
1024	2.834e-3 ± 3.8e-5	3.617e-2 ± 6.7e-5	2.837e-3 ± 3.9e-5

#roads	nVAE-GCN copula	nVAE + GAN	VAE-FC
16	1.189e-3 ± 2.7e-5	1.154e-3 ± 2.7e-5	1.149e-3 ± 2.5e-5
96	1.867e-3 ± 2.9e-5	1.567e-3 ± 2.9e-5	1.561e-3 ± 2.7e-5
256	5.972e-3 ± 3.1e-5	1.972e-3 ± 3.1e-5	1.966e-3 ± 2.9e-5
512	7.317e-3 ± 6.3e-5	2.185e-3 ± 3.3e-5	2.179e-3 ± 3.1e-5
1024	3.624e-2 ± 6.7e-5	2.841e-3 ± 3.9e-5	2.835e-3 ± 3.7e-5

#roads	VAE-GCN	β-lin-GCN-VAE	β-cos-GCN-VAE
16	1.152e-3 ± 2.6e-5	1.153e-3 ± 2.7e-5	1.150e-3 ± 2.6e-5
96	1.564e-3 ± 2.8e-5	1.565e-3 ± 2.8e-5	1.562e-3 ± 2.8e-5
256	1.969e-3 ± 3.0e-5	1.970e-3 ± 3.0e-5	1.967e-3 ± 3.0e-5
512	2.182e-3 ± 3.2e-5	2.183e-3 ± 3.2e-5	2.180e-3 ± 3.2e-5
1024	2.838e-3 ± 3.8e-5	2.839e-3 ± 3.8e-5	2.836e-3 ± 3.8e-5

#roads	β-T-VAE	GCop
16	3.7e-3 ± 8.0e-4	3.01e1 ± 1.90
96	3.1e-3 ± 3.0e-4	3.058e2 ± 1.60e1
256	2.8e-3 ± 1.0e-4	*
512	3.0e-3 ± 4.0e-4	*
1024	3.4e-3 ± 6.0e-4	*

8

CONCLUSION

Urban mobility systems are inherently affected by uncertainty. Many analytical and decision-support processes address this challenge through the use of traffic scenarios, which provide structured representations of the possible states of the network. However, generating such scenarios is challenging: the underlying traffic dynamics are highly complex and strongly shaped by spatial interdependencies, and these characteristics must be accurately captured and reproduced in the generated scenarios.

When the generated scenarios fail to capture the true statistical properties of the data, such as variability, correlations, and structural dependencies, any analysis becomes systematically biased. This affects a wide range of tasks, from simulation and stress testing to optimization under uncertainty, that depend on representative scenario sets to evaluate the performance of proposed strategies across diverse conditions.

For this reason, scenario generation constitutes a core modelling component: it provides a statistically faithful representation of uncertainty that supports reliable simulation, robust evaluation, and the development of data-driven decision tools. In transportation, where system behavior is shaped by complex interactions across the network, high-quality scenarios are crucial.

This thesis has examined two core components of the scenario generation problem: the design of scalable generative AI models capable of capturing the complexity and spatial correlation structure of large-scale urban road networks, and the formulation of an evaluation framework for assessing the statistical fidelity and multivariate distributional consistency of the generated synthetic scenarios.

The proposed models, ranging from naive VAE baselines to GCN-enhanced architectures and a non-causal Transformer-based variant, demonstrated strong effectiveness in approximating the underlying distribution of traffic speeds across large-scale networks. Experimental results provided evidence of the scalability of the proposed approaches in settings where traditional statistical methods become impracticable.

A relevant contribution of this thesis concerns the evaluation of generative models. The thesis highlighted the inadequacy of relying on a single metric to assess distributional fidelity in multivariate

settings. To address this limitation, a multi-metric evaluation framework was introduced, combining complementary statistical distances, low-dimensional visualizations, and computational efficiency measures. This approach provides a more comprehensive understanding of generative performance and supports the identification of strengths and weaknesses across different model families.

Finally, the thesis analized the scalability of the proposed models through extensive empirical validation on real-world road datasets. Our analysis on obtained results reveals that VAE-based architectures, particularly those enriched with graph-based and Transformer mechanisms, can reliably scale to networks of hundreds or thousands of segments, outperforming classical baselines in both fidelity and computational feasibility.

Although the proposed approaches were conceived within the domain of transportation systems, their formulation is generalizable to other high-dimensional, spatially structured, and uncertainty-driven applications, such as the generation of pollutant concentration scenarios across an urban area, as we discuss in A.

FUTURE RESEARCH DIRECTIONS

Several perspectives for future research emerge from this work:

- **Integration of temporal dynamics**
  Extending the proposed models to jointly capture spatial and temporal dependencies would enable the generation of traffic scenarios that explicitly account for the evolution of traffic patterns over time.
- **Incorporation of exogenous factors**
  Weather conditions and special events could be incorporated into the generative process to enhance scenario diversity and support more informed decision making.
- **Advanced latent-space modelling**
  Future research may investigate how the latent space can be structured and controlled to generate scenarios with desired characteristics, thereby improving the modelling of rare or extreme events.
- **Scenario generation for multimodal networks**
  Extending the framework to include public transport, cycling infrastructure, and pedestrian flows would broaden its applicability and support integrated mobility planning in smart cities.
- **Operational integration into optimization pipelines**
  Embedding generative models directly within robust or stochas-

tic optimization frameworks could enable end-to-end learning of scenarios tailored to specific decision-making tasks.

- **Generalization to other spatial domains**
  The methodologies introduced in this thesis can be adapted to environmental monitoring, energy systems, and other networked infrastructures where high-dimensional uncertainty plays a central role.

APPENDIX

A

APPLICATION OF VAE-BASED SCENARIO GENERATION TO AIR POLLUTION

The core contributions of this thesis focus on the development of generative models to synthesize realistic traffic speed scenarios on graph-structured road networks. The proposed family of Variational Autoencoders (VAEs) addresses key challenges in capturing non-Gaussian distribution and multivariate data with spatial correlation.

To explore the versatility of this framework beyond traffic scenario generation, this appendix presents a case study in which one of the best-performing models, the β-lin-GCN-VAE introduced in Section 5.7, is adapted to generate air pollution scenarios.
This adaptation shifts from traffic speeds to pollutant concentrations, testing the model's ability to handle datasets with different scales, distributions, and physical dynamics.

Air pollution data share structural similarities with traffic data: both involve multivariate time series defined over spatially correlated nodes (e.g., road segments vs. monitoring stations), exhibit complex non-linear dependencies, and require generating realistic synthetic instances for applications. The β-lin-GCN-VAE captures spatial correlations among monitoring stations through its Graph Convolutional Network (GCN) layer, which explicitly leverages the underlying graph structure.

Key objectives include evaluating transferability, identifying necessary modifications, and assessing performance on single and joint multi-pollutant generation tasks.
This case study highlights the model domain-agnostic potential, paving the way for applications in environmental science, urban planning, and public health.

A.1 DATASET DESCRIPTION

The dataset considered is a collection of air pollution measurements from Seoul, South Korea, provided by the Seoul Metropolitan Government via the "Open Data Plaza" platform[168]. It covers three consecutive years (2017–2019) and reports hourly average concentrations of six atmospheric pollutants across 25 monitoring stations, each associated with a distinct administrative district of the city. The monitored pollutants are:

- NO_2 (*nitrogen dioxide*): Strongly associated with vehicular emissions and urban traffic.
- CO (*carbon monoxide*): Emitted primarily by incomplete combustion processes.
- O_3 (*ozone*): Formed through photochemical reactions, exhibiting complex diurnal patterns.
- PM_{10} (*particulate matter* $\leqslant 10\,\mu m$, $\mu g/m^3$): Coarse particles originating from dust, construction activities, and traffic.
- $PM_{2.5}$ (*particulate matter* $\leqslant 2.5\,\mu m$, $\mu g/m^3$): Fine particles capable of penetrating deep into the respiratory system, posing significant health risks.
- SO_2 (*sulfur dioxide*): Mainly produced by industrial processes and characterized by high chemical reactivity.

This dataset resulted satisfying in respecting the key criteria: simultaneous measurements across spatially distributed points, long-term continuity, and inherent spatial correlations, i.e., between monitoring stations. These characteristics closely parallel the structure of traffic datasets (Section 6.2), where variables exhibit mutual dependencies driven by underlying physical processes.
For the data to be suitable for use within our modelling framework, it is necessary to subject it to an appropriate preprocessing stage.

A.1.1 *Data Preprocessing*

Data preprocessing is performed through a dedicated pipeline that addresses common sensor issues (e.g., instrumental drift or maintenance downtime), consistent with established methodologies for air-quality time-series analysis [48]:

- **Outlier removal**:
  The Interquartile Range (IQR) method with a 1.5 multiplier is applied, removing values outside $[Q_1 - 1.5 \times IQR,\ Q_3 + 1.5 \times IQR]$ as unrealistic instrumental errors.
- **Missing value imputation**:
  Missing entries are reconstructed via *Nearest Neighbor Interpolation* with $k = 5$. Each point is estimated through distance-weighted regression, preserving local spatio-temporal correlations.
- **Min–max scaling**:
  Each pollutant–station pair is scaled to the range $[-1, 1]$, stabilizing VAE training and preventing high-magnitude variables from dominating the loss.

Figure 23 illustrates the preprocessing workflow applied to the PM_{10} pollution data.

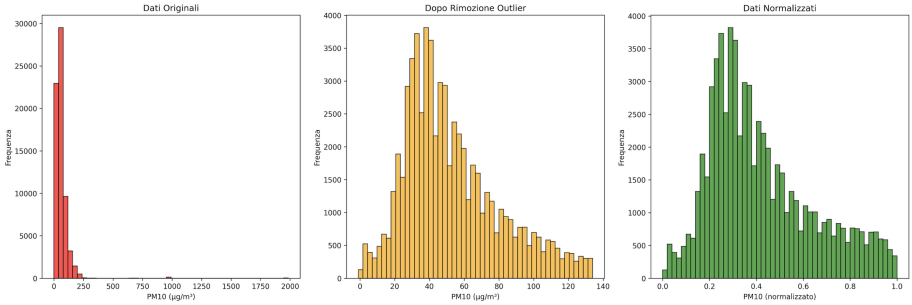


Figure 23 – Preprocessing pipeline applied to the PM_{10} time series: outlier removal, imputation, and min-max scaling.

A.1.2 *Graph Construction*

To capture spatial dependencies among the monitoring stations, the air-quality monitoring network is represented as an undirected graph.
Figure 24 shows both the geographical distribution of the monitoring stations and the corresponding graph structure. The graph is explicitly constructed as an undirected network where each node represents a monitoring station (and its administrative district), while edges encode the expected spatial interactions among stations.

The edges are defined according to three complementary principles reflecting realistic mechanisms of pollutant dispersion and correlation:

- **Geographic proximity**:
  Nearby monitoring stations are connected to model the natural airborne diffusion of pollutants over short distances. Edges are established between these stations whose geographic coordinates are within a predefined proximity threshold (typically computed via the Haversine formula based on the stations' latitude and longitude).
- **District-level correlation**:
  Connections are added between monitoring stations located in districts that share similar urban characteristics (e.g., population density, land use, presence of industrial or commercial zones). This accounts for shared emission sources, traffic patterns, and local meteorological influences that induce similar pollution behaviors even when stations are not immediately adjacent.
- **Road infrastructure connectivity**:
  Major roads, arterial streets, and highways are considered preferential pathways for the transport and dispersion of traffic-related pollutants (in particular NO_2, CO, and particulate matter). Additional edges are therefore created between districts linked by significant road corridors, using information

derived from road network maps (e.g., OpenStreetMap-derived connectivity).

DATA REPRESENTATION Based on this connectivity structure, each node is then enriched with the corresponding pollutant measurements. According to the chosen modelling approach, the graph can be represented in a single-attributed configuration, where each node carries the measurements of a single pollutant, or in a multi-attributed configuration, where each node is associated with a multi-dimensional feature vector containing the values of a set of pollutants recorded at that station at timestamp t.

Multi-attributed graph structure is a node-attributed graph with multiple attributes for each node. Formally, the graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where: $\mathcal{V} = \{v_1, \ldots, v_{25}\}$ is the vertex set with $N = 25$ nodes, each node v_i corresponding to one air-quality monitoring station;
$\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the edge set, with $(v_i, v_j) \in \mathcal{E}$ whenever stations i and j exhibit a spatial interaction induced by geographic proximity, district similarity, and road connectivity.
Each node $v_i \in \mathcal{V}$ is associated with a multivariate feature vector. Let T denote the number of timestamps in the dataset, and F the number of pollutants considered in the feature representation. The node features are therefore represented by the multi-dimensional array

$$X \in \mathbb{R}^{N \times T \times F} \tag{56}$$

where $x_i^t \in \mathbb{R}^F$ denotes the feature vector associated with node v_i at timestamp t.

