Université Gustave Eiffel

PARIS EST SUP — ÉCOLE DOCTORALE
Ville, Transports et Territoires

Deep Generative Models for Synthetic Population Modeling in Mobility Research

Thèse de doctorat de l'Université Gustave Eiffel

École doctorale n°528 : Ville, Transport et Territoires (VTT)
Spécialité de doctorat: Transport
Unité de recherche : Laboratoire SPLOTT, Université Gustave Eiffel

Thèse soutenue le 12/01/2026 par :

Abdoul Razac SANÉ

Composition du Jury

Président du jury :	**Éric CORNÉLIS** Professeur des Universités, HDR, Université de Namur
Rapportrice :	**Juliette ROUCHIER** Directrice de Recherche CNRS, HDR, Aix-Marseille Université
Rapportrice :	**Mounira HARZALLAH** Maîtresse de Conférences, HDR, Université de Nantes

Encadrement de la thèse

Directeur de thèse :	**Pierre-Olivier VANDANJON** Directeur de Recherche, HDR, Université Gustave Eiffel
Encadrant de thèse :	**Pierre HANKACH** Chargé de Recherche, Université Gustave Eiffel
Encadrant de thèse :	**Rachid BELAROUSSI** Chargé de Recherche, Université Gustave Eiffel

Acknowledgements

At the conclusion of this thesis, I would like to express my deepest gratitude to all those who have contributed, directly or indirectly, to its completion.

First and foremost, I would like to thank my thesis supervisor Pierre-Olivier Vandanjon. Beyond the rigorous scientific guidance, I wish to highlight the quality of the human relationship we built throughout these years. Your trust, availability and thoughtful insights have been invaluable, both academically and personally. Alongside you, my two co-supervisors, Pierre Hankach and Rachid Belaroussi, played an essential role in the development of this work. Their continuous support, insightful advice and patience created a working environment based on mutual respect, attentive listening and collaboration. Thank you for the constructive discussions, the human warmth and the availability that greatly contributed to the success of this thesis.

I would also like to extend my sincere gratitude to the examiners, Eric Cornelis, Juliette Rouchier and Mounira Harzallah, who kindly accepted to evaluate this work. Thank you for the time and attention dedicated to reading this manuscript and for the scientific exchange it will foster.

My warmest thanks go to the SPLOTT laboratory for welcoming me and providing the necessary resources to pursue this research. I am grateful to Pétronille Rème-Harnay and Sandrine Vanhoutte for her dedication, efficiency and invaluable assistance throughout this journey.

I would like to express my special thanks to Pascal Gastineau, who has been a true supporter throughout the years. Thank you for your support, your insightful discussions, and your kind daily presence. My gratitude also goes to Jean-Claude Bahati Nzeyi, Léonardo Galassi-Luquezi, Valentin Le Bescond, for their support, their good humor, stimulating discussions and the moments of camaraderie that brightened this experience.

To my family, I owe my deepest appreciation. I would like to thank my mother and late father for their unconditional love and for always encouraging me to pursue my ambitions. To my two brothers, Yaya and Karim, for their support, their confidence and their attentive presence. To my uncle, Bakouamé Ouattara, whose constant encouragement has been a unwavering source of motivation. Thank you for being my strength and my refuge, both in

times of doubt and in moments of joy.

To my wife, I wish to express my most heartfelt words. Thank you for your love, your endless patience, your understanding and your constant presence by my side. You have been my greatest support throughout this journey, and this achievement is as much yours as it is mine.

To all my friends, both near and far, thank you for your encouragement, your messages, your smiles, your advice and the moments of respite and happiness we shared. Each of you has contributed, in your own way, to providing the balance and motivation needed to bring this work to fruition.

To all of you, thank you sincerely.

Abstract

This thesis investigates the use of recent artificial intelligence methods—particularly deep generative models for the generation of synthetic populations and activity plans. Its overarching goal is to reproduce realistic and statistically coherent representations of households, individuals, and their daily behaviors, as required in urban mobility and transport microsimulation frameworks. The research begins with a comprehensive review of synthetic population generation methods, ranging from traditional statistical reweighting and combinatorial optimization techniques to probabilistic, deep, and large language models (LLMs). This review highlights the trade-offs between robustness and flexibility and identifies hybrid models as a promising research direction. Building on this foundation, Variational Autoencoders (VAEs) are introduced as a core methodology for generating high-dimensional mixed data. Empirical experiments demonstrate their capacity to synthesize large, coherent populations from small samples, effectively addressing the issue of sampling zeros while preserving statistical consistency. The approach is then extended to a two-layered framework (*SVAE-Pop2* and *MVAE-Pop2*) that jointly models households and their members, capturing intra-household dependencies crucial for realistic behavioral simulations. To further enhance representativeness, a hybrid *VAE+IPF* model is proposed, combining the VAE's flexibility in learning complex distributions with the Iterative Proportional Fitting procedure to enforce external marginal constraints. This hybridization significantly improves both marginal and joint consistency while reducing structural zeros. Finally, the thesis explores the generation of daily activity plans through two deep architectures: a Conditional VAE (CVAE) and a Transformer-based model termed *ActivityGPT*. Preliminary analyses show that sequential representations combined with GPT architectures hold strong potential for capturing temporal dependencies in daily mobility behaviors. Overall, the thesis contributes a coherent methodological and conceptual framework for integrating deep generative models into synthetic population and activity modeling.

Keywords : *synthetic population, deep generative models, variational autoencoders, hybrid modeling, households, activity plans, microsimulation, AI for urban mobility*

Résumé

Cette thèse explore l'application des approches récentes de l'intelligence artificielle, et plus particulièrement des modèles génératifs profonds, à la génération de populations et de plans d'activités synthétiques. L'objectif général est de reproduire des représentations réalistes et statistiquement cohérentes des individus, des ménages et de leurs comportements quotidiens, indispensables à la modélisation urbaine et à la microsimulation des transports. Le travail débute par une revue comparative des principales méthodes de génération de populations synthétiques, allant des approches classiques de rééchantillonnage et d'optimisation combinatoire aux modèles probabilistes, profonds et fondés sur les grands modèles de langage (LLM). Cette analyse met en évidence les compromis entre robustesse et flexibilité, et souligne le potentiel des approches hybrides combinant la rigueur statistique des méthodes traditionnelles et la puissance expressive des modèles modernes. Sur cette base, les Autoencodeurs Variationnels (VAE) sont introduits comme cadre méthodologique central pour la génération de données mixtes de grande dimension. Les résultats empiriques démontrent leur capacité à produire des populations cohérentes à partir d'échantillons réduits, tout en limitant le problème des zéros d'échantillonnage. L'approche est ensuite étendue à un cadre hiérarchique à deux niveaux (*SVAE-Pop2* et *MVAE-Pop2*), permettant la génération conjointe des ménages et de leurs membres et la prise en compte des dépendances intra-ménages. Pour renforcer la représentativité statistique, un modèle hybride *VAE+IPF* est proposé, combinant la flexibilité du VAE à la rigueur du procédé d'ajustement itératif des marges (IPF). Enfin, la thèse explore la génération de plans d'activités quotidiens à l'aide de deux architectures complémentaires : un VAE conditionnel (CVAE) et un modèle de type Transformer, *ActivityGPT*. Les analyses préliminaires confirment le potentiel de ces modèles pour capturer les dépendances temporelles des comportements de mobilité.

Mots-clés : *population synthétique, modèles génératifs profonds, autoencodeurs variationnels, modélisation hybride, ménages, plans d'activités, microsimulation, intelligence artificielle pour la mobilité urbaine*

Contents

Contents ix

List of Figures xiii

List of Tables xvi

Publications xvii

1 Introduction 1

1.1 Synthetic Population . . . 1
1.2 Agent-Based Models for Transport Policy Evaluation . . . 2
1.3 Territorial Modeling . . . 4
1.4 Synthetic Population and Activity-Plan Generation . . . 6
1.5 Artificial Intelligence as a Source of New Methodologies for Synthetic Population Generation . . . 7
1.6 Positioning . . . 9
1.7 Contributions . . . 10

2 Literature Review 13

Abstract . . . 14
2.1 Key Concepts . . . 15
2.1.1 Synthetic Populations . . . 15
2.1.2 Hierarchical Data . . . 16
2.1.3 Activity Plans . . . 17
2.2 Methods for population synthesis . . . 17
2.2.1 Traditional Methods for Population Synthesis . . . 18
2.2.2 Deep Generative Models . . . 22
2.3 Methods for Activity Plan Synthesis . . . 26
2.3.1 Statistical Matching . . . 26
2.3.2 Machine Learning (ML) . . . 27
2.3.3 Deep Generative Models . . . 28
2.4 Cross-comparison of approaches for population synthesis . . . 28

2.5 Historical Perspectives . . . 30
2.6 Conclusion and Positioning . . . 33

3 VAE for generating a single-layered synthetic population 35
Abstract . . . 36
3.1 Introduction . . . 37
3.2 Variational Autoencoder Models . . . 38
3.2.1 Linear Autoencoder analyzed as a Principal Component Analysis . . . 38
3.2.2 Nonlinear autoencoder . . . 40
3.2.3 Stacked autoencoder . . . 41
3.2.4 Variational Autoencoder (VAE) . . . 41
3.2.5 Model training . . . 44
3.3 Case study data . . . 46
3.4 Unlocking the optimal VAE architectures and hyperparameters . . . 47
3.4.1 Data engineering . . . 47
3.4.2 Technologies introduced . . . 49
3.4.3 Model-building . . . 49
3.5 Assessment of the generated synthetic population . . . 51
3.5.1 Performance on the individual attributes . . . 52
3.5.2 Performance on the household attributes . . . 55
3.6 Influence of the model building dataset size . . . 56
3.6.1 Influence on the accuracy of the generated population . . . 57
3.6.2 Influence on the zero-cell problem . . . 58
3.7 Conclusion . . . 62

4 Two-layered Population Synthesis with VAE 66
Abstract . . . 67
4.1 Introduction . . . 68
4.2 Two Approaches for Modeling Two-Level Synthetic Population Generation using VAEs . . . 69
4.2.1 Single VAE for two layered population generation (SVAE-Pop2) . . . 69
4.2.2 Multi VAE for two layered population generation (modeling-based multiple VAE) (MVAE-Pop2) . . . 70
4.3 Experiments . . . 71
4.3.1 Case study dataset . . . 71
4.3.2 Loss function . . . 72
4.3.3 Model building . . . 73
4.4 Results . . . 74
4.4.1 Performance on the Individual Attributes . . . 75
4.4.2 Performance on the Household Attributes . . . 78

4.4.3 Performance on the Joint Household-Individual Cross-Modalities . . . 79
4.5 Discussion . . . 79
4.6 Conclusions and future work . . . 81

5 A Hybrid VAE+IPF Methodology for Synthetic Populations 84
Abstract . . . 85
5.1 Introduction . . . 86
5.2 Methodology . . . 86
5.2.1 Iterative Proportional Fitting (IPF) . . . 86
5.2.2 Variational Autoencoder (VAE) . . . 88
5.2.3 Model VAE+IPF . . . 88
5.3 Case Study . . . 90
5.3.1 Datasets . . . 90
5.3.2 Training and Assessments . . . 91
5.3.3 Technologies introduced . . . 92
5.4 Results . . . 92
5.4.1 Univariate evaluation of generated data . . . 92
5.4.2 Multivariate evaluation of generated data . . . 93
5.4.3 Sampling and Structural Zeros . . . 95
5.5 Discussion . . . 96
5.6 Conclusion . . . 97

6 Deep Generative Modeling of Activity Plans 99
Abstract . . . 100
6.1 Introduction . . . 101
6.2 Activity Plan Modeling . . . 101
6.2.1 Formalization and Notation . . . 101
6.2.2 Activity Plan Modeling . . . 102
6.3 Conditional Variational Autoencoder (CVAE) . . . 103
6.4 GPT for Activities Synthesis . . . 105
6.4.1 Word Embedding and Positional Encoding/Embedding . . . 105
6.4.2 Transformers . . . 107
6.4.3 ActivityGPT model . . . 110
6.5 Case study . . . 113
6.5.1 Datasets . . . 113
6.5.2 Evaluation Metrics . . . 113
6.6 Implementation and Preliminary Analysis . . . 115
6.6.1 Implemented Models . . . 116
6.6.2 Technological Framework . . . 116
6.6.3 A Priori Analysis of the Different Approaches . . . 116

6.7 Conclusion 118

7 Conclusion and perspectives **120**

7.1 Conclusion 120

7.2 Operational Perspectives 121

7.3 Academic Perspectives 122

Bibliography **125**

List of Figures

1.1 Synthetic populations for mobility studies: (a) localised socio-economic representation of households and individuals; (b) associated daily activity plans. 3

1.2 Example use of a synthetic population for urban worksite management: scheduling noisy machinery according to the people effectively present within noise influence zones (65/55/45 dB) and their noise sensitivity. 5

1.3 Example sequencing of operations to produce a synthetic population with spatialised daily activity plans: household and individual generation, attribute enrichment, activity-plan generation, spatialisation, then definition of utility functions for modal choice. Orange boxes indicate the specific contributions of the thesis. 6

1.4 Illustration of the Variational Autoencoder (VAE) principle: an input individual (left) is encoded into a probabilistic latent space where each dimension follows a Gaussian distribution. Random sampling within this space generates new latent points that are decoded into predicted individuals (right). 8

2.1 Example of a contingency table $x_1 \times x_2$ showing cell counts, row and column margins, and overall total. Green cells correspond to sampling zeros, red cells to structural zeros, and the margins appear in the outer gray cells. 16

2.2 Historical evolution of population synthesis methods. 32

3.1 The autoencoder is an unsupervised model trained to copy its inputs. It compresses the input vector X with dimensions n into the latent representation Z with fewer dimensions and then reconstructs it back into the data space. . 39

3.2 Variational Autoencoder architecture : (a) During training the random variable ε is injected into the latent space Z as an external input; (b) After training, input a Z following a normal random law in order to generate a new sample. 42

3.3 Loire-Atlantique Department/France 46

3.4 Best Variational Autoencoder architecture 53

3.5 Performance on the individual attributes. 54

3.6 Performance on individual attributes: link with the household reference, gender, educational level, and socio-professional class 55
3.7 Performance of VAE on the household attributes: Marginal distributions . . . 56
3.8 Performance on the household attributes: housing type, house occupation type, number of vehicles, internet availability 57
3.9 Values of the various metrics on the 41.6 k-dimensional joint representation with a growing training set size 58
3.10 Zero-cell problem in the case of samples with two attributes x_1, x_2. Sampling zeros are possible representation of individual features (x_1, x_2) not presented during the training. Structural zeros are impossible combinations of features that are still generated by the model. 59
3.11 Fraction of total sampling zeros and total structural zeros generated in the synthetic population for each VAE and for a growing total population size. . 61
3.12 Ratio of the number of structural zeros to sampling zeros versus the size of the generated population. 62

4.1 (a) General architecture of a VAE for population modeling with two layers (Household, Individuals) ; (b) Inputs for SVAE and MVAE representations. . 70
4.2 Marginal distributions for the true population (green) and samples generated using VAE (orange and blue) for individual attributes 76
4.3 Goodness of fit of the marginal distributions of generated individual data to the test values 76
4.4 Multivariate comparison between predicted and test data for individual attributes (link_ref_person, age, sex, has_license, socioprofessional_class) . . 76
4.5 Marginal distributions for the true population (green) and samples generated using VAE (orange and blue) for household attributes 77
4.6 Goodness of fit of the marginal distributions of generated household data to the test values 77
4.7 Multivariate comparison between predicted and test data for household attributes (number of vehicles, has Internet) 77
4.8 Combined multivariate comparison between predicted and test data (household_size, age, sex) 80

5.1 Architecture of the VAE+IPF method. 89
5.2 Goodness of fit of the univariate distributions for all attributes. 93
5.3 Goodness of fit of the univariate distributions for attributes not included in the IPF procedure. 94
5.4 Goodness of fit of the multivariate distributions for attributes : *house type*, *internet access*, *gender*, *age*, *education level*, and *socio-professional class* . . . 95
5.5 Proportion of zero-cells before and after the IPF procedure. 96

6.1 Illustration of the two common representations of an activity plan: TimeGrid (a) and SeqRep (b and c). 102
6.2 Conceptual diagram of the Conditional Variational Autoencoder (CVAE). . . 104
6.3 Illustration of the embedding process. Categorical activity attributes are projected into a continuous latent space through word embeddings 106
6.4 Internal structure of a *Transformer* block, including the masked multi-head attention and feed-forward network. 107
6.5 Multi-head attention mechanism within a Transformer block: linear projection into Query–Key–Value spaces, computation of attention weights, concatenation, and causal masking. 109
6.6 Conceptual architecture of the *ActivityGPT* model: input and positional embeddings, stacked Transformer blocks, and multiple output layers, each corresponding to an activity attribute. 112

List of Tables

2.1 Comparative summary of the main families of methods for population synthesis 29

3.1 Overview of household and individual features 48
3.2 Overview of the models hyper-parameters 50
3.3 Samples of investigated architectures and hyperparameters investigated, along with their performance on the validation set 52
3.4 Total number of sampling zeros and structural zeros and Fraction of agents whose cross modalities are sampling zeros and structural zeros over the 1.4M generated agents according to the training size (18k-dimensional joint)(27k-dimensional joint) 61

4.1 Overview of household and individual features 72
4.2 Models' Hyper-Parameters Overview 73
4.3 Overview of the best-performing models 74
4.4 Overview of how the MVAE sub-models performed on household attributes. . 79

5.1 IPF example 87
5.2 Overview of household and individual features 91

6.1 List of activity and individual attributes used for generating activity plans. . 114

Publications

Journals

1. Sané, A. R., Vandanjon, P.-O., Belaroussi, R. et Hankach, P., A comprehensive investigation of variational auto-encoders for population synthesis. *Journal of Computational Social Science*, 2025. 8(1):13

2. Sané, A. R., Belaroussi, R., Hankach, P. et Vandanjon, P.-O., Population synthesis with deep generative model: a joint household-individual approach. *Computational Urban Science*, 2025, 5(1):34

International Conferences

1. Abdoul Razac Sané, Pierre-Olivier Vandanjon, Pierre Hankach, Rachid Belaroussi, Intégration d'un VAE et de l'IPF pour la génération de populations synthétiques par IRIS : une application à l'enquête ménage déplacement, *7e édition des Rencontres Francophones Transport Mobilité*, Dunkerk, France, 2025

2. Abdoul Razac Sané, Pierre-Olivier Vandanjon, Pierre Hankach, Rachid Belaroussi, Population synthesis using Variational AutoEncoders, *Eqasim Workshop*, Paris, France, 2024

National Conferences

1. Abdoul Razac Sané, Pierre-Olivier Vandanjon, Pierre Hankach, Rachid Belaroussi, Variational Auto Encoder for synthetic population generation, *Journée Population Synthétique*, Nantes, France, 2025

2. Abdoul Razac Sané, Pierre-Olivier Vandanjon, Pierre Hankach, Rachid Belaroussi, Génération de Population Synthétique avec un Variational Auto-Encoder (VAE), *Meet up FAISTOS/IRSTV - IA appliquée aux environnements urbains*, Nantes, France, 2025

Chapter 1

Introduction

1.1 Synthetic Population

In general terms, a synthetic population refers to a set of artificially generated agents designed to reproduce the structure and diversity of a real population. These agents may represent various entities—individuals, households, firms, or even vehicles—depending on the application domain. In the context of this thesis, the term synthetic population specifically denotes a population of individuals organized into households and described by a set of socio-economic attributes. Nevertheless, the methodological contributions developed here are sufficiently generic to be transferable to other types of agents.

The generation of a synthetic population aims to digitally reconstruct the composition of a territory from partial information derived from surveys, censuses, or other statistical sources. It consists in producing a set of individuals whose aggregated characteristics are consistent with observed data, while introducing a degree of variability that allows the representation of attribute combinations not present in available samples. The synthetic population thus becomes an operational approximation of the real population, compliant with known statistical constraints.

Such artificial populations constitute a fundamental component of many modeling and simulation frameworks. They enable the evaluation of public policies and the exploration of hypothetical scenarios that cannot be directly observed. Depending on the domain, they are used to simulate epidemic propagation (Gómez-Marín *et al.*, 2023), assess healthcare demand (Dumont *et al.*, 2017), analyze population evacuation (Daudé *et al.*, 2019), evaluate public transport performance (Delhoum *et al.*, 2024), or anticipate the evolution of mobility systems (Hörl et Balac, 2021b). Their main strength lies in their capacity to bridge individual and collective levels, offering a detailed and coherent representation of behaviors and interactions within a given territory. To be genuinely beneficial for public decision-making, and ultimately for citizens, such models must be understandable and interpretable by those who use or are affected by them, enabling a critical assessment of their assumptions,

advantages, and limitations (Rouchier, 2024).

In the field of mobility, a synthetic population exhibits three essential properties, illustrated by Figure 1.1(a) and (b). First, it is structured at two levels: individuals are grouped into households, each characterized by specific attributes such as size, dwelling type, or level of motorization. Second, each individual is associated with an activity plan describing a typical day—for example, accompanying children to school, commuting to work, shopping, or engaging in leisure activities. Third, the population is spatially anchored: individuals and their activities are located on a specific territory, which allows mobility demand to be linked to existing infrastructures and the spatial organization of resources.

Within this framework, a synthetic population for mobility serves as an input to agent-based transport simulators. It provides a disaggregated representation of travel demand by connecting socio-economic characteristics of individuals with their activity and mode-choice patterns. Beyond this operational role, it also serves as a tool for territorial analysis: the synthetic population offers a coherent representation of the social and spatial fabric of a region and thus becomes a direct decision-support tool for urban planning, transport policy, and environmental assessment.

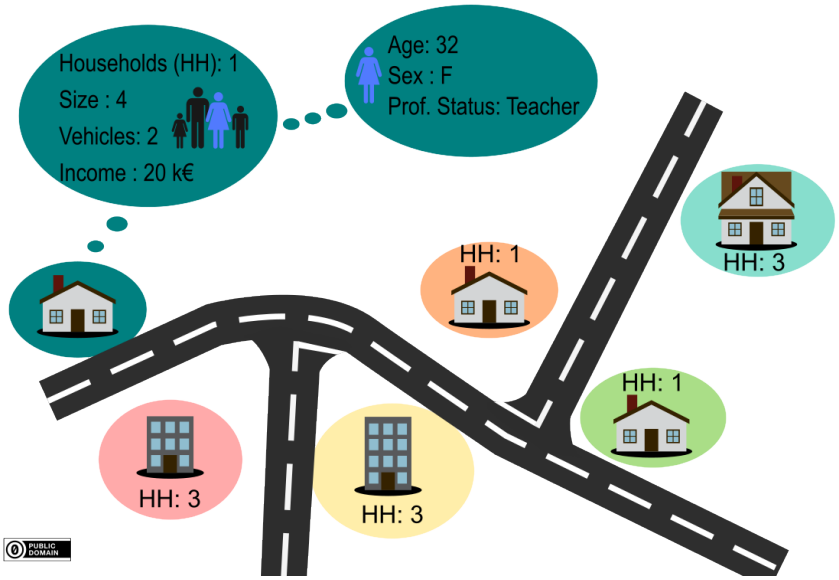
This thesis adopts this perspective. It focuses on generating, for a given territory, a synthetic population of individuals and households endowed with daily activity plans, with the objective of analyzing mobility behaviors and their determinants.

In the mobility domain, a synthetic population serves two roles. First, it constitutes the required input to agent-based transport simulators. Second, it offers a tractable model of the territory in its own right, from which indicators and scenarios can be derived for planning and policy analysis. The next two paragraphs briefly outline, respectively, agent-based modeling for transport systems and territorial modeling.

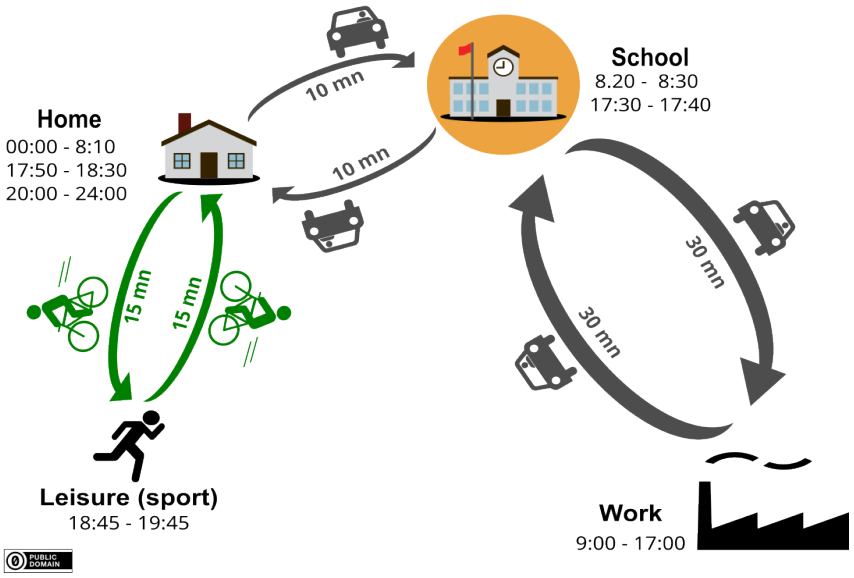
1.2 Agent-Based Models for Transport Policy Evaluation

Agent-Based Models (ABMs) have become a central framework for modeling and simulating transportation systems. Their core principle lies in the explicit representation of individuals and their interactions, linking mobility demand and transport supply within a unified simulated environment. Each agent represents an individual endowed with specific characteristics, preferences, and constraints, who interacts with other agents and with the transport system according to predefined behavioral rules. This approach makes it possible to reproduce the complexity of mobility phenomena and to capture the emergent effects resulting from the combination of individual decisions.

Historically, the first ABMs applied to transportation were developed to analyze road



(a) Example of a localised synthetic population with households and individuals characterised by socio-economic attributes.



(b) Example of an activity plan describing the daily sequence of activities and travel for an individual.

Figure 1.1: Synthetic populations for mobility studies: (a) localised socio-economic representation of households and individuals; (b) associated daily activity plans.

congestion and assess the balance between travel demand and network capacity (Horni *et al.*, 2016). By simulating the behavior of thousands of interacting individuals, these models were able to reproduce collective phenomena such as traffic jams, modal split, and the temporal dynamics of travel demand. Over time, their scope has expanded to include environmental, energy, and public health dimensions associated with mobility.

One of the main strengths of ABMs lies in their ability to simulate public policy scenarios prior to their actual implementation. They enable ex ante evaluation of measures such as the establishment of low-emission zones, congestion pricing schemes, the introduction of new public transport services, or the promotion of active modes. By reproducing individual behaviors in a realistic setting, these models generate aggregated indicators—such as emissions, travel times, or accessibility—that support evidence-based decision-making.

ABMs rely on a synthetic population representing the travel demand of the simulated territory. Each agent is associated with an activity plan and travel choices consistent with their socio-economic characteristics and the constraints of the network. One of the most widely used models in this domain is MATSim (Multi-Agent Transport Simulation), an open and extensible platform designed to simulate mobility decisions at the individual level. MATSim operates through an iterative optimization process: agents adjust their activity plans and travel choices across iterations to maximize their utility, gradually converging toward a system-level behavioral equilibrium.

In this perspective, ABMs constitute a powerful framework for transport policy evaluation. By coupling a realistic synthetic population with a behavioral simulator, they link the social and spatial characteristics of a territory to the mobility dynamics they generate. They therefore provide a coherent analytical framework for assessing the distributive, energy, and environmental impacts of public policies at fine spatial and temporal scales (Gastineau *et al.*, 2023).

1.3 Territorial Modeling

Territorial modeling aims to represent population, activities, and space jointly in order to analyze their interdependencies. It describes how socio-economic characteristics of residents, the distribution of urban functions, and spatial organization contribute to local dynamics. In this perspective, the synthetic population provides a micro-data set consistent with available information and anchored to a given territory.

Using a synthetic population makes it possible to produce disaggregated representations of daily practices and their locations. By combining individual attributes, household structures, and spatial positions, one can build maps of presence, attendance, or potential demand and identify territorial contrasts. These outputs are then used to formulate scenario

hypotheses and construct indicators that are comparable over time and across space.

Synthetic populations can be used to estimate differentiated exposure and sensitivity to environmental nuisances (noise, air quality, heat) (Hankach *et al.*, 2024; Galassi Luquezi *et al.*, 2025). The key point is the ability to link, at the individual level, (i) effective presence on site according to activity plans, (ii) the intensity of a nuisance at a given place and time, and (iii) the presumed or stated sensitivity of individuals based on their socio-economic characteristics.

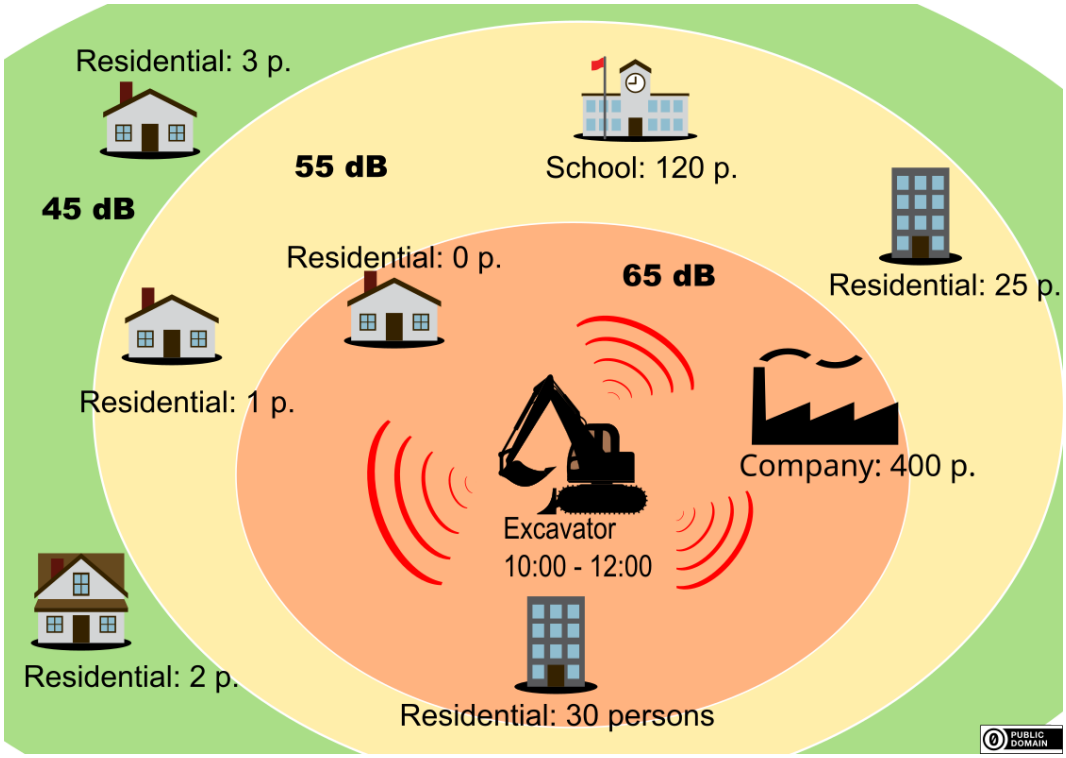


Figure 1.2: Example use of a synthetic population for urban worksite management: scheduling noisy machinery according to the people effectively present within noise influence zones (65/55/45 dB) and their noise sensitivity.

An example is shown in Figure 1.2: urban worksite management seeks to operate noisy machinery at times when it will disturb the fewest nearby people. The synthetic population allows hour-by-hour estimation of persons effectively present within noise influence zones (e.g., 65 dB, 55 dB, 45 dB), distinguishing place types (dwellings, schools, companies) and weighting impacts by noise sensitivity profiles (age, employment situations, schedules, etc.). Worksite planning can then minimize an aggregated, sensitivity-weighted exposure indicator rather than the nominal count of resident inhabitants.

To move from territorial representation to micro-data usable in simulation, the next section details the generation process step by step.

1.4 Synthetic Population and Activity-Plan Generation

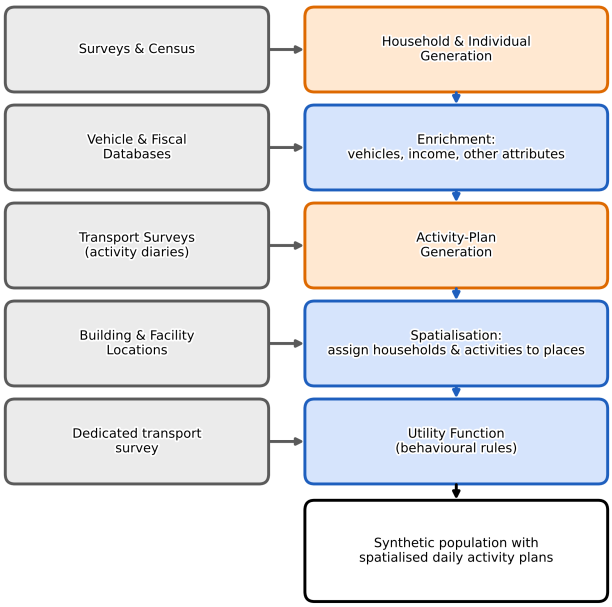


Figure 1.3: Example sequencing of operations to produce a synthetic population with spatialised daily activity plans: household and individual generation, attribute enrichment, activity-plan generation, spatialisation, then definition of utility functions for modal choice. Orange boxes indicate the specific contributions of the thesis.

The objective is to produce, for a given territory, a database of individuals and households endowed with socio-economic attributes, a spatial anchoring, and daily activity plans. Figure 1.3 illustrates one possible sequencing of the operations. The process unfolds sequentially as follows:

- Household and individual generation. Based on surveys and census data, construction of a population consistent with observed distributions (household structure, ages, statuses, etc.).
- Attribute enrichment. Integration of information from secondary sources (e.g., vehicle and fiscal databases) to add attributes such as car ownership, income level, or other relevant characteristics.

- Activity-plan generation. Using transport surveys (activity diaries), production of daily sequences consistent with socio-economic profiles.
- Spatialisation. Assignment of households, individuals, and activities to observed places (buildings, facilities) to establish an explicit correspondence with space.
- Utility function. Assignment to each individual of a utility function that formalises modal-choice trade-offs, preparing subsequent integration into an agent-based simulator.

This sequencing yields a synthetic population with spatialised daily activity plans, suitable for territorial analysis and ex-ante evaluation.

The sequencing shown in Figure 1.3 is indicative rather than prescriptive: it adapts to the available data, their level of detail, and the constraints of each case study.

Conventional approaches for generating such synthetic populations are often designed in an ad hoc manner, depending on the available data sources. They do not fully exploit the potential recently opened by Artificial Intelligence (AI) to learn complex dependencies and to generate new individuals in a controlled and statistically consistent way. The following section presents these methodological advances.

1.5 Artificial Intelligence as a Source of New Methodologies for Synthetic Population Generation

Recent advances in Artificial Intelligence (AI) have introduced generative models capable of learning multivariate dependencies from observed samples and producing unseen individuals while preserving target statistical properties. Their relevance for population synthesis lies in their ability to handle mixed data (categorical and numerical after encoding or discretization), to capture latent structures, and to generate new profiles sampled from these representations.

Deep Generative Models: General Principle

Deep generative models learn a latent representation of the data in a reduced-dimensional space. Two components are central: (i) the construction of this latent representation from observations, and (ii) a sampling mechanism that draws new points in the latent space and projects them back into the attribute space. In the following, the feature of the Variational Autoencoder (VAE) is presented because this methodology was the main used during this thesis.

Variational Autoencoder (VAE) to generate individuals

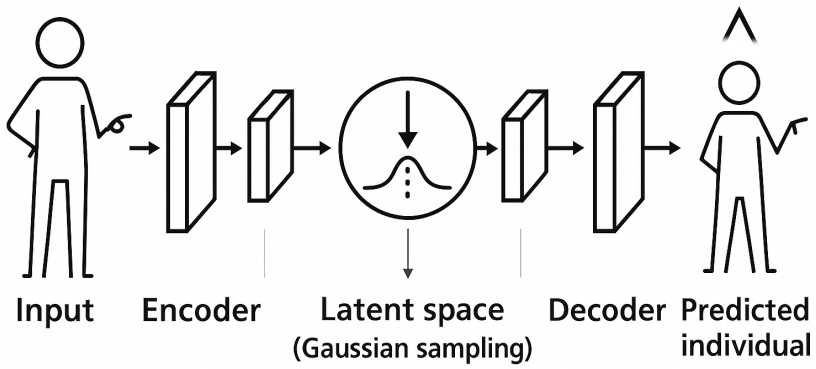


Figure 1.4: Illustration of the Variational Autoencoder (VAE) principle: an input individual (left) is encoded into a probabilistic latent space where each dimension follows a Gaussian distribution. Random sampling within this space generates new latent points that are decoded into predicted individuals (right).

The Variational Autoencoder (VAE) consists of an encoder that projects each observation into a probabilistic latent space (typically Gaussian) and a decoder that reconstructs the attributes from a randomly sampled latent vector. Figure 1.4 illustrates this general mechanism. The stochastic component introduced in the latent space enables the controlled generation of new, unseen individuals that remain statistically similar to the training data. More precisely:

- the encoder produces, for each observation, a pair (mean, standard deviation) defining a probability density in the latent space;
- a latent sample is randomly drawn and passed to the decoder;
- training minimizes both a reconstruction term (fidelity to the original data) and a regularization term (alignment with a prior latent distribution, generally a centered normal law).

In this thesis, this framework is used to generate individuals (and, when applicable, households) with socio-economic attributes. The latent space acts as a continuous manifold where new profiles can be sampled, decoded, and integrated into the synthetic population.

Activity Plans and Sequential Dimension: Conditional VAE and GPT

The generation of activity plans requires modeling both the sequential structure and the dependencies between successive activities. Two complementary approaches are considered:

- using **Conditional Variational Autoencoders (CVAE)** to sequential data (activity representations, durations, transitions) by conditioning on socio-demographic variables and incorporating temporal or spatial features into the latent space;
- the *ActivityGPT* model, which is specifically designed for modelling activity chains based on Generative Pre-trained Transformers (GPT) architecture. This autoregressive architecture uses self-attention mechanisms to capture long-term temporal dependencies and the sequential order of daily activities, producing coherent, context-aware synthetic plans.

In both cases, the synthetic population maintains internal consistency with individual socio-economic characteristics and spatial contexts, ensuring that generated activity plans remain behaviorally coherent and contextually plausible.

1.6 Positioning

This thesis aims to explore recent approaches from Artificial Intelligence (AI), in particular Variational Autoencoders (VAE), for the generation of synthetic populations of individuals grouped into households and endowed with daily activity plans, with the objective of studying mobility behaviors.

It has been carried out within the Systems, Logistics, Organization of Transport, and Work research group (Systèmes Productifs, Logistique, Organisation des Transports et Travail - Splott) of the Planning, Mobility and Environment (Aménagement, Mobilité, Environnement - Ame) laboratory at University Gustave Eiffel. This institutional context has allowed the research to benefit from cross-disciplinary influences, particularly from economics, geography, and psychology, which contribute to a broader understanding of travel behaviors and territorial dynamics.

This research was also conducted within the interdisciplinary CNRS Research Federation, the Institute for Research on Urban Science and Technology (Institut de Recherche en Sciences et Techniques de la Ville - IRSTV). This affiliation fostered exchanges with other disciplines, including sociology, biology, and again economics and psychology.

The work presented here is part of a collective scientific project aimed at developing decision-support tools based on agent-based territorial modeling, in the perspective of ecodesign and the sustainable management of transport systems, hosted in Splott. In this

context, the thesis has also drawn on complementary advances from applied statistics, notably in population generation and spatial filtering.

In return, the contributions developed in this research will directly benefit, in the short term, this scientific project by strengthening the methodological expertise of the host team. In the medium term, they will contribute to another research program of the SPLOTT team, which focuses on urban logistics modeling through agent-based systems. In this perspective, the methods explored in this thesis may be extended to the generation of synthetic business populations, integrating delivery demand and supply as well as daily delivery tour plans, in order to better represent interactions between passenger mobility and freight transport in urban environments.

1.7 Contributions

In the process of generating synthetic populations and spatialised activity plans, the work conducted in this thesis focuses on two main components: (i) the generation of individuals and households, and (ii) the generation of activity plans, as illustrated in Figure 1.3.

The academic contributions derived from this research are as follows:

- Generation of individuals using VAEs. A pedagogical and reproducible approach was developed to promote the use of Variational Autoencoders (VAE) for population synthesis. This work was published in the Journal of Computational Social Science Sané *et al.* (2025b).
- Joint and coherent generation of individuals and households. Two new methods were proposed to simultaneously produce individuals and households while ensuring statistical consistency. These results were published in Computational Urban Science Sané *et al.* (2025a).
- Household generation using a VAE constrained by external aggregate statistics. A new method was developed to generate households consistent with external aggregated data from secondary sources. Its application to the Île-de-France region was presented at the Rencontres Francophones Transport et Mobilité (RFTM) in 2025 in Dunkirk (France). A methodological paper is being written.
- Generation of activity chains. Two different techniques were explored for generating daily activity plans: Conditional VAEs and ActivityGPT. Each technique was modelled independently, opening the way for further investigations.

All source codes supporting the published results are freely available to foster reproducibility and reuse of the work. The remaining codes will be released as soon as the corresponding

papers are published. Special attention was paid to the clarity and structure of the programming to ensure that the implementations are easily reusable by other researchers.

In the longer term, after their industrialisation, these modules will be integrated into the Eqasim software pipeline, an open-source process designed for the generation of synthetic populations and activity plans based on French data (Hörl et Balac, 2021a).

Each contribution forms the core of a dedicated chapter in this manuscript, while the final chapter presents a synthesis of the results and the research perspectives opened by this work.

Chapter 2

Synthetic Population: A Literature Review

Contents

Abstract **14**

2.1 Key Concepts **15**

2.1.1 Synthetic Populations 15

2.1.2 Hierarchical Data 16

2.1.3 Activity Plans 17

2.2 Methods for population synthesis **17**

2.2.1 Traditional Methods for Population Synthesis 18

2.2.2 Deep Generative Models 22

2.3 Methods for Activity Plan Synthesis **26**

2.3.1 Statistical Matching 26

2.3.2 Machine Learning (ML) 27

2.3.3 Deep Generative Models 28

2.4 Cross-comparison of approaches for population synthesis . . . **28**

2.5 Historical Perspectives **30**

2.6 Conclusion and Positioning **33**

Abstract

Abstract

This chapter provides a comparative review of the main approaches to synthetic population generation, ranging from classical reweighting methods (*Sample Reweighting*) to recent deep generative models, as well as combinatorial optimization techniques, probabilistic models, and large language models (LLMs). Each family is analyzed with respect to its generative mechanisms, strengths (e.g., fidelity to marginals, flexibility, diversity of agents), and limitations (e.g., computational cost, sensitivity to dimensionality, difficulties in integrating structural constraints).

The cross-cutting analysis shows that while traditional approaches ensure robustness and simplicity, they struggle to handle the complexity of multi-level data. Deep generative models, particularly VAEs, GANs, and diffusion models, open promising perspectives for balancing fidelity and diversity, though they still face challenges in calibration and structural coherence. LLM-based approaches represent an emerging alternative, characterized by their flexibility and ease of use, but requiring complementary validation mechanisms.

In conclusion, no single paradigm emerges as a universal solution. The future of synthetic population generation appears to lie in hybrid approaches, combining the rigor of classical methods with the expressive power of modern generative models, in order to meet the demands of statistical fidelity, multi-level coherence, and diversity in synthetic populations.

Keywords : *synthetic population, synthetic reconstruction, combinatorial optimization, statistical learning, Bayesian networks, variational autoencoders, GAN, diffusion models, LLM, sampling zeros, structural zeros, household–individual*

2.1 Key Concepts

This chapter begins by introducing the fundamental notions underlying synthetic population generation. This section aims to establish a common vocabulary, avoid ambiguities, and clarify the main challenges associated with the statistical modeling and representation of populations.

2.1.1 Synthetic Populations

The term *population* refers to the entire set of statistical units of interest, whether they are individuals or households. In practice, practitioners generally have access only to partial data drawn from a *sample*, often collected through surveys or censuses. Synthetic population generation seeks to reconstruct an artificial population that reproduces the characteristics of the real population, using both the available sample and auxiliary information from official sources.

A population can be represented in two complementary ways.

The first representation takes the form of an *individuals* $\times$ *variables* table, also known as "tabular data", where each row corresponds to a statistical unit and each column to an *attribute* (i.e., a descriptive characteristic of the unit), such as age, sex, or education level. An attribute may be either *categorical* or *numerical*. In the categorical case, its values are drawn from a finite set of *modalities*. For example, for the attribute *socio-professional category*, the modalities may include "Executive", "Student", or "Retired". In contrast, a numerical attribute is defined on a continuous or discrete scale. For instance, the attribute *monthly income* may take values such as 1 500 or 1 800.

The second representation relies on the use of a *contingency table*, in which each cell contains the count associated with a specific combination of attribute modalities, where a modality represents a possible value of an attribute. When more than two attributes are involved, this contingency table becomes a multidimensional table or tensor. Each dimension of the tensor corresponds to an attribute, and the overall size of the tensor equals the total number of modality combinations. To give an order of magnitude, it is common to work with populations comprising around 20 attributes. If each attribute can take four modalities, the corresponding contingency table would contain $20^4 = 160\,000$ cells.

Figure 2.1 illustrates this case for two attributes, x_1 and x_2. Each cell represents the number of observed units, while the outer margins display the one-dimensional distributions of each attribute.

These two representations convey two types of statistical information. The *marginal data*, derived from the margins of the table (in gray), describe the distribution of each

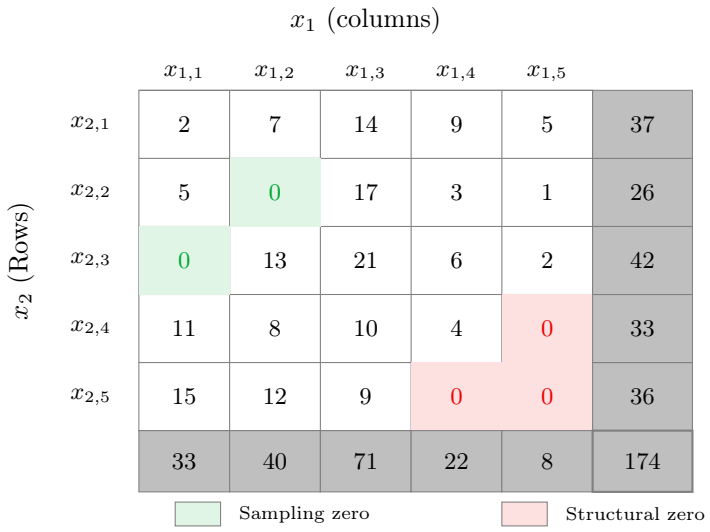


Figure 2.1: Example of a contingency table $x_1 \times x_2$ showing cell counts, row and column margins, and overall total. Green cells correspond to sampling zeros, red cells to structural zeros, and the margins appear in the outer gray cells.

attribute taken individually and serve as essential constraints for many population synthesis methods. In practice, the term "marginal data" refers to auxiliary information obtained from official sources. They are generally treated as constraints. For example, if the total number of inhabitants in a city is known, this constitutes a marginal datum and a constraint: the total size of the generated synthetic population must match this number.

The *joint data*, denoted $P(x_1, x_2)$ in the case of two attributes, describe the joint distribution of these attributes and capture the dependencies between them.

Sample Samples can also be represented as synthetic populations in the form of a table of individuals, or in the form of a contingency table. Some cells may be empty, i.e. absent from the sample, without this implying that they are not present in the population from which the sample was drawn. In this case, we refer to *sampling zeros* (shown in green in the figure). Similarly, some cells—combinations of modalities—are impossible by construction in both the population and the sample and correspond to *structural zeros* (shown in red in the figure). For example, a 5-year-old child with a university degree.

2.1.2 Hierarchical Data

Populations are not limited to isolated individuals: they are organized into hierarchical structures. The first level typically consists of *households*, domestic units defined by attributes such as size, dwelling type, location, or motorization. Each household groups together several

individuals, characterized by personal attributes such as age, sex, education level, occupational category, or employment status. The term "two-layered synthetic population" is often used to describe a synthetic population composed of households and their corresponding individuals.

Beyond this first relationship between households and individuals, additional levels of organization may exist, such as the assignment of households to geographical zones (e.g., neighborhoods, municipalities, regions). Such structures are referred to as *hierarchical* or *multi-level data*. A synthetic population must preserve coherence across these levels: individual attributes must remain consistent with household characteristics (for example, the number of children with the mother's age), and household features must in turn align with the socio-economic properties of their geographical area.

2.1.3 Activity Plans

In certain application domains—particularly in transportation and mobility studies—the objectives extend beyond the generation of synthetic populations that merely describe the socio-demographic characteristics of individuals and households. They also encompass a behavioral dimension through the inclusion of *activity plans*.

An *activity plan* describes the sequence of an individual's daily actions—such as work, school, shopping, or leisure—along with their temporal and spatial characteristics: activity locations, start and end times, durations, and the modes of transport used. This representation—combining *synthetic population* and *activity plan*—enables the modeling not only of population composition but also of the interactions between individual behaviors and collective constraints (e.g., household resource sharing, transport infrastructure, social rhythms). It thus constitutes a central component for simulating complex phenomena such as urban congestion, the diffusion of mobility innovations, or the spread of infectious diseases.

2.2 Methods for population synthesis

Since the 1940s, population synthesis has continuously evolved, gradually moving from traditional statistical methods to modern approaches based on machine learning and artificial intelligence. This evolution reflects not only the increasing availability of data and computational power but also the growing diversity of needs in modeling synthetic populations.

Over time, each generation of methods has sought to overcome the limitations of its predecessors or to introduce a new conceptual framework for population synthesis. Research efforts have focused on improving fidelity to marginal data, handling high-dimensional feature spaces, accommodating both numerical and categorical (i.e., mixed-type) attributes, explicitly modeling hierarchical structures, and, more recently, incorporating behavioral

components such as activity plans.

It is presented here an overview of the main families of population synthesis methods, drawing on the classification proposed by Fabrice Yaméogo *et al.* (2021), which distinguishes three major categories: (i) *Synthetic Reconstruction* (SR); (ii) *Combinatorial Optimization* (CO); and (iii) *Statistical Learning* or *Machine Learning* (ML). This taxonomy is extended by introducing a fourth category: *Deep Learning*. Although it can be regarded as a subset of ML, the deep learning family deserves to be treated separately due to its distinct methodological principles and its recent, rapid development.

This section is organized into two main parts. First, the so-called traditional approaches, encompassing SR, CO, and ML methods are presented. Second, the more recent population synthesis methods based on *deep learning* are discussed.

2.2.1 Traditional Methods for Population Synthesis

2.2.1.1 Synthetic Reconstruction (SR)

Synthetic Reconstruction (SR) is among the earliest methods for generating a synthetic population. It takes as input both a micro-sample with categorical attributes[1] and aggregate marginal data. Through a deterministic process, SR reconstructs the population by reproducing all known constraints from the marginals (Zhu et Ferreira, 2014; Lim et Gargett, 2013). The two main synthetic reconstruction methods are: iterative proportional fitting and iterative proportional updating (Ye *et al.*, 2022; Lovelace *et al.*, 2015; Ye *et al.*, 2016);

The initial studies conducted on synthetic data generation date back to the 1940s with the work of Deming et Stephan (1940) who laid the foundations for Iterative Proportional Fitting (IPF) methods. Subsequent studies, such as those by Ireland et Kullback (1968), Fienberg (1970) and Deville *et al.* (1993), have helped to consolidate the theoretical foundations such as Generalized Raking.

As its name suggests, IPF entails iteratively adjusting the weights of the individuals in the sample such that the attribute contingency table constructed from these individuals corresponds to the marginal distributions. In other words, IPF minimizes the discrepancy between the values in the contingency tables and the marginal data. The synthetic population is then obtained by duplicating each individual proportionally to their weight.

This method offers the dual advantage of being relatively simple while ensuring that the aggregated values of the synthetic population correspond to the marginal data (Beckman *et al.*, 1996). However, the IPF approach, as originally proposed by Deming et Stephan (1940) and later adapted by Beckman *et al.* (1996), also presents several limitations. First,

[1] Numerical attributes are discretized into categories

it requires a sufficiently large number of individuals to achieve satisfactory results. Second, it becomes problematic when a large number of attributes are considered. Indeed, the number of contingency table cells grows exponentially with the number of attributes: for example, three attributes with five categories each yield $5 \times 5 \times 5 = 5^3 = 125$ cross-classified cells, whereas four such attributes already result in $5^4 = 625$ cells. This exponential growth not only produces a substantial proportion of empty cells but also entails significant computational and memory demands. Finally, the resulting weights are generally non-integer values.

Faced with these limitations, Pritchard et Miller (2012) have proposed a new approach that combines the classical IPF with the sparse data structure proposed in Williamson *et al.* (1998). This structure enables an efficient use of computing resources through storage proportional to the number of nonzero cells in the initial array. By reducing the table to nonzero cells, this method allows increasing the number of attributes while maintaining reasonable computational complexity (Pritchard et Miller, 2012). Furthermore, Lovelace et Ballas (2013) proposed an improvement to the IPF method, called "Truncate, Replicate, Sample", which takes non-integer weights into consideration.

The original IPF algorithm, as proposed by Deming et Stephan (1940), does not account for hierarchical data. Variants of IPF, such as the Iterative Proportional Updating (IPU) introduced by Ye *et al.* (2009) and the Hierarchical IPF (HIPF) proposed by Müller et Axhausen (2012, 2011), extend the method to handle hierarchical population structures. Both algorithms iteratively estimate the weights of households and individuals based on a cross-classification of household- and individual-level attributes (Yameogo *et al.*, 2021). The HIPF algorithm alternates between the household and the individual level through an adjustment step based on entropy maximization (Müller et Axhausen, 2012). By contrast, the IPU algorithm first adjusts the weights to satisfy household-level constraints and then updates them to ensure that individual-level constraints are also met.

2.2.1.2 Combinatorial Optimization (CO)

Combinatorial Optimization (CO) methods can generate synthetic households and individuals directly, typically by replicating units from a reference sample while ensuring consistency with aggregate population data (Voas et Williamson, 2000; Ma et Srinivasan, 2015). Like Synthetic Reconstruction (SR) methods, they require both a micro-sample (with categorical attributes) and marginal data. However, important differences exist: CO approaches are stochastic and produce complete households and individuals, thereby avoiding the need for a separate allocation step as in SR methods. Moreover, their data requirements are generally less restrictive (Templ *et al.*, 2017). A major drawback, however, lies in their computational complexity. As the size of the target population increases, these methods cannot guarantee an optimal solution and demand substantial processing time, which limits their widespread

use (Lee et Fu, 2011).

In practice, CO methods construct the synthetic population by selecting or swapping units from the reference sample in order to optimize a fitness function relative to the known marginals (Harland *et al.*, 2012; Ma et Srinivasan, 2015; Templ *et al.*, 2017; Chapuis *et al.*, 2022). A range of optimization schemes have been applied, from greedy heuristics and hill-climbing to simulated annealing and genetic algorithms. Standard goodness-of-fit measures, such as the χ^2 statistic, entropy, or normalized deviations, guide the elementary operations (addition, replacement, or permutation of units) until convergence. These approaches are highly flexible, capable of accommodating multiple constraints simultaneously, and are relatively memory-efficient since they do not rely on large tensor operations. Nevertheless, they remain computationally intensive, do not ensure convergence to a global optimum, and are restricted to replicating observed units.

2.2.1.3 Machine Learning (ML)

Machine learning models aim to capture the underlying statistical structure of training data in order to generate new instances resembling the observed ones. This section begins by introducing Markov Chains and hierarchical mixture models, both of which sequentially update variables according to plausible conditional distributions. It then turns to Bayesian networks. Because these approaches seek to recover the probability distribution from which the data were sampled, this class of methods—often grouped under *machine learning*—is sometimes referred to as *statistical learning*.

Farooq *et al.* (2013) proposed a Gibbs sampling based on the Markov Chain Monte Carlo (MCMC) to simulate a draw of the joint distributions of attributes conditional upon the available marginal distributions. In cases where the conditional distributions where incomplete, the authors proposed estimating them using a discrete-choice model. The results obtained demonstrated that this approach outperforms the IPF. However, it is important to note that the efficiency of this method depends on the availability and reliability of conditional distributions.

Following this initial approach, Saadi *et al.* (2016b) proposed a new method based on Hidden Markov Models (HMM), that proves to be more scalable. According to this approach, the characteristics of individuals are assimilated to a sequence of hidden states, which implies a specific order between the attributes. Numerical attributes are discretized, with the different modalities of each attribute being considered as hidden states. Thanks to this modeling set-up, it becomes easier to estimate the conditional distributions of the ith hidden state of the sequence by taking the previous hidden states into account. This approach thus offers better data management and greater processing capacity.

Other works, such as those of Casati *et al.* (2015) and Sun *et al.* (2018), have proposed

synthetic population generation methods to capture the hierarchical structure that exists between individuals in a household. The Casati *et al.* (2015) method, referred to as "hierarchical MCMC (hMCMC)", follows a multi-step process. First of all, individuals in the same household are ordered according to their role (i.e. owner, spouse-children, others). Second, using the MCMC (Farooq *et al.*, 2013), the characteristics of households and owners are generated. Third, the spouses and children are generated conditionally based on the joint probabilities of the household and owner data generated during the second step. Fourth and last, the other members are generated conditionally on the joint probabilities of the second and third stage data.

The hierarchical mixture (HM) model for population synthesis, as presented in Sun *et al.* (2018), combines three distinct models: probabilistic tensor factorization, multilevel latent class modeling, and rejection sampling. This approach makes it possible to generate a synthetic population while capturing the complex interdependencies between households and individuals. However, the proposed model does not take numerical attributes into account. Moreover, as the number of attributes increases, HM method implementation becomes increasingly complex, and the choice of latent variables can influence its robustness (Ilahi et Axhausen, 2019; Borysov *et al.*, 2019a).

The Bayesian network approach is a model that incorporates both causal relationships and probabilistic semantics; it is used to efficiently approximate or reproduce the joint distribution of various attributes in population systems by identifying the relationships between them. The work of Sun et Erath (2015) introduced this approach into population synthesis. In this context, Bayesian networks are used to estimate local conditions and produce a realization of the population by sampling the estimated network. These authors showed that such networks allow for population syntheses to effectively identify the basic structure of the underlying population system.

By combining traditional survey data with regional social network information, Zhang *et al.* (2019) proposed an algorithm based on a Bayesian network approach to generate connected synthetic populations. Their methodology comprises several stages, including Bayesian network modeling, graphical structure learning, and population synthesis.

Despite their effectiveness in reproducing joint distributions of attributes, Bayesian networks feature some limitations depending on the application case, in particular: the curse of dimensionality, the need for a large sample size, the assumption of conditional independence, and the difficulty in handling numerical variables (Sun et Erath, 2015).

2.2.2 Deep Generative Models

Deep generative models, based on deep learning principles, offer several advantages in managing the high dimensionality of data. They are able to learn the joint distribution of attributes, whether numerical or categorical, and generate new individuals (sampling-zero) not present in the learning sample. In recent years, several studies have investigated the use of these models for synthetic population generation (Borysov *et al.*, 2018, 2019a; Garrido *et al.*, 2020; Johnsen *et al.*, 2022; Aemmer et MacKenzie, 2022).

2.2.2.1 Variational Autoencoders (VAE) and Generative Adversarial Networks (GAN)

Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) are two of the most widely used deep generative models. A VAE is a probabilistic framework composed of two components: an encoder and a decoder. The encoder maps input data into a low-dimensional latent space, where each point corresponds to a probability distribution, while the decoder reconstructs input data from this representation Kingma et Welling (2013). By contrast, a GAN is built on an adversarial game between two networks: a generator that produces synthetic data resembling real samples, and a discriminator that attempts to distinguish between authentic and generated instances. Through this competition, the generator progressively improves its capacity to produce highly realistic data Goodfellow *et al.* (2014).

The pioneering work of Borysov *et al.* (2018, 2019a,b) laid the foundation for the use of generative models in population synthesis. The results of their study indicated that for small-scale problems, common population generation methods such as the Gibbs sampler and Bayesian networks (BN), outperform the VAE in terms of standardised root mean square error (SRMSE). However, as the number of attributes increases, the advantages of the VAE become apparent. Moreover, their study revealed that both VAEs and Bayesian networks can generate new samples that differ from the individuals in the training set, while preserving the same statistical properties and dependencies observed in the original data.

Other studies have also drawn comparisons between Generative Adversarial Network (GAN) and VAE deep generative models. These studies include the work of Garrido *et al.* (2020), Kim et Bansal (2022) and Johnsen *et al.* (2022). In their studies, Garrido *et al.* (2020) and Kim et Bansal (2022) investigated the ability of Wasserstein Generative Adversarial Networks (WGAN) and Variational Autoencoders (VAE) to generate instances of *sampling zeros* from high-dimensional distributions. They further proposed new metrics to address the zero-cell problem, distinguishing between *sampling zeros* and *structural zeros*. The main findings indicate that, although the WGAN generally outperforms other models in predictive accuracy across several metrics, it tends to generate less diverse populations than the VAE.

Conversely, the VAE was found to be more stable than the GAN (Garrido *et al.*, 2020; Johnsen *et al.*, 2022). While the GAN produces synthetic populations that more closely resemble the training data—a property that can be advantageous—it may also result in excessive fidelity to the sample, thereby limiting diversity. These studies thus highlight the importance of striking a balance between fidelity to observed data and diversity in the generated populations.

The work by Aemmer et MacKenzie (2022) was among the first to attempt to synthesise hierarchical data – household and individual-level – using a deep generative model. Their method generates a synthetic population of households and individuals using a VAE/CVAE framework in two steps: first, a synthetic population of households is generated; then, synthetic individuals are generated one by one, conditioned on the household data. However, one major limitation of this approach is its inability to capture the hierarchical structure within households, nor does it model the interdependencies between individuals. As a result, the individuals generated are independent of one another and may not adhere to the constraints inherent in a realistic household structure (e.g. a household composed of a couple with two children). This limitation can have significant implications for downstream applications, such as activity-based transport modelling, where interactions within households can greatly impact individual behaviours. For instance, creating inconsistent households — such as a household which includes only two children — can result in inaccurate simulations of daily mobility patterns or resource allocation. Similarly, the timing, destination and mode of transport chosen by one individual often depend on the activities and constraints of other household members, such as coordinating school drop-offs, shared commutes or joint use of a family vehicle. Capturing these interdependencies is essential in order to produce synthetic populations that are both realistic and policy-relevant.

Recent work by Qian et al. Qian *et al.* (2024) sought to address this limitation by proposing a novel approach that combines VAE with fine-tuning. Their method transforms both the household and its individuals into a single vector and then employs a VAE to generate the synthetic population. To account for the hierarchical structure between household members, the authors introduced a new loss function, called *Decoupled Binary Cross-entropy (D-BCE)*, that quantifies the relationships between each attribute of the household and its individuals. Although this approach has shown some promising results, it has not been validated at a large scale. In fact, the D-BCE loss function is computationally expensive, with a complexity of $O(n^2)$ compared to the standard Binary Cross-Entropy (BCE), whose complexity is $O(n)$. Moreover, this function requires a large amount of data in order to be effective. This computational burden poses a significant challenge to the practical implementation of large-scale urban simulations or planning tools, where the rapid generation of diverse and realistic populations is crucial. Without scalable solutions, it remains difficult to apply such methods in operational contexts, which limits their impact on real-world

decision-making.