In the study case of joint multi-polluttants, where we consider the five pollutants (NO_2, CO, O_3, PM_{10}, $PM_{2.5}$), the node-level feature vector takes the form

$$x_i^t = \begin{bmatrix} x_{i,NO_2}^t \\ x_{i,CO}^t \\ x_{i,O_3}^t \\ x_{i,PM_{10}}^t \\ x_{i,PM_{2.5}}^t \end{bmatrix} \tag{57}$$

where each entry $x_{i,p}^t$ represents the preprocessed concentration of pollutant p at station i and timestamp t.

The resulting graph structure enables the model to exploit spatial dependencies driven by atmospheric transport, urban morphology, and emission propagation patterns. The graph construction aligns

with the methodological principles established in the thesis for traffic networks, ensuring coherence and facilitating the transfer of the generative framework across different multivariate settings.

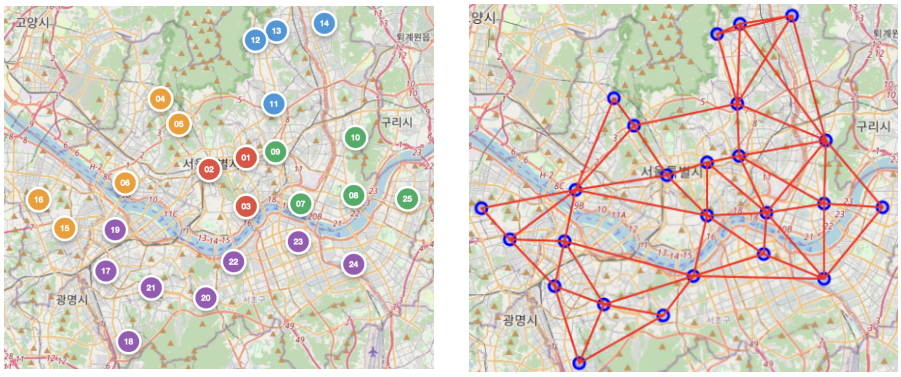


Figure 24 – On the left: geographic distribution of the air quality monitoring stations in Seoul. On the right: graph structure connecting the 25 monitoring stations.

A.2 EXPERIMENTS

To investigate the generative capabilities of β-lin-GCN-VAE model under different modelling assumptions, two complementary experimental settings are considered:

Experiment a: **Single-pollutant scenario generation**.
For each pollutant, a β-lin-GCN-VAE model is trained on its corresponding measurements and generates pollutant-specific scenarios. This setting enables an evaluation of the models' ability to capture spatial dependencies for single pollutants and serves as a baseline for the subsequent multi-pollutant configuration.

Experiment b: **Joint multi-pollutant scenario generation**.
A single β-lin-GCN-VAE model is trained jointly on all pollutants and generates realistic scenarios for all the considered pollutants simultaneously. In this configuration, the model learns shared factors that capture cross-pollutant dependencies. This joint setting reflects more accurately real-world air-quality dynamics, where pollutants interact rather than evolve independently, and supports the generation of physically plausible multi-pollutant fields across the Seoul monitoring network.

These two experimental settings enable a systematic comparison between single-attribute and multi-attribute generative performance, thereby highlighting the advantages of explicitly modelling cross-pollutant dependencies within a graph-structured VAE framework.

A.2.1 *Single-pollutant Scenario Generation*

Single-pollutant scenario generation experiment constitutes the baseline and involves training and evaluating the model independently for each of the six pollutants, treating each of them as a separate single-channel graph attribute.
For every pollutant $p \in \{NO_2, CO, O_3, PM_{10}, PM_{2.5}, SO_2\}$, the node features are defined as:

$$\mathbf{X}^{(p)} \in \mathbb{R}^{N \times T \times 1} \qquad (N = 25,\ T = \text{number of timestamps}), \quad (58)$$

where the scalar value $\mathbf{x}_i^{(t,p)} \in \mathbb{R}$ represents the preprocessed concentration of pollutant p measured at monitoring station i at timestamp t.

This approach enables:

- An assessment of the model's capability to reproduce spatial correlations (atmospheric dispersion, proximity effects, road-corridor influences) induced by the underlying graph structure for each pollutant.
- Direct comparison of performance across pollutants with markedly different dynamics (traffic-dominated NO_2 and CO, photochemically-driven O_3, industrial/episodic SO_2, coarse/fine particulate matter $PM_{10}/PM_{2.5}$).
- Establishment of a clear quantitative benchmark for assessing the added value of the joint multivariate modelling in multi-pollutant scenario generation.

A.2.1.1 *Experimental results*

The experimental results provide a comprehensive assessment of the generative capabilities of β-lin-GCN-VAE across the broader set of atmospheric parameters monitored in Seoul. The evaluation follows the framework introduced in this thesis, as detailed in Section 6.1, which integrates quantitative metrics, such as Wasserstein distance (WD) and Bhattacharyya distance (BD) with qualitative visual analysis based on UMAP. The evaluation is complemented by a comparison with a state-of-the-art baseline, the Gaussian Copula (GCop).

Table 14 – Wasserstein Distance (WD) per pollutant for β-lin-GCN-VAE and GCop.

Pollutant	**β-lin-GCN-VAE**	**GCop**
SO_2	0.206 ± 0.012	0.088 ± 0.013
NO_2	0.049 ± 0.003	0.045 ± 0.003
CO	0.047 ± 0.003	0.043 ± 0.003
O_3	0.056 ± 0.004	0.053 ± 0.004
PM_{10}	0.055 ± 0.004	0.051 ± 0.004
$PM_{2.5}$	0.070 ± 0.005	0.065 ± 0.005

Table 15 – Bhattacharyya Distance (BD) per pollutant for β-lin-GCN-VAE and GCop.

Pollutant	**β-lin-GCN-VAE**	**GCop**
SO_2	0.140 ± 0.010	0.070 ± 0.011
NO_2	0.032 ± 0.003	0.021 ± 0.003
CO	0.030 ± 0.003	0.020 ± 0.003
O_3	0.037 ± 0.003	0.024 ± 0.003
PM_{10}	0.036 ± 0.003	0.023 ± 0.003
$PM_{2.5}$	0.046 ± 0.004	0.030 ± 0.004

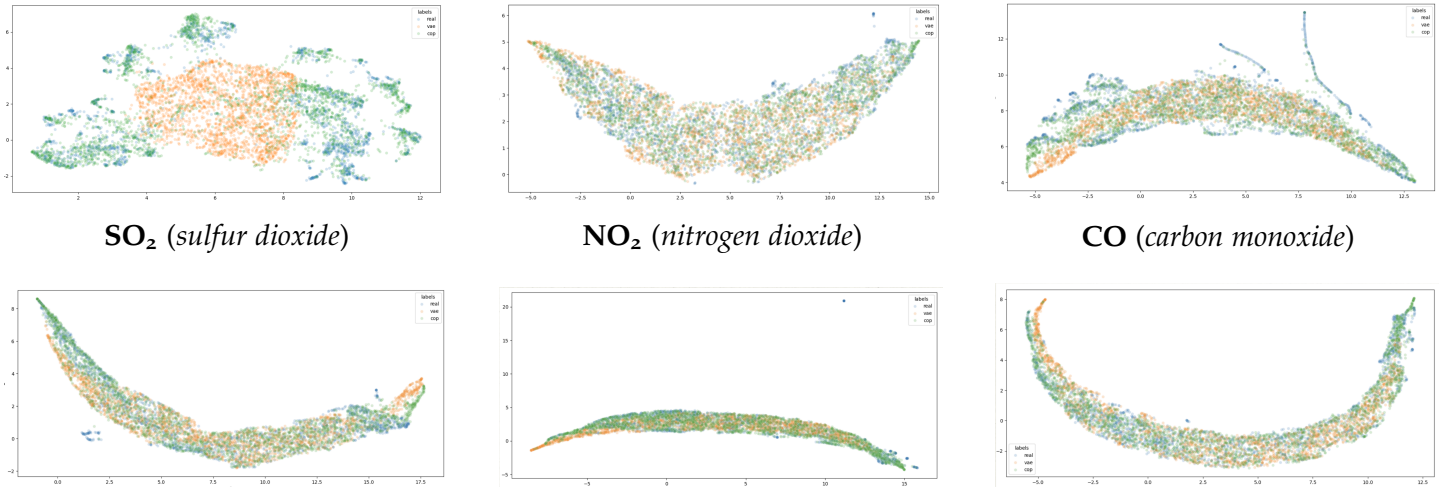


Figure 25 – UMAP plots of pollutant distributions across models for six key air quality indicators. The plot shows the real data in blue, the corresponding model-generated data in orange, and the Copula-based generated data in green.

The quantitative comparison (tables 14 and 15) between the β-lin-GCN-VAE and the GCop model provides a clear assessment of their generative accuracy across all six atmospheric pollutants. Although GCop achieves slightly lower Wasserstein and Bhattacharyya distances for all pollutants, the differences relative to the β-lin-GCN-VAE remain consistently low for five out of six cases. This demonstrates that the proposed model is able to reproduce the empirical distributions with a fidelity comparable to that of the Copula-based generator.

For pollutants such as NO_2, CO, O_3, PM_10, and $PM_2.5$, the performance gap between the two models is narrow, typically within a few thousandths, indicating that the β-lin-GCN-VAE reliably captures the essential statistical properties of these variables despite the intrinsic complexity of the generative task and the structural constraints imposed by the graph-based architecture.

SO_2 is the only pollutant for which the β-lin-GCN-VAE yields a significantly higher discrepancy. This behaviour is consistent across both metrics and reflects the irregular physical dynamics and distinctive chemical properties of SO_2, which make it particularly challenging to model with deep generative architectures.

Overall, the quantitative results show that while GCop provides the best marginal fidelity, the β-lin-GCN-VAE achieves competitive performance across the majority of pollutants, confirming its suitability as a flexible and accurate generative model.

The UMAP visualizations complement the quantitative findings by illustrating how effectively the model reproduces the global structure of the data across all pollutants. For NO_2, CO, O_3, PM_10, and $PM_2.5$ visualizations display a continuous and coherent distribution in which real data, β-lin-GCN-VAE generated scenarios, and Copula-generated scenarios overlap, with no isolated regions, discontinuities, or cluster separations. This consistent overlap indicates that β-lin-GCN-VAE architecture is capable of capturing the overall geometric organization of the empirical distributions, preserving their continuity and large-scale structural patterns.

As expected from the quantitative results, SO_2 behaves differently. UMAP projections show partial separation between real and generated instances, with synthetic points concentrating in regions that do not fully overlap with the empirical distribution. This reinforces the conclusion that SO_2 is the pollutant for which the model exhibits the largest deviation.

Despite GCop's slightly better marginal fidelity, the β-lin-GCN-VAE offers greater modelling flexibility, the ability to incorporate graph-structured information, and a more expressive generative mechanism.

A.2.2 *Joint Multi-pollutant Scenario Generation*

Joint multi-pollutant scenario generation experiment constitutes an advancement with respect to the baseline introduced in Section A.2.1. In this setting, the model is designed to generate scenarios characterized by multiple pollutants simultaneously, thereby capturing spatial correlation and cross-dependencies between the pollutants.

For each monitoring station, we consider the five pollutants: NO_2, CO, O_3, PM_{10} and $PM_{2.5}$. SO_2 pollutant has been excluded because, unlike the others, it is not primarily traffic-related and exhibits a distinct physical behaviour driven by industrial sources and episodic emissions. Its empirical distribution shows high instability and irregular temporal patterns that are incompatible with the statistical structure observed for traffic-driven pollutants, making it unsuitable for inclusion in the multivariate generative framework.

Following Section A.2.1, we compare the scenarios generated by our proposed β-lin-GCN-VAE with those generated by the state-of-the-art baseline Gaussian Copula (GCop).
For GCop, the input data is flattened across stations and pollutants. Each monitoring station provides simultaneous measurements for five pollutants, resulting in a multivariate variable composed of $25 \times 5 = 125$ marginal component. This representation allows the Copula to capture both spatial and cross-pollutant dependencies, leveraging the dataset described in Section A.1.

In contrast, the β-lin-GCN-VAE employs the multi-attributed graph representation introduced in Section A.1.2, preserving the spatial topology of the monitoring network and encoding pollutant concentrations as node attributes.

By construction, GCNs generate node embeddings by combining each node's features with those of its local neighborhood[100]. The GCN layers used in the β-lin-GCN-VAE are specifically designed to operate on graphs with multiple attributes.
While in the traffic speed scenario generation task the node representation was limited to a single feature (the traffic speed), in the multi-attribute pollution-generation setting each node is associated with a vector of several pollutant concentrations. This enables the construction of rich graph embeddings that integrate both spatial and

multi-attribute information.
Figure 26 illustrates the architecture of the modified β-lin-GCN-VAE adapted for joint multi-pollutant scenario generation.