2.2.2.2 Diffusion Models

Diffusion models are a class of deep generative models that learn to synthesize data by reversing a gradual noising process. The central idea is to start from real data and progressively add small amounts of Gaussian noise over many steps until the original structure is entirely destroyed, resulting in pure noise. A neural network is then trained to approximate the reverse process, i.e., to denoise step by step and recover the underlying data distribution. Once trained, the model can generate new samples by beginning with random noise and iteratively applying the learned denoising transformations. This approach has demonstrated state-of-the-art performance in producing high-quality and diverse synthetic data, particularly in image generation tasks, while also offering a well-grounded probabilistic formulation (Ho *et al.*, 2020). The application of these models for generating synthetic populations is very recent.

One of the earliest applications of this approach is the *TabDDPM* model, which effectively handles the heterogeneity of tabular variables by means of a *Denoising Diffusion Probabilistic Model (DDPM)*. It has been shown to outperform both VAEs and GANs on numerous datasets, particularly in the context of privacy-preserving data generation (Kotelnikov *et al.*, 2023).

More recently, *TabDiff*, introduced by Shi *et al.* (2025), advances the modeling of mixed data. It implements a continuous-time diffusion process that jointly addresses numerical and categorical attribut, featuring a *feature-wise learnable diffusion mechanism* combined with an end-to-end Transformer. This approach significantly improves the reconstruction of attributes correlations, achieving relative gains of up to +22.5% on certain pairwise correlation metrics compared to state-of-the-art models such as GAN (Shi *et al.*, 2025).

These models offer high generative quality and remarkable sample diversity, while remaining stable during training. However, they come with a very high computational cost, both during training and generation, mainly due to the multi-step sampling procedure and the tuning of the noise schedule (Li *et al.*, 2025). Moreover, the direct incorporation of marginal constraints or the hierarchical structure of data (e.g., household–individual) is not inherently supported and remains an open challenge.

2.2.2.3 Large Language Model (LLM)-based Approaches

An emerging trend in synthetic data generation leverages Large Language Models (LLMs) for the generation of tabular data. Unlike VAEs or GANs, LLMs exploit their in-context learning capabilities to produce realistic synthetic data from textual descriptions, often without requiring task-specific training.

It is worth noting that, apart from the first cited work (Borisov *et al.*, 2022), most studies employing LLMs do not generate synthetic populations in the sense defined in this thesis. They do not aim to reproduce a population anchored to a well-defined territory but rather to create synthetic datasets for medical or clinical applications that are not geographically grounded. These studies are nevertheless included because they may serve as valuable inspiration in very specific contexts—for instance, when generating toy populations to test new methodological frameworks.

Borisov *et al.* (2022) introduced *GReaT*, an autoregressive LLM-based model for realistic tabular data generation. GReaT conditions generation on a subset of observed variables, thereby enabling efficient simulation of the remaining attributes. Experimental results confirm its strong generative quality on mixed datasets (categorial and numerical attributes). In this thesis, the LLM autoregressive feature will be leveraged to generate activity plans (see Chapter 6).

Smolyak *et al.* (2024) proposed a *text-to-tabular* approach that employs a pre-trained, *non-fine-tuned*[2] LLM to generate synthetic clinical data from a simple textual description of the desired dataset. This method does not require access to original data, representing a major advantage in ethically or regulatorily constrained contexts. It demonstrates promising performance in terms of fidelity, utility, and preservation of clinical correlations.

Barr *et al.* (2025) examined the ability of GPT-4o to generate clinical tabular datasets in a *zero-shot* setting. From simple descriptions, GPT-4o was shown to produce statistically consistent datasets aligned with benchmarks such as VitalDB, achieving high fidelity to real distributions without requiring explicit training or calibration .

Finally, Ilaty *et al.* (2025) conducted a comparative analysis of several open-source LLMs for the generation of medical tabular data. Their findings emphasize the critical importance of prompt engineering: different prompting strategies (e.g., exemplar-based or rule-based) allow for fine-grained control over generation without requiring model fine-tuning. A multi-metric evaluation (statistical fidelity,, privacy preservation) showed that rule-based prompts yield the best balance between quality and safety.

LLM-based approaches applied to tabular data generation offer several distinctive advantages. First, they enable the synthesis of data *without the need for a training micro-sample*, relying instead on pre-trained language models that can generalize from textual or structural context. They also allow for fine-grained control of the generation process through *prompt engineering*, eliminating the need for additional fine-tuning. However, these strengths come with significant challenges. The robustness of the generated results remains highly sensitive to prompt design — poorly formulated prompts may lead to hallucinations

[2]Fine-tuning a model means adapting a pre-trained model to a specific task by training it on a smaller, task-specific dataset. This improves performance on the target task without training from scratch.

or logically inconsistent attribute combinations. Moreover, the absence of an explicit learning sample implies that the generated data are not guaranteed to be *representative at the territorial or demographic level*, which is particularly problematic in domains such as urban microsimulation, where accurate reproduction of spatial and population structures is essential. Finally, several studies underline intrinsic structural limitations of these models: due to their autoregressive architecture, LLMs struggle to enforce functional dependencies or permutation-invariant relationships among attributes, and they do not inherently incorporate marginal or aggregate data constraints (Xu *et al.*, 2025; Ilaty *et al.*, 2025).

2.3 Methods for Activity Plan Synthesis

The synthesis of activity plans is an important step in modelling individual behaviour within transport and mobility models. It aims to generate, for each individual in a synthetic population, a sequence of daily activities (work, study, leisure, shopping, etc.) and the associated journeys, in line with their socio-demographic, economic and spatial characteristics. Various approaches have been proposed in the literature, ranging from rule-based models to statistical or machine learning methods.

2.3.1 Statistical Matching

The most commonly used approach remains *Statistical Matching*. Originally developed by D'orazio *et al.* (2006) for handling missing data in survey sampling, this method relies on the principle of matching two distinct datasets that share a common set of variables. The idea is to identify, for each individual in the first sample, one or several similar individuals in a second sample, based on the proximity of observed attributes. The variables missing from the first dataset are then imputed from the matched individuals in the second dataset according to a statistical similarity rule (e.g., Euclidean distance or conditional probability).

In the context of activity plan synthesis, this approach has been adapted to assign each individual in a synthetic population an activity plan drawn from an observed database. The matching is performed conditionally on the individual's socio-demographic and economic characteristics such as age, gender, employment status, or household composition. Thus, each synthetic individual is associated with an observed activity plan exhibiting comparable profiles (Arentze *et al.*, 2007; Namazi-Rad *et al.*, 2017; Hörl et Balac, 2021b).

However, this method has several limitations. First, it depends on the availability of a sufficiently rich and representative sample of observed activities, which is not always guaranteed. Second, the matching principle often leads to a partial reproduction of behavioral diversity: synthetic individuals are assigned pre-existing plans rather than genuinely new behaviors. Finally, *Statistical Matching* does not allow for the explicit modeling of

intra-household dependencies or dynamic adjustments related to the temporal structure of activities.

2.3.2 Machine Learning (ML)

Beyond *statistical matching*, two major families of probabilistic methods have been applied to activity plan synthesis: *Markov chains* (and their hidden variants) and *Bayesian networks*. Each provides mechanisms to model temporal or causal dependencies, yet both exhibit certain limitations that merit discussion.

2.3.2.1 Markov Chains

The Markov chain approach is based on the idea that an individual's activity state at a given moment depends (at least) on the preceding state according to transition probabilities. This method models sequences of activities as stochastic state transitions (Saadi *et al.*, 2016a; Liu *et al.*, 2015). Liu *et al.* (2015) introduced an advanced version—the *profile Hidden Markov Model* (pHMM)—designed to better capture both frequent and rare activity sequences observed in survey data. The pHMM approach enhances the ability to represent diverse chains of activities without resorting to excessive cloning of observed sequences.

The Markov/HMM framework has the advantage of explicitly modeling temporal dependency (each state at time t depends on the state at $t-1$) and integrating transition probabilities. However, it becomes limited when the number of states or activity modalities is large (leading to combinatorial explosion) or when activity sequences are long.

2.3.2.2 Bayesian Networks for Activity Chain Generation

Bayesian Networks (BNs) provide an alternative paradigm for modeling activity plans by explicitly representing conditional dependencies between individual attributes and activity characteristics. The underlying principle is as follows: socio-demographic and economic variables are defined as *root nodes* that influence the attributes of successive activities. The structure of the network and the conditional probability tables are learned from a survey sample under structural constraints—for example, enforcing that activity nodes influence only subsequent activities in temporal order. The generation of an activity plan for a synthetic individual then consists of sequentially sampling activity attributes conditional on the individual's characteristics (Joubert et De Waal, 2020; Sallard et Balać, 2023).

This approach presents several advantages. It enables the generation of new, previously unobserved sequences through the use of conditional distributions. It can also capture non-linear and multivariate dependencies among attributes while offering causal interpretability, as it becomes possible to analyze the probabilistic influence of individual characteristics on activity choices.

However, Bayesian networks also have certain limitations. The network structure must often be constrained or directed to avoid combinatorial explosion. Moreover, the quality of the generated sequences strongly depends on the size and richness of the training sample. Finally, Bayesian networks may produce unrealistic activity chains if conditional relationships are poorly estimated or inadequately calibrated.

2.3.3 Deep Generative Models

The recent work by Shone et Hillel (2025) is one of the first to use a deep generative model for synthesising activity schedules. Building on this, it applies a Variational Autoencoder (VAE) to generate individual activity schedules. Their approach relies on encoding attributes such as travel mode, duration, and distance into vector representations using word embedding techniques, which are then fed into the VAE. This method stands out for its ease of implementation, rapid generation, and ability to produce realistic, diverse, and usable samples for both activity-data synthesis and anonymization. However, a key limitation highlighted by the authors is the model's inability to generate activity schedules conditional on other information, such as socio-economic or demographic characteristics. This shortcoming restricts its applicability within behavioral modeling frameworks. These findings underscore the importance of conditional approaches—such as Conditional VAEs (CVAEs) or hierarchical architectures—which are specifically designed to introduce explicit dependencies between demographic structures, contextual constraints, and individual behaviors.

2.4 Cross-comparison of approaches for population synthesis

Table 2.1 provides an overview of the main families of synthetic population generation methods, highlighting their strengths as well as their main limitations.

The different families of methods (SR, CO, ML, and deep generative models) provide contrasting answers to the challenges of population synthesis. They differ primarily in the type of generative mechanism they rely on, as well as in the trade-offs they establish between *fidelity to the marginals of micro-sample used*, *diversity of generated individuals*, *computational cost*, and ability to *incorporate external marginal data as constraints*.

Classical *Sample Reweighting* approaches (IPF, IPU) are characterized by their simplicity, speed, and ability to reproduce marginals exactly. However, they become fragile as dimensionality increases, particularly in the presence of zero cells, and they adapt poorly to mixed data.

Combinatorial Optimization (CO) methods offer increased flexibility, especially for in-

Table 2.1: Comparative summary of the main families of methods for population synthesis

Method family	Examples	Main strengths	Limitations
Synthetic Reconstruction (SR)	IPF, IPU	• Simple and fast • Exact fidelity to marginal distributions	• Fragile in high-dimensional spaces • Sensitive to zero cells • Cannot generate new agents
Combinatorial Optimization (CO)	Simulated annealing, genetic algorithms	• Flexible for complex constraints • Approximates marginal distributions	• High computational cost • No guarantee of optimality • Based on sample replication
Machine Learning (ML)	MCMC, HMM, Bayesian networks	• Generation of unseen agents • Captures complex dependencies • Allows causal integration	• Requires large training samples • Sensitive to dimensionality issues • Trade-off between fidelity and feasibility
Deep Generative Models	VAE, GAN, Diffusion	• Generates diverse agents • Strong expressivity in high-dimensional data • Allows differentiable constraint handling	• Costly calibration process • Sensitive to hyperparameter tuning • Difficult handling of structural zeros
Large Language Models (LLM)	GPT, LLaMA, etc.	• Fast *zero-shot* generation • Control via *prompt engineering* • No dedicated training required	• Strong dependence on prompt formulation • Risk of hallucinations • Requires validation and post-adjustment

corporating logical or hierarchical constraints. Based on heuristic algorithms, they enable a satisfactory approximation of marginals, but at the expense of rapidly growing computational cost and without guarantees of optimality. Moreover, they rely on replicating observed units and do not generate new agents.

Machine learning approaches, such as MCMC, Hidden Markov Models, or Bayesian networks, overcome this limitation by enabling sampling-zeros to be created. They capture rich structural dependencies and facilitate the integration of causal relations. Nevertheless, their robustness depends heavily on the size of the available sample and the ability to manage high-dimensional complexity. Their expressive power remains constrained by trade-offs between local fidelity, parsimony of structures, and computational feasibility.

Deep generative models (VAEs, GANs and related variants) represent a significant development in the processing of mixed and high-dimensional data. By learning latent representations, they reconcile diversity and fidelity while producing sampling-zeros. They also open the way to the integration of external constraints or post-adjustment using classical methods. However, their calibration remains demanding due to high training costs, and difficulties in handling structural zeros.

Diffusion models build on this dynamic and often surpass VAEs and GANs in terms of fidelity and diversity. Their adoption in population synthesis is still in its infancy, and more work is needed to effectively incorporate marginal and hierarchical data, albeit at a high computational cost.

Finally, approaches based on Large Language Models (LLMs) further enrich the methodological landscape by offering a flexible and rapidly deployable framework, notably through *prompt engineering*. Their main originality lies in their capacity to generate entirely new agents *without requiring a dedicated training sample*, in contrast to most deep generative models that rely on microdata. However, the reliability of these models remains highly dependent on the quality and precision of textual instructions, and their use necessitates validation and calibration mechanisms to guarantee the statistical and structural consistency of the generated populations. Furthermore, since these approaches operate without explicit reference to territorial or demographic sampling frames, the resulting synthetic populations are generally *not representative of specific geographical areas*—a limitation that can be critical in applications such as urban microsimulation, where spatial representativeness is essential for policy evaluation and scenario analysis.

2.5 Historical Perspectives

The evolution of population synthesis methods can be traced through three major periods, each reflecting distinct scientific and technological dynamics. The first, referred to as

the *Foundation Era* (1940–1995), focused on the mathematical foundations of population synthesis and the emergence of the first proportional adjustment methods. The second, the *Development Era* (1995–2012), marked a phase of algorithmic refinement and methodological diversification, characterized by the consolidation of practical applications and the growing use of combinatorial optimization techniques. Finally, the *AI/ML Revolution* (2012–2025) ushered in the widespread adoption of machine learning and, more recently, deep learning approaches for the generation of synthetic data.

Figure 2.2 illustrates this historical trajectory. The horizontal axis represents the period from 1940 to 2025, while the vertical axis distinguishes the main methodological families. Each point corresponds to a publication, where the color indicates the methodological category and the point size reflects the number of citations.

Several key milestones structure this evolution. The foundational work of Deming et Stephan (1940) introduced the *Iterative Proportional Fitting* (IPF) method, which remains a cornerstone for marginal adjustment procedures. In the early 1990s, Deville *et al.* (1993) formalized calibration methods, or *generalized raking*, thereby providing a robust theoretical framework for integrating auxiliary information into survey samples.

The early 2000s marked the emergence of combinatorial optimization, initiated by Voas et Williamson (2000), who laid the groundwork for this approach to synthetic population generation. During the following decade, the contributions of Müller et Axhausen (2010, 2012) expanded the field by explicitly introducing hierarchical household–individual modeling, enabling a more accurate representation of inter-level dependencies.

Between 2013 and 2016, new advances based on Markov chains and stochastic simulation techniques (MCMC, HMM) strengthened the ability of models to represent population heterogeneity while reducing the cloning of individuals (Farooq *et al.*, 2013; Saadi *et al.*, 2016b). At the same time, Bayesian networks emerged as a promising tool for formalizing complex dependencies, as demonstrated by the work of Sun et Erath (2015).

Finally, from 2018 onward, a new phase began with the introduction of deep generative models—such as *Variational Autoencoders* (VAEs), *Generative Adversarial Networks* (GANs), and their variants (e.g., CTGAN)—marking the entry of population synthesis into the domain of *deep generative modeling*. The work of Borysov *et al.* (2018) exemplifies this transition, highlighting the ability of neural models to generate synthetic populations that are both diverse and statistically coherent.

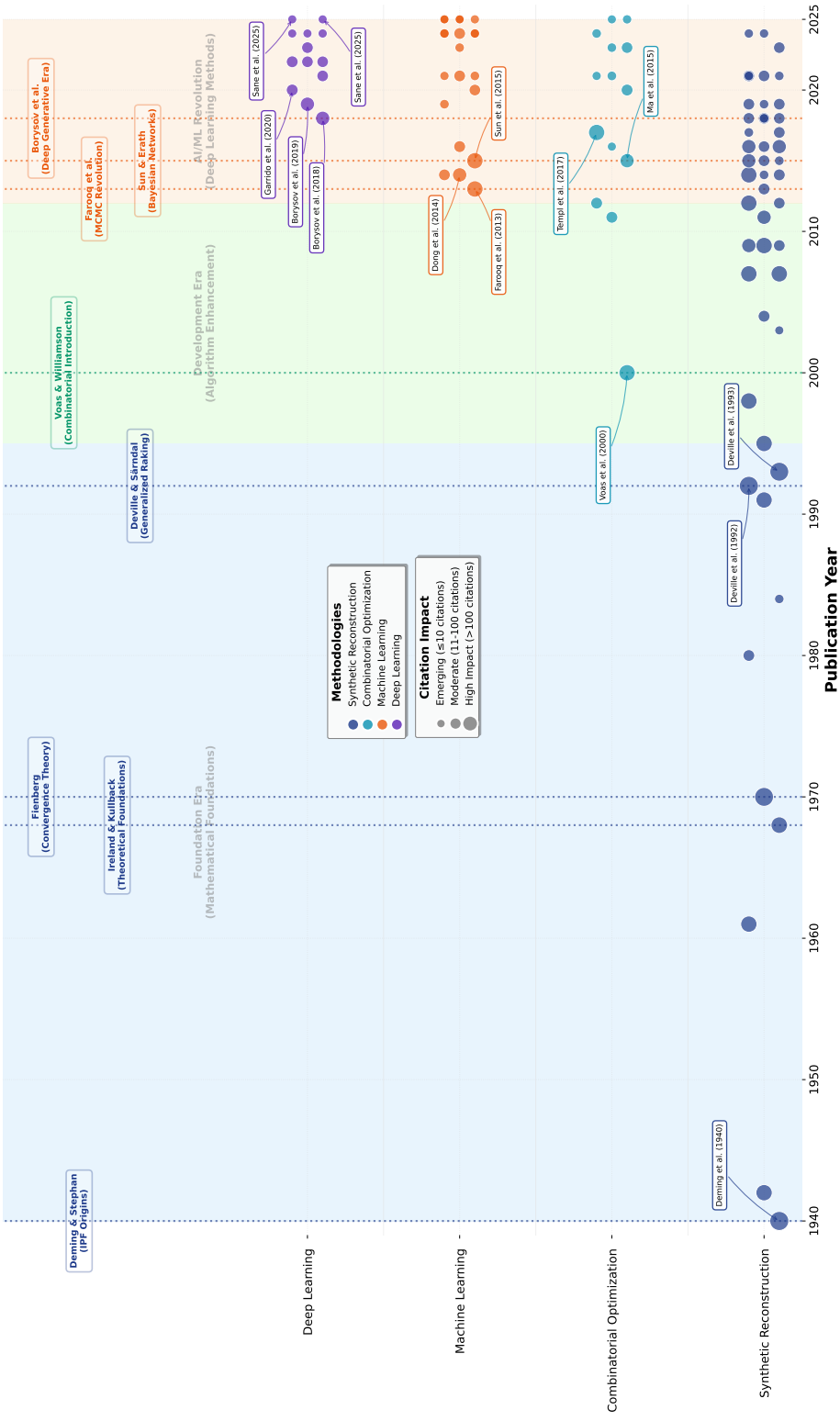


Figure 2.2: Historical evolution of population synthesis methods.

2.6 Conclusion and Positioning

This review highlights the wealth of existing approaches for population synthesis. Each family of methods has specific advantages, but also limitations that restrict their applicability in isolation. Among these methods, the Variational Autoencoder (VAE) stands out as particularly promising. Thanks to its probabilistic nature and its representation in latent space, it offers both flexibility and stability, while allowing mixed data and large-scale data to be processed at a relatively low computational cost.

The following chapters explore this perspective in greater depth by examining several complementary approaches. An educational presentation of the VAE, accompanied by an evaluation of its performance in generating synthetic data is first presented. How to adapt this approach to hierarchical individual-household data, overcoming certain limitations identified in the literature is the question of the second step. A third step consists of proposing a hybrid method that combines the advantages of VAE with those of Iterative Proportional Fitting (IPF), in order to reconcile generative flexibility with strict adherence to margins. Finally, the discussion is opened to large language models (LLMs), whose ability to process sequences makes them natural candidates for the synthesis of activity plans.

Chapter 3

A comprehensive investigation of variational auto encoders for generating a single-layered synthetic population

Contents

Abstract **36**

3.1 Introduction **37**

3.2 Variational Autoencoder Models **38**

3.2.1 Linear Autoencoder analyzed as a Principal Component Analysis 38

3.2.2 Nonlinear autoencoder 40

3.2.3 Stacked autoencoder 41

3.2.4 Variational Autoencoder (VAE) 41

3.2.5 Model training 44

3.3 Case study data **46**

3.4 Unlocking the optimal VAE architectures and hyperparameters 47

3.4.1 Data engineering 47

3.4.2 Technologies introduced 49

3.4.3 Model-building 49

3.5 Assessment of the generated synthetic population **51**

3.5.1 Performance on the individual attributes 52

3.5.2 Performance on the household attributes 55

3.6 Influence of the model building dataset size **56**

3.6.1 Influence on the accuracy of the generated population 57

3.6.2 Influence on the zero-cell problem 58

3.7 Conclusion **62**

Abstract

Abstract

This chapter provides an in-depth exploration of the use of Variational Autoencoders (VAEs) for generating synthetic populations from mixed data, combining both categorical and numerical attributes. It addresses a gap in the literature by offering not only a theoretical overview but also a practical implementation framework, supported by a reproducible Python notebook. The chapter examines key aspects of VAE modeling, including hyperparameter tuning, sample size effects, and performance evaluation metrics, with a specific focus on the treatment of zero cells—agents absent from the training dataset.

Empirical results demonstrate that VAEs are capable of producing large-scale, statistically consistent synthetic populations from relatively small samples. The model achieved strong accuracy, with marginal attribute errors below three percentage points and low SRMSE values across both simple and cross-distributions. These findings highlight the robustness of VAEs in handling high-dimensional data and their capacity to generate unseen individuals (sampling zeros), although a small share of structural zeros remains and requires post-processing.

The analysis further shows that VAEs trained on categorical data exhibit greater stability than those handling numerical variables, suggesting that discretizing continuous features can enhance model performance. Beyond their predictive capacity, VAEs also ensure data confidentiality and can be integrated into broader simulation pipelines such as EQASIM for urban modeling applications.

Overall, the chapter establishes a comprehensive and reproducible methodological foundation for synthetic population generation using VAEs. It also outlines several future research directions: integrating marginal data to enhance representativeness, extending the framework to multi-level modeling (e.g., individuals within households), and incorporating sequential dimensions such as daily activity plans to achieve richer and more behaviorally coherent synthetic populations.

Keywords : *Synthetic population, machine learning, deep generative model, variational autoencoders, sampling zeros, structural zeros*

3.1 Introduction

Building on the comparative review of synthetic population generation methods presented in the previous chapter, this chapter turns to a more focused exploration of one particular approach: the Variational Autoencoder (VAE). It is a promising new ML generation method that has emerged in recent years. From a review of the literature, it appears that Variational Autoencoders (VAEs) have strong potential for the generation of synthetic populations. However, most recent works do not detail how this method is to be implemented. Researchers and practitioners seeking to test and implement this technology often find the literature lacking in practical information. The present chapter aims to fill this void by providing a practical guide to generating a synthetic population using a VAE and then evaluating the resulting performance. This chapter is accompanied by a Python notebook, thus allowing interested readers to reproduce the results. Special care has been taken with the code to ensure that practitioners and researchers can adapt it to their specific problems.

This chapter conducts a thorough exploration of the implementation of a VAE based method for single-layered synthetic population generation (i.e. either individual-centered or household-centered). Notably the various parameters of the VAE architecture are explored. In addition, this chapter examines the impact of the size of the training sample on model performance, while focusing on an evaluation of zero-cells (i.e agents not present in the training dataset), including zero-sampling (agents not present in the training dataset yet present in the population), and structural zeros (agents not present in the training dataset and not present in the population) .

In sum, the contributions of this chapter are sixfold:

1. a pedagogical presentation of the VAE techniques;
2. Python notebook, allowing interested readers to adapt it to their specific problems;
3. a complete analysis of the hyperparameters involved, including the weights modulating the different parts of the cost function;
4. an assessment of the strengths and limitations of this method through an actual case study;
5. a spotlight on the effect of the sampling dataset size on VAE performance; and
6. an in-depth analysis of both the generation of sampling sets and structural zeros.

The remainder of this chapter is organized as follows. First, a detailed presentation of the Variational Autoencoder (VAE) is provided to give the reader a comprehensive understanding of its functioning. Next, a case study illustrates the VAE modeling process in

practice. Finally, the chapter concludes with the presentation of results, focusing on model performance across different attributes, the effect of training sample size on this performance, and an assessment of zero cells in the generated data.

3.2 Variational Autoencoder Models

Autoencoders are a type of neural network designed to learn efficient data representations, usually serving purposes like dimensionality reduction or feature learning. To better understand Variational Autoencoders (VAEs), let's begin with a brief presentation of linear autoencoders, which can be assimilated to a Principal Component Analysis (PCA). The discussion will turn next to stacked autoencoders, which refer to a sequence of autoencoders, before exploring the details of VAEs themselves.

Formally, let $X = (X_1, ..., X_n) \in \mathbb{R}^n$ be a set of random variables defining the characteristics of individuals. In the database, each sample represents an individual of the population with n features related to his/her particular situation (age, gender, employment, education, driver's license status, etc.) or his/her household (house type, number of persons, number of vehicles, etc.).

3.2.1 Linear Autoencoder analyzed as a Principal Component Analysis

Geometrically speaking, the set of examples from the database can be seen as a cloud point of $\mathbb{R}^n$, in the form of a hyperellipsoid, whose principal axes are the eigenvectors of its covariance matrix. Principal Component Analysis (PCA) performs a reduction in the dimensionality of the inputs, by focusing attention on the directions along which the cloud point is most highly diffused, i.e. the axes on which data variance is maximum. During this reduction operation from input dimension n to dimension m of the compressed data representation, m is the dimension of the subspace best explaining the "signal", with the remaining $n-m$ dimensions explaining the "noise".

PCA can be performed by an auto-associator neural network, called a linear autoencoder, with both a hidden layer and linear activation functions. An activation function defines the transformation applied to the weighted sum of inputs of a neuron before passing it to the next layer, thereby introducing non-linearity—or, in the linear case, preserving linear relationships—between layers. The number of inputs and outputs corresponds to the dimension n of the training examples, while the number of hidden neurons equals m. The autoencoder uses a latent variable model, whereby it relates the observable variables X to a multivariate latent variable Z. The latent variables are unobserved yet still capture underlying patterns or representations in the data. The shape of the network is illustrated

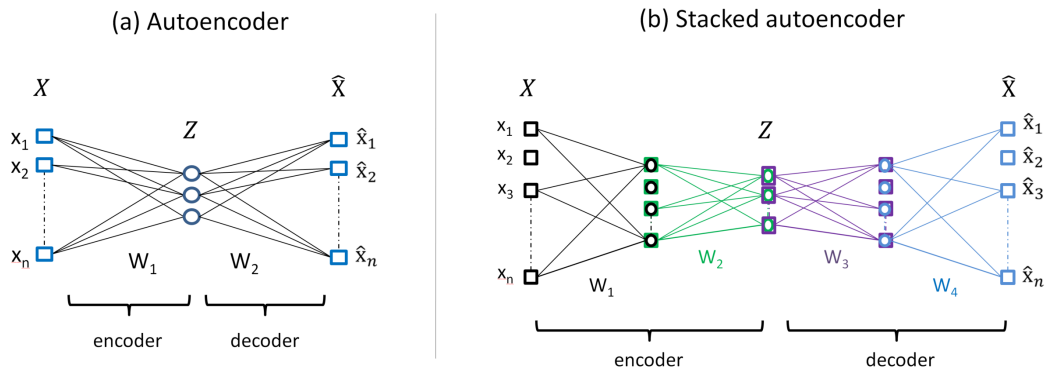


Figure 3.1: The autoencoder is an unsupervised model trained to copy its inputs. It compresses the input vector X with dimensions n into the latent representation Z with fewer dimensions and then reconstructs it back into the data space.

in Figure 3.1(a), in which the hidden layer is small compared to the number of inputs and outputs.

Learning is accomplished by backpropagating the gradient of the reconstruction error; also, it is unsupervised because, for each example, the desired output is identical to the input. When the activation functions are linear, the network learns a linear model of the examples, which is optimal from the standpoint of least squares. The output of the m hidden neurons can then, under certain conditions, correspond to the principal components of the examples adopted as autoencoder input, as mentioned in Kramer (1991). This set-up represents a compact coding of the input vector: the network thus performs a dimensional reduction.

The weights connecting the hidden layer to the network input are called encoder weights; the weights connecting the hidden layer to the output layer are often referred to as generator weights since they reconstruct the example shown. Both encoder and decoder weights are called synaptic weights by analogy with biological neural networks.

The mapping function $Q_\theta(X) = Z$, with parameters θ representing the encoder weights W_1 in Figure 3.1(a) is often referred to as an "encoder" since it maps the inputs X to the feature space Z.

The mapping function $P_\phi(Z) = \hat{X}$ is often referred to as a "decoder" given that it decodes the latent variable representation into the observable space of X. ϕ represents the decoder weights (W_2 in Fig. 3.1(a)).

3.2.2 Nonlinear autoencoder

When the activation functions of the hidden layer are nonlinear, the network operates a nonlinear PCA (Kramer (1991)), with the output Z of the hidden layer then being the mapping (or the nonlinear projection) of the input X into a subspace of dimension m:

$$z_j = Q_{\theta,j}(\mathbf{X}) = s\Big(\sum_{k=1}^{n} w_k^j x_k + b^j\Big), \quad j = \{1, 2, \ldots, m\} \tag{3.1}$$

with X being an input sample vector, Q_θ a nonlinear function, the activation function s usually being a sigmoid, a tanh or a Rectified Linear Unit ($\text{ReLU}(x) = \max(0, x)$) function, x_k the kth attribute of observation X, w_k^j the weight linking input k to the hidden neuron j, and b^j the bias of this neuron.

The reconstruction $\hat{X}$ of sample X is computed from the feature representation to the linear output layer of the network:

$$\hat{x}_k = P_{\phi,k}(Z) = \sum_{j=1}^{m} w_j^k z_j + b^k, \quad k = \{1, 2, \ldots, n\} \tag{3.2}$$

where w_j^k denotes the generator weight linking the output neuron j to the hidden neuron k, and b^k its bias.

In PCA, the choice of the number m of principal components is made on the spectrum of the covariance matrix: by sorting the eigenvalues in decreasing order, a significant difference between successive values indicates a natural number for the dimension of the eigen-subspace. By analogy, when increasing the number of hidden neurons, the learning cost function, in general a quadratic error, no longer decreases significantly from a given number of neurons indicating the most consistent dimension m.

When training an autoencoder, the minimized cost function is the sum, on the training database with N samples, of the square of the errors between the input $\mathbf{X}^k$ (corresponding to the desired output) and the computed output $\hat{\mathbf{X}}^k$ for observation number i ($i = \{1, 2, \ldots, N\}$):

$$L(\theta, \phi) = \sum_{i=1}^{N} \sum_{k=1}^{n} (\hat{x}_i^k - x_i^k)^2 \tag{3.3}$$

where θ and ϕ are the synaptic weights of encoder and decoder, respectively.

For a nonlinear model, the cost function L is not quadratic with respect to the parameters; therefore, the least squares method does not apply, thus making it necessary to resort

to an iterative method to minimize L : the gradient descent with an adaptive step, stopped by cross-validation. The first step consists of estimating the gradient of the cost function with respect to the parameters, by means of backpropagation. The second step entails modifying the parameters according to the estimated gradient, which strictly speaking gives rise to the gradient descent.

3.2.3 Stacked autoencoder

Autoencoders can contain multiple hidden layers, in which case they are often called a stacked autoencoder or deep autoencoder. A stacked autoencoder is a deep extension of the architecture of the presented auto-associator networks (linear autoencoder) : it consists of several layers of autoencoders, wherein the output of each hidden layer is connected to the input of the successive hidden layer, as illustrated in Figure 3.1(b) with three hidden layers.

The lth layer of the autoencoder computes its output as $\mathbf{y_l} = s(W_l \mathbf{y_{l-1}} + \mathbf{b_l})$, where W_l and b_l are the weight matrix and bias vector associated with the layer, and s is a nonlinear activation function.

Furthermore, depending on the nature of the data, whether numerical or categorical, the use of an appropriate activation function proves to be necessary. For example, for a categorical attribute, a function such as "softmax" may be employed, with outputs ranging between 0 and 1, for the purpose of representing the probabilities of the different modalities of this attribute (Sharma *et al.*, 2020).

Various ways exist to train such deep neural networks. However, their general principle remains the same as that for the simple autoencoder, which has already been described above. In accordance with Borysov *et al.* (2019a), this category will simply be called autoencoders, and this study will be limited to the case of standard training by means of error backpropagation.

3.2.4 Variational Autoencoder (VAE)

This section will set forth the main ideas behind the VAE model. For a more in-depth understanding, the reader is referred to the original article on image recognition by Kingma et Welling (2013) which introduced a type of generative model that combines techniques from neural networks with probabilistic graphical models.

VAE is a powerful unsupervised generative model that learns to represent complex data by mapping latent variables to observable variables using a cleverly chosen nonlinear transformation. It allows generating new data samples by sampling the latent space and mapping it onto the observed data space via a decoder function. The encoder and decoder have parameters that are learned during the model building process in order to optimize the

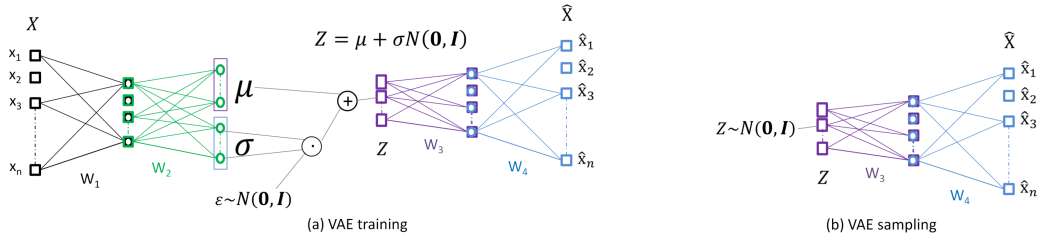


Figure 3.2: Variational Autoencoder architecture : (a) During training the random variable ε is injected into the latent space Z as an external input; (b) After training, input a Z following a normal random law in order to generate a new sample.

generation of accurate samples.

The main goal of the VAE is to learn the joint probability distribution $P(X)$ of the observable variables X representing the attributes of the population. To approximate $P(X)$, the VAE identifies a cleverly chosen nonlinear transformation of a known distribution $P(Z)$. One common choice for $P(Z)$ is a multivariate Gaussian distribution. The VAE maps samples from this distribution using a nonlinear transformation to generate data that approximate the original data distribution $P(X)$.

The VAE generates new data samples by sampling the latent variable Z from the chosen distribution (e.g. multivariate Gaussian) and then mapping this variable onto the observed space of X using the nonlinear transformation. Unlike traditional autoencoders, which learn a fixed and deterministic representation of the input, the VAE introduces a probabilistic formulation of the latent space. Each input is encoded not as a single point but as a distribution, typically Gaussian, characterized by its mean and variance. This stochastic encoding enables the model to capture uncertainty and variability in the data, allowing the generation of diverse and realistic samples. By incorporating this probabilistic structure, VAEs extend the capabilities of classical autoencoders beyond mere reconstruction toward genuine data generation.

Figure 3.2(a) illustrates the architecture of a VAE; note that variables μ, σ and ε all have the same dimension m as the feature space Z.

During training, the encoder takes a data sample X as input and maps it into two vectors, μ and σ, both of dimension m. These vectors are then used to parameterize the multivariate normal distribution for generating the latent variable $Z = \mu + \sigma \cdot \varepsilon$, with the $\cdot$ symbol denoting element-wise multiplication and where ε is a random input of size m according to a multivariate normal distribution: $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The insertion of ε to produce the feature space is known as the re-parameterization trick, since it enables backpropagating the gradient without involving a stochastic variable during the update.

Z is then transformed using the decoder, and the similarity between the original data X and its reconstructed counterpart $\hat{X}$ is measured. The optimization of the following loss function (Section 3.4) takes place using the backpropagation algorithm with respect to the encoder and decoder parameters:

$$\begin{aligned} L(\phi,\theta) = & \sum_{i=1}^{N} \underbrace{c_1 \times \sum_{k\in\{num\}} (\hat{x}_i^k - x_i^k)^2}_{\text{Numerical cost}} \underbrace{- c_2 \times \sum_{k\in\{cat\}} \sum_{d=1}^{D_i} x_i^{kd} \log \hat{x}_i^{kd}}_{\text{Categorical cost}} \\ & \underbrace{+ \beta \times \mathbb{KL}(\mathcal{N}(\mu^l, \sigma^l) \| \mathcal{N}(\mathbf{0}, \mathbf{I}))}_{\text{KL cost}} \end{aligned} \tag{3.4}$$

The reconstruction error measures the distance between the input X and the generated samples $\hat{X} = P_\phi(Q_\theta(X))$. Quantifying this error is typically done through the quadratic error for numerical data and the categorical cross-entropy for categorical attributes, given the often mixed nature of the data (Borysov *et al.*, 2019b; Aemmer et MacKenzie, 2022).

The cost function differs from the case of the autoencoder by virtue of including a regularization term, as represented by the Kullback-Leibler divergence between the m-dimensional latent variable distribution, as parameterized by the encoder output (mean μ and standard deviation σ), and the standard Gaussian prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$. This term ensures that the data distribution in the latent space conforms to the desired form, thus enabling efficient sampling from the latent space. This feature stands in contrast with deterministic Autoencoder models, which lack a probabilistic interpretation in their latent space, in turn making sampling less meaningful.

Hyperparameters $(c_1, c_2, \beta) \in {\mathbb{R}_+}^3$ represent scalar weighting coefficients between the reconstruction losses of categorical and numerical inputs and the regularization strength (Higgins *et al.*, 2017). In previous research on tabular data including both numerical and categorical attributes, such as that conducted by Borysov *et al.* (2019a), the coefficients (c_1, c_2) were set to $(\frac{1}{2}, 1)$ and the coefficient β variable to values lying between 0 and 1. As part of this study, different values of (c_1, c_2) will be explored.

Once the model has been trained, it becomes possible to generate new samples by sampling the latent variable from the prior distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and then transforming this variable through the decoder to the data space, as illustrated in Figure 3.2. By exploring random regions of the latent space following a multivariate normal law, it also becomes possible to create new agents that differ from those seen in the original data used for training, yet with these created agents following their distribution $P(X)$. The sampled points are passed through the decoder, which maps them back onto the original data space, in

generating synthetic data points that approximate the characteristics and patterns of the original population data. By repeating this sampling and decoding process, the VAE can generate a large number of synthetic data points, effectively creating a synthetic population that shares statistical similarities with the real-world population.

Note that the quality and accuracy of the synthetic population generated by the VAE depend on both the quality and diversity of the original training data. The VAE is capable of capturing complex data distributions and generating realistic synthetic data, thus making it a powerful tool in generating representative synthetic datasets for various applications. However, it is essential to carefully validate and assess the quality of the synthetic population before using it for any real-world applications.

3.2.5 Model training

The training of the model aims to optimize its parameters so that it can accurately reproduce and generalize the relationships observed in the data. In practice, this process involves the minimization of a cost function that quantifies the discrepancy between the model's predictions and the observed data. Because evaluating this function over the entire dataset of size N can be computationally expensive, it is generally estimated over small subsets of data, called *batches*. This stochastic approximation, known as *mini-batch gradient descent*, updates the model parameters (weights and biases) iteratively, one batch at a time, until convergence is reached.

A complete pass through the entire training dataset constitutes an *epoch*. During training, multiple epochs are performed so that the model can progressively adjust its internal parameters and improve its predictive capacity. The number of epochs, along with the batch size, network architecture, and type of activation function, are part of the model's *hyperparameters*, which must be carefully tuned to ensure both convergence and generalization.

In a stochastic gradient estimation, the cost function is evaluated by summing on a *batch* of training samples, instead of the entire N samples. The cost function is minimized with respect to its parameters (weights and biases), one batch at a time, in an iterative manner until reaching convergence.

Model calibration depends on :

- *parameters*: weights and biases
- *hyperparameters*: type of nonlinear function, batch size, epoch number, number of neurons, etc.

These parameters and hyperparameters are calibrated using the *model building dataset*, which is a part of the dataset. This model building dataset is separated into two sets as

follows :

1. The *learning dataset*: weights and biases are learned (or identified or trained)[1] with this dataset, which itself is divided into two distinct sets :

   (a) The *training dataset*: weights and biases are identified by minimizing the cost function using this training dataset, which has been divided into smaller batches. The model's parameters (weights and biases) are updated incrementally after processing each batch. This process then continues until the model has seen and learned from every batch in the training dataset, through completing one epoch. The Epoch number constitutes one of the hyperparameters.

   (b) The *cross-validation dataset* is used to stop the training process, referred to as early stopping. It serves to avoid overfitting, in reference to a situation whereby the VAE has learned to model the training data too closely or too precisely, to such an extent that it fails to generalize effectively to new unseen data. Concretely, the training process is halted when the reconstruction error on the cross-validation dataset stops decreasing or begins to increase, indicating that further training would degrade the model's generalization ability.

   Since the initial values of weights and biases are random, this minimization process is iterated several times starting from different initial values. The number of iterations is also one of the hyperparameters.

2. The *validation dataset* or *test dataset*: The hyperparameters are calibrated using this dataset. Each hyperparameter combination specifies an autoencoder whose weights and biases are identified during the learning process. The generalization capacity of a trained autoencoder is evaluated on this validation dataset according different metrics such as the standardised root mean square error (RMSE). The hyperparameter calibration typically follows an optimization or search strategy—such as grid search, random search, or Bayesian optimization—aimed at finding the combination that minimizes the validation error and maximizes the model's ability to generalize to unseen data. The best assessment yields the optimal hyperparameter configuration.

   Combining Cross-Validation and Validation Datasets In some cases, the same dataset can be used for both cross-validation and validation. This combined dataset then serves the dual purpose of:

   - Stopping Training: Helping stop the training process at the right time to prevent overfitting.

[1] The Deep Learning community might have its own vocabulary but their concepts are very close to those of other scientific communities.

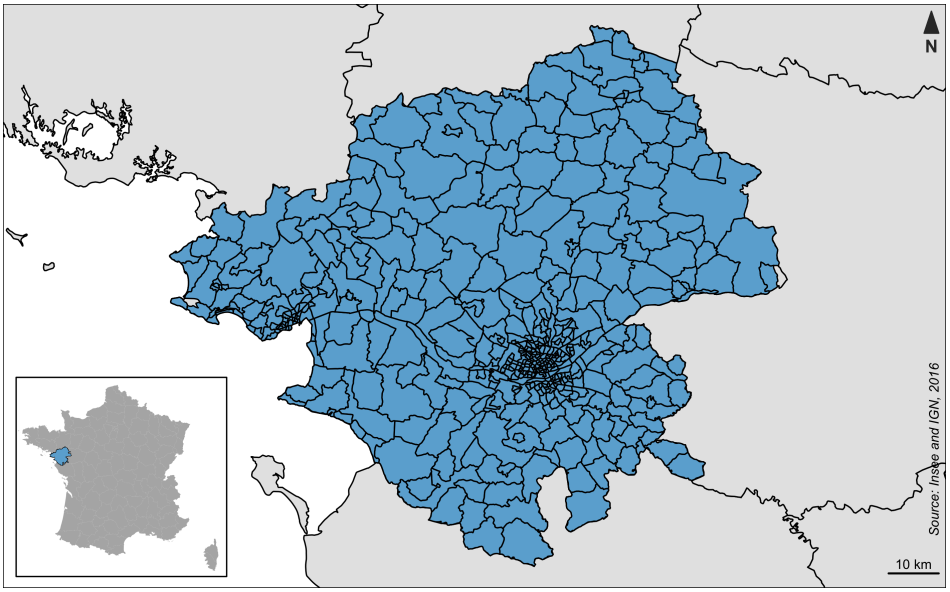


Figure 3.3: Loire-Atlantique Department/France

- Hyperparameter Tuning: Evaluating the predictive performance of different model configurations in order to select the best one.

3.3 Case study data

To generate a synthetic population, machine learning methods make use of a sample of individuals or households that accurately reflect the characteristics and relationships present in the target population. This sample can be compiled from various relevant data sources, such as surveys, census data and administrative records. The attributes used to represent individuals or households in the synthetic population can be classified as either categorical or numerical. These attributes may encompass age, gender, income, household size, education level and other sociodemographic features. For this study, the data sample has been derived from a Household Travel Survey (HTS) conducted in Nantes, France (Département-Loire-Atlantique, 2015). It is important to note however that this sample is comparable to data samples utilized in other studies, including Borysov *et al.* (2019a) and Borysov *et al.* (2019b); Aemmer et MacKenzie (2022), applying the VAE methodology. Consequently, the model development and results remain valid for case studies conducted in other areas.

More specifically, the objectives of this HTS were to understand urban and suburban mobility, and moreover evaluate and develop transportation policies in these areas. A total of 230 municipalities were concerned, with 96% of them located in Loire-Atlantique, which is an administrative jurisdiction known as a "department". Approximately 20 800 people aged 5

and older, distributed across over 12 700 households, were interviewed to obtain information on 29 476 individuals. The main information collected included household characteristics, sociodemographic characteristics, socioeconomic characteristics, and travel characteristics of the previous day.

Household characteristics, sociodemographic and socioeconomic attributes in the construction of our models are presented in the Table 4.1. The 6 total household characteristics, comprise 3 categorical attributes and 3 numerical attributes. The individual attributes total 10, with 8 categorical and 2 numerical.

Since the purpose of this chapter is to demonstrate the interest and limitations of Variational Autoencoders (VAE) in the context of synthetic population generation, true population statistics are needed in order to assess VAE capacity. For purposes herein, the data being separated into two sets, i.e. :

1. the *model-building dataset* used to calibrate the parameters and hyperparameters as explained in Subsection 3.2.1 ;

2. the *final test dataset* used to analyze the interests and limitations of the VAE methodology in generating the synthetic population. It is assumed that this data shares the same statistical properties as the true population.

In all, 29 476 samples are available in this HTS: 80% (23 600 individuals) were used as the model-building dataset to set up the VAE, and the remaining 20% (5 876 individuals) used in the final test. The model-building dataset of 23 600 samples was divided into two parts: 70% of these samples served to train the VAE (16 500) and update the neural weights during backpropagation (i.e. these samples constitute the learning dataset), and the 30% (7 100) were used to validate estimation of the hyperparameters c_1, β and the best architecture (together constituting the validation dataset).

3.4 Unlocking the optimal VAE architectures and hyperparameters

3.4.1 Data engineering

Standardization of numerical features

Most neural network algorithms rely on gradient descent for model parameter optimization. To ensure that gradient descent proceeds smoothly toward potential minima, it is important to normalize the data. This standardization of attributes offers several advantages, namely allowing for faster convergence toward the minima and avoiding local optima.

Table 3.1: Overview of household and individual features

Groups	Attributes	Type[1]	Description
Household (HH)			
	house_type	cat(5)	House type
	house_occupation_type	cat(4)	House occupation type
	household_size	num	Number of persons in the HH
	number_of_vehicles	num	Number of vehicles in the HH
	number_of_bikes	num	Number of bicyles in the HH
	has_internet	cat(2)	Has internet access ?
Individuals			
	link_ref_person	cat(5)	Link with the HH reference person
	age	num	Age
	sex	cat(2)	Gender
	is_adolescent	cat(2)	Is adolescent ?
	school_level	cat(5)	School level
	employed	cat(2)	Is employed ?
	studies	cat(2)	Is a student ?
	socioprofessional_class	cat(9)	Socio-professional status
	has_license	cat(2)	Has a driver's license ?
	number_of_trips	num	Number of previous day's trips

[1] Indicates whether the attribute is numerical (num) or categorical (cat). When the attribute is categorical, the number in parentheses represents the number of possible values.

One-hot encoding for categorical features

Machine learning algorithms only take into account numerical values as input. However, several attributes in our dataset are categorical, hence the need to transform them to enable their effective use. One-hot encoding method are chosen because it allows for categorical values to be encoded in a simple manner, without altering the information contained in the initial data.

The one-hot encoding technique consists of generating a new Boolean attribute (0 or 1) for each category, as distinct from the categorical attribute (Potdar *et al.*, 2017). Each new binary attribute represents a specific category, assuming the value 1 if the observation belongs to that category and 0 otherwise. Keep in mind that, this method can significantly increase model complexity should the attributes have a large number of values. Hence, after applying one-hot-encoding, the number of columns in the data table increased from 16 to 45.

Missing values

Values are often missing in the data tables, due to multiple potential reasons: data corruption, error during recording, a lack of information, incomplete results, refusal to answer, etc. However, these shortfalls still need to be addressed since they can exert a negative impact by biasing the training result of the model (Markey *et al.*, 2006). Typically, when the number of missing data values is very small, one common approach calls for removing rows from the data table that contain these missing values. Yet, it is also possible to replace these missing data with new values using specific imputation techniques (Donders *et al.*, 2006; Alabadla *et al.*, 2022). With regard to the data contained in this work, missing values were not an issue because of their zero proportion across all attributes.

3.4.2 Technologies introduced

All tasks, from data preparation to model evaluation, were carried out using Python, primarily utilizing the libraries Pandas 1.5, Numpy 1.23, and Tensorflow 2.10. The graphs were created with the Plotnine 0.10 library. The source code and data are available on Gitlab[2] for readers wishing to implement this model. On average, training a model takes about 30 seconds. The computer used for this task was equipped with an Intel Xeon 11th generation processor, 128 GB of RAM, and an NVIDIA RTX A5000 graphics card.

3.4.3 Model-building

To obtain the best model, a set of values for each hyperparameter are defined. The values adopted for the hyperparameters are presented in Table 3.2.

We opted for latent spaces with dimensions of less than one-third of the total number of features ($n = 45$ and $m \leq \frac{n}{3} = 15$); these latent spaces thus have dimensions of $\{5; 10; 15\}$. The size of the hidden layers is generally decreasing. Each transition from one hidden layer to another entails a loss of information, hence a high number of hidden layers can lead to a greater loss of information, not to mention the computational complexity this entails. As a result, architectures with one, two or three hidden layers are defined. With regard to batch size, which represents the number of inputs considered for each synaptic weight update, we chose from among the values[3]$\{128; 256; 512; 1\,024\}$. The number of epochs was set at 100, i.e. reflecting the number of times the dataset passes through the model. Coefficient c_1 was used to regulate the importance of numerical attributes versus categorical attributes, while coefficient β regulated the importance of the Kullback-Leibler divergence cost versus the error.

A set of models was implemented by combining different values of these hyperparame-

[2] https://gitlab.univ-eiffel.fr/population-synthesis/vae-vector-emd
[3] Powers of 2 are recommended for improved computational performance

Table 3.2: Overview of the models hyper-parameters

Number of layers	Number nodes	Latent dim	Mini-Batch size	c_1	β	Node activations
1; 2; 3	60; 40; 30; 20	5; 10; 15	128; 256; 512; 1 024	0,4; 0,5; 0,7	0,01; 0,05; 0,005	ReLU; tanh

ters. In all, 1 512 models were created, with 10 iterations for each model. Indeed, the random initialization of the synaptic weights of a model can exert an influence on its learning and, consequently, on its performance. Thus, the performances calculated on a single model are subject to influence by this random initialization. By performing several iterations, it is possible to calculate an average performance for each model and thus avoid the bias due to the initialization of synaptic weights.

Furthermore, when the number of hyperparameter configurations is too large for numerical capacities, heuristic methods are available to explore just a portion of the configurations, at the risk of not finding the optimal one. These methods involve randomly selecting a certain number of configurations, choosing the best one, and then exploring the hyperparameter space around that configuration.

Various metrics have been employed to quantify the model performance. Each of these metrics provides a different perspective on model performance and a better understanding of model strengths and weaknesses in synthetic population generation tasks. This performance is evaluated by comparing the frequencies of the real cross-modalities, denoted π, with those of the cross-modalities of the generated data, denoted $\hat{\pi}$. Among these metrics, let's note :

- Standardized Root Mean Square Error (SRMSE)

  The SRMSE makes it possible to quantify the average error between the frequency of the cross-modalities generated by the model $\hat{\pi}$ and the frequency of the real cross-modalities π, while taking into account the standard deviation of the residuals. This metric is commonly employed when exponential importance has been assigned to errors based on their magnitude. With a lower SRMSE, the model is considered to be more accurate.

$$SRMSE = \frac{\sqrt{\frac{1}{N}\sum_k (\pi_k - \hat{\pi_k})^2}}{\frac{1}{N}\sum \pi_k} \tag{3.5}$$

- Coefficient of correlation ($Corr$)

This metric quantifies the level of correlation between $\hat{\pi}$ and π

$$Corr = \frac{cov(\pi, \hat{\pi})}{sd(\pi) \times sd(\hat{\pi})} \tag{3.6}$$

Table 3.3 lists the scores obtained for some of the architectures investigated. For each architecture, 10 VAEs were trained with a different weight initialization; the mean performance was recorded. For instance, for architecture $40 \times 10 \times 40$, ten VAE were trained with hyperparameters $c_1 = 0.4$ and $c_2 = 1.0$: their mean SRMSE equaled 3,551. Using this same architecture but with $c_1 = 0.5$ the mean SRMSE was 1,996, which is significantly better. The best architecture, according to the SRMSE criterion, is shown in bold in the table. More complex architectures, such as $60 \times 40 \times 20 \times 10 \times 20 \times 40 \times 60$, offer less interesting cases (with an SRMSE of around 2) because they stick closely to the learning dataset and cannot generalize. More frugal architectures, like $40 \times 10 \times 40$ (also with an SRMSE of around 2), are too simple to appropriately grasp the complexity of the probability density and therefore cannot generalize.

Figure 3.4 illustrates the architecture of the best trained VAE, i.e. the one with the lowest SRMSE value among all trained VAE; this architecture is $40 \times 30 \times 5 \times 30 \times 40$.

All the VAEs have 45 inputs: 5 of them are numerical attributes and represented by one input each; also, 11 attributes are categorical and represented by a total of 40 inputs (see Table 4.1 for the number of possible values for each attribute). In the decoder part of Figure 3.4, the five fully connected neurons with an ReLU activation function correspond to the five numerical attributes. The next five fully connected neurons correspond to the first categorical attribute, *house_type*, which has five possible values. The last two fully connected neurons correspond to the attribute *has_license*.

The neurons corresponding to categorical attributes are activated by means of a *softmax* function that ensures the consistency of the outputs, e.g., an agent either has or does not have a driver's license; a mixed attribute is not allowed (see Sharma *et al.* (2020) for a detailed explanation of activation functions).

3.5 Assessment of the generated synthetic population

Once the VAE architecture has been chosen, VAE performance on a test set needs to be described. The trained VAE is used to generate a synthetic population of 1.4 million agents, with the final test dataset being utilized to verify that their individual attributes and household attributes are consistent with those of the available microsample. This verification is characterized by marginals, relating to the level of each attribute independently, as well as joint distributions, constituting a multivariate analysis of a set of attributes taken together.

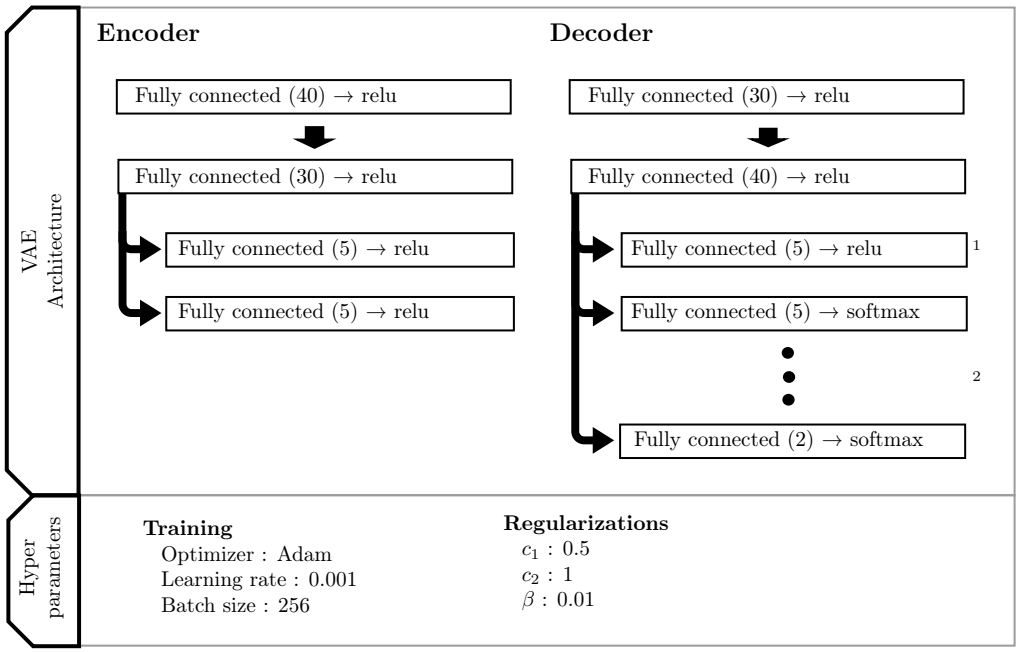
Table 3.3: Samples of investigated architectures and hyperparameters investigated, along with their performance on the validation set

Architecture	srmse	corr	c_1	c_2
40 x 10 x 40	3.551	0.588	0.4	1.0
40 x 10 x 40	1.996	0.694	0.5	1.0
40 x 10 x 40	2.115	0.687	0.7	1.0
40 x 20 x 10 x 20 x 40	2.374	0.608	0.7	1.0
40 x 20 x 15 x 20 x 40	3.809	0.572	0.4	1.0
40 x 20 x 15 x 20 x 40	2.272	0.626	0.7	1.0
40 x 20 x 5 x 20 x 40	3.281	0.650	0.4	1.0
40 x 20 x 5 x 20 x 40	2.010	0.675	0.5	1.0
40 x 20 x 5 x 20 x 40	2.271	0.631	0.7	1.0
40 x 30 x 10 x 30 x 40	4.202	0.553	0.4	1.0
40 x 30 x 10 x 30 x 40	2.022	0.686	0.5	1.0
40 x 30 x 10 x 30 x 40	2.168	0.653	0.7	1.0
40 x 30 x 5 x 30 x 40	3.835	0.563	0.4	1.0
40 x 30 x 5 x 30 x 40	**1.658**	**0.782**	**0.5**	**1.0**
40 x 30 x 5 x 30 x 40	2.515	0.522	0.7	1.0
40 x 30 x 20 x 5 x 20 x 30 x 40	3.980	0.527	0.4	1.0
40 x 30 x 20 x 5 x 20 x 30 x 40	1.757	0.757	0.5	1.0
40 x 30 x 20 x 5 x 20 x 30 x 40	1.912	0.716	0.7	1.0
60 x 40 x 20 x 10 x 20 x 40 x 60	2.311	0.607	0.4	1.0
60 x 40 x 20 x 10 x 20 x 40 x 60	2.192	0.621	0.5	1.0
60 x 40 x 20 x 10 x 20 x 40 x 60	2.082	0.662	0.7	1.0

Regarding the evaluation of joint distributions, the set of attributes forms more than 8.6×10^8 cross-modalities. Evaluating all these attributes together is not feasible. Consequently, the most commonly used attributes in transportation modeling from the literature are chosen, on which this evaluation is based (Bouzouina *et al.*, 2021; Plaut, 2006; Sun et Zacharias, 2021; Brückmann *et al.*, 2021; Wang et Cao, 2017; Wang *et al.*, 2022). In line with this approach, a first evaluation is conducted on the direct individual characteristics, followed by a second assessment focusing on the household features associated with each individual.

3.5.1 Performance on the individual attributes

A synthetic population of 1.4 M individuals was generated with the VAE. Figure 3.5 illustrates the performance of the VAE on marginals at the individual level. For each attribute, the proportion of agents generated is shown in blue and the proportion of individuals in the *final test dataset* in gray. These attributes include "link-ref-person", which denotes the type of link the agent has with the reference person of the household: spouse, child, housemate, other, or else the reference person him/herself. Age is categorized into size classes, each featuring a 15-year range. School level indicates five categories of education from



[1] : Five numerical attributes: one input/output for each
[2] : Other categorical attributes

Figure 3.4: Best Variational Autoencoder architecture

primary through Master's level or higher, or other for those without a formal education. Socio-professional class contains nine categories, including student, worker, unemployed and retired. Gender, adolescent status, employment, under-education and driver's license status are all described with two possibilities each.

Figure 3.5 indicates that the VAE performs very well on each attribute independently, with an error of less than 4% for all attributes except Age. The age categories of between 15-29 years and 75+ years are underestimated, with 5% less than the true population, whereas the 30-44 and 45-59 age categories are overestimated by 5% compared to the test set.

More specifically, the socio-professional class has been well modeled despite containing nine categories. This performance is indeed attractive for the application of synthetic population generation, since socio-professional category is a strong indicator of the mode of transportation used by agents during their daily routines. Moreover, both employment and educational status have been well modeled.

The marginal distributions of interest are attributes describing the household or person solely along a single dimension. However, when generating a synthetic population, it is often desirable to obtain a synthetic population whose characteristics mimic the joint distribution of attributes in the true population.

Figure 3.6 illustrates the performance of the VAE on a joint distribution of four attributes. The following four attributes are analyzed: Gender, Educational level, Socio-professional category, and Link with the reference person in the household. These factors are strong and do affect the choice of a person's daily schedule planning; the mode of transportation, activity chain and hours of activities all heavily depend on these agents' characteristics (Hörl et Balac, 2021b; Delhoum *et al.*, 2021).