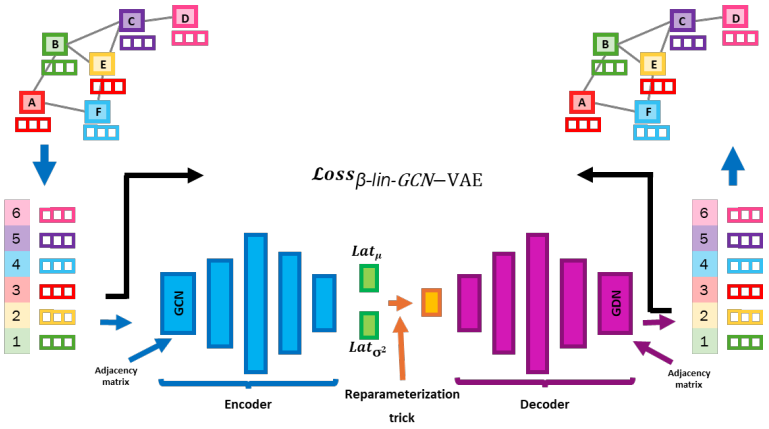


Figure 26 – Architecture of the β-lin-GCN-VAE for multi-pollutant scenario generation.

A.2.2.1 *Experimental results*

The framework introduced in Section 6.1 has been adapted to evaluate performance in the multi-attribute setting, where each node is characterized by multiple pollution variables. In this context, we consider two complementary evaluation strategies:

- **Single pollutant evaluation**: although in the joint multi-pollutant scenario all pollutants are generated simultaneously, this evaluation examines each pollutant independently. This evaluation aims to analyze the model's ability to capture spatial correlations of each pollutant when considered individually, ignoring any cross-attribute dependencies.

- **Joint pollutant evaluation**: the values of all pollutants are evaluated jointly. This analysis considers the distance between the generated and real joint distributions. In this way, the metric captures both spatial correlations and cross-pollutants dependencies .

Table 16 – Wasserstein Distance (WD) per pollutant.

Pollutant	**β-lin-GCN-VAE**	**GCop**
NO_2	0.061 ± 0.004	0.067 ± 0.004
CO	0.053 ± 0.004	0.056 ± 0.004
O_3	0.073 ± 0.004	0.075 ± 0.004
PM_{10}	0.087 ± 0.004	0.093 ± 0.004
$PM_{2.5}$	0.093 ± 0.005	0.087 ± 0.005
mean	0.073 ± 0.004	0.076 ± 0.004

Table 17 – Bhattacharyya Distance (BD) per pollutant.

Pollutant	**β-lin-GCN-VAE**	**GCop**
NO_2	0.053 ± 0.003	0.049 ± 0.003
CO	0.051 ± 0.003	0.049 ± 0.003
O_3	0.038 ± 0.003	0.035 ± 0.003
PM_{10}	0.057 ± 0.003	0.044 ± 0.003
$PM_{2.5}$	0.068 ± 0.004	0.071 ± 0.004
mean	0.053 ± 0.003	0.050 ± 0.003

Table 18 – Joint Wasserstein and Bhattacharyya Distance computed jointly over all components.

Metric	**β-lin-GCN-VAE**	**GCop**
Joint WD	0.068 ± 0.004	0.071 ± 0.004
Joint BD	0.048 ± 0.003	0.046 ± 0.003

The single-pollutant evaluation allows us to assess whether the joint multi-attribute scenario generation model improves the marginal-distribution fidelity of each pollutant, or whether the model preserves the performance observed when pollutants are generated independently.
To evaluate this, we compare the marginal results obtained in the multi-attribute generation setting, where each pollutant is evaluated separately and reported in Tables 16–17, with those obtained in the single-attribute generation setting, where each attribute is generated independently (Tables 14–15).

Across pollutants, the single-attribute setting yields consistently lower discrepancies (e.g., for NO_2: WD 0.049 vs. 0.061, BD 0.032 vs.

0.053; for PM_{10}: WD 0.055 vs. 0.087, BD 0.036 vs. 0.057), indicating that marginal fidelity is generally better preserved when pollutants are generated independently.
Comparisons suggest that joint generation does not offer a clear marginal benefit, indicating that the quality of each individual pollutant tends to be better preserved in the single-attribute setting.

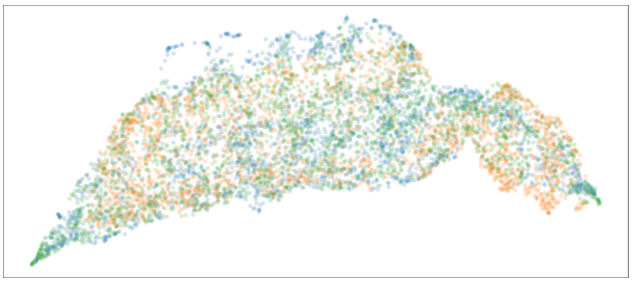


Figure 27 – UMAP embedding in the multi-pollution setting. The plot shows the real data (blue), the samples generated by the β-VAE (orange), and those produced by the Copula-based model (green).

Joint pollutant evaluation allows us to assess whether the model is able to capture spatial correlations and cross-attributes dependencies.
While the single pollutant analysis captures spatial correlations of each individual pollutants, the joint metric (Table 18) is computed on the full multivariate distribution (i.e., without considering pollutants separately) and therefore reflects both spatial structure and cross-attribute dependencies.

A meaningful comparison emerges when compare the joint distances with the average marginal distances reported in Tables 16 and 17.

For instance, the mean marginal Wasserstein Distance in the multi-attribute setting is $\bar{D}_{WD} = 0.073$, whereas the corresponding joint value is $D_{WD}^{joint} = 0.068$. Similarly, the mean marginal Bhattacharyya Distance is $\bar{D}_{BD} = 0.053$, compared to a joint value of $D_{BD}^{joint} = 0.048$.

Since in both cases the BD and WD values for the joint-attribute generation setting are lower than the mean values observed in the single-attribute setting, this indicates that the model effectively capturing the correlations between pollutants as well as their spatial structure. This behaviour confirms that the multi-attribute generative process leverages the underlying joint dependence structure rather than treating each attribute as an independent variable.

This observation is further supported by the UMAP embedding shown in Figure 27, where the real data (blue) forms coherent clusters that are closely approximated by the samples generated by the β-VAE

(orange) and the Copula-based model (green). The visual overlap between real and generated samples suggests that both models are able to reproduce the joint distribution of multiple pollutants with reasonable fidelity.

Overall, the case study confirms that the β-lin-GCN-VAE can be successfully transferred to application domains beyond traffic speed scenario generation, such as the modelling of distributed air pollution scenarios, in a domain-agnostic manner. Despite the substantial differences in scale, distributional properties, and physical behaviour between the two domains, the model maintains stable performance and effectively captures spatial correlations across monitoring stations.

In the single-pollutant setting, the model achieves marginal fidelity comparable to the Gaussian Copula baseline for five out of six pollutants, demonstrating its ability to reproduce pollutant-specific spatial patterns and empirical distributions. In the multi-pollutant setting, although joint generation does not yield a clear improvement in marginal quality, the joint evaluation shows that the model successfully learns cross-pollutant dependencies and the underlying multivariate structure of the data.

These findings validate the transferability and domain-agnostic nature of the proposed VAE framework, highlighting its potential for broader applications in environmental modelling, urban analytics, and data-driven scenario generation beyond the traffic domain.

B

EXTENDED RESULT

This appendix provides the full set of quantitative and qualitative results that support the analyses presented in the thesis. All metrics, generation times, and UMAP-based qualitative results are reported here in their complete form.

B.1 BHATTACHARYYA DISTANCE RESULTS

This section reports the Bhattacharyya distances computed across all models and network sizes, providing a quantitative assessment of generative fidelity. The complete set of values is presented in Table 19.

B.2 WASSERSTEIN DISTANCE RESULTS

This section presents the Wasserstein distances between real and generated samples, enabling a direct comparison of distributional alignment across models. Detailed results are provided in Table 20.

B.3 TIME GENERATION RESULTS

This section documents the generation times for each model, reporting how computational cost scales with network size. The completed measurements are reported in Table 21.

B.4 UMAP RESULTS

This section presents UMAP projections that visualise the structure of real and generated data, offering a qualitative comparison of model behavior. The corresponding plots are shown in Figures 28–33.

Table 19 – Bhattacharyya Distance across models and network sizes. The symbol '*' indicates no reported measurement due to excessive training time (≫ 1 hour 30 minutes)

#roads	nVAE-FC Noise	nVAE-FC Copula	nVAE-GCN Noise	nVAE-GCN+Copula	nVAE+GAN	VAE-FC
16	2.217 ± 0.105	1.703 ± 0.085	1.421 ± 0.071	1.198 ± 0.060	4.012 ± 0.201	0.823 ± 0.345
32	3.312 ± 0.166	2.918 ± 0.146	2.637 ± 0.132	1.823 ± 0.091	6.048 ± 0.302	1.216 ± 0.074
64	6.107 ± 0.305	5.689 ± 0.284	5.325 ± 0.266	4.812 ± 0.241	9.093 ± 0.455	4.477 ± 0.093
96	6.423 ± 0.321	6.015 ± 0.301	5.623 ± 0.281	5.107 ± 0.255	13.084 ± 0.654	4.044 ± 0.112
128	6.918 ± 0.346	6.123 ± 0.306	5.847 ± 0.292	5.398 ± 0.270	15.176 ± 0.759	4.630 ± 0.093
192	15.134 ± 0.757	13.027 ± 0.651	12.823 ± 0.641	10.312 ± 0.515	21.093 ± 1.055	8.299 ± 0.136
256	25.398 ± 1.270	21.046 ± 1.052	20.689 ± 1.034	18.537 ± 0.927	28.217 ± 1.411	12.004 ± 0.198
512	27.513 ± 1.376	24.623 ± 1.231	25.048 ± 1.252	23.298 ± 1.165	35.284 ± 1.764	16.707 ± 0.290
768	39.842 ± 1.992	35.617 ± 1.781	34.228 ± 1.711	31.406 ± 1.570	46.372 ± 2.318	28.914 ± 0.412
1024	52.176 ± 2.609	46.903 ± 2.345	45.112 ± 2.256	41.587 ± 2.079	58.493 ± 2.925	41.127 ± 0.528

#roads	VAE+GCN	β-lin-GCN-VAE	β-cos-GCN-VAE	GCop	β-T-VAE
16	0.380 ± 0.062	0.296 ± 0.017	0.453 ± 0.023	0.081 ± 0.016	**0.010 ± 0.001**
32	0.439 ± 0.094	0.777 ± 0.049	0.897 ± 0.045	0.111 ± 0.011	**0.028 ± 0.020**
64	2.160 ± 0.119	1.957 ± 0.091	2.087 ± 0.104	0.163 ± 0.008	**0.018 ± 0.004**
96	2.455 ± 1.333	1.997 ± 0.251	2.362 ± 0.118	0.206 ± 0.016	**0.059 ± 0.053**
128	3.062 ± 1.137	2.855 ± 0.677	2.964 ± 0.294	0.256 ± 0.016	**0.042 ± 0.017**
192	7.003 ± 1.542	4.236 ± 1.912	5.873 ± 0.294	0.307 ± 0.016	**0.064 ± 0.031**
256	9.084 ± 1.823	4.903 ± 0.177	8.093 ± 0.405	*	**0.050 ± 0.011**
512	12.276 ± 2.107	5.884 ± 0.243	9.701 ± 0.485	*	**0.054 ± 0.022**
768	15.348 ± 2.450	6.732 ± 0.310	11.284 ± 0.528	*	**0.061 ± 0.025**
1024	18.507 ± 2.790	7.581 ± 0.378	12.693 ± 0.635	*	**0.069 ± 0.028**

Table 20 – Wasserstein distance across models and network sizes. The symbol '*' indicates no reported measurement due to excessive training time ($\gg$ 1 hour 30 minutes)

#roads	nVAE-FC Noise	nVAE-FC Copula	nVAE-GCN Noise	nVAE-GCN Copula	nVAE+GAN	VAE-FC
16	3.487 ± 0.174	3.213 ± 0.161	3.098 ± 0.155	3.023 ± 0.151	5.984 ± 0.299	2.930 ± 1.010
32	6.224 ± 0.311	5.942 ± 0.297	5.716 ± 0.286	5.611 ± 0.281	8.176 ± 0.409	5.378 ± 0.289
64	18.932 ± 0.947	17.842 ± 0.892	17.316 ± 0.866	16.982 ± 0.849	18.304 ± 0.915	16.576 ± 0.196
96	15.481 ± 0.774	14.826 ± 0.741	14.312 ± 0.716	14.102 ± 0.705	14.984 ± 0.749	13.792 ± 0.197
128	16.512 ± 0.826	15.893 ± 0.795	15.476 ± 0.774	15.216 ± 0.760	16.984 ± 0.849	14.784 ± 0.146
192	18.273 ± 0.914	17.164 ± 0.858	16.812 ± 0.840	16.182 ± 0.809	21.688 ± 1.099	15.411 ± 0.170
256	21.476 ± 1.074	20.312 ± 1.016	19.874 ± 0.994	19.316 ± 0.966	26.107 ± 1.305	17.932 ± 0.198
512	25.173 ± 1.259	23.929 ± 1.199	23.212 ± 1.161	22.418 ± 1.121	32.978 ± 1.649	20.384 ± 0.243
768	32.914 ± 1.646	30.784 ± 1.549	30.127 ± 1.506	29.457 ± 1.473	39.954 ± 1.999	26.197 ± 0.298
1024	40.127 ± 2.006	37.854 ± 1.899	36.842 ± 1.842	35.671 ± 1.783	47.733 ± 2.399	31.842 ± 0.354

#roads	VAE+GCN	β-lin-GCN-VAE	β-cos-GCN-VAE	GCop	β-T-VAE
16	1.817 ± 0.219	0.613 ± 0.027	0.793 ± 0.039	0.193 ± 0.017	**0.145 ± 0.005**
32	3.748 ± 0.521	0.661 ± 0.025	1.024 ± 0.051	**0.184 ± 0.011**	0.247 ± 0.142
64	2.461 ± 0.203	1.242 ± 0.104	1.684 ± 0.084	**0.224 ± 0.015**	0.523 ± 0.002
96	2.273 ± 0.940	0.843 ± 0.087	1.984 ± 0.099	**0.193 ± 0.011**	0.571 ± 0.001
128	3.022 ± 0.982	1.051 ± 0.325	2.176 ± 0.108	**0.205 ± 0.005**	0.488 ± 0.001
192	5.603 ± 1.102	1.422 ± 0.748	2.984 ± 0.149	**0.263 ± 0.028**	0.563 ± 0.004
256	8.098 ± 1.386	1.253 ± 0.236	3.612 ± 0.203	*	**0.508 ± 0.002**
512	12.004 ± 1.726	1.187 ± 0.218	4.176 ± 0.209	*	**0.560 ± 0.009**
768	15.893 ± 2.104	1.342 ± 0.241	4.984 ± 0.249	*	**0.612 ± 0.011**
1024	19.782 ± 2.482	1.496 ± 0.264	5.612 ± 0.281	*	**0.664 ± 0.013**

Table 21 – Generation time for different models on the Roads dataset. The symbol '*' indicates no reported measurement due to excessive training time (≫ 1 hour 30 minutes)

#roads	nVAE-FC Noise	nVAE-FC copula	nVAE-GCN noise	nVAE-GCN copula	nVAE + GAN	VAE-FC
16	1.148e-3 ± 2.6e-5 s	1.191e-3 ± 2.7e-5 s	1.151e-3 ± 2.7e-5 s	1.189e-3 ± 2.7e-5 s	1.154e-3 ± 2.7e-5 s	1.149e-3 ± 2.5e-5 s
32	1.120e-3 ± 2.3e-5 s	1.167e-3 ± 2.4e-5 s	1.123e-3 ± 2.4e-5 s	1.174e-3 ± 2.4e-5 s	1.125e-3 ± 2.4e-5 s	1.121e-3 ± 2.2e-5 s
64	1.130e-3 ± 2.6e-5 s	1.214e-3 ± 2.7e-5 s	1.134e-3 ± 2.7e-5 s	1.236e-3 ± 2.7e-5 s	1.136e-3 ± 2.7e-5 s	1.132e-3 ± 2.5e-5 s
96	1.560e-3 ± 2.8e-5 s	1.873e-3 ± 2.9e-5 s	1.563e-3 ± 2.9e-5 s	1.867e-3 ± 2.9e-5 s	1.567e-3 ± 2.9e-5 s	1.561e-3 ± 2.7e-5 s
128	1.595e-3 ± 2.9e-5 s	2.897e-3 ± 5.6e-5 s	1.599e-3 ± 3.0e-5 s	2.903e-3 ± 5.6e-5 s	1.602e-3 ± 3.0e-5 s	1.596e-3 ± 2.8e-5 s
192	1.780e-3 ± 2.7e-5 s	4.223e-3 ± 2.8e-5 s	1.783e-3 ± 2.8e-5 s	4.197e-3 ± 2.8e-5 s	1.787e-3 ± 2.8e-5 s	1.781e-3 ± 2.6e-5 s
256	1.965e-3 ± 3.0e-5 s	5.968e-3 ± 3.1e-5 s	1.968e-3 ± 3.1e-5 s	5.972e-3 ± 3.1e-5 s	1.972e-3 ± 3.1e-5 s	1.966e-3 ± 2.9e-5 s
512	2.178e-3 ± 3.2e-5 s	7.312e-3 ± 6.3e-5 s	2.181e-3 ± 3.3e-5 s	7.317e-3 ± 6.3e-5 s	2.185e-3 ± 3.3e-5 s	2.179e-3 ± 3.1e-5 s
768	2.507e-3 ± 3.5e-5 s	1.645e-2 ± 6.5e-5 s	2.510e-3 ± 3.6e-5 s	1.668e-2 ± 6.5e-5 s	2.514e-3 ± 3.6e-5 s	2.508e-3 ± 3.4e-5 s
1024	2.834e-3 ± 3.8e-5 s	3.617e-2 ± 6.7e-5 s	2.837e-3 ± 3.9e-5 s	3.624e-2 ± 6.7e-5 s	2.841e-3 ± 3.9e-5 s	2.835e-3 ± 3.7e-5 s

#nodes	VAE + GCN	β-lin-GCN-VAE	β-cos-GCN-VAE	GCop	β-T-VAE
16	1.152e-3 ± 2.6e-5 s	1.153e-3 ± 2.7e-5 s	1.150e-3 ± 2.6e-5 s	3.01e1 ± 1.90 s	3.7e-3 ± 8.0e-4 s
32	1.124e-3 ± 2.3e-5 s	1.125e-3 ± 2.3e-5 s	1.122e-3 ± 2.3e-5 s	4.35e1 ± 2.10 s	3.2e-3 ± 1.0e-4 s
64	1.135e-3 ± 2.6e-5 s	1.132e-3 ± 2.6e-5 s	1.133e-3 ± 2.6e-5 s	1.303e2 ± 1.30e1 s	3.3e-3 ± 3.0e-4 s
96	1.564e-3 ± 2.8e-5 s	1.565e-3 ± 2.8e-5 s	1.562e-3 ± 2.8e-5 s	3.058e2 ± 1.60e1 s	3.1e-3 ± 3.0e-4 s
128	1.600e-3 ± 2.9e-5 s	1.598e-3 ± 2.9e-5 s	1.597e-3 ± 2.9e-5 s	6.127e2 ± 1.84e1 s	3.0e-3 ± 1.0e-4 s
192	1.784e-3 ± 2.7e-5 s	1.784e-3 ± 2.7e-5 s	1.782e-3 ± 2.7e-5 s	2.743e3 ± 2.10e2 s	3.3e-3 ± 5.0e-4 s
256	1.969e-3 ± 3.0e-5 s	1.970e-3 ± 3.0e-5 s	1.967e-3 ± 3.0e-5 s	*	2.8e-3 ± 1.0e-4 s
512	2.182e-3 ± 3.2e-5 s	2.183e-3 ± 3.2e-5 s	2.180e-3 ± 3.2e-5 s	*	3.0e-3 ± 4.0e-4 s
768	2.511e-3 ± 3.5e-5 s	2.512e-3 ± 3.5e-5 s	2.509e-3 ± 3.5e-5 s	*	3.2e-3 ± 5.0e-4 s
1024	2.838e-3 ± 3.8e-5 s	2.839e-3 ± 3.8e-5 s	2.836e-3 ± 3.8e-5 s	*	3.4e-3 ± 6.0e-4 s

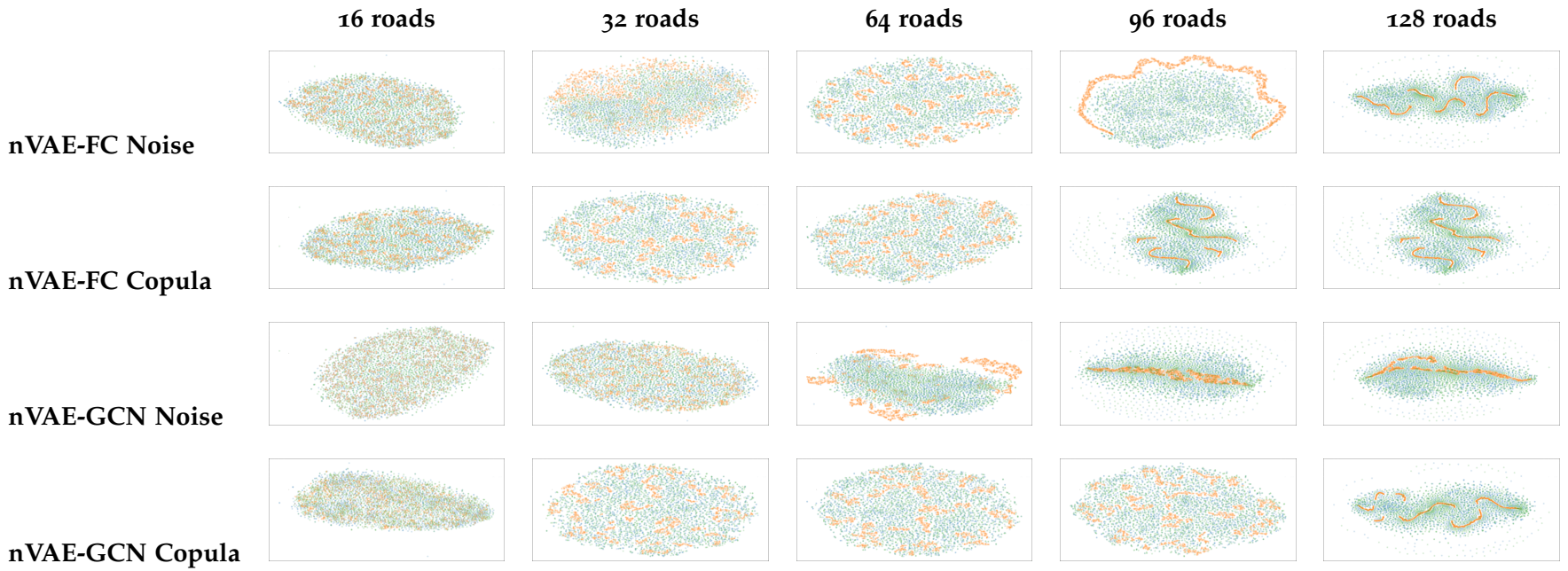


Figure 28 – UMAP for models across five road configurations (16, 32, 64, 96, 128).The plot shows the real data in blue, the corresponding model-generated data in orange, and the copula-based generated data in green.

Figure 29 – UMAP for models across five road configurations (192, 256, 512, 768, 1024).The plot shows the real data in blue, the corresponding model-generated data in orange, and the copula-based generated data in green.