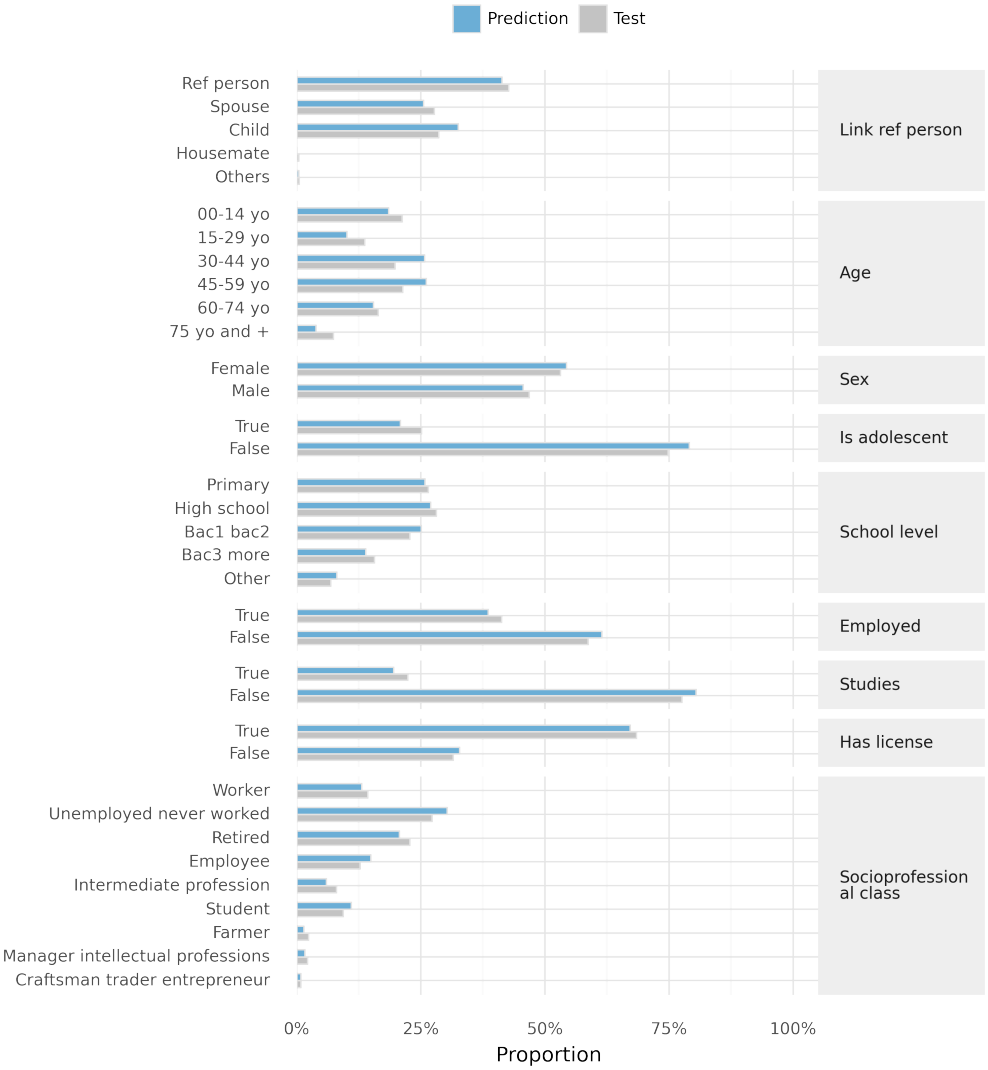


Figure 3.5: Performance on the individual attributes.

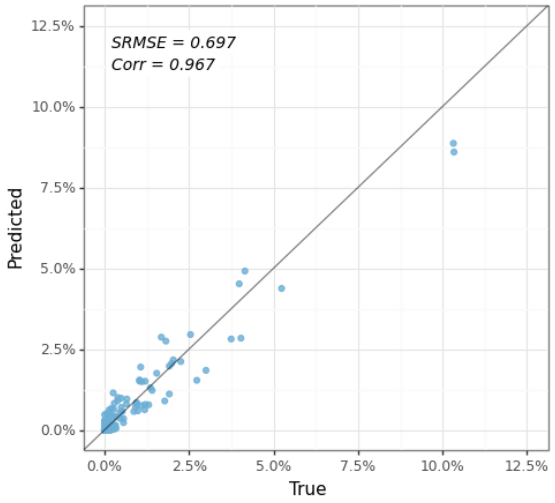


Figure 3.6: Performance on individual attributes: link with the household reference, gender, educational level, and socio-professional class

3.5.2 Performance on the household attributes

The model performance in terms of marginal proportions relative to household attributes is illustrated in Figure 3.7. The blue bars represent the generated data, while the gray bars show the test data. Several attributes are presented in this figure, namely: type of housing with the modalities "embracing", "isolated", "big co", "little col" and "other type"; type of home occupation with the modalities "public housing", "owner", "renter" and "others"; the number of vehicles; the number of bicycles; and internet availability in the house.

It can be observed that the VAE model generally produces better performance for all household-related attributes, with an error of below 3%, except for certain modalities of the attributes "number of vehicles" and "number of bicycles". Indeed, the households with two vehicles are slightly overestimated, while those with more than four are underestimated. Similarly, households with 1 or 2 bicycles are overestimated, while those without any bicycles are underestimated.

Figure 3.8 presents VAE model performance in terms of joint probability of the attributes "type of housing", "type of house occupancy", "number of vehicles" and "Internet availability", in yielding a total of 150 possible combinations. These attributes are often chosen for socioeconomic studies on transportation (with the "Internet availability" attribute being correlated with telecommuting). Let's note an average error of 0,5% between the joint probabilities of these attributes, of which over 60% reveal an error of less than 1%.

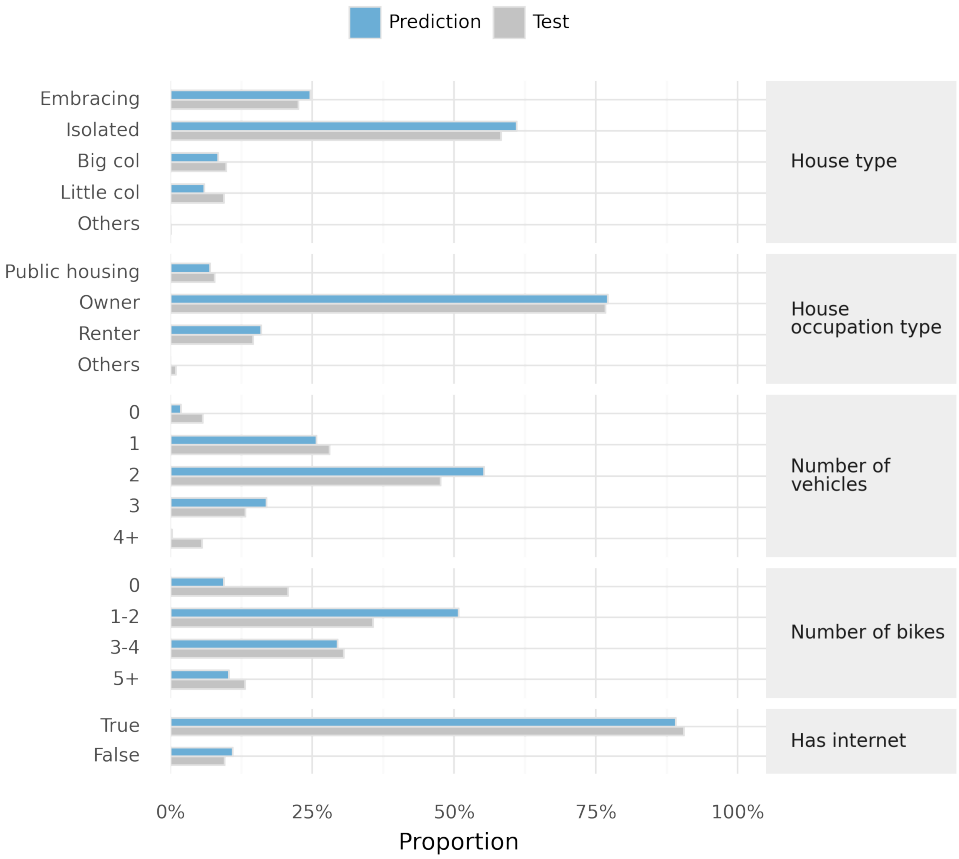


Figure 3.7: Performance of VAE on the household attributes: Marginal distributions

3.6 Influence of the model building dataset size

Another aspect of this analysis is VAE performance relative to the size of its model-building dataset. The importance of the training samples required has been established relative to the so-called zero-cell problem. This problem pertains to two kinds of events that can occur during population generation: 1) the capability of the VAE to generate agents that were not encountered during its training yet still exist in the actual population; and 2) the propensity to generate new agents that do not correspond to reality. These incorrect cell values significantly diminish the representativeness of the synthesized population.

For the following experiments, in addition to the best VAE characterized in Section 3.5 above, we have built three other VAE models using the same chosen architecture (depicted

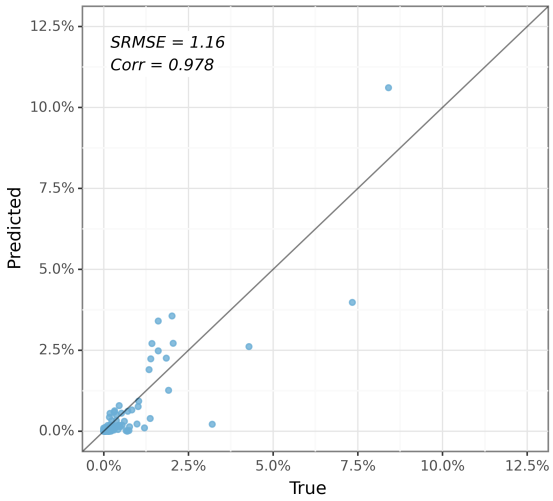


Figure 3.8: Performance on the household attributes: housing type, house occupation type, number of vehicles, internet availability

in Figure 3.4), yet trained on a model-building database sized respectively 1 000, 5 000 and 10 000. From each of these models, a synthetic population of 1.4 M agents has been randomly generated, thus yielding four distinct synthetic populations generated by four VAEs, whose parameters were estimated on a growing training database.

3.6.1 Influence on the accuracy of the generated population

The following attributes are modeled jointly by the VAE: 'house type' (5 levels), 'link with ref person' (5 levels), 'has internet' (2 levels), 'age' (6 levels), 'sex' (2 levels), 'school level' (5 levels), 'socio-professional class' (9 levels). The output yields a 27 000-dimensional joint representation meaning there are 27 000 possibilities for combining such attributes. We then conducted multiple generations of 1 400 000 individuals and consistently obtained the same results (to within 0.1%). This stability in the results is, probably, due to a phenomenon similar to that of the Law of Large Numbers.

The lower the SRMSE, the better the model: according to Figure 3.9, the optimal model-building set size comprises approximately 10 000 samples. A reduced number of 1 000 training examples performs the poorest on the test database. For the Corr metric, the higher the value, the better the model. For this metric, Figure 3.9 indicates that the optimal model-building set size is also 10 000 samples. According to these metrics, the larger the training sample size, the better the model performance, as would be expected. The results obtained with the 5 000 training examples are quite acceptable with a correlation coefficient of 0.6.



Figure 3.9: Values of the various metrics on the 41.6 k-dimensional joint representation with a growing training set size

3.6.2 Influence on the zero-cell problem

3.6.2.1 Zero-cell : more details

As mentioned in the section 2.1.1, the zero-cell problem focuses on samples featuring a rare combination of attributes within a population. Rarity is a relative measure that refers to the survey data and not to the final population generated. With a population micro-sample, some of the feature combinations have a nonzero probability of being present in the general population, yet still go unobserved. These combinations are called *sampling zeros*: they are present in the test set but not in the model-building set. They are especially numerous if the micro-sample is small and the number of attributes and their modalities is increasing. For instance, a young person with a low level of education but a high socio-professional status might be infrequent but not altogether impossible to generate. Machine learning approaches provide solutions to the zero-cell problem, along with the ability to generate data that are not present during the training phase, although such added flexibility does come at a cost. Attribute combinations with a zero probability in the real population may be assigned a positive probability by the trained model. Such feature combinations are commonly referred to as *structural zeros*. They represent the aberrant samples, a hallucination in the deep learning vocabulary, that are generated by the model, such as a one-year old person with a driver's license or a retiree who is an adolescent.

Figure 3.10 illustrates the concepts of sampling zeros and structural zeros in the case of a database described by two features x1 and x2. A micro-sample is divided into model-building and test databases. The first two tables are the model-building (or, more concisely put, training) dataset and the test set used in this example. After training a model, let's suppose it produces the third data table depicting the generated population. Two sampling zeros are shown in green in the test set but not in the model-building set. In the generated population, the model was able to fill one representation of these cells but not the other, in which no example was generated. According to the metric introduced in Garrido *et al.* (2020), the fraction of sampling zeros equals 50%.

The red cells shown in Figure 3.10 are the structural zeros. These samples are in neither the model-building set nor the test set; they represent impossible feature combinations. Three such cells appear in this example: in the generated population, two cells were filled by the model, which means that the model generated agents for an impossible combination of features. A ratio of 7/7 structural zeros per sampling zeros was derived, according to the metric proposed in Garrido *et al.* (2020). In the following evaluation, this absolute ratio measurement #struct / #sampl will be used as a metric to characterize the performance of the built VAE when addressing the zero-cell problem. In particular, no attention is paid to the actual content of the generated sampling zero or structural zero-cells.

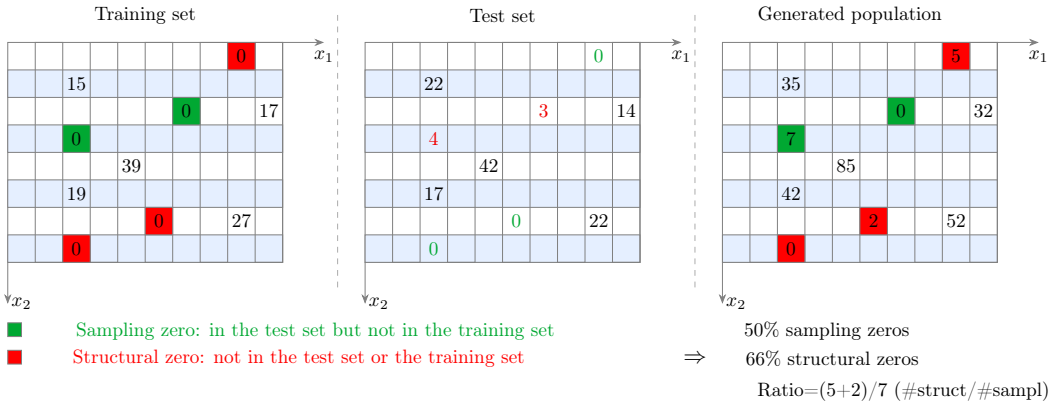


Figure 3.10: Zero-cell problem in the case of samples with two attributes x_1, x_2. Sampling zeros are possible representation of individual features (x_1, x_2) not presented during the training. Structural zeros are impossible combinations of features that are still generated by the model.

Classical methods such as IPF and its derivatives are limited to replicating individuals from the training data table, meaning they do not generate new individuals, including sampling zeros and structural zeros. Under these specific conditions, for these methods to produce a high-quality and diverse synthetic population, the input data must be large and

contain a sufficient variety of individuals. In contrast, the VAE not only replicates individuals from the sample but also generates new individuals (zero-cells), thereby enhancing the diversity of the synthetic population.

3.6.2.2 Zero-cell assessment

Table 3.4 reports the total number of zero-cells and the fraction of zero-cell agents generated in a 27 000-dimensional joint representation including the following attributes : 'house type', 'house occupation type', 'has Internet', 'link with ref person', 'gender', 'educational level', 'socio-professional status' for the various training set sizes. The dataset is composed of 27 000 possible cross-modality. The test set size comprises 5 900 samples, with a large number of these being identical. For instance, the total number of possible structural zeros for a model-building dataset size of 5 000 is 25 728 out of the total 27 000 possible combinations of attributes : 96% of the representation space is composed of impossible cross-modalities. The representation space of the attribute is heavily dominated by impossible cross-modalities. For this model-building dataset size, the total number of possible sampling zeros is 386; these samples need to be generated by the VAE even if not trained on them.

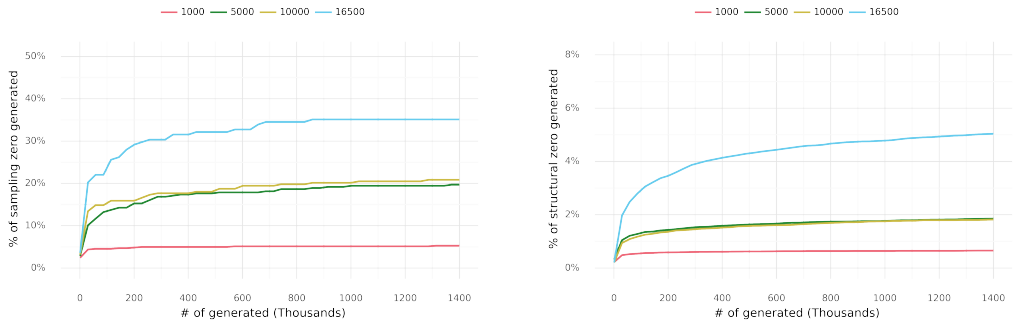
For a model-building set size of 5 000 samples, 42 000 of the 1.4 M generated agents (or 3%) have a cross-modality being a sampling zero, i.e. a cross-modality that is a realistic combination of attributes that were not presented to the VAE during the training phase. This criterion demonstrates the capacity of VAE generalization, i.e. the ability to generate agents whose cross-modalities are not in the training set yet present in the actual population. Classical generation methods do not possess this ability. 238 000 agents (17% of the 1.4 M synthetic population) correspond to structural zeros, which reflect impossible combinations of attributes. This number may seem high, although one needs to consider that 96% of the representation space (of dimensional joint modalities equal to 27 000) is composed of such unrealistic values. For this reason, the recount of zero-cells in Garrido *et al.* (2020) uses fractions of the total possible sampling zeros and total possible structural zeros.

Figure 3.11 displays these fractions, plotted versus the number of agents generated for a growing agent population from 0 to 1.4 M. Figure 3.11(a) plots the fraction of sampling zeros generated for the four VAEs. For instance, the VAE trained on the training database of 5 000 samples was used to generate 800 000 agents: 20% of the possible sampling zeros were correctly generated, while less than 2% of the possible structural zeros were generated, as shown in Figure 3.11(b). From this point of view, the VAE outperforms classical methods like IPF, which are unable to generate agents whose attributes are not present in the model-building dataset.

Another relevant indicator to examine is the ratio between the number of structural zeros and the number of sampling zeros; this indicator assesses the number of structural

Table 3.4: Total number of sampling zeros and structural zeros and Fraction of agents whose cross modalities are sampling zeros and structural zeros over the 1.4M generated agents according to the training size (18k-dimensional joint)(27k-dimensional joint)

Train set size	Number of possible cross-modalities		Fraction of agents with these cross-modalities	
	# sampling	# structural	% of agents with sampling	% of agents with structural
1 000	663	25 964	15,56	60,24
5 000	386	25 728	2,94	17,42
10 000	283	25 514	1,57	9,44
16 500	168	25 093	0,71	7,28



(a) **(b)**

Figure 3.11: Fraction of total sampling zeros and total structural zeros generated in the synthetic population for each VAE and for a growing total population size.

zeros generated for a given sampling zero. A lower ratio generally indicates a better model. Figure 3.12 illustrates the evolution of this ratio as a function of the number of agents generated as well as for different training sample sizes. It can be seen that the ratio value stabilizes after the generation of 800 000 individuals. Average values lie on the order of 5, 6, 7 and 20 for training sample sizes of 1 000, 5 000, 10 000 and 16 500 respectively. Indeed, with a small training sample, the number of sampling zeros is higher, since the model has access to only a limited subset of the possible cross-modalities in the data. Conversely, a larger training sample provides the model with a more complete representation of the joint attribute space, thereby reducing the occurrence of sampling zeros. The VAE model trained on a sample of 5 000, for example, generates 6 structural zeros for each sampling zero in a space of 27 000 joint modalities where 95% are structural zeros.

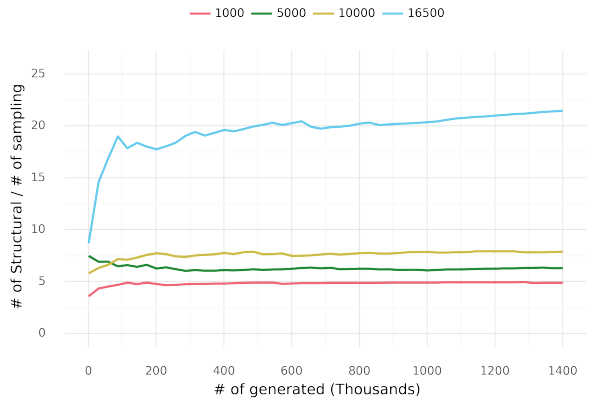


Figure 3.12: Ratio of the number of structural zeros to sampling zeros versus the size of the generated population.

The previous analysis has demonstrated that a training sample size of 5,000 individuals offers a good compromise in obtaining a suitable representation of the actual population.

This approach reveals the benefit, but also the limitations, of such methods. One possible way to limit structural zeros would be to post-process the results in order to eliminate obvious "aberrations" like retired adolescents. The treatment of aberrations remains an open issue. One potential solution could entail the use of external data, such as marginal data, during model construction so as to eliminate some of these aberrations. It is also possible to combine VAE with other methods, such as IPF, in a complementary manner, capitalizing on their respective strengths while addressing their shortcomings. For instance, a two-step approach, as demonstrated by Ilahi et Axhausen (2019); Rahman et Fatmi (2023) in combining a Bayesian Network (BN) with IPF, could first generate a diverse synthetic population using a VAE. This would then be followed by adjusting the population to align with the desired marginal distributions through IPF or its variations. Such an approach would ensure diversity in the synthetic population while maintaining consistency with the marginal constraints. A more detailed exploration of this combined approach is presented in Chapter 5.

3.7 Conclusion

The objectives of this chapter have been to provide a methodological framework for implementing and evaluating a VAE model aimed at generating a synthetic population of agents using mixed data, i.e. both numerical and categorical. After presenting the VAE model, the parameters to consider when training the model are explored. Various indicators to consider when evaluating performance are investigated. Special focus has been placed on evaluating

the ability of VAEs to address a well-known challenge in generating synthetic populations, namely that of zero cells.

The main results demonstrate the remarkable performance of VAEs. Using a model trained on a sample of 16 500 individuals, 1,4 M synthetic agents were able to be generated. The average errors for attribute marginal data were less than 3 percentage points, while the SRMSE values were 0,6 for individual attributes and 1,16 for cross attributes.

Furthermore, VAEs are capable of processing high-dimensional data while maintaining good performance when the sample size is small. In our case, we were able to obtain a high-performance model with a sample of 5 000 individuals, with each individual described by 16 attributes (representing 45 model inputs after a "one-hot" encoding of the categorical attributes). However, the larger the training sample size, the better the model.

Another notable advantage of VAE is its ability to generate new individuals not present in the training sample. In this study, with a training sample size of 5 000, the model was able to generate around 20% of sampling zeros and a small proportion of structural zeros, within a space including 27 000 cross-modalities, of which approximately 96% were structural zeros.

Although VAE performance is indeed promising, there are several areas for improvement. First, comparing VAEs with numerical versus categorical attributes suggests that VAEs with categorical attributes are more robust. The modeling of categorical attributes can be likened to a classification problem, while that of numerical attributes is more akin to a regression, as characterized by a greater variability of outputs. Therefore, a categorization of numerical attributes could improve overall model performance. Second, when generating new individuals not present in the training sample, the proportion of individuals representing structural zeros remains non-negligible. Consequently, post-processing has proven to be necessary to reduce or even eliminate these individuals.

VAEs offer several advantages that enable practitioners to address a variety of challenges, including those arising from limited sample sizes. They also provide a means to preserve data confidentiality, allowing experiments and model testing to be conducted without direct access to sensitive or proprietary data. In the context of urban modeling, for instance, VAEs can be employed to generate synthetic yet statistically consistent representations of city residents, which can subsequently be used in simulation frameworks such as *MatSim*.

There are many interesting perspectives that can be considered by continuing the work discussed in this chapter. For one thing, the integration of VAE models into computer libraries covering the entire processing chain for synthetic data generation, like the EQASIM pipeline (Hörl et Balac, 2021a), offers additional possibilities. This approach would enable more sophisticated data generation for urban simulations, with a wide range of potential

applications. Moreover, incorporating marginal data could further enhance model performance by reducing aberrations or hallucinations, aspects that will be discussed in detail in Chapter 5. Another promising direction involves extending the modeling framework to generate individuals within households, thereby capturing both the specific characteristics of each individual and the interdependencies among household members. For instance, in urban transportation modeling, the travel mode choices of household members are often correlated. This dimension will also be examined in Chapter 4. Integrating such intra-household relationships could substantially improve the behavioral realism and predictive accuracy of transportation models. Finally, introducing sequential data—such as daily activity plans—opens the way to even richer and more dynamic representations of synthetic populations, a topic that will be explored in Chapter .

Chapter 4

Two-layered Population Synthesis with VAE: A Joint Household-Individual Approach

Contents

Abstract **67**

4.1 Introduction **68**

4.2 Two Approaches for Modeling Two-Level Synthetic Population Generation using VAEs **69**

4.2.1 Single VAE for two layered population generation (SVAE-Pop2) 69

4.2.2 Multi VAE for two layered population generation (modeling-based multiple VAE) (MVAE-Pop2) 70

4.3 Experiments **71**

4.3.1 Case study dataset 71

4.3.2 Loss function 72

4.3.3 Model building 73

4.4 Results **74**

4.4.1 Performance on the Individual Attributes 75

4.4.2 Performance on the Household Attributes 78

4.4.3 Performance on the Joint Household-Individual Cross-Modalities 79

4.5 Discussion **79**

4.6 Conclusions and future work **81**

Abstract

Abstract

This chapter extends the use of Variational Autoencoders (VAEs) to the joint generation of households and their members, introducing a hierarchical framework for two-layered synthetic population generation. Building on the results of Chapter 3, which focused on single-layered populations, the work addresses the challenge of modeling interdependencies between individuals within the same household—an essential aspect for applications such as urban mobility and transportation microsimulation.

Two architectures were proposed and evaluated: SVAE-Pop2, a unified single-VAE model capable of generating complete households and individuals jointly, and MVAE-Pop2, a multi-VAE approach where separate models are trained for each household size. Both methods produced coherent and realistic synthetic populations, closely replicating observed distributions. The MVAE-Pop2 architecture achieved slightly higher accuracy for joint attribute distributions, while SVAE-Pop2 proved more efficient and robust when data are limited or unevenly distributed across household sizes.

The comparative analysis highlighted the trade-offs between both strategies. MVAE-Pop2 requires sufficiently large samples for each household size, which can be limiting for underrepresented configurations, whereas SVAE-Pop2 leverages the entire dataset but introduces potential biases due to the use of padding for incomplete households. The choice between the two approaches therefore depends on data availability and the desired balance between accuracy and flexibility.

Keywords : *Synthetic population, machine learning, deep generative model, variational autoencoders, Household-Individual, two-layered*

4.1 Introduction

Chapter 3 demonstrated the potential of the Variational Autoencoder (VAE) for generating single-layered synthetic populations, focusing either on individuals or on households taken independently. The results confirmed that VAEs can efficiently reproduce complex joint distributions while maintaining high performance even with limited training data. However, this approach treated households and individuals as separate entities, without explicitly accounting for the hierarchical dependencies that link them. In many real-world contexts—particularly in urban modeling and transportation microsimulation—these interdependencies play a crucial role in shaping behavioral patterns and aggregate system dynamics.

As highlighted in the literature review, only a few studies have addressed the use of deep generative models for hierarchical population synthesis. Their application to the joint generation of households and individuals remains very recent, with notable contributions from Aemmer et MacKenzie (2022) and Qian *et al.* (2024). The former proposed a two-step VAE/CVAE framework in which synthetic households are generated first, followed by individuals conditionally created based on household attributes. The latter introduced a unified approach encoding both household- and individual-level information into a single vector processed by a VAE to jointly generate the population. To better capture intra-household dependencies, they also designed a specialized loss function—the *Decoupled Binary Cross-Entropy (D-BCE)*—to quantify the relationships between household attributes and those of individual members. As discussed in Section 2.2.2, both methods present limitations: the two-step architecture of Aemmer et MacKenzie (2022) may lead to inconsistencies in household composition, while the unified representation of Qian *et al.* (2024) incurs high computational costs that hinder scalability and practical use.

In response, this chapter introduces a novel deep generative framework for two-layered synthetic population synthesis. Its contributions are fourfold:

- **SVAE-Pop2**: a unified architecture based on a single VAE to jointly generate complete households and their members in an end-to-end process;
- **MVAE-Pop2**: an alternative multi-VAE framework, where distinct VAEs are trained for each household size, better capturing structure-specific dependencies;
- **Empirical evaluation**: a systematic case study assessing the realism and consistency of the generated populations;
- **Open and reusable codebase**: publicly available, well-documented code to ensure reproducibility and facilitate adaptation to diverse application contexts.

The remainder of this chapter is structured as follows. First, the proposed methodology is described in detail. Next, the experimental results are presented, together with a comparative analysis of model performance. Finally, the main contributions and limitations of the approach are discussed, and directions for future research are outlined.

4.2 Two Approaches for Modeling Two-Level Synthetic Population Generation using VAEs

One of the major challenges encountered when generating a synthetic population that integrates both households and their individuals lies in data modeling. In fact, the number of individuals varies from one household to another, while VAE architectures require fixed-sized inputs. To overcome this constraint, two distinct approaches are proposed (see Fig 4.1):

- A single-model approach, in which each household and its individuals are represented as an extended vector.
- A household size-specific approach, whereby the representation of each household and its individuals is also provided as an extended vector, with a dedicated model for each household size.

It is important to note that the underlying principles of the Variational Autoencoder—its architecture, training process are identical to those presented in Section 3.2.

4.2.1 Single VAE for two layered population generation (SVAE-Pop2)

This approach, which relies on a single VAE, requires that the input be of fixed size. To satisfy this requirement, a vector with a maximum size corresponding to a household composed of S members (in the present example, $S = 5$) is defined. When actual households contain fewer than S members, the corresponding vectors are padded with fictitious individuals, each characterized by specific attributes.

$$X = \left(\underbrace{x_1, x_2, ..., x_{k_h}}_{household}, \underbrace{x_1^1, x_2^1, ..., x_{k_i}^1}_{individual_1}, \underbrace{x_1^2, x_2^2, ..., x_{k_i}^2}_{individual_2}, \cdots, \underbrace{x_1^S, x_2^S ... x_{k_i}^S}_{individual_S} \right) \tag{4.1}$$

where k_h is the number of attributes describing the household, and k_i the number of attributes for each individual. x_k is the kth household attribute, while x_j^i is the jth individual attribute of individual i, as described in Table 4.1.

For example, for $S = 5$, a household with three individuals is padded with two fictitious

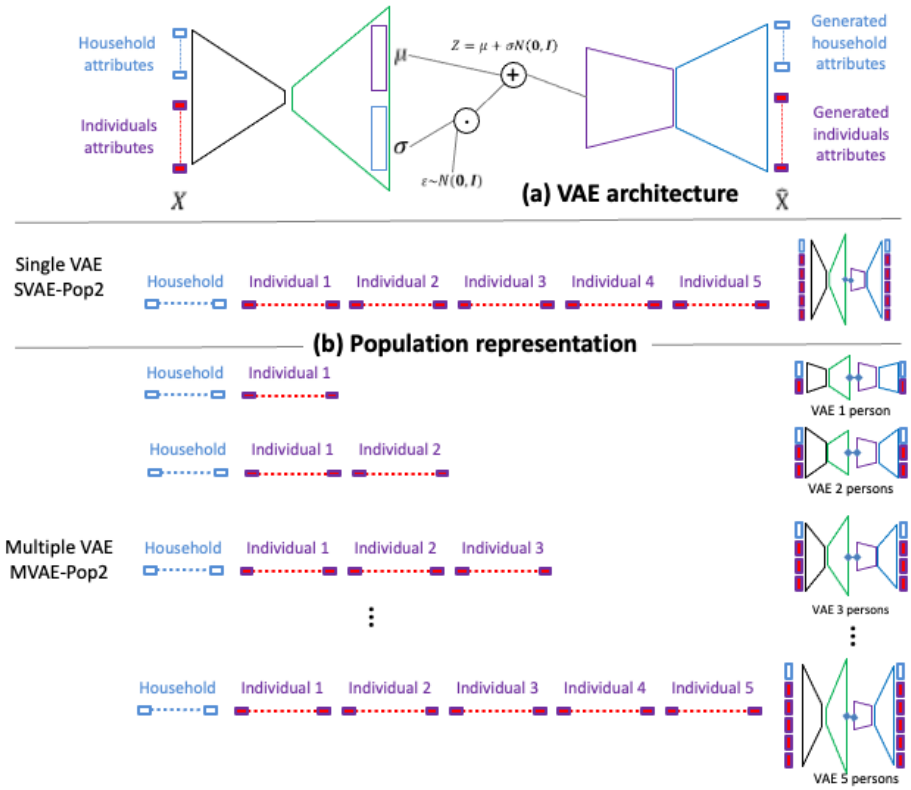


Figure 4.1: (a) General architecture of a VAE for population modeling with two layers (Household, Individuals) ; (b) Inputs for SVAE and MVAE representations.

individuals, who in turn are assigned additional modalities for each attribute. As an illustration, for the gender attribute (called *sex*), which originally comprised two modalities, a third modality, referred to as a *fictional gender*, is introduced to represent the attributes of these fictitious individuals.