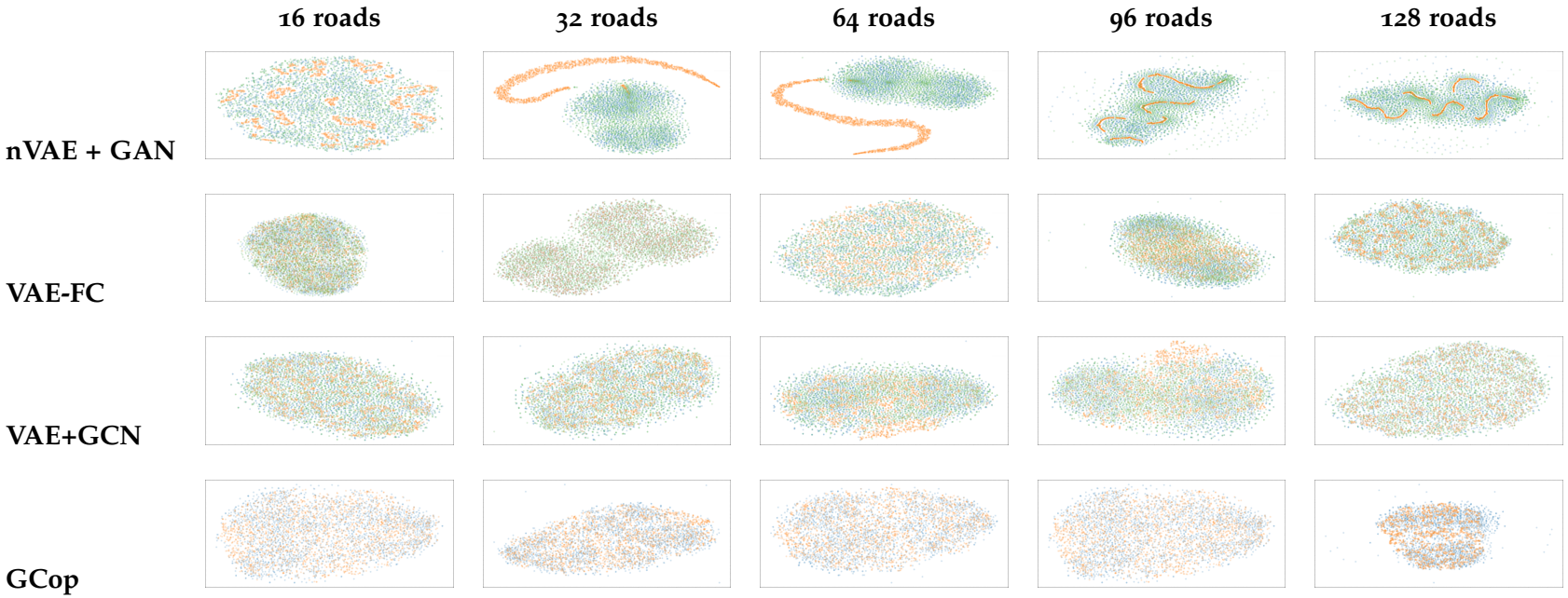


Figure 30 – UMAP for models across five road configurations (16, 32, 64, 96, 128).The plot shows the real data in blue, the corresponding model-generated data in orange, and the copula-based generated data in green.

Figure 31 – UMAP for models across five road configurations (192, 256, 512, 768, 1024).The plot shows the real data in blue, the corresponding model-generated data in orange, and the copula-based generated data in green.

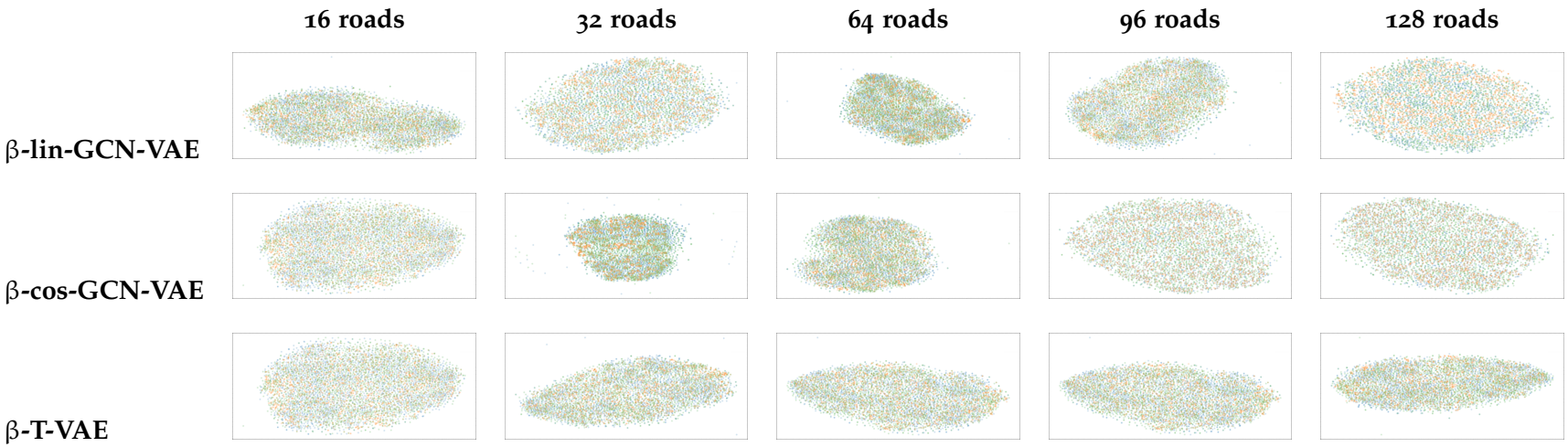


Figure 32 – UMAP for models across five road configurations (16, 32, 64, 96, 128). The plot shows the real data in blue, the corresponding model-generated data in orange, and the copula-based generated data in green.

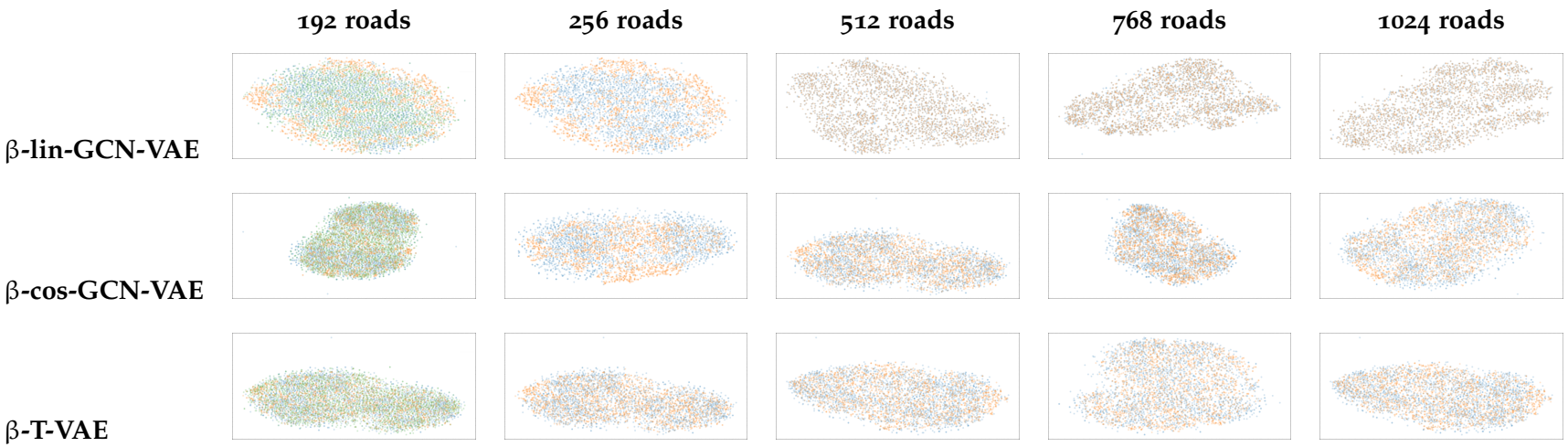


Figure 33 – UMAP for models across five road configurations (192, 256, 512, 768, 1024). The plot shows the real data in blue, the corresponding model-generated data in orange, and the copula-based generated data in green.

BIBLIOGRAPHY

[1] Kjersti Aas, Claudia Czado, Arnoldo Frigessi, and Henrik Bakken. « Pair-copula constructions of multiple dependence. » In: *Insurance: Mathematics and Economics* 44.2 (2009), pp. 182–198. ISSN: 0167-6687.

[2] Mohammad Abudayah, Omar Alomari, and Torsten Sander. « Incidence matrices and line graphs of mixed graphs. » In: *Special Matrices* 11 (Jan. 2023).

[3] Syed M Ali and Samuel D Silvey. « A general class of coefficients of divergence of one distribution from another. » In: *Journal of the Royal Statistical Society. Series B (Methodological)* 28.1 (1966), pp. 131–142.

[4] Paulo Anciaes, Yan Cheng, and Stephen John Watkins. « Policy measures to reduce road congestion: What worked? » In: *Journal of Transport & Health* 41 (2025), p. 101984.

[5] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I. Jordan. « An Introduction to MCMC for Machine Learning. » In: *Machine Learning* 50.1 (2003), pp. 5–43.

[6] Martin Arjovsky, Soumith Chintala, and Leon Bottou. « Wasserstein GAN. » In: *International Conference on Learning Representations (ICLR 2017)*. 2017.

[7] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. « Layer normalization. » In: *arXiv preprint arXiv:1607.06450* (2016).

[8] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. « End-to-end Optimized Image Compression. » In: *5th International Conference on Learning Representations, ICLR 2017*. 2017.

[9] Güzin Bayraksan and David P Morton. « Assessing solution quality in stochastic programs. » In: *Mathematical Programming* 108 (2006), pp. 189–209.

[10] Raphael Becker. « Generation of Time-Coupled Wind Power Infeed Scenarios Using a Gaussian Copula Approach. » In: *IEEE Transactions on Sustainable Energy* 9.3 (2018), pp. 1298–1306.

[11] Lowell W. Beineke and Jay S. Bagga. « Line graphs and line digraphs. » In: *Selected topics in graph theory*. Academic Press, 1978, pp. 271–305.

[12] Lowell W. Beineke, Martin Charles Golumbic, and Robin Wilson. *Topics in algorithmic graph theory*. Cambridge University Press, 2021.

[13] Yoshua Bengio. « Practical recommendations for gradient-based training of deep architectures. » In: *Neural networks: Tricks of the trade*. Springer, 2012, pp. 437–478.

[14] A. Bhattacharyya. « On a Measure of Divergence between Two Statistical Populations Defined by Their Probability Distributions. » In: *Bulletin of the Calcutta Mathematical Society* 35 (1943), pp. 99–109.

[15] Guillem Boquet, Antoni Morell, Javier Serrano, and Jose Lopez Vicario. « A variational autoencoder solution for road traffic forecasting systems: Missing data imputation, dimension reduction, model selection and anomaly detection. » In: *Transportation Research Part C: Emerging Technologies* 115 (June 2020), p. 102622.

[16] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis. « Compressed sensing using generative models. » In: *International conference on machine learning*. PMLR. 2017, pp. 537–546.

[17] Léon Bottou, Frank E Curtis, and Jorge Nocedal. « Optimization methods for large-scale machine learning. » In: *SIAM Review* 60.2 (2018), pp. 223–311.

[18] Hervë Bourlard and Yves Kamp. « Auto-association by multilayer perceptrons and singular value decomposition. » In: *Biological Cybernetics* 59.4 (1988), pp. 291–294.

[19] Samuel R Bowman, Luke Vilnis, Oriol Vinyals Ma, Jun Dai, Rafal Jozefowicz, and Samy Bengio. « Generating sentences from a continuous space. » In: *International Conference on Learning Representations (ICLR 2016)*. 2016.

[20] George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. *Time Series Analysis: Forecasting and Control*. 5th. Hoboken, NJ: John Wiley & Sons, 2015.

[21] Piet Broersen and Stijn De Waele. « Finite Sample Properties of ARMA Order Selection. » In: *Instrumentation and Measurement, IEEE Transactions on* 53 (July 2004), pp. 645 –651.

[22] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. « Spectral networks and deep locally connected networks on graphs. » In: *International Conference on Learning Representations (ICLR 2014)*. 2014.

[23] Michele Carbonera, Michele Ciavotta, and Enza Messina. « Driving into Uncertainty: An Adversarial Generative Approach for Multivariate Scenario Generation. » In: *Proceedings of the 2023 IEEE International Conference on Big Data (BigData)*. 2023, pp. 2578–2587.

[24] Michele Carbonera, Michele Ciavotta, and Enza Messina. « Variational Autoencoders and Generative Adversarial Networks for Multivariate Scenario Generation. » In: *Data Science for Transportation* 6.3 (2024).

[25] Michele Carbonera, Michele Ciavotta, and Enza Messina. « Generative AI for Traffic Scenarios: A GCN-VAE Model. » In: *Book of Short Papers 2025, Conference of the 12th Scientific Meeting of the Statistics for the Evaluation and Quality of Services Group of the Italian Statistical Society (SVQS)*. 2025, pp. 1288–1294.

[26] Cristiano Cervellera and Danilo Macciò. « A copula-based scenario generation framework for urban traffic models. » In: *Transportation Research Part C: Emerging Technologies* 135 (2022), p. 103492.

[27] Athanasios Chartsias, Thomas Joyce, George Papanastasiou, Scott Semple, Mark Williams, David E. Newby, Rohan Dharmakumar, and Sotirios A. Tsaftaris. « Disentangled representation learning in cardiac image analysis. » In: *Medical Image Analysis* 58 (2019), p. 101535.

[28] Peiyuan Chen, Troels Pedersen, Birgitte Bak-Jensen, and Z. Chen. « ARIMA-Based Time Series Model of Stochastic Wind Power Generation. » In: *Power Systems, IEEE Transactions on* 25 (June 2010), pp. 667 –676.

[29] Yize Chen, Yishen Wang, Daniel S. Kirschen, and Baosen Zhang. « Model-Free Renewable Scenario Generation Using Generative Adversarial Networks. » In: *IEEE Transactions on Power Systems* 33.3 (2018), pp. 3265–3275.