4.2.2 Multi VAE for two layered population generation (modeling-based multiple VAE) (MVAE-Pop2)

The second approach consists of developing a specific model for each household size. Thus, when considering a maximum household size of $S = 5$, a total of five specialized VAEs is obtained. This method effectively overcomes the challenges associated with variations in household size. The following discussion presents the inputs of the VAEs specialized in generating households of size 1 (Eq. 4.2) and of size s (Eq. 4.3). The number of household attributes equals $k_h - 1$ because household size does not need to be taken into account.

For one individual:

$$X = [\underbrace{x_1, x_2, ..., x_{k_h-1}}_{household}, \underbrace{x_1^1, x_2^1, ..., x_{k_i}^1}_{individual_1}] \tag{4.2}$$

For a household with s individuals:

$$X = \left(\underbrace{x_1, x_2, ..., x_{k_h-1}}_{household}, \underbrace{x_1^1, x_2^1, ..., x_{k_i}^1}_{individual_1}, \cdots, \underbrace{x_1^s, x_2^s ... x_{k_i}^s}_{individual_s} \right) \tag{4.3}$$

4.3 Experiments

4.3.1 Case study dataset

The dataset used in this chapter is the same as that employed in the previous one, namely the *Household Travel Survey of France's Loire-Atlantique Département* (Département-Loire-Atlantique, 2015). This publicly available dataset contains detailed information on both household-level and individual-level attributes, acknowledging that each household may comprise multiple individuals. Such surveys are typically conducted at the national or regional scale to analyze travel behavior and transportation demand, making this dataset representative of mobility-oriented microdata commonly used in transportation modeling.

For this case study, the maximum household size is set at $S = 5$, corresponding to 12 575 households and 29 496 individuals. The variables selected for this analysis, as presented in Table 4.1 below, pertain to both individuals and their households.

Some originally numerical attributes were recoded as categorical, resulting in a dataset consisting entirely of categorical attributes. This decision was inspired by the findings of Sané *et al.* (2025b); Garrido *et al.* (2020); Johnsen *et al.* (2022), which show that Variational Autoencoders (VAEs) perform better on categorical attributes than numerical ones. The rationale behind this transformation is justified in more detail in the section discussion.

Moreover, 70% of the data is used to train the models, while the remaining 30% serves as the final test dataset. We hypothesise that the characteristics of the final test dataset are the same as those of the real population. This population is referred to as the 'reference population' throughout the remainder of this chapter.

Table 4.1: Overview of household and individual features

Groups	Attributes	Type[1]	Description
Household (HH)			
	house_type	cat(5)	House type
	house_occupation_type	cat(4)	House occupation type
	household_size	cat(5)	Number of persons in the HH
	number_of_vehicles	cat(5)	Number of vehicles in the HH
	number_of_bikes	cat(4)	Number of bicyles in the HH
	has_internet	cat(2)	Has Internet access ?
Individuals			
	link_ref_person	cat(5)	Link with the HH ref. person
	age	cat(6)	Age
	sex	cat(2)	Gender
	is_adolescent	cat(2)	Is an adolescent ?
	school_level	cat(5)	Educational level achieved
	employed	cat(2)	Is employed ?
	studies	cat(2)	Is a student ?
	socioprofessional_class	cat(9)	Socioprofessional status
	has_license	cat(2)	Holds a driver's license ?

[1] All attributes are categorical. The number in parentheses reflects the number of possible values.

4.3.2 Loss function

Given that all attributes are categorical, the cost function differs slightly from that introduced in Chapter 3 (Eq. 3.4). The loss function of the proposed VAE architecture combines two terms, as expressed in Equation 4.4:

$$L(\phi,\theta) = \sum_{a=1}^{N} \underbrace{\sum_{k=1}^{K}\sum_{d=1}^{D_k} x_a^{kd} \log \hat{x}_a^{kd}}_{\text{Reconstruction cost}} + \underbrace{\beta \times \mathbb{KL}(\mathcal{N}(\mu,\sigma)\|\mathcal{N}(\mathbf{0},\mathbf{I}))}_{\text{KL cost}} \tag{4.4}$$

where N is the number of agents (individuals or households), K is the number of attributes, and D_k is the number of modalities for attributes k. μ (mean) and σ (standard deviation), both of size m constitutes the encoder outputs.

The first term accounts for the *reconstruction error*, which measures how well the VAE reproduces the input data X in its output $\hat{X}$. The second term, a *regularization term* represented by the *Kullback-Leibler (KL) divergence*, quantifies how much the learned latent space distribution deviates from the standard Gaussian prior $\mathcal{N}(\mathbf{0},\mathbf{I})$.

The hyperparameter $\beta \in \mathbb{R}_+$ serves as weighting coefficients, intended to balance the

trade-off between reconstruction accuracy and regularization (Higgins *et al.*, 2017).

4.3.3 Model building

4.3.3.1 Identification of hyperparameters

To determine the optimal architecture for SVAE-Pop2, a range of hyperparameters has been explored. Table 4.2 provides an overview of the hyperparameter space examined to identify the most effective models. This table lists value ranges for several key hyperparameters, including the number of layers, the number of neurons per layer, the latent dimension, the number of epochs, the batch size, the β value, and the activation function. In all, 210 models were implemented for the SVAE-Pop2 architecture.

Table 4.2: Models' Hyper-Parameters Overview

Layers	Nodes	Latent Dim	Epochs	Batch Size	β	Activation
1–3	50–100	10–40	200	256	0.0005–0.01	sigmoid, tanh

This same hyperparameter optimization procedure was similarly applied to determine the optimal architecture for MVAE-Pop2.

4.3.3.2 Comparison metrics

To evaluate model performance, the same set of quantitative indicators introduced in Chapter 3 are employed, namely the Correlation Coefficient (*corr*) and the Standardized Root Mean Square Error (*SRMSE*), complemented here by a third metric, the Mean Absolute Error (*MAE*). These metrics measure the distance between two agents: π, the modality vector of the true agent, and $\hat{\pi}$, the vector predicted by the model.

The mean absolute error (*MAE*) (Eq. 4.5) quantifies the average magnitude of errors in a set of predictions, without considering their direction; it is calculated as the average of the absolute differences between predicted and actual values. The MAE is expressed in the same units as the original data, making it easy to interpret.

$$MAE = \frac{1}{N_{all}} \sum_{l} |\pi_l - \hat{\pi}_l| \tag{4.5}$$

The SRMSE was used to compare architectures because it is the most robust metric and thus the most commonly employed for this purpose (Borysov *et al.*, 2018; Aemmer et MacKenzie, 2022; Sané *et al.*, 2025b). Other metrics were also used to interpret the results of the methodologies applied to the representative case study.

After optimizing the hyperparameters, the best-performing architectures based on SRMSE

for both modeling approaches (SVAE-Pop2 and MVAE-Pop2) are presented in Table 4.3. The optimal SVAE-Pop2 model follows an architecture of $245 \times 80 \times 40 \times 80 \times 245$, with a β value of 0.005 and a *sigmoid* activation function. This model consists of an encoder and a decoder, each composed of three layers. The encoder includes an input layer of size 245, a single hidden layer of size 80, and a latent space layer of size 40. The decoder mirrors the encoder's structure, with three layers of sizes $40 \times 80 \times 245$. For the MVAE-Pop2 modeling approach, which employs a specialized model for each household size, the optimal architecture varies accordingly. For instance, the best-performing model for households of size 4 adopts an architecture of $176 \times 100 \times 50 \times 10 \times 50 \times 100 \times 176$ with a β value of 0.005 and a *tanh* activation function.

Table 4.3: Overview of the best-performing models

Model	Input size	Architecture	β	Activation
SVAE-Pop2	245	$245 \times 80 \times 40 \times 80 \times 245$	0.005	sigmoid
MVAE-Pop2				
$size = 1$	59	$59 \times 100 \times 40 \times 100 \times 59$	0.0005	sigmoid
$size = 2$	98	$98 \times 50 \times 20 \times 50 \times 98$	0.001	sigmoid
$size = 3$	127	$127 \times 80 \times 20 \times 80 \times 127$	0.005	tanh
$size = 4$	176	$176 \times 100 \times 50 \times 10 \times 50 \times 100 \times 176$	0.005	tanh
$size = 5$	215	$215 \times 100 \times 20 \times 100 \times 215$	0.0005	tanh

4.3.3.3 Software Stack and Computational Resources

Data processing for model evaluation was carried out using Python. The primary libraries used were: Pandas 1.5, NumPy 1.23 and TensorFlow 2.10. Graphs were created using the Plotnine 0.10 library. Both the source code and the data are available via GitLab link[1]. These tasks were carried out using the Cines-Adastra[2] supercomputer with two AMD EPYC 7A53 (Trento) processors with 64 cores at 2.0 GHz and 128 GB of RAM.

4.4 Results

Once the optimal architectures have been determined at the end of the training phase, a synthetic population is generated for each approach (SVAE-Pop2 and MVAE-Pop2). The total number of generated entities amounts to approximately 1,4 M individuals distributed among 650 k households, which corresponds to the size of the population in the study area. These synthetic populations served as the basis for evaluating and comparing the ability of each model to faithfully reproduce the characteristics of the actual population, with these

[1] https://gitlab.univ-eiffel.fr/population-synthesis/VAE-joint-houshold-individuals/
[2] https://www.cines.fr/calcul/espace-utilisateur/

characteristics assumed to be the same as those of the reference population described in the previous section.

This section provides a detailed analysis of the results obtained, in highlighting the respective performance of each approach, their differences and their practical implications. This evaluation focuses on: individual characteristics, household-specific attributes, and the overall coherence between these two levels of information.

4.4.1 Performance on the Individual Attributes

The evaluation of how well both approaches perform on individual attributes was carried out in two steps. First the marginal distributions of each attribute is analyzed, then the joint distribution of these attributes was examined.

Figure 4.2 presents the marginal distributions of the individual attributes. The orange and blue bars correspond to the synthetic populations generated by the SVAE-Pop2 and MVAE-Pop2 approaches, respectively, while the green bars represent the test data used as a reference. Overall, the marginal distributions of the synthetic populations align well with those of the test data, although some discrepancies do remain. The mean absolute error is 3,5 percentage points for SVAE-Pop2 and 2,9 percentage points for MVAE-Pop2, with standard deviations of 2,9 and 2,7, respectively. Moreover, 75% of the modalities exhibit an absolute error below 6 percentage points for SVAE-Pop2 and below 4,9 percentage points for MVAE-Pop2. The maximum errors observed reach 8,6 percentage points for SVAE-Pop2 (for the attribute *studies*) and 8,8 percentage points for MVAE-Pop2 (for the attribute *has_license*).

The adequacy of the marginal distributions is further confirmed by Figure 4.3, which compares the marginal distributions of the test data (on the x-axis) with those of the synthetic populations (on the y-axis). The points are generally aligned along the line with a slope of 1, thus indicating a good level of correspondence between the distributions. However, the SVAE-Pop2 approach (in orange) exhibits a slightly higher dispersion compared to MVAE-Pop2 (in blue), which is corroborated by the SRMSE values: 0,17 for SVAE-Pop2 vs. 0,15 for MVAE-Pop2. The linear correlation coefficient is identical for both approaches, at 0,98.

The second evaluation focuses on comparing the joint distributions of individual attributes. For this purpose, five attributes, frequently used in mobility studies, are selected: Link with household reference person (*link_ref_person*), Age, Gender, Holds a driver's license (*has_license*), socioprofessional class (Bouzouina *et al.*, 2021; Plaut, 2006; Sun et Zacharias, 2021; Brückmann *et al.*, 2021; Wang et Cao, 2017; Wang *et al.*, 2022). Figure 4.4 illustrates this comparison; it can be observed that the MVAE-Pop2 approach (in blue)

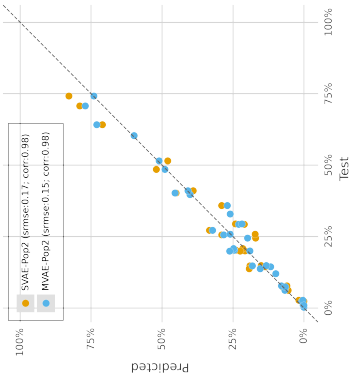


Figure 4.3: Goodness of fit of the marginal distributions of generated individual data to the test values

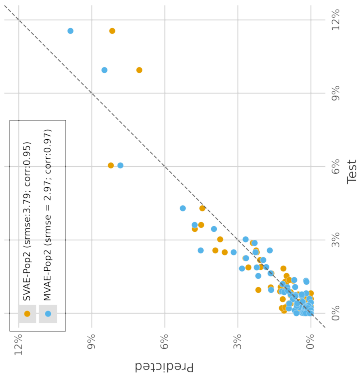


Figure 4.4: Multivariate comparison between predicted and test data for individual attributes (link_ref_person, age, sex, has_license, socioprofessional_class)

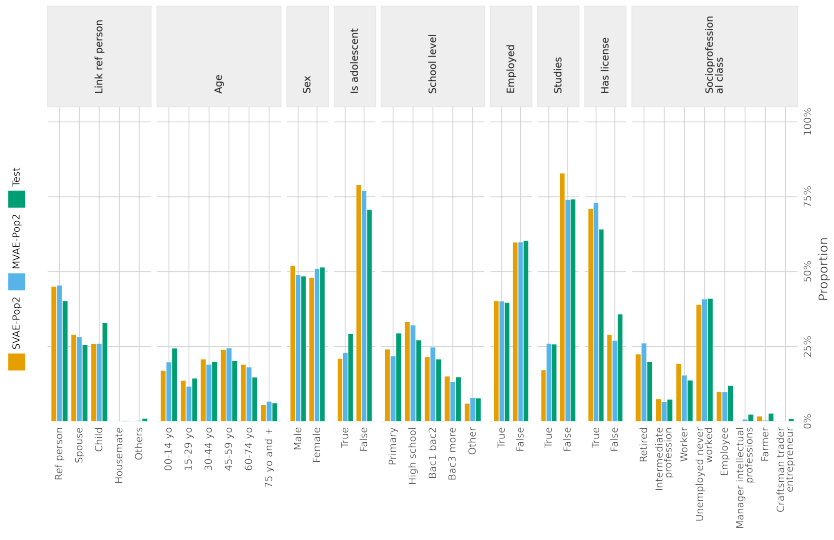


Figure 4.2: Marginal distributions for the true population (green) and samples generated using VAE (orange and blue) for individual attributes

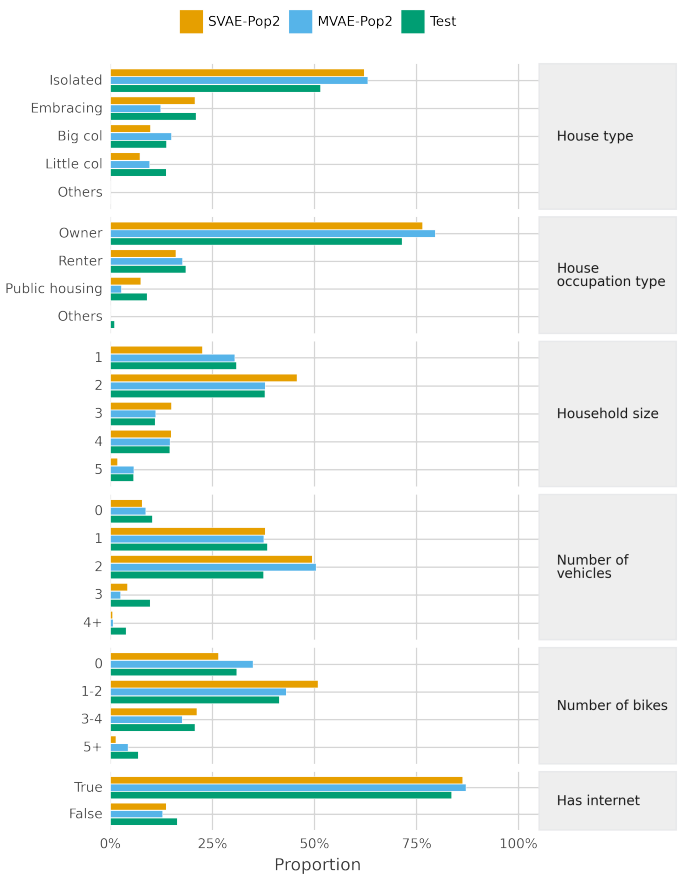


Figure 4.5: Marginal distributions for the true population (green) and samples generated using VAE (orange and blue) for household attributes

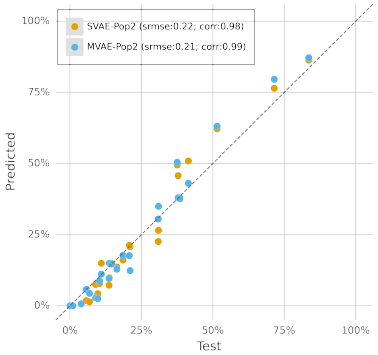


Figure 4.6: Goodness of fit of the marginal distributions of generated household data to the test values

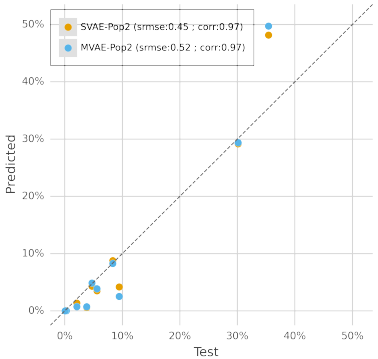


Figure 4.7: Multivariate comparison between predicted and test data for household attributes (number of vehicles, has Internet)

exhibits a lower dispersion compared to SVAE-Pop2 (in orange), with an SRMSE of 3,79 for SVAE-Pop2 versus 2,97 for MVAE-Pop2. Furthermore, the linear correlation coefficient equals 0,95 for SVAE-Pop2 and 0,97 for MVAE-Pop2, thus confirming a higher fidelity of the joint distributions with the latter approach.

4.4.2 Performance on the Household Attributes

The evaluation of how well both approaches perform on household attributes relied on a methodology similar to that used for individual attributes. Figures 4.5 and 4.6 respectively present the marginal distributions and the univariate comparison between the test data and the synthetic populations generated by the two approaches.

The performance metrics indicate a mean absolute error of 4,2 for SVAE-Pop2 and 3,4 for MVAE-Pop2, with standard deviations of 3,3 and 3,7, respectively. The third quartile of absolute errors is 5,5 for SVAE-Pop2 and 4,1 for MVAE-Pop2, meaning that 75% of the absolute errors lie below these values. The maximum errors observed are 12 for SVAE-Pop2 and 12,9 for MVAE-Pop2, reaching their peaks on the attribute *number_ of_ vehicles* in both cases.

Figure 4.6 illustrates this comparison in the form of a scatter plot. A nearly similar dispersion is observed for both approaches, although MVAE-Pop2 (in blue) appears to exhibit a slightly lower dispersion than SVAE-Pop2 (in orange). The SRMSE values are 0,22 for SVAE-Pop2 and 0,21 for MVAE-Pop2, while the linear correlation coefficients are 0,98 and 0,99, respectively.

Regarding the joint household attributes, Figure 4.7 compares the distributions of the attributes *number of vehicles* and availability of internet connection (*has_ internet*). For this analysis, we selected the variables *number of vehicles* and *has Internet*, which are recognized in the literature as household factors influencing mobility (Bouzouina *et al.*, 2021; Plaut, 2006; Sun et Zacharias, 2021; Brückmann *et al.*, 2021; Wang et Cao, 2017; Wang *et al.*, 2022). In this case, MVAE-Pop2 (in blue) shows a higher dispersion than SVAE-Pop2 (in orange), with an SRMSE of 0,52 compared to 0,45. However, the linear correlation coefficient is identical for both approaches, at 0,97.

In order to analyze this relatively lower performance of the MVAE, we also evaluated the performance of each sub-model within the MVAE framework on household-level attributes. Table 4.4 presents these results along with the training sample size for each household size. The findings indicate that sub-models trained on smaller samples perform less accurately (compare the sub-models for Household size 3,5 to the sub-models for Household size 1,2,4). In other words, the performance of each sub-model tends to improve as the size of its training dataset increases. This explains that the MVAE performs slightly worse than the SVAE on

the Household attributes.

Table 4.4: Overview of how the MVAE sub-models performed on household attributes.

Household size	N[1]	srmse	corr
1	3058	0.330	0.985
2	3822	0.623	0.979
3	1122	1.218	0.911
4	1478	0.666	0.983
5	580	1.564	0.900

[1] Size of the training sample.

4.4.3 Performance on the Joint Household-Individual Cross-Modalities

The final stage of the evaluation involves comparing the joint distributions that integrate both household and individual attributes. For this purpose, the attributes *household size, age and sex*, which are common to both levels of information, are selected. The cross-analysis of these attributes allows us to assess the coherence between household characteristics and individual features by examining the gender distribution across different age groups and household sizes.

Figure 4.8 illustrates this comparison. It can be observed that the MVAE-Pop2 approach (in blue) exhibits a lower dispersion than SVAE-Pop2 (in orange), thus indicating a better fidelity of the joint distributions. The SRMSE values are 0,49 for SVAE-Pop2 and 0,36 for MVAE-Pop2, while the linear correlation coefficients equal 0,93 and 0,96, respectively.

4.5 Discussion

The main contribution of this work is the application of VAE for generating two-layered synthetic population. By proposing two innovative approaches, SVAE-Pop2 and MVAE-Pop2, these methods are capable of generating synthetic populations that faithfully replicate the test data at both the individual and household levels. The experimental results indicate that, although both approaches deliver a convincing performance, the MVAE-Pop2 model yields slightly superior results, particularly in terms of the accuracy of the joint attribute distributions.

However, these approaches also present certain limitations. The MVAE-Pop2 model, which relies on a dedicated model for each household size, requires a sufficiently large sample for each of these sizes. This requirement can be problematic for underrepresented household



Figure 4.8: Combined multivariate comparison between predicted and test data (household_size, age, sex)

sizes, which may explain the less satisfactory performance for certain household attributes compared to SVAE-Pop2. Moreover, the MVAE-Pop2 approach necessitates additional information, such as household size, for its proper application.

Conversely, the SVAE-Pop2 approach offers the advantage of employing a single model, thus leveraging the entire dataset during training regardless of household size. However, to ensure a fixed-size input, this method resorts to padding incomplete households with fictitious individuals. While this strategy is practical, it may introduce biases, particularly when the sample size is limited, thereby affecting the model's robustness and generalization capacity. This explains the less satisfactory performance for individual attributes compared to MVAE-Pop2.

The choice between these two approaches should therefore be guided by data availability. If the sample is sufficiently rich for each household size, then the MVAE-Pop2 approach appears to be more suitable, especially when a large number of attributes must be generated, i.e. necessitating a substantial number of individuals per household size. On the other hand,

in contexts where data are limited by household size, the SVAE-Pop2 approach provides a more relevant alternative since it allows inputting the entire dataset without the need for partitioning by household size.

Although both approaches show promising results, their application will ultimately depend on the specific context and availability of data. Future research could explore adaptive strategies or hybrid combinations of these methods to further optimize the quality of the generated synthetic populations while mitigating the limitations identified in this study.

Regarding the categorization of originally numerical attributes, this choice was motivated not only by the findings of previous studies (Sané *et al.* (2025b); Aemmer et MacKenzie (2022); Johnsen *et al.* (2022); Garrido *et al.* (2020)), which consistently show that Variational Autoencoders (VAEs) tend to perform better on categorical data in synthetic population generation tasks, but also by broader technical considerations. Specifically, handling mixed data types with VAEs requires a heterogeneous reconstruction loss—typically combining binary cross-entropy for categorical variables, mean squared error for numerical variables, and the Kullback-Leibler divergence for latent space regularization. These loss components operate on different scales, which can complicate optimization, destabilize training, and hinder convergence. By using a fully categorical representation, the loss formulation is homogeneous, that simplifies optimization and enhances model robustness, particularly important in our two-level (household–individual) architecture. Moreover, in the context of transportation modeling, discretized variables (such as age groups or income brackets) are common practice and align with the level of granularity typically required for applications such as travel demand estimation or scenario-based planning.

The architectures presented in this chapter enable extending a sample to generate a complete synthetic population. However, in many practical applications, practitioners also have access to aggregated or marginal data describing key population attributes. Chapter 5 explores how to fully exploit this additional information by integrating these data.

4.6 Conclusions and future work

This chapter has presented two methods for generating individuals within households using VAEs: the SVAE-Pop2 method, based on a single VAE, and the MVAE-Pop2 method, which employs a dedicated VAE for each household size. In a representative case study, both architectures produced a synthetic population consistent with the actual population, with a slight advantage observed for the MVAE-Pop2 approach.

SVAE-Pop2 is a unified generic solution when data are limited or unevenly distributed across household sizes. In contrast, MVAE-Pop2 proves more effective in settings with large, well-balanced samples, especially when modeling complex intra-household dependen-

cies. These differentiated strengths make each approach suitable for distinct application scenarios.

The two-layered synthetic population is particularly well-suited for real-world applications, especially in domains such as urban traffic modeling and mobility analysis. By explicitly modeling the interdependencies between individuals within the same household, our methods allow for a more holistic representation of household behavior. This is essential in transport modelling, where the activities of one household member often influence those of others (e.g. joint trips, school runs, coordinated commuting or car sharing within the household). Furthermore, these methods (SVAE-pop2 and MVAE-pop2) offer the same advantages as VAE, as discussed in the previous chapter, but applied to the two-layered model.

To promote reproducibility and practical use, the source code is publicly available and well-documented, facilitating adaptation to a variety of datasets and use cases.

The use of VAEs in generating synthetic populations has reached a new milestone by evolving from generating individuals or households to generating individuals within households. This is a significant step towards producing synthetic populations that more closely mirror real-world social structures and mobility patterns. The next step, as it will be discussed in Chapter 5, will be to incorporate aggregated data into the VAE generation process. This will further enhance the fidelity and robustness of the synthetic populations.

Chapter 5

Balancing Fidelity and Diversity: A Hybrid VAE+IPF Methodology for Synthetic Populations

Contents

Abstract **85**

5.1 Introduction **86**

5.2 Methodology **86**

5.2.1 Iterative Proportional Fitting (IPF) 86

5.2.2 Variational Autoencoder (VAE) 88

5.2.3 Model VAE+IPF 88

5.3 Case Study **90**

5.3.1 Datasets 90

5.3.2 Training and Assessments 91

5.3.3 Technologies introduced 92

5.4 Results **92**

5.4.1 Univariate evaluation of generated data 92

5.4.2 Multivariate evaluation of generated data 93

5.4.3 Sampling and Structural Zeros 95

5.5 Discussion **96**

5.6 Conclusion **97**

Abstract

Abstract

This chapter introduces a hybrid approach for synthetic population generation that combines the flexibility of Variational Autoencoders (VAEs) with the statistical rigor of the Iterative Proportional Fitting (IPF) procedure. The VAE model is used to learn the multidimensional structure of the data from a sample, while the IPF acts as a post-processing step to adjust the generated marginal distributions so that they align with external statistical constraints. This method aims to overcome the respective limitations of both approaches—namely, the VAE's inability to incorporate known marginals and the limited expressiveness of IPF in high-dimensional spaces.

Using several indicators to compare the synthetic data with the reference observations, including the SRMSE, MAE, and Pearson correlation coefficient, the empirical evaluation demonstrates that the VAE+IPF combination produces synthetic data that are coherent both univariately (marginal consistency) and multivariately (preservation of internal dependencies), while effectively limiting the generation of structural zeros.

Keywords : *synthetic population, Iterative Proportional Fitting (IPF), Variational Autoencoder (VAE), hybrid generative models, marginal constraints, structural zeros, sampling zeros, microsimulation, urban planning, transport modeling*

5.1 Introduction

The previous chapters have highlighted the relevance of deep generative models—particularly the Variational Autoencoder (VAE)—for the generation of synthetic populations. Thanks to their ability to model complex latent structures, VAEs have proven to be especially effective in contexts where the original sample is partial or limited in size (Borysov *et al.*, 2019a; Garrido *et al.*, 2020; Aemmer et MacKenzie, 2022; Sané *et al.*, 2025b). However, a major limitation of these models lies in their inability to directly incorporate aggregate constraints—such as known marginal data obtained from official sources (Garrido *et al.*, 2020; Sané *et al.*, 2025b,a).

In many application domains, such as mobility modeling, demographic microsimulation, or urban planning, practitioners have access to both individual-level samples and aggregate statistics derived from census data. For example, in France, the *Enquêtes Ménages Déplacements* (Household Travel Surveys, EMD) collect detailed information on a sample of households, while data from the National Institute of Statistics and Economic Studies (INSEE) provide marginal distributions for attributes such as age, sex, or occupational category at the municipal or departmental level.

The literature review (Section 2.2.1) also showed that classical iterative reweighting methods, such as the *Iterative Proportional Fitting* (IPF), make it possible to generate distributions that strictly comply with imposed marginals. These approaches are robust and well established, yet they present their own limitations, such as difficulties in handling high-dimensional data (Chapuis *et al.*, 2022; Yameogo *et al.*, 2021).

To combine the respective strengths of these two paradigms—the generative flexibility of the VAE and the constraining rigor of IPF—this chapter proposes a hybrid approach, referred to as VAE+IPF. The objective of this method is to generate realistic synthetic individuals while ensuring full consistency with known marginal distributions.

5.2 Methodology

5.2.1 Iterative Proportional Fitting (IPF)

The principle of *Iterative Proportional Fitting* (IPF) is based on the iterative adjustment of a prior probability distribution, typically derived from a contingency table observed in a sample. This adjustment relies on marginal data, often obtained from official sources, which serve as exogenous constraints. The objective is to produce a posterior probability distribution that strictly satisfies the imposed margins while remaining as close as possible to the initial distribution. In practice, IPF updates the cell counts of the contingency table so that their totals exactly match the target marginals, while minimizing the deviation from

the original counts (Deming et Stephan, 1940; Deville *et al.*, 1993; Yameogo *et al.*, 2021).

For example, Table 5.1 presents the cross-frequency modalities of the socio-professional category (in the rows) and the possession of a motor vehicle (in the columns). The prior distribution represents the observed sample frequencies in each cell.

Table 5.1: IPF example

A priori probability

Prof. \ Vehi.	0	1	2+
Agri	0.010	0.050	0.040
Craf	0.020	0.060	0.060
Exec	0.030	0.070	0.080
Inter	0.050	0.100	0.100
Emp	0.060	0.110	0.110
Work	0.040	0.080	0.080
Total	**0.21**	**0.47**	**0.47**
Goal	**0.20**	**0.30**	**0.50**

A posteriori probability

Prof. \ Vehi.	0	1	2+
Agri	0.017	0.026	0.043
Craf	0.025	0.038	0.063
Exec	0.032	0.049	0.080
Inter	0.036	0.054	0.090
Emp	0.045	0.066	0.109
Work	0.045	0.066	0.109
Total=Goal	**0.20**	**0.30**	**0.50**

IPF computes an adjusted distribution that satisfies these marginals while deviating minimally from the original distribution. Mathematically, this can be formulated as a constrained optimization problem, expressed in Equation 5.1.

$$p^{\text{post}} = \arg\min_{p} \sum_{i,j} p_{i,j} \ln\left(\frac{p_{i,j}}{p_{i,j}^{\text{prior}}}\right) \tag{5.1}$$

The solution corresponds to the distribution that minimizes the cross-entropy relative to the prior distribution, subject to the marginal constraints. These constraints are formalized in Equations 5.2 for the three marginals of the illustrative example.

$$\sum_{\text{prof}} p^{\text{post}}(v = 0 \cap \text{prof}) = 0.2 \tag{5.2}$$

$$\sum_{\text{prof}} p^{\text{post}}(v = 1 \cap \text{prof}) = 0.3 \tag{5.3}$$

$$\sum_{\text{prof}} p^{\text{post}}(v = 2 + \cap \text{prof}) = 0.5 \tag{5.4}$$

5.2.2 Variational Autoencoder (VAE)

As discussed in more detail in Section 3.2, the Variational Autoencoder (VAE) is a probabilistic generative model that learns a continuous latent representation of the data. It consists of two main components: an *encoder*, which projects input data into a lower-dimensional latent space, and a *decoder*, which reconstructs or generates new data from this latent representation (Kingma et Welling, 2013; Borysov *et al.*, 2019a). Training the VAE seeks to maximize the likelihood of the observed data while constraining the latent space to follow a multivariate normal distribution, thus facilitating sampling and synthetic data generation. This regularization is achieved by optimizing a loss function (Eq. 5.5) that combines reconstruction quality with the Kullback–Leibler divergence between the learned latent distribution and the target prior distribution (Kingma et Welling, 2013; Higgins *et al.*, 2017; Kingma et Welling, 2019).

$$\mathcal{L}(\phi,\theta) = \sum_{a=1}^{N} \underbrace{\sum_{k=1}^{K}\sum_{d=1}^{D_k} x_a^{kd} \log \hat{x}_a^{kd}}_{\text{Reconstruction cost}} + \underbrace{\beta \times \mathbb{KL}(\mathcal{N}(\mu,\sigma)\|\mathcal{N}(\mathbf{0},\mathbf{I}))}_{\text{KL cost}} \tag{5.5}$$

where N is the number of agents (individuals or households), K is the number of attributes, and D_k is the number of modalities for attributes k. μ (mean) and σ (standard deviation), both of size m constitutes the encoder outputs.

As mentioned in the previous chapter, the first term accounts for the *reconstruction error*, which measures how well the VAE reproduces the input data X in its output $\hat{X}$. The second term, a *regularization term* represented by the *Kullback-Leibler (KL) divergence*, quantifies how much the learned latent space distribution deviates from the standard Gaussian prior $\mathcal{N}(\mathbf{0},\mathbf{I})$. The hyperparameter $\beta \in \mathbb{R}_+$ serves as weighting coefficients, intended to balance the trade-off between reconstruction accuracy and regularization.

5.2.3 Model VAE+IPF

The proposed method relies on the sequential combination of two processes, as illustrated in Figure 5.1. The first step consists in generating synthetic data using a *Variational Autoencoder* (VAE), while the second applies the *Iterative Proportional Fitting* (IPF) procedure to adjust marginal distributions and ensure consistency with external data.

The VAE+IPF approach aims to reconcile two often conflicting objectives: faithfulness to the multivariate structures present in the model's training data and strict compliance with known marginal distributions.

Let $\hat{p}_{\text{VAE}}(x)$ denote the empirical distribution of individuals generated by the VAE, ob-

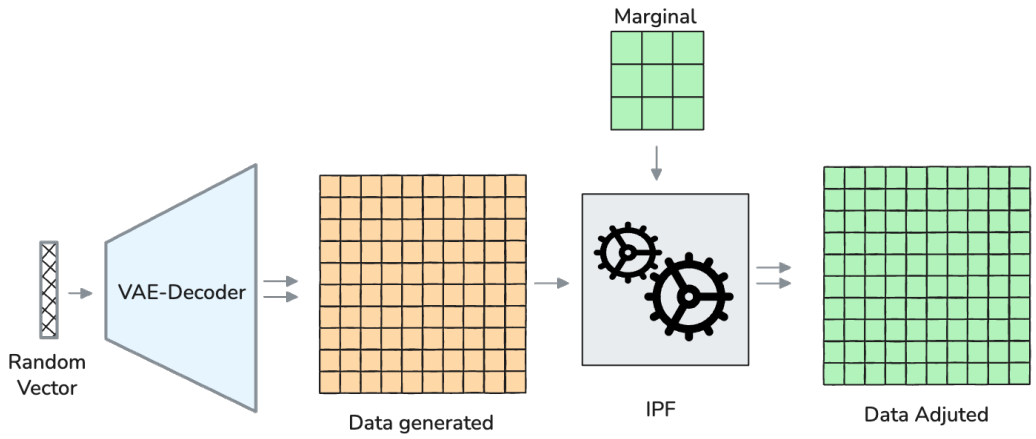


Figure 5.1: Architecture of the VAE+IPF method.

The figure illustrates the two main steps of the process: (1) generation of synthetic data from the VAE latent space via the decoder; (2) adjustment of marginal distributions through the Iterative Proportional Fitting (IPF) procedure, ensuring consistency with observed marginals.

tained by sampling from the latent space and decoding the resulting representations. This distribution, unconstrained by external information, may diverge from official marginal data. The application of IPF therefore consists in projecting $\hat{p}_{\text{VAE}}(x)$ onto the set $\mathcal{P}_{\text{marg}}$ of distributions satisfying the marginal constraints $\{m_i\}_{i=1}^{K}$, by minimizing the Kullback–Leibler divergence:

$$p^* = \arg\min_{p \in \mathcal{P}_{\text{marg}}} \mathbb{D}_{\text{KL}}(p \,\|\, \hat{p}_{\text{VAE}}) \tag{5.6}$$

$$= \arg\min_{p \in \mathcal{P}_{\text{marg}}} \sum_{x} p(x) \ln\left(\frac{p(x)}{\hat{p}_{\text{VAE}}(x)}\right) \tag{5.7}$$

where:

- p^* denotes the adjusted distribution;
- $\mathbb{D}_{\text{KL}}(p \,\|\, \hat{p}_{\text{VAE}})$ is the Kullback–Leibler divergence;

- $p(x)$ refers to the adjusted distribution to be estimated;
- $\hat{p}_{\text{VAE}}(x)$ is the unconstrained distribution generated by the VAE;
- $\mathcal{P}_{\text{marg}}$ represents the set of all distributions satisfying the marginal constraints.

In practice, the VAE is first trained on the available data to learn a compact latent representation, and then used to generate a synthetic population that reproduces the observed multivariate dependencies. In the second step, the IPF procedure adjusts the weights associated with the generated individuals so that marginal totals (e.g., by age, sex, or education) match those from external statistical sources. Thus, the VAE ensures internal coherence and diversity among individuals, while IPF guarantees global conformity with observed constraints.

5.3 Case Study

5.3.1 Datasets

As in the previous chapters, the case study presented here aims to illustrate the application of the proposed method using a real-world dataset. It is based on the 2014 Household Travel Survey (*Enquête Ménages Déplacements*, EMD) conducted in Loire-Atlantique, France (Département-Loire-Atlantique, 2015). About 29 500 persons have been interviewed. The analysis relies on a set of both household and individual-level attributes, including variables of categorical and numerical nature. Table 5.2 provides the detailed list: a total of 15 attributes were selected, of which 11 are categorical and 4 numerical.

Based on insights from the literature, we identified the most frequently used attributes in transportation modeling, which form the basis of this application (Bouzouina *et al.*, 2021; Plaut, 2006; Sun et Zacharias, 2021; Brückmann *et al.*, 2021; Wang et Cao, 2017; Wang *et al.*, 2022). At the household level, six attributes were selected, including three categorical and three numerical variables, such as House type, housing tenure, number of household members, and number of vehicles. At the individual level, nine attributes were retained, comprising eight categorical and one numerical variable, including age, relationship to the household reference person, gender, and socio-professional category.

Although this study is based on the EMD dataset, the proposed methodology is fully transferable to other data sources with a comparable *Individuals* × *Attributes* structure, provided that external marginals are available for adjustment.

Table 5.2: Overview of household and individual features

Groups	Attributes	Type[1]	Description
Household (HH)			
	house_type	cat(5)	House type
	house_occupation_type	cat(4)	House occupation type
	household_size	num	Number of persons in the HH
	number_of_vehicles	num	Number of vehicles in the HH
	number_of_bikes	num	Number of bicyles in the HH
	has_internet	cat(2)	Has internet access ?
Individuals			
	link_ref_person	cat(5)	Link with the HH reference person
	age	num	Age
	sex	cat(2)	Gender
	is_adolescent	cat(2)	Is adolescent ?
	school_level	cat(5)	School level
	employed	cat(2)	Is employed ?
	studies	cat(2)	Is a student ?
	socioprofessional_class	cat(9)	Socio-professional status
	has_license	cat(2)	Has a driver's license ?

[1] Indicates whether the attribute is numerical (num) or categorical (cat). When the attribute is categorical, the number in parentheses represents the number of possible values.

5.3.2 Training and Assessments

The dataset was divided into two parts: a test set representing 20% of the observations, and a model-building set (80%), which was itself split into two subsets—70% for model training and 30% for validation. In this configuration, the characteristics of the test set are considered to represent those of the real population and therefore serve as the benchmark for all evaluations. The results obtained from the generated data (using either the VAE alone or the VAE combined with IPF) are compared to the distributions derived from this test set, both at the univariate level (marginal distributions) and the multivariate level (cross-distributions).

The VAE training process follows the same principles as those presented in Chapter 3. Regarding the adjustment phase using IPF, the marginal data are derived from the test dataset, and the choice of attributes is based on previous studies highlighting their importance in household structure and mobility dynamics (Brückmann *et al.*, 2021; Wang et Cao, 2017; Wang *et al.*, 2022). The attributes selected for adjustment are the *number of vehicles* within the household, *gender*, *age*, and *socio-professional category*.

In addition to the evaluation metrics introduced in Section 3.4.3, notably the Stan-

dardized Root Mean Square Error (SRMSE) and the Pearson Correlation Coefficient, the Mean Absolute Error (MAE), which quantifies the average difference between observed and estimated frequencies, is also used. Its formulation is given by Equation 5.8.

$$\text{MAE} = \frac{1}{N} \sum_{k=1}^{N} |\pi_k - \hat{\pi}_k| \tag{5.8}$$

where π_k represents the observed frequency for modality k, $\hat{\pi}_k$ the estimated frequency, and N the total number of modalities considered.

5.3.3 Technologies introduced

All stages of the project—from data preprocessing to model validation—were performed in Python, mainly leveraging the Pandas (v1.5), NumPy (v1.23), and TensorFlow (v2.10) libraries. Visualizations were generated using Plotnine (v0.10). Both the dataset and the complete source code are available on Gitlab[1] for readers interested in reproducing or extending the model. On average, the training process for a single model requires approximately 30 seconds. The experiments were conducted on a workstation featuring an 11th-generation Intel Xeon processor, 128 GB of RAM, and an NVIDIA RTX A5000 GPU.

5.4 Results

This section presents an empirical evaluation of the proposed method by comparing the performance of the standalone VAE model with that of the combined VAE+IPF approach. The analysis is structured along three complementary dimensions: a univariate comparison, a multivariate comparison, and an assessment of sampling zeros and structural zeros.

5.4.1 Univariate evaluation of generated data

The univariate analysis examines the ability of both approaches to accurately reproduce the observed marginal distributions. It also allows us to assess the impact of applying IPF on attributs that are not directly constrained.

Figure 5.2 illustrates the quality of the marginal adjustment for all attributes, before and after applying IPF. Each point represents a modality of an attribute: the x-axis indicates the frequency observed in the real data, while the y-axis shows the corresponding frequency in the generated data. Points lying on the dashed diagonal indicate a perfect match between the two distributions. Yellow points correspond to frequencies from VAE-generated data, while green points represent those obtained after adjustment with IPF.

[1] https://gitlab.univ-eiffel.fr/population-synthesis/vae-vector-emd

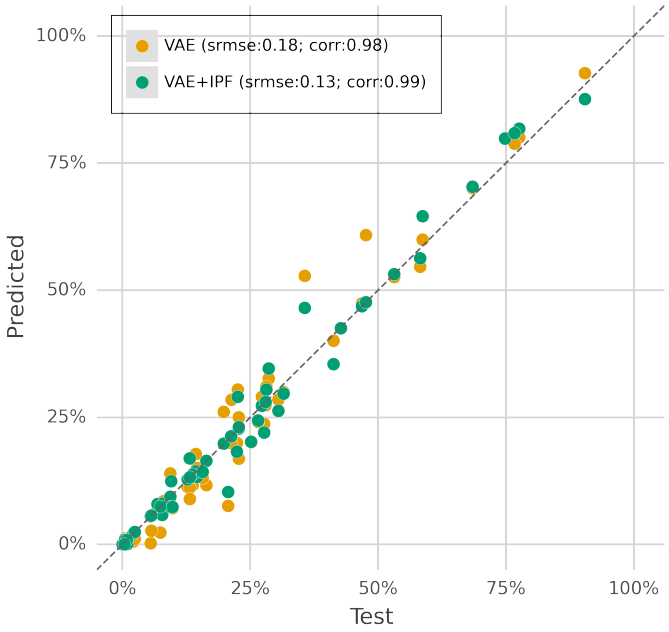


Figure 5.2: Goodness of fit of the univariate distributions for all attributes.

Visually, the green points are generally closer to the diagonal than the yellow ones, indicating improved adjustment. This is confirmed by quantitative indicators: the Standardized Root Mean Squared Error (SRMSE) decreases from 0,18 for the VAE-generated data to 0,13 after IPF, representing an improvement of 27,8%. The Pearson correlation coefficient, which measures the strength of the linear relationship between observed and simulated frequencies, increases slightly from 0,98 to 0,99. It should be noted, however, that this figure also includes attributes explicitly constrained during the IPF procedure, which may influence the overall assessment of the adjustment.

Figure 5.4 focuses specifically on the attributes not constrained by IPF, thereby allowing an evaluation of its indirect effect on these variables. A similar pattern is observed, although the improvement is more moderate. For these attributes only, the SRMSE decreases from 0,15 before adjustment to 0.14 after IPF, corresponding to a 6,7% improvement. The Pearson correlation coefficient remains stable at 0,99, indicating a consistently strong overall coherence.

5.4.2 Multivariate evaluation of generated data

The multivariate evaluation aims to assess the extent to which relationships between variables are preserved or altered in the generated synthetic data, particularly following the

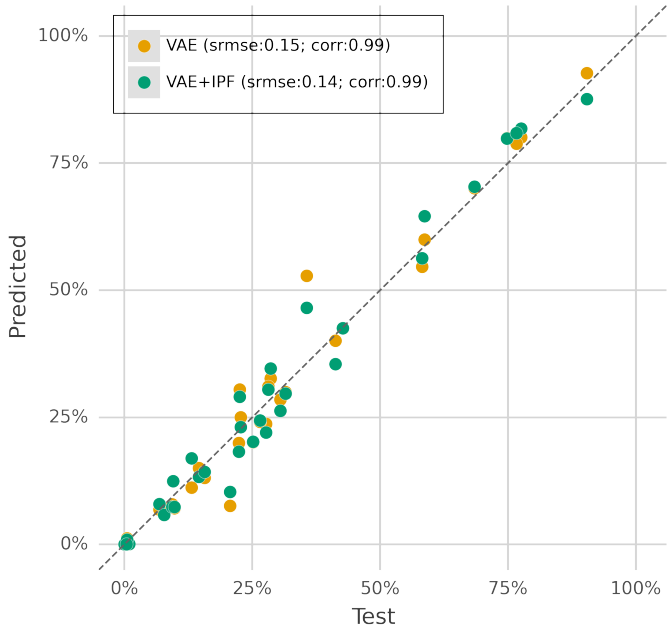


Figure 5.3: Goodness of fit of the univariate distributions for attributes not included in the IPF procedure.

IPF adjustment. Unlike the univariate analysis, which focuses on marginal distributions, this approach examines cross-combinations of attribute modalities, reflecting the structural interactions between variables.

For this purpose, six variables were selected: *house type*, *internet access*, *gender*, *age*, *education level*, and *socio-professional class*, yielding a total of approximately 5 400 cross-modalities. This selection is grounded in previous research in transportation behavior modeling, which has highlighted the key role of these variables in shaping lifestyle patterns and mobility practices (Bouzouina *et al.*, 2021; Plaut, 2006; Sun et Zacharias, 2021; Brückmann *et al.*, 2021; Wang et Cao, 2017; Wang *et al.*, 2022).

Figure 5.4 shows the quality of the joint distribution reproduction obtained from the synthetic data generated by the VAE (in yellow) and by the VAE+IPF combination (in green). Each point in the scatterplot represents one cross-modality among the six selected variables. The x-axis indicates the observed proportion of each combination in the test dataset, while the y-axis corresponds to the proportion obtained in the synthetic data. Most points align closely around the diagonal, indicating good consistency between the observed and generated distributions. The Pearson correlation coefficient reaches 0,93 for both approaches, suggesting a satisfactory overall structural fidelity. However, the SRMSE is slightly lower for the standalone VAE (1,47) than for the VAE+IPF model (1,52).

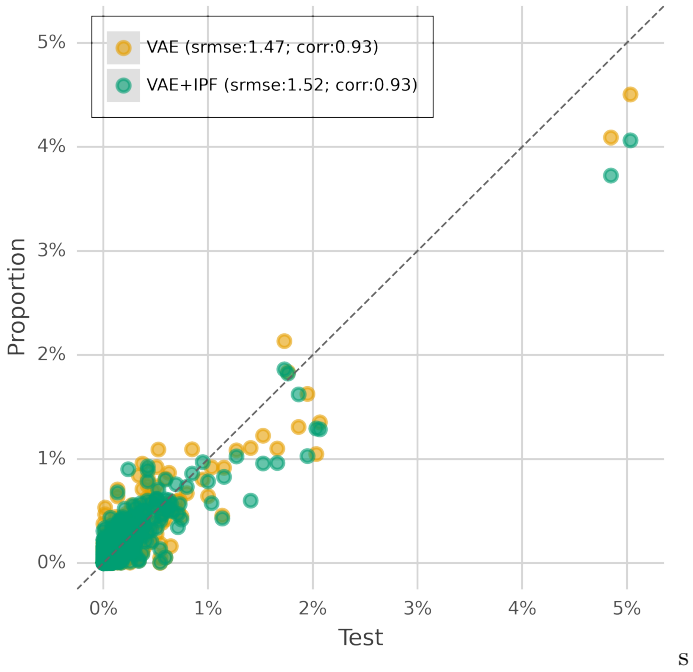


Figure 5.4: Goodness of fit of the multivariate distributions for attributes : *house type*, *internet access*, *gender*, *age*, *education level*, and *socio-professional class*

5.4.3 Sampling and Structural Zeros

Another key criterion for assessing generative models lies in their ability to correctly handle *sampling zeros* and *structural zeros*, as introduced in Section 3.6.2. In this analysis, we rely on the same six attributes as in the previous section. The resulting combinatorial space includes approximately 5 400 cross-modalities, of which about 2% correspond to sampling zeros and 78% to structural zeros.

Figure 5.5 presents the proportions of structural and sampling zeros in the data generated by the VAE alone and after post-processing with IPF. Yellow bars represent the results from the VAE, while green bars indicate those obtained after applying IPF.

For sampling zeros, the VAE generates around 35%. After applying IPF, this proportion decreases slightly to about 34%. Regarding structural zeros, the VAE produces approximately 12%, which is reduced to about 11% following IPF adjustment. Expressed in relative terms, these decreases correspond to a reduction of about 2% for sampling zeros and 8% for structural zeros.

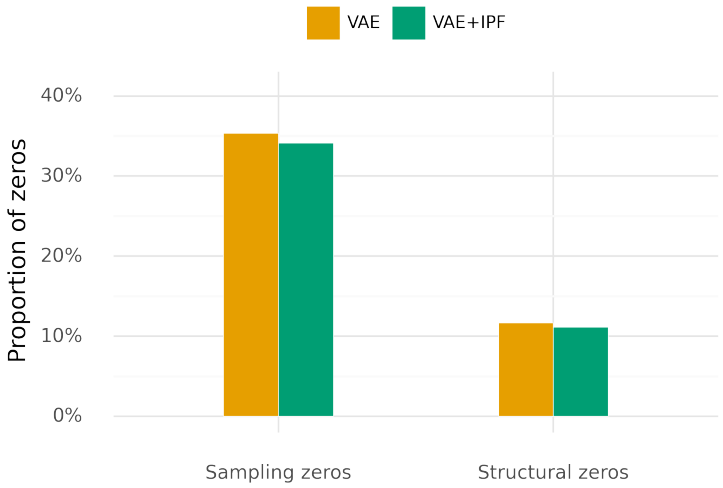


Figure 5.5: Proportion of zero-cells before and after the IPF procedure.

5.5 Discussion

The results highlight the combined ability of the VAE and the IPF procedure to produce coherent synthetic data that respect marginal constraints while limiting the generation of structural zeros. The evaluation of both household- and individual-level attributes shows that the standalone VAE successfully captures the multidimensional structures present in the original data, as reflected by the high correlation coefficients observed between real and simulated frequencies (0.98 for aggregated attributes and 0.99 for individual attributes).

However, this reconstruction quality alone does not ensure that the observed marginal distributions are properly reproduced. In this regard, the IPF plays a decisive role. By imposing constraints on selected margins, it significantly improves the overall adjustment, as evidenced by a substantial error reduction (SRMSE decreasing from 0.18 to 0.13, i.e., a 28% improvement). Interestingly, variables not directly included in the adjustment retain statistical properties similar to those obtained before applying IPF, confirming that the procedure does not introduce major distortions in unconstrained attributes; on the contrary, it slightly enhances their consistency (SRMSE decreasing from 0.15 to 0.14, corresponding to a 6.7% improvement).

Regarding the analysis of joint distributions, the results demonstrate good preservation of multivariate structures in both configurations. The standalone VAE accurately reproduces complex interactions among variables, achieving a Pearson correlation of 0.93 and an SRMSE of 1.47. The inclusion of IPF does not significantly degrade this performance: the correlation

remains stable at 0.93, and the SRMSE increases marginally by about 3%, reaching 1.52. This slight deviation can be explained by the fact that IPF adjusts univariate frequencies without explicitly accounting for inter-variable dependencies, which may slightly alter certain cross-structures. Nonetheless, this minor loss in multivariate precision appears to be an acceptable trade-off given the improvement in marginal consistency. Further analyses on additional case studies would be required to confirm this finding.

Another important aspect concerns the treatment of structural and sampling zeros. By design, the VAE's generative nature introduces diversity into the synthetic data by producing combinations not present in the training sample. This capability is valuable for addressing the issue of sampling zeros, as it ensures better coverage of the space of plausible combinations. However, it also leads to the creation of structural zeros—combinations that are impossible or non-existent in reality. Integrating the IPF helps reduce these occurrences slightly (an 8% decrease in structural zeros), though it does not eliminate them entirely. The proportion of sampling zeros remains relatively stable, showing only a slight decrease of about 2% after adjustment.

These initial results will need to be confirmed on case studies involving larger populations and model building datasets that are smaller in proportion than in the proposed example. The challenge is to create realistic test cases of this nature.

5.6 Conclusion

This study proposed a hybrid approach combining a generative model—the Variational Autoencoder (VAE)—with a margin adjustment procedure, the Iterative Proportional Fitting (IPF), for generating synthetic populations that are coherent both in terms of internal structure and external marginal distributions. The results demonstrate that the VAE effectively captures the multidimensional structure of the data, while the IPF successfully aligns the generated population with known marginals and slightly improves the consistency of unconstrained variables. Moreover, this combined approach reduces the occurrence of structural zeros while maintaining diversity among sampling zeros.

Looking ahead, directly integrating marginal constraints into the generative learning process represents a promising direction to avoid post-hoc adjustments while ensuring conformity with external distributions. Nevertheless, the findings reported here would benefit from further validation through additional case studies to confirm their robustness and generalizability.

Chapter 6

Deep Generative Modeling of Activity Plans: Preliminary Insights with CVAE and ActivityGPT Architectures

Contents

Abstract **100**

6.1 Introduction **101**

6.2 Activity Plan Modeling **101**

6.2.1 Formalization and Notation 101

6.2.2 Activity Plan Modeling 102

6.3 Conditional Variational Autoencoder (CVAE) **103**

6.4 GPT for Activities Synthesis **105**

6.4.1 Word Embedding and Positional Encoding/Embedding 105

6.4.2 Transformers 107

6.4.3 ActivityGPT model 110

6.5 Case study **113**

6.5.1 Datasets 113

6.5.2 Evaluation Metrics 113

6.6 Implementation and Preliminary Analysis **115**

6.6.1 Implemented Models 116

6.6.2 Technological Framework 116

6.6.3 A Priori Analysis of the Different Approaches 116

6.7 Conclusion **118**

Abstract

Abstract

This chapter explores preliminary work on the generation of daily activity plans using deep generative models. Activity plans describe an individual's sequence of daily activities (e.g., work, school, shopping, leisure) together with the corresponding travel modes, forming a important input for microsimulation and urban mobility modeling. While recent advances in generative deep learning have improved population synthesis, few studies have addressed the direct generation of activity plans.To provide additional insight, two approaches were investigated: a Conditional Variational Autoencoder (CVAE) to generate activity plans conditioned on socio-demographic attributes, and an adaptation of the Generative Pre-trained Transformer (GPT) architecture—here termed ActivityGPT—to model activity sequences as temporally dependent data.

Four model configurations were proposed by combining these two architectures (CVAE and ActivityGPT) with two alternative activity representations: TimeGrid, a fixed temporal discretization of the day, and SeqRep, an explicit sequential encoding of activities. The resulting configurations—CVAE+TimeGrid, CVAE+SeqRep, ActivityGPT+TimeGrid, and ActivityGPT+SeqRep—were analyzed conceptually to assess their potential performance, computational requirements, and representational suitability.

The preliminary analysis suggests a clear hierarchy of expected performance. The ActivityGPT+SeqRep configuration appears the most promising, as the sequential representation aligns naturally with the autoregressive nature of GPT models, enabling them to capture both short- and long-term dependencies between activities. The CVAE+TimeGrid configuration is also coherent, offering a good trade-off between structural simplicity and expressiveness, though it may suffer from heavy computational loads due to long input vectors. The CVAE+SeqRep model, while less expressive, is computationally lighter and easier to train. Finally, ActivityGPT+TimeGrid is deemed the least appropriate, since the repetitive structure of TimeGrid conflicts with the autoregressive design of GPT, resulting in potential inefficiencies during training.

Keywords : *activity-based modeling, synthetic population, activity plan generation, Conditional Variational Autoencoder (CVAE), Generative Pre-trained Transformer (GPT), ActivityGPT, sequential data modeling, urban mobility, travel behavior, deep generative models, temporal dependencies, microsimulation*

6.1 Introduction

The synthesis of activity plans constitutes a key step in the detailed modeling of individual mobility dynamics. Beyond socio-demographic description, it enables the generation of temporal sequences of actions for each individual (e.g., work, school, shopping, leisure), along with their corresponding travel modes. Such plans are essential for simulating transport flows, evaluating mobility policies, and assessing the impact of new infrastructures.

The literature review (Chapter 2) shows that, despite the rise of deep generative methods in many domains, very few studies have addressed the direct generation of activity plans using deep learning models. One of the few references in this area is Shone et Hillel (2025), who employed a VAE to generate activity plans. However, this approach presents notable limitations, particularly its inability to account for contextual attributes of individuals—such as education level, household structure, or employment status—which are known to influence travel motivations and behavioral patterns.

To address this gap, this chapter introduces two original approaches for synthesizing activity plans conditioned on socio-demographic and economic attributes:

- a *Conditional Variational Autoencoder* (CVAE), which generates activity plans by conditioning the latent sampling process on individual attributes such as age, gender, and professional category;
- an adaptation of the *Generative Pre-trained Transformer* (GPT) architecture to model activity plans as temporally dependent sequences. Transformers are well known for their ability to capture both short- and long-term dependencies in sequential data, making them promising candidates for generating realistic and dynamic activity plans (Vaswani *et al.*, 2017; Borisov *et al.*, 2022).

The remainder of this chapter is organized into four parts. First, the notion of an activity plan is formalized, and the main modeling challenges are identified. The proposed architecture of the CVAE is then described, with a detailed specification of the latent conditioning mechanism. The following section presents the adaptation of a GPT model for activity plans synthesis. Next, the case study and evaluation metrics used are presented. The final section presents the implementation and preliminary analysis.

6.2 Activity Plan Modeling

6.2.1 Formalization and Notation

An activity plan A is defined as an ordered sequence of T activities:

$$A = (a_1, a_2, \ldots, a_T) \tag{6.1}$$

Each activity a_t is characterized by a vector of attributes:

$$a_t = (x_{1;t}\ x_{2;t}\ \ldots\ x_{K;t}) \tag{6.2}$$

where $x_{k;t}$ denotes the k-th attribute of activity a_t, and K is the total number of attributes per activity. In the context of mobility modeling, typical attributes include the *activity type* (e.g., work, school, leisure), the *mode of transport*, and the *time-related variables* (e.g., start time, end time, duration).

6.2.2 Activity Plan Modeling

Problem statement:
In the dataset, each individual i is associated with an activity plan $A^{(i)} = (a_1^{(i)}, \ldots, a_{T_i}^{(i)})$, whose length T_i varies from one person to another ($T_i \neq T_j$ in general). This variability in sequence length poses a practical challenge for neural models, which most often operate on tensors of fixed size.

Two common representation schemes are typically used to address this constraint while preserving the temporal structure: (i) the *discretized 24-hour representation* (time grid, hereafter referred to as *TimeGrid*), and (ii) the *fixed-length sequential representation* with padding (hereafter referred to as *SeqRep*) (Arentze *et al.*, 2007; Liu *et al.*, 2015; Saadi *et al.*, 2016a; Namazi-Rad *et al.*, 2017; Hörl et Balac, 2021b; Shone et Hillel, 2025).

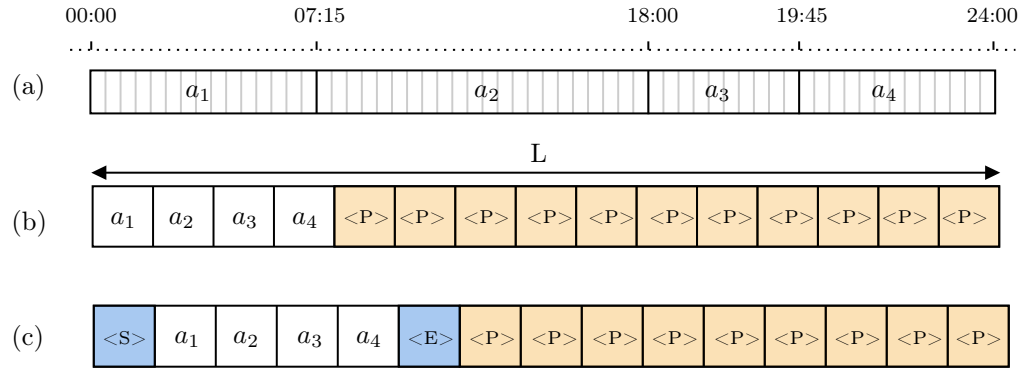


Figure 6.1: Illustration of the two common representations of an activity plan: TimeGrid (a) and SeqRep (b and c).

6.2.2.1 Option A — Discretized 24-hour Representation (TimeGrid)

Principle:
The daily time horizon is discretized into H regular intervals (e.g., $H = 96$ slots of 15 minutes each), and for every slot, the state of the individual during that period is specified. Figure 6.1.a illustrates this modeling scheme. For instance, activity a_1, which starts at *00:00* and ends at *07:15*, consists of 29 time slots of 15 minutes each.

Advantages / Limitations:
TimeGrid provides a fixed-size input/output tensor (independent of the number of activities). However, the discretization introduces a *resolution bias* (quantization of time intervals) and redundancy for long-duration activities.

6.2.2.2 Option B — Sequential Representation (SeqRep)

Principle:
A maximum sequence length L is defined for activity plans. Then, a padding operation, denoted Pad_L, extends any sequence $A = (a_1, a_2, \ldots, a_T)$ of length $T \leq L$ to length L by appending a special padding token `<P>` at the end of the sequence (see Figure 6.1.b). In practice, for sequential models, special start and end tokens — `<S>` and `<E>` — are inserted before the padding operation (see Figure 6.1.c). These tokens explicitly delimit the sequence and guide the model during generation by indicating when to start and stop the prediction.