[30] Rewon Child. « Very deep vaes generalize autoregressive models and can outperform them on images. » In: *International Conference on Learning Representations (ICLR 2021)*. 2021.

[31] Xiaochen Chou and Enza Messina. « Problem-driven scenario generation for stochastic programming problems: a survey. » In: *Algorithms* 16.10 (2023), p. 479.

[32] Ondřej Cífka, Alexey Ozerov, Umut Simsekli, and Gaël Richard. « Self-Supervised VQ-VAE for One-Shot Music Style Transfer. » In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2021, pp. 3385–3389.

[33] Gabriele Corso, Luca Cavalleri, Davide Beaini, Pietro Liò, and Cesare Alippi. « Principal neighborhood aggregation for graph neural networks. » In: *International Conference on Neural Information Processing Systems (NeurIPS 2020)*. 2020.

[34] Imre Csiszár. « Information-type measures of difference of probability distributions and indirect observations. » In: *Studia Scientiarum Mathematicarum Hungarica* 2 (1967), pp. 299–318.

[35] George Cybenko. « Approximation by superpositions of a sigmoidal function. » In: *Mathematics of control, signals and systems* 2.4 (1989), pp. 303–314.

[36] Hanjun Dai, Chengtao Li, Connor W. Coley, Bo Dai, and Le Song. « Retrosynthesis Prediction with Conditional Graph Logic Network. » In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 32. 2019, pp. 8876–8886.

[37] Abdelkader Dairi, Fouzi Harrou, and Ying Sun. « Deep Generative Learning-based 1-SVM Detectors for Unsupervised COVID-19 Infection Detection Using Blood Tests. » In: *IEEE Transactions on Instrumentation and Measurement* PP (2021), pp. 1–11.

[38] Abdelkader Dairi, Fouzi Harrou, Ying Sun, and Sofiane Khadraoui. « Short-Term Forecasting of Photovoltaic Solar Power Production Using Variational Auto-Encoder Driven Deep Learning Approach. » In: *Applied Sciences* 10 (Nov. 2020).

[39] Michaël Defferrard, Xavier Bazerque, and Pierre Vandergheynst. « Convolutional neural networks on graphs with fast localized spectral filtering. » In: *Advances in neural information processing systems (NeurIPS 2016)*. 2016, pp. 3844–3852.

[40] Boris Defourny, Damien Ernst, and Louis Wehenkel. « Scenario Trees and Policy Selection for Multistage Stochastic Programming Using Machine Learning. » In: *INFORMS Journal on Computing* 25.3 (2012), pp. 488–501.

[41] Andreas Demetriou, Henrik Allsvag, Sadegh Rahrovani, and Morteza Haghir Chehreghani. « Generation of Driving Scenario Trajectories with Generative Adversarial Networks. » In: *Proceedings of the 23rd International Conference on Intelligent Transportation Systems (ITSC)*. Sept. 2020, pp. 1–6.

[42] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. « ImageNet: A large-scale hierarchical image database. » In: *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255.

[43] Nico Di Domenica, Cormac Lucas, Gautam Mitra, and Patrick Valente. « Scenario generation for stochastic programming and simulation: A modelling perspective. » In: *Ima Journal of Management Mathematics - IMA J MANAG MATH* 20 (Aug. 2007), pp. 1–38.

[44] Guzmán Díaz, Javier Gómez-Aleixandre, and Jose Coto. « Wind power scenario generation through state-space specifications for uncertainty analysis of wind power plants. » In: *Applied Energy* 162 (Jan. 2016), pp. 21–30.

[45] Wei Dong, Xianqing Chen, and Qiang Yang. « Data-driven scenario generation of renewable energy production based on controllable generative adversarial networks with interpretability. » In: *Applied Energy* 308 (2022), p. 118387. ISSN: 0306-2619.

[46] Marie-Christine Düker, David S. Matteson, Ruey S. Tsay, and Ines Wilms. « Vector AutoRegressive Moving Average Models: A Review. » In: *WIREs Computational Statistics* 16.1 (2024), e1635.

[47] Andreas Eichhorn and Werner Römisch. « Stability of multistage stochastic programs incorporating polyhedral risk measures. » In: *Optimization* 57.2 (2008), pp. 295–318.

[48] Osman Elmenawy. « Detecting Environmental Anomalies: Variational Autoencoder-Based Analysis of Air Quality Time Series Data. » In: *International Journal of Intelligent Systems and Applications in Engineering* 12.4 (2024), pp. 3687–3695.

[49] Paul Embrechts, Filip Lindskog, and Alexander McNeil. « Modelling dependence with copulas and applications to risk management. » In: *Handbook of heavy tailed distributions in finance* 8.1 (2003), pp. 329–384.

[50] Paul Embrechts, Alexander McNeil, and Daniel Straumann. « Correlation and Dependence in Risk Management: Properties and Pitfalls. » In: *Risk Management: Value at Risk and Beyond* (2002), pp. 176–223.

[51] Mohammad Fattahi, Ehsan Keyvanshokooh, Devika Kannan, and Kannan Govindan. « Resource Planning Strategies for Healthcare Systems During a Pandemic. » In: *European Journal of Operational Research* 304.1 (2023), pp. 192–206.

[52] Lan Feng, Quanyi Li, Zhenghao Peng, Yunfei Zhang, and Hang Zhao. « TrafficGen: Learning to Generate Diverse and Realistic Traffic Scenarios. » In: *IEEE Transactions on Intelligent Transportation Systems* 24.12 (2023), pp. 14405–14418.

[53] George S. Fishman. *Monte Carlo: Concepts, Algorithms, and Applications*. New York: Springer-Verlag, 1996.

[54] Solveig Flaig and Gero Junike. « Scenario Generation for Market Risk Models Using Generative Neural Networks. » In: *Risks* 10.11 (2022), p. 199.

[55] Chen Gao, Hongliang Guo, and Wenda Sheng. *Travel Time Data of Chengdu Road Network*. 2021.

[56] Zhong Xu Gao, An Jia Mao, De Zhi Chen, and Yun Ting Song. « A Wind Farm Capacity Credibility Calculation Method Based on Parabola. » In: *Applied Mechanics and Materials* 472 (Jan. 2014), pp. 953–957.

[57] Leijiao Ge and Yuanzheng Li. « Coordinated Operation Between Electric Vehicle Charging Stations and Distribution Power Network Considering Energy and Reserve. » In: *Smart Power Distribution Network: Situation Awareness, Planning, and Operation*. Springer Nature Singapore, 2023, pp. 183–211.

[58] Christian Genest, Michel Gendron, and Michaël Bourdeau-Brien. « The Advent of Copulas in Finance. » In: *The European Journal of Finance* 15.7-8 (2009), pp. 609–618.

[59] Xavier Glorot and Yoshua Bengio. « Understanding the difficulty of training deep feedforward neural networks. » In: *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*. 2010, pp. 249–256.

[60] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[61] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. « Generative adversarial nets. » In: *Advances in neural information processing systems (NeurIPS 2014)*. 2014, pp. 2672–2680.

[62] Feng Guo, Dongqing Zhang, Yucheng Dong, and Zhaoxia Guo. « Urban Link Travel Speed Dataset from a Megacity Road Network. » In: *Scientific Data* 6.1 (2019), p. 61.

[63] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. « Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. » In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01 (2019), pp. 922–929.

[64] Ziyuan Guo, Qingyi Lin, and Xuhui Meng. « A Comparative Study on Deep Learning Models for COVID-19 Forecast. » In: *Healthcare* 11 (Aug. 2023), p. 2400.

[65] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. « Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks. » In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2255–2264.

[66] Will Hamilton, Rex Ying, and Jure Leskovec. « Inductive representation learning on large graphs. » In: *Advances in neural information processing systems (NeurIPS 2017)*. 2017, pp. 1024–1034.

[67] Fouzi Harrou, Abdelkader Dairi, Abdelhafid Zeroual, and Ying Sun. « Forecasting of Bicycle and Pedestrian Traffic Using Flexible and Efficient Hybrid Deep Learning Approach. » In: *Applied Sciences* 12.9 (2022).

[68] Elaine K. Hart and Mark Z. Jacobson. « A Monte Carlo approach to generator portfolio planning and carbon emissions assessments of systems with large penetrations of variable renewables. » In: *Renewable Energy* 36.8 (2011), pp. 2278–2286.

[69] William Harvey, Saeid Naderiparizi, and Frank Wood. « Conditional image generation by conditioning variational autoencoders. » In: *arXiv preprint arXiv:2102.12037* (2021).

[70] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. « Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. » In: *Proceedings of the IEEE international conference on computer vision (ICCV)*. 2015, pp. 1026–1034.

[71] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. « Deep residual learning for image recognition. » In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 2016, pp. 770–778.

[72] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. New York: Wiley, 1949.

[73] Holger Heitsch and Werner Römisch. « Stability and scenario trees for multistage stochastic programs. » In: *Stochastic Programming: The State of the Art In Honor of George B. Dantzig*. Springer, 2010, pp. 139–164.

[74] Jon C Helton and Freddie J Davis. « Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. » In: *Reliability Engineering & System Safety* 81.1 (2003), pp. 23–69.

[75] Dan Hendrycks and Kevin Gimpel. « Gaussian error linear units (GELUs). » In: *arXiv preprint arXiv:1606.02515* (2016).

[76] Tom Henighan et al. « Scaling laws for autoregressive transformers. » In: *arXiv preprint arXiv:2001.08361* (2020).

[77] Irina Higgins, Leon Matthey, Xavier Glorot, Arka Pal, Benigno Uria, Charles Blundell, Shakir Mohamed, and Alexander Lerchner. « β-VAE: Learning basic visual concepts with a constrained variational framework. » In: *International Conference on Learning Representations (ICLR 2017)*. 2017.

[78] Geoffrey E. Hinton and Ruslan R. Salakhutdinov. « Reducing the Dimensionality of Data with Neural Networks. » In: *Science* 313.5786 (2006), pp. 504–507.

[79] Jonathan Ho, Ajay Jain, and Pieter Abbeel. « Denoising diffusion probabilistic models. » In: *Advances in neural information processing systems (NeurIPS 2020)*. 2020.

[80] Sepp Hochreiter and Jürgen Schmidhuber. « Long Short-Term Memory. » In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[81] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Bucilá, Sandeep Purushwalkam, Oriol Clark, Philip Bachman, Trevor Brown, Michael M Bronstein, and Daniele de Freitas. « Training compute-optimal large language models. » In: *arXiv preprint arXiv:2203.15556* (2022).

[82] Serge Hoogendoorn and P. Bovy. « State-of-the-art of vehicular traffic flow modeling. » In: *J. Syst. Cont. Eng.* 215 (June 2001), pp. 283–303.

[83] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. « Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. » In: *Neural networks* 4.2 (1991), pp. 251–257.

[84] Harold Hotelling. « Analysis of a complex of statistical variables into principal components. » In: *Journal of Educational Psychology* 24.6 (1933), pp. 417–441.

[85] Kjetil Höyland and Stein W. Wallace. « Generating Scenario Trees for Multistage Decision Problems. » In: *Management Science* 47.2 (2001), pp. 295–307.

[86] Huaibo Huang, zhihang li, Ran He, Zhenan Sun, and Tieniu Tan. « IntroVAE: Introspective Variational Autoencoders for Photographic Image Synthesis. » In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., 2018.

[87] Siyuan Huang, Zan Wang, Puhao Li, Baoxiong Jia, Tengyu Liu, Yixin Zhu, Wei Liang, and Song-Chun Zhu. « Diffusion-based Generation, Optimization, and Planning in 3D Scenes. » In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023.

[88] Zhiyu Huang, Zixu Zhang, Ameya Vaidya, Yuxiao Chen, Chen Lv, and Jaime Fernández Fisac. « Versatile Behavior Diffusion for Generalized Traffic Agent Simulation. » In: *arXiv preprint arXiv:2404.02524* (2024).

[89] Miltiadis Iatrou, Christos Karydas, Xanthi Tseni, and Spiros Mourelatos. « Representation learning with a variational autoencoder for predicting nitrogen requirement in rice. » In: *Remote Sensing* 14.23 (2022), p. 5978.

[90] Mustafa Inci, OzgUr Celik, Abderezak Lashab, Kamil Bayindir, Juan Vasquez, and Josep Guerrero. « Power System Integration of Electric Vehicles: A Review on Impacts and Contributions to the Smart Grid. » In: *Applied Sciences* 14 (Mar. 2024), p. 2246.

[91] Sergey Ioffe and Christian Szegedy. « Batch normalization: Accelerating deep network training by reducing internal covariate shift. » In: *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*. 2015, pp. 448–456.

[92] Touseef Iqbal and Shaima Qureshi. « Reconstruction probability-based anomaly detection using variational auto-encoders. » In: *International Journal of Computers and Applications* 45 (Nov. 2022), pp. 1–7.

[93] Tobias Skovgaard Jepsen, Christian S. Jensen, and Thomas Dyhre Nielsen. « Relational Fusion Networks: Graph Convolutional Networks for Road Networks. » In: *IEEE Transactions on Intelligent Transportation Systems* 23.1 (Jan. 2022), 418—429. ISSN: 1558-0016.

[94] Harry Joe. *Dependence Modeling with Copulas*. 1st ed. New York: Chapman and Hall/CRC, 2014.

[95] Tero Karras, Samuli Laine, and Timo Aila. « A style-based generator architecture for generative adversarial networks. » In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 4401–4410.

[96] Michal Kaut and Stein Wallace. « Evaluation of Scenario-Generation Methods for Stochastic Programming. » In: *Pacific Journal of Optimization* 3 (June 2003).

[97] Mansoor Khan, Muhammad Rashid Naeem, Essam A. Al-Ammar, Wonsuk Ko, Hamsakutty Vettikalladi, and Irfan Ahmad. « Power Forecasting of Regional Wind Farms via Variational Auto-Encoder and Deep Hybrid Transfer Learning. » In: *Electronics* 11 (2022).

[98] Diederik P Kingma and Jimmy Ba. « Adam: A method for stochastic optimization. » In: *International Conference on Learning Representations (ICLR 2015)*. 2015.

[99] Diederik P Kingma and Max Welling. « Auto-encoding variational bayes. » In: *International Conference on Learning Representations (ICLR 2014)*. 2014.

[100] Thomas N Kipf and Max Welling. « Semi-supervised classification with graph convolutional networks. » In: *International Conference on Learning Representations (ICLR 2017)*. 2017.

[101] Genshiro Kitagawa. « Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. » In: *Journal of Computational and Graphical Statistics* 5.1 (1996), pp. 1–25.

[102] Robert Krajewski, Julian Bock, Lennart Kloeker, and Lutz Eckstein. « The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways. » In: *Proceedings of the 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2118–2125.

[103] Mark A Kramer. « Nonlinear principal component analysis using autoassociative neural networks. » In: *AIChE journal* 37.2 (1991), pp. 233–243.

[104] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. « ImageNet classification with deep convolutional neural networks. » In: *Advances in neural information processing systems (NIPS 2012)*. 2012, pp. 1097–1105.

[105] Anders Krogh and John A Hertz. « A simple weight decay can improve generalization. » In: *Advances in neural information processing systems 4 (NIPS 1991)*. 1992, pp. 950–957.

[106] Alex Kuefler, Jeremy Morton, Tim Wheeler, and Mykel Kochenderfer. « Imitating driver behavior with generative adversarial networks. » In: *2017 IEEE intelligent vehicles symposium (IV)*. IEEE. 2017, pp. 204–211.

[107] Solomon Kullback and Richard Arthur Leibler. « On information and sufficiency. » In: *The Annals of Mathematical Statistics* 22.1 (1951), pp. 79–86.

[108] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. « Efficient backprop. » In: *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[109] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. « Backpropagation applied to handwritten zip code recognition. » In: *Neural computation* 1.4 (1989), pp. 541–551.

[110] Simon Leglaive, Xavier Alameda-Pineda, Laurent Girin, and Radu Horaud. « A Recurrent Variational Autoencoder for Speech Enhancement. » In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020, pp. 371–375.

[111] Jure Leskovec and Christos Faloutsos. « Sampling from large graphs. » In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2006, pp. 631–639.

[112] Chang Li and Dan Goldwasser. « Encoding Social Information with Graph Convolutional Networks for Political Perspective Detection in News Media. » In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2019, pp. 2594–2604.

[113] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. « Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. » In: *International Conference on Learning Representations (ICLR)* (2018).

[114] Jianhua Lin. « Divergence measures based on the Shannon entropy. » In: *IEEE Transactions on Information theory* 37.1 (1991), pp. 145–151.

[115] Seppo Linnainmaa. « The representation of the cumulative rounding error of an algorithm in terms of the local rounding errors. » MA thesis. University of Helsinki, 1970.

[116] Ilya Loshchilov and Frank Hutter. « Decoupled weight decay regularization. » In: *International Conference on Learning Representations (ICLR 2019)*. 2019.

[117] Hailin Lu, Dongchen Sun, and Jing Hao. « Random Traffic Flow Simulation of Heavy Vehicles Based on R-Vine Copula Model and Improved Latin Hypercube Sampling Method. » In: *Sensors* 23.5 (2023), pp. 1–18.

[118] Xiaolei Ma, Sen Luan, Bowen Du, and Bin Yu. « Spatial Copula Model for Imputing Traffic Flow Data from Remote Microwave Sensors. » In: *Sensors* 17.10 (2017), pp. 1–19.

[119] Zhiao Ma, Xin Yang, Anthony Chen, Tianlei Zhu, and Jianjun Wu. « Assessing the resilience of multi-modal transportation networks with the integration of urban air mobility. » In: *Transportation Research Part A: Policy and Practice* 195 (2025), p. 104465.

[120] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. « Rectifier nonlinearities improve neural network object recognition. » In: *ICASSP* 30 (2013), p. 3.

[121] Prasanta Chandra Mahalanobis. « On the Generalised Distance in Statistics. » In: *Proceedings of the National Academy of Sciences of India* 2 (1936), pp. 49–55.

[122] Warren S. McCulloch and Walter Pitts. « A logical calculus of the ideas immanent in nervous activity. » In: *The Bulletin of Mathematical Biophysics* 5.4 (Dec. 1943), pp. 115–133. ISSN: 1522-9602.

[123] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. « UMAP: Uniform Manifold Approximation and Projection. » In: *Journal of Open Source Software* 3 (2018), p. 861.

[124] Michael D McKay, Richard J Beckman, and William J Conover. « A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. » In: *Technometrics* 21.2 (1979), pp. 239–245.

[125] Peter Meibom, Ruediger Barth, Bernhard Hasche, Heike Brand, Christoph Weber, and Mark O'Malley. « Stochastic Optimization Model to Study the Operational Impacts of High Wind Penetrations in Ireland. » In: *IEEE Transactions on Power Systems* 26.3 (2011), pp. 1367–1379.

[126] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. « Equation of State Calculations by Fast Computing Machines. » In: *The Journal of Chemical Physics* 21.6 (1953), pp. 1087–1092.

[127] Marvin Minsky and Seymour Papert. *Perceptrons*. MIT Press, 1969.

[128] Mehdi Mirza and Simon Osindero. « Conditional generative adversarial nets. » In: *arXiv preprint arXiv:1411.1784* (2014).

[129] Diganta Misra. « Mish: A self regularized non-monotonic neural activation function. » In: *Proceedings of the winter conference on applications of computer vision (WACV)*. 2020, pp. 200–207.

[130] Zhaobin Mo, Yongjie Fu, Daran Xu, and Xuan Di. « TrafficFlowGAN: Physics-Informed Flow Based Generative Adversarial Network for Uncertainty Quantification. » In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2022, Grenoble, France, September 19–23, 2022, Proceedings, Part III*. Springer-Verlag, 2022, 323–339. ISBN: 978-3-031-26408-5.

[131] Alfred Muller. « Integral probability metrics and their generating classes of functions. » In: *Advances in Applied Probability* 29.2 (1997), pp. 429–443.

[132] Vinod Nair and Geoffrey E. Hinton. « Rectified linear units improve restricted boltzmann machines. » In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML 10. Omnipress, 2010, 807–814. ISBN: 9781605589077.

[133] Roger B. Nelsen. *An Introduction to Copulas*. 2nd. New York: Springer, 2006.

[134] Augustus Odena, Christopher Olah, and Jonathon Shlens. « Conditional image synthesis with auxiliary classifier GANs. » In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML'17. JMLR.org, 2017, 2642–2651.

[135] Aaron Van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. « Pixel recurrent neural networks. » In: *International Conference on Machine Learning (ICML)*. 2016, pp. 1747–1756.

[136] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst. *An introduction to signal processing on graphs*. Cambridge University Press, 2022.

[137] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst. « Graph signal processing: Overview, challenges, and applications. » In: *Proceedings of the IEEE* 106.5 (2018), pp. 808–828.

[138] Christidis P, Vega Gonzalo M, Ulpiani G, and Vetters N. *Post-pandemic trends in urban mobility*. JRC Science for Policy Report. Joint Research Centre, European Commission, 2023.

[139] Bruno K. Pagnoncelli, Shabbir Ahmed, and Alexander Shapiro. « Sample Average Approximation Method for Chance Constrained Programming: Theory and Applications. » In: *Journal of Optimization Theory and Applications* 142.2 (2009), pp. 399–416.

[140] Hossein Panamtash, Qun Zhou, Tao Hong, Zhihua Qu, and Kristopher O. Davis. « A copula-based Bayesian method for probabilistic solar power forecasting. » In: *Solar Energy* 196 (2020), pp. 336–345.

[141] George Papaefthymiou and Bernd Klöckl. « MCMC for Wind Power Simulation. » In: *IEEE Transactions on Energy Conversion* 23.1 (2008), pp. 234–240.

[142] Karl Pearson. « On lines and planes of closest fit to systems of points in space. » In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572.

[143] Karl Pearson and Alice Lee. « Note on regression and inheritance in the case of two parents. » In: *Proceedings of the Royal Society of London* 58 (1895), pp. 240–242.

[144] Georg Ch Pflug and Alois Pichler. *Multistage stochastic optimization*. Vol. 1104. Springer, 2014.

[145] Bob Pishue. *2025 Global Traffic Scorecard*. INRIX, Inc. Bellevue, WA, 2025.

[146] Boris Teodorovich Polyak. « Some methods of speeding up the convergence of iteration processes. » In: *USSR Computational Mathematics and Mathematical Physics* 4.5 (1964), pp. 1–17.

[147] Warren B Powell. « A unified framework for stochastic optimization. » In: *European Journal of Operational Research* 275.3 (2019), pp. 795–821.

[148] Maurice Quach, Jiahao Pang, Dong Tian, Giuseppe Valenzise, and Frédéric Dufaux. « Survey on Deep Learning-Based Point Cloud Compression. » In: *Frontiers in Signal Processing* 2 (2022), p. 846972.

[149] Maurice Quach, Giuseppe Valenzise, and Frederic Dufaux. « Learning Convolutional Transforms for Lossy Point Cloud Geometry Compression. » In: *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. 2019, pp. 4320–4324.

[150] Alec Radford, Luke Metz, and Soumith Chintala. « Unsupervised representation learning with deep convolutional generative adversarial networks. » In: *arXiv preprint arXiv:1511.06434* (2015).

[151] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. *Improving Language Understanding by Generative Pre-training*. Tech. rep. OpenAI, 2018.

[152] Hamed Rahimian and Sanjay Mehrotra. « Distributionally robust optimization: A review. » In: *arXiv preprint arXiv:1908.05659* (2019).

[153] Prajit Ramachandran, Barret Zoph, and Quoc V Le. « Searching for activation functions. » In: *arXiv preprint arXiv:1710.05941* (2017).

[154] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. « Generating diverse high-fidelity images with VQ-VAE-2. » In: *arXiv preprint arXiv:1906.00446* (2019).

[155] Danilo Jimenez Rezende, Shakir Mohamed, and Laurent Van Der Wiele. « Stochastic backpropagation and variational inference on graphs and manifolds. » In: *International Conference on Machine Learning (ICML 2014)*. 2014.

[156] Nick Rhinehart, Kris M. Kitani, and Paul Vernaza. « Prediction Conditioned on Goals in Visual Multi-Agent Settings. » In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 2821–2830.

[157] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. 2nd. Springer, 2004.

[158] Ralph Tyrrell Rockafellar and Roger Jean Baptiste Wets. « Scenarios and Policy Aggregation in Optimization under Uncertainty. » In: *Mathematics of Operations Research* 16.1 (1991), pp. 119–147.

[159] Robin Rombach, Andreas Blattmann, Dominique Lorenz, Patrick Esser, and Bjorn Ommer. « High-resolution image synthesis with latent diffusion models. » In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 10684–10695.

[160] Frank Rosenblatt. « The perceptron: A probabilistic model for information storage and organization in the brain. » In: *Psychological Review* 65.6 (1958), pp. 386–408.

[161] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. « Learning representations by back-propagating errors. » In: *Nature* 323.6088 (1986), pp. 533–536.

[162] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. « Improved techniques for training gans. » In: *Advances in neural information processing systems (NeurIPS 2016)*. 2016, pp. 1205–1213.

[163] G. Salvadori and C. De Michele. « Frequency analysis via Copulas: Theoretical aspects and applications to hydrological events. » In: *Water Resources Research* 40.12 (2004).

[164] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. « The graph neural network model. » In: *IEEE Transactions on Neural Networks* 20.1 (2009), pp. 61–80.

[165] Jürgen Schmidhuber. « Deep learning in neural networks: An overview. » In: *Neural Networks* 61 (2015), pp. 85–117. ISSN: 0893-6080.

[166] Patrick Schober, Christa Boer, and Lothar A Schwarte. « Correlation coefficients: appropriate use and interpretation. » In: *Anesthesia & Analgesia* 126.5 (2018), pp. 1763–1768.

[167] Paul Schoemaker. « Scenario Planning: A Tool for Strategic Thinking. » In: *Sloan Management Review* 36 (Jan. 1995), pp. 25–40.

[168] Seoul Metropolitan Government. *Air pollution dataset*. Data retrieved on 8 November 2019. 2019. URL: https://data.seoul.go.kr (visited on 09/15/2025).

[169] Iman Seyedi, Antonio Candelieri, Vincenzina Messina, and Francesco Archetti. « Wasserstein Distributionally Robust Optimization for Chance Constrained Facility Location Under Uncertain Demand. » In: *Mathematics* 13 (June 2025), p. 2144.

[170] Alexander Shapiro. « Monte Carlo Sampling Methods. » In: *Stochastic Programming*. Vol. 10. Handbooks in Operations Research and Management Science. Elsevier, 2003, pp. 353–425.

[171] Alexander Shapiro. « Analysis of stochastic dual dynamic programming method. » In: *European Journal of Operational Research* 209.1 (2011), pp. 63–72.

[172] David I Shuman, Siddharth Narang, Parfaite Faraji, Mengran Zhao, Antonio Ortega, and Pierre Vandergheynst. « The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. » In: *IEEE Signal Processing Magazine* 30.3 (2013), pp. 83–98.

[173] Abe Sklar. « Fonctions de Répartition à n Dimensions et Leurs Marges. » In: *Publications de l'Institut Statistique de l'Université de Paris* 8 (1959), pp. 229–231.

[174] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. « Practical bayesian optimization of machine learning algorithms. » In: *Advances in neural information processing systems (NeurIPS 2012)*. 2012, pp. 2951–2959.

[175] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Kevin Swersky. « Deep unsupervised learning using nonequilibrium thermodynamics. » In: *International Conference on Machine Learning (ICML)*. 2015, pp. 2258–2267.

[176] Alberto Solera-Rico, Carlos Sanmiguel Vila, Miguel Gómez-López, Yuning Wang, Abdulrahman Almashjary, Scott T. M. Dawson, and Ricardo Vinuesa. « β-Variational Autoencoders and Transformers for Reduced-Order Modelling of Fluid Flows. » In: *Nature Communications* 15.1 (2024), p. 1361.

[177] Charles Spearman. « The proof and measurement of association between two things. » In: *The American Journal of Psychology* 15.1 (1904), pp. 72–101.

[178] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. « Dropout: a simple way to prevent neural networks from overfitting. » In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[179] Michael Stein. « Large sample properties of simulations using Latin hypercube sampling. » In: *Technometrics* 29.2 (1987), pp. 143–151.

[180] Shuo Sun, Zekai Gu, Tianchen Sun, Jiawei Sun, Chengran Yuan, Yuhang Han, Dongen Li, and Marcelo H. Ang. « DriveSceneGen: Generating Diverse and Realistic Driving Scenarios From Scratch. » In: *IEEE Robotics and Automation Letters* 9.8 (2024), pp. 7007–7014.

[181] Department of Economic United Nations and Population Division Social Affairs. *World Urbanization Prospects 2025: Summary of Results*. Tech. rep. United Nations, 2025.

[182] Arash Vahdat and Jan Kautz. « NVAE: A multi-scale hierarchical variational autoencoder. » In: *Advances in neural information processing systems (NeurIPS 2020)*. 2020.

[183] Hamed Valizadeh Haghi and Saeed Lotfifard. « Spatiotemporal Modeling of Wind Generation for Optimal Energy Storage Sizing. » In: *IEEE Transactions on Sustainable Energy* 6.1 (2015), pp. 113–121.

[184] Laurens Van Der Maaten and Geoffrey Hinton. « Visualizing Data using t-SNE. » In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605.

[185] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. « Attention is all you need. » In: *Advances in neural information processing systems (NIPS 2017)*. 2017, pp. 5998–6008.

[186] Petar Veličković, Guillem Cucurull, Arñau Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. « Graph attention networks. » In: *International Conference on Learning Representations (ICLR 2018)*. 2018.

[187] Cédric Villani. « The Wasserstein distances. » In: *Optimal Transport: Old and New* (2009), pp. 93–111.

[188] Stein W Wallace. « Decision making under uncertainty: is sensitivity analysis of any use? » In: *Operations Research* 48.1 (2000), pp. 20–25.

[189] Zhiqiang Wan, Yazhou Zhang, and Haibo He. « Variational autoencoder based synthetic data generation for imbalanced learning. » In: *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. 2017, pp. 1–7.

[190] Prince Zizhuang Wang and William Yang Wang. « Neural Gaussian Copula for Variational Autoencoder. » In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, 2019, pp. 4333–4343.

[191] Shuo Wang, Yanran Li, Jiang Zhang, Qingye Meng, Lingwei Meng, and Fei Gao. « PM2.5-GNN: A Domain Knowledge Enhanced Graph Neural Network For PM2.5 Forecasting. » In: *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*. SIGSPATIAL '20. 2020, pp. 163–166.

[192] Yishen Wang, Yize Chen, Baosen Zhang, and Daniel S. Kirschen. « Model-Free Renewable Scenario Generation Using Generative Adversarial Networks. » In: *IEEE Transactions on Power Systems* 33.3 (2018), pp. 3265–3275.

[193] Yuning Wang, Alberto Solera-Rico, Carlos Sanmiguel Vila, and Ricardo Vinuesa. « Towards Optimal β-Variational Autoencoders Combined with Transformers for Reduced-Order Modelling of Turbulent Flows. » In: *International Journal of Heat and Fluid Flow* 105 (2024), p. 109254.

[194] Wijarn Wangdee and Roy Billinton. « Probing the Intermittent Energy Resource Contributions From Generation Adequacy and Security Perspectives. » In: *Power Systems, IEEE Transactions on* 27 (Nov. 2012), pp. 2306–2313.

[195] Benjamin Weggenmann, Valentin Rublack, Michael Andrejczuk, Justus Mattern, and Florian Kerschbaum. « DP-VAE: Human-Readable Text Anonymization for Online Reviews with Differentially Private Variational Autoencoders. » In: *Proceedings of the ACM Web Conference 2022 (WWW '22)*. 2022, pp. 721–731.

[196] Ruoqi Wei and Ausif Mahmood. « Recent Advances in Variational Autoencoders With Representation Learning for Biomedical Informatics: A Survey. » In: *IEEE Access* 9 (2021), pp. 68478–68496.

[197] Haomin Wen, Youfang Lin, Yutong Xia, Huaiyu Wan, Qingsong Wen, Roger Zimmermann, and Yuxuan Liang. « DiffSTG: Probabilistic Spatio-Temporal Graph Forecasting with Denoising Diffusion Models. » In: *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL '23)*. Association for Computing Machinery, 2023.

[198] Paul J. Werbos. « Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. » PhD thesis. Cambridge, MA: Harvard University, 1974.

[199] Bernard Widrow and Marcian E. Hoff. « Associative Storage and Retrieval of Digital Information in Networks of Adaptive Neurons. » In: *Biological Prototypes and Synthetic Systems*. Vol. 1. Plenum Press, 1962, p. 160.

[200] Chao Wu, Lei Chen, Guibin Wang, Songjian Chai, Hui Jiang, Jianchun Peng, and Zhouzhenyan Hong. « Spatiotemporal Scenario Generation of Traffic Flow Based on LSTM-GAN. » In: *IEEE Access* 8 (2020), pp. 186191–186198.

[201] Zonghan Wu, Shirui Pan, Feng Chen, Guodong Long, Chengqi Zhang, and Sydney Yu Philip. « A comprehensive survey on graph neural networks. » In: *IEEE transactions on neural networks and learning systems* 32.1 (2021), pp. 4–24.

[202] Chen Xu, Qiang Wang, and Lijun Sun. « Likelihood-Free Variational Autoencoders. » In: *arXiv preprint arXiv:2504.17622* (2025).

[203] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. « How powerful are graph neural networks? » In: *International Conference on Learning Representations (ICLR 2019)*. 2019.

[204] Huan Yan and Yong Li. « Generative AI for Intelligent Transportation Systems: Road Transportation Perspective. » In: *ACM Computing Surveys* 57.12 (2025).

[205] Chen Yang, Yangfan He, and Tianyu Shi. « WcDT: World-centric Diffusion Transformer for Traffic Scene Generation. » In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. 2025.

[206] Byeongchan Yoo, Jongmin Lee, Jaesik Ju, Sungyong Chung, Soyoun Kim, and Jinwoo Choi. « Conditional temporal neural processes with covariance loss. » In: *Proceedings of the 38th International Conference on Machine Learning (ICML)*. Vol. 139. PMLR. 2021, pp. 12051–12061.

[207] Yang You, Jing Li, Sashank Reddi, Srinivas Bhojanapalli, Manzil Kumar, Vinay Kanade, and Jonathan Goodman. « Large batch optimization for deep learning: Training bert in 76 minutes. » In: *International Conference on Learning Representations (ICLR 2020)*. 2020.

[208] Han Yu, C. Y. Chung, K. P. Wong, H. W. Lee, and J. H. Zhang. « Probabilistic Load Flow Evaluation With Hybrid Latin Hypercube Sampling and Cholesky Decomposition. » In: *IEEE Transactions on Power Systems* 24.2 (2009), pp. 661–667.

[209] James J. Q. Yu, Christos Markos, and Shiyao Zhang. « Long-term urban traffic speed prediction with deep learning on graphs. » In: *IEEE Transactions on Intelligent Transportation Systems* 23.7 (2022), 7359–7370.

[210] Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. « PhysDiff: Physics-Guided Human Motion Diffusion Model. » In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023.

[211] Matthew D Zeiler and Rob Fergus. « Visualizing and understanding convolutional networks. » In: *European conference on computer vision (ECCV)*. Springer, Cham, 2014, pp. 818–833.

[212] Yacine Zerrouki, Fouzi Harrou, Nabil Zerrouki, Abdelkader Dairi, and Ying Sun. « Desertification Detection Using an Improved Variational Autoencoder-Based Approach Through ETM-Landsat Satellite Data. » In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* PP (2020), pp. 202–213.

[213] Wei Zhan et al. « INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps. » In: *arXiv preprint arXiv:1910.03088* (2019).

[214] Chengyang Zhang, Yong Zhang, Qitan Shao, Bo Li, Yisheng Lv, Xinglin Piao, and Baocai Yin. « ChatTraffic: Text-to-Traffic Generation via Diffusion Model. » In: *Trans. Intell. Transport. Sys.* 26.2 (2025), pp. 2656–2668.

[215] Liang Zhang and Cheng Long. « Road Network Representation Learning: A Dual Graph-based Approach. » In: *ACM Trans. Knowl. Discov. Data* 17.9 (2023). ISSN: 1556-4681.

[216] Weijia Zhang, Hao Liu, Yanchi Liu, Jingbo Zhou, and Hui Xiong. « Semi-Supervised Hierarchical Recurrent Graph Neural Network for City-Wide Parking Availability Prediction. » In: *AAAI Conference on Artificial Intelligence*. 2019.

[217] Xiurong Zhang, Daoliang Li, and Xueqian Fu. « A novel wasserstein generative adversarial network for stochastic wind power output scenario generation. » In: *IET Renewable Power Generation* 18 (Feb. 2024), 3731–3742.

[218] Shengjie Zhao, Jiaming Liu, Jiawei Zhang, Glen Fung, Paul Li, and Yajing Zhang. « InfoVAE: Information maximizing variational autoencoders. » In: *arXiv preprint arXiv:1706.02262* (2017).

[219] Jie Zhou, Ganqu Cui, Zhengyan Hu, Zhenyu Zhang, Changdong Yang, and Zhiyuan Liu. « Graph neural networks: A review of methods and applications. » In: *AI Open* 1 (2020), pp. 57–81.

[220] Marco Zugno, Juan M Morales, Pierre Pinson, and Henrik Madsen. « Pool strategy of a price-maker wind power producer. » In: *IEEE Transactions on Power Systems* 28.3 (2013), pp. 3440–3450.

Tesi di dottorato realizzata nell'ambito del progetto **MOST** finanziato dal *PNRR Missione 4 Componente 2 Investimento 1.4*, finanziato dall'Unione Europea - NextGenerationEU
CUP H43C22000510001