Advantages / Limitations:
SeqRep faithfully preserves the temporal order of events and is well suited for autoregressive architectures (e.g., Transformers). However, the choice of L implies a trade-off: a small L may truncate plans, while a large L increases both *padding* and computational cost. Moreover, modeling continuous attributes requires either discretization (risk of resolution bias) or probabilistic output modeling.

6.3 Conditional Variational Autoencoder (CVAE)

The *Conditional Variational Autoencoder* (CVAE) is a direct extension of the classical VAE model, originally introduced by Sohn *et al.* (2015) to explicitly incorporate conditional attributes into the generative process. While the standard VAE aims to model the joint distribution $p(x \mid z)$ between the observed data x and the latent variables z, the CVAE seeks to model the conditional distribution $p(x \mid c, z)$, where c represents a set of contextual or explanatory attributes. In the context of activity plan generation, x corresponds to the activity plan A, and the conditional attributes may include socio-economic characteristics of the individual (e.g., age, gender, occupational category), as well as spatial factors such as home location or accessibility conditions.

The architecture of the CVAE, illustrated in Figure 6.2, is based on two main neural networks, similar to those of the standard VAE:

- a **conditional encoder** $q_\phi(z \mid x, c)$, which projects the observed data x and the contextual attributes c into a continuous latent space by learning the parameters of the Gaussian distribution (μ, σ);
- a **conditional decoder** $p_\theta(x \mid z, c)$, responsible for reconstructing the data from the latent vector z and the conditioning variables c.

Thus, unlike the standard VAE, where generation depends solely on the latent variable z, the CVAE explicitly conditions the reconstruction process on contextual attributes.

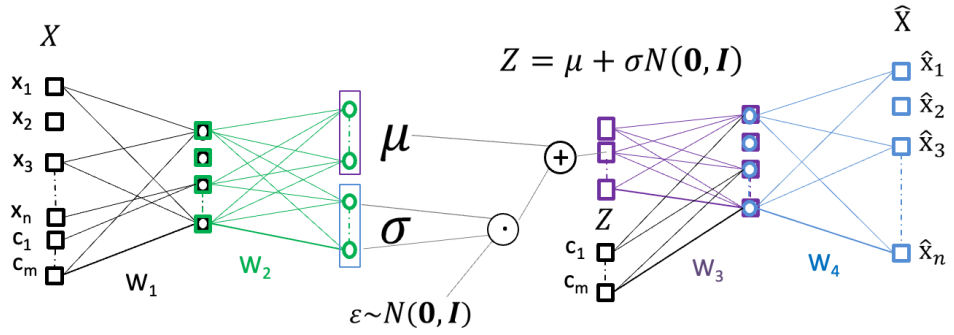


Figure 6.2: Conceptual diagram of the Conditional Variational Autoencoder (CVAE).

The loss function of the CVAE retains the canonical structure of the standard VAE but integrates the conditioning variables c into both the encoding and decoding distributions. It is defined as follows:

$$\mathcal{L}(\phi, \theta) = \sum_{a=1}^{N} \underbrace{\sum_{k=1}^{K} \sum_{d=1}^{D_k} x_a^{kd} \log \hat{x}_a^{kd}}_{\text{Reconstruction cost}} + \underbrace{\beta \times \mathbb{KL}(\mathcal{N}(\mu, \sigma) \| \mathcal{N}(\mathbf{0}, \mathbf{I}))}_{\text{KL cost}} \tag{6.3}$$

The first term in Equation 6.3, referred to as the *reconstruction cost*, measures how accurately the reconstruction $\hat{x}$ matches the real data x, typically through a cross-entropy or mean squared error loss depending on the variable type. The second term, the *KL cost*, regularizes the learned latent distribution $q_\phi(z \mid x, c)$ by bringing it closer to a standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The coefficient β controls the trade-off between reconstruction accuracy and latent space regularization (Higgins *et al.*, 2017). Once the model has been trained, the generation of an individual's activity plan, conditioned on their contextual attributes, is performed by feeding these attributes into the decoder to produce the

corresponding activity sequence.

The main advantage of the CVAE in the context of activity plan generation lies in its ability to produce sequences that are coherent with specific individual profiles. For example, a working adult living in a suburban area is more likely to generate a plan including a motorized home–work trip, while an urban student will tend to produce a plan characterized by shorter trips and greater diversity of activities.

6.4 GPT for Activities Synthesis

Generative Pre-trained Transformer (GPT) models represent a class of autoregressive approaches based on the *Transformer* architecture, originally proposed by Vaswani *et al.* (2017). Their ability to model both short- and long-term sequential dependencies, without relying on recurrent structures, makes them particularly well suited for the generation of sequences such as activity plans.

Unlike variational models such as the VAE or CVAE, which rely on an explicit probabilistic modeling of the latent space, GPT models learn the conditional distribution $p(a_t \mid c, a_{<t})$ directly from large volumes of observed sequences. In other words, the activity a_t at time t is predicted as a function of the socio-economic characteristics c and the preceding activities $a_{<t}$.

This section introduces the fundamentals of GPT models, beginning with word embedding and transformers, before presenting the adaptation of the GPT model for generating activity plans.

6.4.1 Word Embedding and Positional Encoding/Embedding

In traditional data preprocessing, categorical attributes are often transformed into numerical representations using methods such as *Label Encoding* or *One-Hot Encoding*. While these techniques allow categorical data to be processed by machine learning models, they present significant limitations when applied to sequential data such as activity plans.

Label Encoding assigns an arbitrary integer to each category, implicitly introducing an artificial ordinal relationship (e.g., *work* = 1, *shopping* = 2, *home* = 3), which has no semantic meaning. Conversely, *One-Hot Encoding* represents each category as a sparse, high-dimensional vector with all components equal to zero except one. Although this avoids the problem of artificial ordering, it leads to two major issues:

- *High dimensionality:* the vector size grows linearly with the vocabulary, quickly becoming inefficient when dealing with numerous activity types, modes, or time slots;

- *Absence of semantic relationships:* all vectors are orthogonal and equidistant, meaning that *work* and *school*—which often share similar temporal and behavioral contexts—are treated as completely unrelated.

These limitations prevent traditional encodings from capturing structural and semantic regularities within sequences of activities.

To overcome these drawbacks, deep learning approaches rely on **word embeddings**, which map each discrete symbol to a dense, continuous vector in a latent space of reduced dimension ($d_{\text{model}} \ll |V|$). Instead of manually defining categories, embeddings are *learned* from data: the model automatically organizes the latent space so that elements appearing in similar contexts are close to each other.

Formally, in the context of activity plan synthesis, each activity a_t is described by a set of categorical attributes (e.g., activity type, transport mode, duration, etc.). These attributes are encoded, concatenated, or linearly projected to form a single vector representation:

$$\mathbf{e}_t = \text{WordEmbed}(a_t) \in \mathbb{R}^{d_{\text{model}}} \tag{6.4}$$

The embedding layer corresponds to a learnable matrix $\mathbf{E} \in \mathbb{R}^{|V| \times d_{\text{model}}}$, where each row represents the latent vector of one category (or token) in the vocabulary. During training, this matrix is optimized jointly with the model parameters to minimize prediction error. Through gradient-based learning, the embedding space self-organizes to reflect semantic and functional similarities: activities occurring in comparable temporal or modal contexts (e.g., *work* and *school*) converge toward neighboring regions in the vector space.

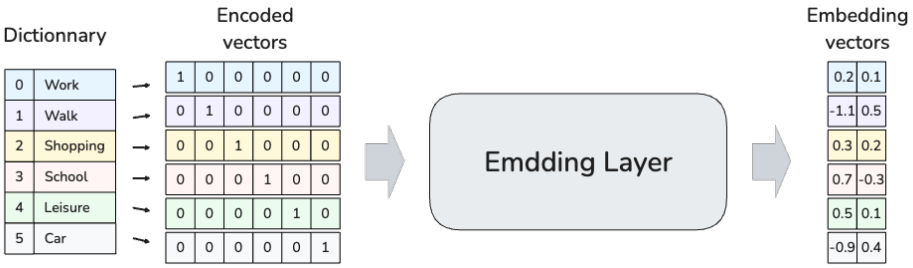


Figure 6.3: Illustration of the embedding process. Categorical activity attributes are projected into a continuous latent space through word embeddings

However, standard embeddings do not encode positional information: two identical activities occurring at different times of day are represented by the same vector. Yet, in activity sequences, temporal ordering is essential for behavioral consistency: the sequence *work* → *shopping* → *home* differs fundamentally from *shopping* → *work* → *home*.

To address this, Vaswani *et al.* (2017) introduced **positional encodings**, which add deterministic sinusoidal functions to embeddings, injecting information about relative positions while allowing generalization to unseen sequence lengths. Later, Shaw *et al.* (2018) proposed **positional embeddings**, in which positions are represented by learnable vectors rather than predefined ones. This approach enables the model to capture more complex and non-linear positional dependencies—particularly relevant for structured sequences such as daily activity plans.

The resulting representation combines semantic and positional information:

$$\mathbf{h}_t = \mathbf{e}_t + \mathbf{p}_t \tag{6.5}$$

6.4.2 Transformers

The core of the GPT model relies on a stack of *Transformer* blocks, each composed of three main sub-layers: (1) a *masked multi-head attention* layer, (2) a *normalization and residual connection* layer, and (3) a *feed-forward* layer. Figure 6.4 presents the architecture and illustrates the internal structure of a *Transformer* block, centered on the attention mechanism.

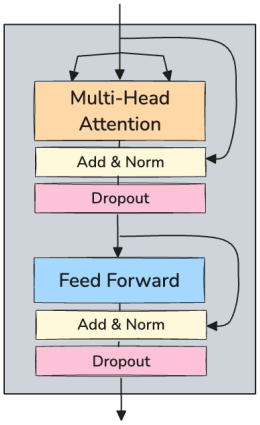


Figure 6.4: Internal structure of a *Transformer* block, including the masked multi-head attention and feed-forward network.

(1) Masked Multi-Head Attention Layer

The key idea behind the attention mechanism is to allow the model, for each position t, to determine which parts of the contextual features c and of the preceding sequence $(a_1, \ldots, a_{t-1})$ are most relevant for predicting the next activity a_t.

Each input vector $\mathbf{h}_t$ is linearly projected into three distinct subspaces using weight matrices learned during model training:

$$\mathbf{Q} = \mathbf{H}\mathbf{W}_Q \tag{6.6}$$

$$\mathbf{K} = \mathbf{H}\mathbf{W}_K \tag{6.7}$$

$$\mathbf{V} = \mathbf{H}\mathbf{W}_V \tag{6.8}$$

where:

- **Q** (**Query**) represents the query, i.e., the question posed by the current position regarding the context;
- **K** (**Key**) represents the keys, i.e., the identifiers of the information available within the sequence;
- **V** (**Value**) represents the values associated with these keys, i.e., the actual content to be aggregated.

The attention mechanism then computes a similarity score between each pair (Q_t, K_j) to determine how much position t should attend to position j. This score is normalized using a *softmax* function and applied to the corresponding values, yielding the attention output denoted as Z:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = Z = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \tag{6.9}$$

where d_k is the dimensionality of the key latent space, used to stabilize the gradients during training.

In the case of **multi-head attention**, several sets of matrices $(\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V)$ are used in parallel, each learning to capture a different type of dependency between activities (e.g., temporal relationships, recurrent transport modes, or behavioral transitions). The outputs from these multiple attention heads Z_i are then concatenated and projected back into the original latent space, producing the final output Z_O:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = Z_O \tag{6.10}$$

$$= \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\mathbf{W}_O \tag{6.11}$$

$$= \text{Concat}(Z_1, \ldots, Z_h)\mathbf{W}_O \tag{6.12}$$

Finally, a **masking** operation (*masked self-attention*) is applied to ensure that position t cannot access information from future positions ($t' > t$). This guarantees that, when predicting an activity at time t, the model relies only on past activities, thereby reproducing the sequential logic of a typical day. This constraint is essential to preserve the autoregressive nature of the model.

The diagram in Figure 6.5 summarizes these steps: linear projection into the Query–Key–Value spaces, parallel computation of multi-head attention, and application of the causal mask.

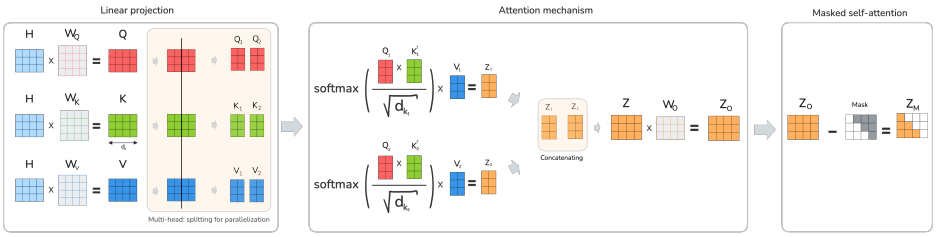


Figure 6.5: Multi-head attention mechanism within a Transformer block: linear projection into Query–Key–Value spaces, computation of attention weights, concatenation, and causal masking.

(2) Normalization Layer and Residual Connection

In each *Transformer* block, after the attention layer (or after the feed-forward layer), a residual connection is applied, followed by layer normalization. The residual connection consists of adding the input vector of the Transformer to its output vector, then applying normalization:

$$Z_N = Norm(H + Z_O) \tag{6.13}$$

This structure preserves the original information while stabilizing the scale and distribution of vector values, thereby improving learning stability in architectures composed of multiple stacked Transformer blocks. A *dropout* layer is also often added, randomly deactivating parts of the neural network to prevent overfitting.

(3) Feed-Forward Layer

The *feed-forward* sub-layer operates after the attention layer and the residual connection within each Transformer block. It acts on a position-wise basis: each vector at position t in the sequence is processed independently but in the same manner. Mathematically, if Z_n^t denotes the input vector at a given position t, the feed-forward transformation is expressed as:

$$\mathrm{FFN}(Z_n^t) = W_2\big(\max(0,\ Z_n^t \times W_1 + b_1)\big) + b_2 \tag{6.14}$$

where:

- W_1 and b_1 are the weight matrix and bias of the first linear layer,
- W_2 and b_2 are the weight matrix and bias of the second linear layer.

This sub-layer introduces a local non-linearity at each position, enriching the vectors output by the attention layer through individual transformations before passing them to the next block. A **dropout** layer is also commonly added after the feed-forward operation to improve generalization and prevent overfitting.

6.4.3 ActivityGPT model

The *ActivityGPT* model proposed in this work adapts the GPT architecture to the specific context of activity plan modeling. Each individual is represented by their contextual attributes c and an ordered sequence of activities $\{a_1, a_2, \ldots, a_T\}$, where both c and a_t are described by attribute vectors. The model learns the following conditional distribution:

$$p(a_1, \ldots, a_T \mid c) = \prod_{t=1}^{T} p(a_t \mid c, a_{<t}) \tag{6.15}$$

where c denotes a vector of socio-demographic or spatial conditioning variables (e.g., age, gender, income, area type, etc.).

Thus, the generation of a complete activity plan is performed in an *autoregressive* manner: each activity a_t is predicted based on the previous activities $a_{<t}$ and the individual's contextual attributes c, according to the following probability:

$$p(a_t|c, a_{<t}) = \prod_{t'=1}^{t-1} p(a_{t'}|c, a_{<t'}) \tag{6.16}$$

Figure 6.6 illustrates the overall architecture of the *ActivityGPT* model, which is struc-

tured into three main components:

(i) **Positional embeddings**: Each activity a_t is transformed into a continuous vector $\mathbf{e}_t$ through an *embedding* layer that encodes its attributes (activity type, transport mode, duration, location, etc.). A *positional embedding* $\mathbf{p}_t$ is then added to this vector to incorporate the temporal position of the activity within the day:

$$\mathbf{h}_t^{(0)} = \mathbf{e}_t + \mathbf{p}_t \tag{6.17}$$

This combination allows the model to distinguish between identical activities occurring at different times within the daily sequence.

(ii) **Transformer blocks**: The enriched representations $\mathbf{h}_t^{(0)}$ are then processed by a stack of N_l *Transformer* blocks. These blocks enable the model to learn temporal and behavioral dependencies between successive activities, such as the likelihood of the sequence *work* $\rightarrow$ *shopping* $\rightarrow$ *home*. The masked attention mechanism ensures the autoregressive property of the model: the prediction of an activity a_t depends only on the socio-economic characteristics c and the preceding activities $(a_1, \ldots, a_{t-1})$.

(iii) **Output layers and generation**: The final linear layer projects the hidden representations into the space of possible activity attributes. The probabilities associated with each attribute are then obtained via a *softmax* function, allowing the model to predict the next activity in the sequence.

The loss function of the *ActivityGPT* model is computed over all the attributes that compose each activity. Each activity a_t is represented by a vector of attributes $\{x_{t,1}, x_{t,2}, \ldots, x_{t,K}\}$, and the total loss corresponds to the sum of cross-entropies between the predicted distributions $\hat{x}_{t,k}$ and the true values $x_{t,k}$ for all attributes k and all temporal positions t in the sequence:

$$\mathcal{L}_{\text{CE}} = -\sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{d=1}^{D_k} x_{t,k}^{(d)} \log \hat{x}_{t,k}^{(d)} \tag{6.18}$$

where:

- T is the total number of activities in the sequence,
- K is the number of attributes describing each activity,
- D_k is the number of possible modalities for attribute k,
- $x_{t,k}^{(d)}$ is the true value (one-hot encoded) of modality d for attribute k at position t,

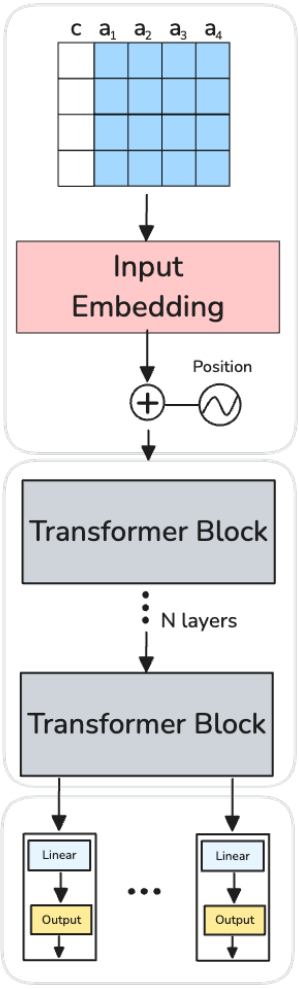


Figure 6.6: Conceptual architecture of the *ActivityGPT* model: input and positional embeddings, stacked Transformer blocks, and multiple output layers, each corresponding to an activity attribute.

- $\hat{x}_{t,k}^{(d)}$ is the corresponding probability predicted by the model.

During the generation phase, the model produces new activity plans by sequentially sampling each activity according to the learned conditional distribution $p_\theta(a_t \mid a_{<t}, c)$. The process continues until a complete plan is generated, i.e., until the prediction of the end-of-sequence token `<E>` or the predefined maximum sequence length L is reached.

6.5 Case study

6.5.1 Datasets

As in the previous case studies, we used the *Enquête Mobilité des Déplacements* (EMD) of the Loire-Atlantique region Département-Loire-Atlantique (2015), which includes a detailed component on daily mobility patterns. Approximately 29 500 individuals were surveyed. For each respondent, the dataset provides detailed information on all activities carried out during a typical day, including their temporal and spatial characteristics, as well as the transport modes used to connect them. It therefore serves as a reference database for the modeling and generation of realistic activity plans.

Each record corresponds to an elementary activity, associated with a set of attributes describing both the activity content (type, duration, location, transport mode, etc.) and the socio-demographic context of the individual performing it. These data allow observed behaviors to be linked with specific individual profiles.

The list of attributes used for modeling is presented in Table 6.1. It distinguishes two main groups: (i) *activity attributes*, describing behavioral characteristics of daily activities, and (ii) *individual attributes*, encompassing socio-demographic and economic variables used as conditioning inputs in the generative models (e.g., age, gender, employment status, household composition, or income level).

6.5.2 Evaluation Metrics

The evaluation of synthetic activity plans requires metrics capable of assessing not only their statistical quality, but also their behavioral coherence and spatio-temporal plausibility. The indicators considered in this section are grouped into four main categories: reproduction of marginal distributions, coherence of transitions between activities, spatio-temporal plausibility, and structural diversity of the generated plans.

6.5.2.1 Marginal Distributions

The first evaluation step consists in comparing the marginal distributions of the main descriptive variables of the generated plans (activity type, transport mode, start time, duration, location, etc.) with those observed in the real sample (EMD). This comparison can be performed using classical divergence measures between distributions, such as:

- the *Jensen–Shannon Distance* (JSD), a symmetric and bounded measure that quantifies the similarity between empirical and synthetic distributions;
- the *Kullback–Leibler Divergence* (KL), which is more sensitive to discrepancies in low-

Table 6.1: List of activity and individual attributes used for generating activity plans.

Groups	Attributes	Type[1]	Description
Activities			
	purpose	cat(6)	Activity purpose
	purpose_duration	num	Activity purpose duration
	mode	cat(5)	Travel mode
	travel_duration	num	Travel duration
	travel_distance	num	Travel distance
Individual			
	household_size	num	Number of persons in the HH
	number_of_vehicles	num	Number of vehicles in the HH
	has_internet	cat(2)	Has internet access ?
	link_ref_person	cat(5)	Link with the HH reference person
	age	num	Age
	gender	cat(2)	Gender
	school_level	cat(5)	School level
	socioprofessional_class	cat(9)	Socio-professional status
	has_license	cat(2)	Has a driver's license ?

[1] Indicates whether the attribute is numerical (num) or categorical (cat). When the attribute is categorical, the number in parentheses represents the number of possible values.

probability regions;

- or the *Mean Absolute Error* (MAE) between marginal proportions.

6.5.2.2 Activity Transitions

The behavioral coherence of the generated plans is then examined through the *inter-activity transition matrices*. For each pair of successive activities (a_t, a_{t+1}), the observed and simulated transition probabilities are estimated, and their difference is measured. A synthetic indicator can be obtained using:

- the *Pearson correlation coefficient* between the real and simulated transition matrices;
- or the *Frobenius distance*:

$$d_F = \sqrt{\sum_{i,j} (P_{ij}^{\text{real}} - P_{ij}^{\text{sim}})^2} \tag{6.19}$$

This metric assesses the model's ability to reproduce typical daily activity patterns (e.g., *home* → *work*, *work* → *shopping*, *home* → *leisure*). Recent studies emphasize that the

evaluation of sequential models should consider not only marginal distributions but also the internal dynamics of sequences (Sai *et al.*, 2020).

6.5.2.3 Spatial and Temporal Distances

The models must also preserve observed spatio-temporal regularities, in particular travel distances and activity durations. To this end, we evaluate:

- the **distribution of distances** between consecutive activity locations, compared between real and generated plans;
- the **distribution of activity durations** and start/end times, analyzed across temporal classes;
- and the **daily temporal structure**, measured by the average number of activities and the total time spent outside the home.

These indicators verify that the models reproduce plausible daily rhythms that remain consistent with observed behavioral patterns.

6.5.2.4 Diversity and Overall Coherence

Finally, the structural diversity and internal coherence of the generated plans are important criteria for assessing the quality of a generative model. They can be evaluated through:

- the **mean and variance of the number of activities** per individual, compared with the observed data;
- the **Shannon diversity index (H_s)** applied to the distribution of unique generated sequences, measuring sequence diversity:

$$H_s = \sum_{i}^{S} p(A_i) \times \ln(p(A_i)) \tag{6.20}$$

  where A_i denotes a unique sequence and S the total number of distinct sequences;
- the **rate of invalid plans**, i.e., those exhibiting temporal inconsistencies or unrealistic sequences (for example, reversed activity order or trips without an origin).

6.6 Implementation and Preliminary Analysis

In the context of this chapter, no numerical results could be obtained due to computational constraints. Indeed, the models explored are highly promising but require substantial computational resources and long training times. For instance, the **ActivityGPT + SeqRep**

configuration involves the training of more than 250 000 parameters on a dataset exceeding 1.5 million records, each corresponding to a pair $(c, a_{<t}) \rightarrow a_t$, where c denotes the conditioning variables and $a_{<t}$ represents the sequence of preceding activities.

These computational requirements considerably increase both training time and memory consumption, particularly for autoregressive architectures such as GPT, which process sequences iteratively. Despite the use of the *CINES Adastra* supercomputer, operational constraints limited model training sessions to a maximum duration of 24 hours. As a result, it was not possible to complete the training phase and obtain quantitative results at this stage.

6.6.1 Implemented Models

Four model configurations were implemented to explore and compare different approaches to generating activity plans, combining the *CVAE* and *ActivityGPT* architectures with the two types of activity representations, *TimeGrid* and *SeqRep*:

- *CVAE + TimeGrid*: a conditional variational model (non-autoregressive) using a discretized time representation;
- *CVAE + SeqRep*: a conditional variational model operating on a sequential representation of activities;
- *ActivityGPT + TimeGrid*: an autoregressive ActivityGPT model applied to a time-grid structure;
- *ActivityGPT + SeqRep*: an ActivityGPT model trained on a sequential representation of activities.

6.6.2 Technological Framework

Data processing for model evaluation was carried out using Python. The primary libraries used were: Pandas 1.5, NumPy 1.23 and Pytorch 2.5.1. Graphs were created using the Plotnine 0.10 library. These tasks were carried out using the Cines-Adastra[1] supercomputer with two AMD EPYC 7A53 (Trento) processors with 64 cores at 2.0 GHz and 128 GB of RAM.

6.6.3 A Priori Analysis of the Different Approaches

Although empirical results are not yet available, it is possible to propose an *a priori* analysis of the expected performance of the different model configurations. This reflection is based on the intrinsic nature of the architectures (CVAE vs. ActivityGPT) and on the type of

[1] https://www.cines.fr/calcul/espace-utilisateur/

activity-sequence representation (TimeGrid vs. SeqRep), each offering specific advantages and limitations in terms of modeling capacity, sequential coherence, and computational complexity.

(1) CVAE + TimeGrid

The combination of the **CVAE** with the **TimeGrid** representation constitutes a conceptually coherent approach. The TimeGrid scheme, based on a fixed temporal discretization of the day, provides a regular structure compatible with the non-autoregressive nature of the CVAE. This representation allows the generation of complete activity plans without the explicit introduction of sequence tokens (`<S>`, `<E>`, `<P>`) or the need to manage a strict order between activities. However, the length of the input vector associated with TimeGrid — potentially very large depending on the chosen temporal granularity — may increase training time and slow model convergence. Hence, a trade-off is expected between temporal expressiveness and computational complexity.

(2) CVAE + SeqRep

The **CVAE + SeqRep** configuration uses the CVAE with an explicit sequential representation of activities. While this approach preserves the chronological order of activities, it is likely to yield slightly less satisfactory results than the **CVAE + TimeGrid** combination. Indeed, the non-autoregressive nature of the CVAE limits its ability to properly capture dependencies between successive activities. Moreover, the introduction of padding tokens (`<P>`) required to standardize sequence lengths may introduce bias, encouraging the model to assign artificial importance to these filled positions. Nevertheless, this approach remains computationally lighter than GPT-based configurations, with shorter training times and lower hardware requirements.

(3) ActivityGPT + TimeGrid

The combination of the **ActivityGPT** model with the **TimeGrid** representation appears, *a priori*, less suitable. The autoregressive nature of ActivityGPT relies on the explicit modeling of sequential dependencies, whereas the TimeGrid representation, based on a regular time grid, induces numerous repetitions of the same activity over consecutive time slots. Such redundancy may hinder the model's ability to identify meaningful transitions and to generate coherent sequences. Moreover, the large size of the input vectors would make training particularly expensive in both memory and computation time. Consequently, this configuration is expected to deliver lower performance than non-autoregressive models, at a higher computational cost.

(4) ActivityGPT + SeqRep

The **ActivityGPT + SeqRep** configuration represents the most conceptually promising approach. The sequential nature of the **SeqRep** representation aligns perfectly with the autoregressive design of the GPT model, which directly learns the conditional distribution $p(a_t|c, a_{<t})$. This structure enables the model to capture long-term temporal and behavioral dependencies and to ensure coherent generation of successive activities. Therefore, this configuration is expected to yield higher-quality results. However, it is also the most computationally demanding.

In summary, the expected hierarchy of performance can be expressed as follows:

$$\text{ActivityGPT+SeqRep} > \text{CVAE+TimeGrid} > \text{CVAE+SeqRep} > \text{ActivityGPT+TimeGrid}.$$

6.7 Conclusion

This chapter has explored several modelling approaches for the generation of individual activity plans, focusing in particular on Conditional Variational Autoencoders (CVAEs) and Transformer-based architectures (ActivityGPT). A preliminary analysis suggests that ActivityGPT, when combined with a sequential representation of activities, appears to be the most promising configuration for generating activity plans. However, the computation times required to calibrate these models were not compatible with the duration of the thesis. The next step will therefore be to optimise the implementation using the *C* or *C++* languages, distributed learning strategies, or hardware acceleration. This will allow us to carry out a full quantitative evaluation and comparison of the proposed generative frameworks.

Conclusion and perspectives

7.1 Conclusion

The field of Artificial Intelligence (AI) is currently experiencing rapid growth, driven in particular by the recent progress of generative models capable of producing images, music, text, or even videos that were not present in the datasets used for their training.

In comparison, the generation of synthetic populations for the evaluation of transport systems remains a more restricted scientific field. Such generations require the construction of individuals with specific attributes — socio-economic, spatial, and behavioural — in order to represent mobility demand and the interactions between population and territory. The methods currently employed still rely largely on classical statistical approaches.

This thesis transposes recent approaches from AI to the generation of synthetic populations and activities, in order to benefit from the fundamental property of generative models: producing realistic individuals or households that do not appear in the training datasets but nevertheless belong to the real population.

Among the diversity of available generative models, this research has focused on Variational Autoencoders (VAE), a representative technique of deep generative models. The VAE methodology was applied to the generation of individuals in order to extend a household travel survey. The results show that the model is capable of generating plausible individuals, present in the population but absent from the training data (sampling zero problem).

Two modelling methods were then proposed to generate individuals within households in a statistically coherent manner, an essential prerequisite for any study of mobility demand. The first, called *SVAE-Pop2*, uses a single unified VAE to jointly generate complete households and their members within a common latent representation. This global structure allows it to capture interdependencies between individuals while maintaining robustness when data are sparse or unevenly distributed. The second, *MVAE-Pop2*, adopts a modular approach in which a specific VAE is trained for each household size, thus enhancing

the model's capacity to reproduce joint attribute distributions with greater accuracy. A comparative analysis revealed that while *MVAE-Pop2* performs better on well-represented household types, *SVAE-Pop2* offers greater flexibility and efficiency when confronted with limited data.

A hybrid modeling strategy was then introduced to integrate external statistical constraints into the generative process. This approach, termed *VAE+IPF*, combines the flexibility of VAEs—capable of learning multivariate structures—with the statistical rigor of the Iterative Proportional Fitting (IPF) procedure, which ensures alignment of the generated data with known marginal distributions. The VAE captures latent relationships and correlations between attributes, while IPF acts as a post-processing correction to guarantee the coherence of marginal totals. Empirical evaluations confirmed that this hybrid model produces synthetic populations consistent both at the individual and aggregate levels, effectively reducing structural zeros and improving overall statistical fidelity.

Finally, the research addressed the generation of individual activity plans. Two modelling approaches were explored: the first, based on the Conditional Variational Autoencoder (CVAE), refines the VAE by introducing conditioning on selected socio-demographic attributes; the second relies on a conceptually different technique, *ActivityGPT*, inspired by the Generative Pre-trained Transformer architecture. Preliminary results were obtained by combining these models with two types of temporal representation: a fixed daily discretization (*TimeGrid*) and a sequential encoding of activities (*SeqRep*). Among the four resulting configurations, *ActivityGPT+SeqRep* emerged as the most promising, as it naturally captures both short- and long-term temporal dependencies in activity sequences. The *CVAE+TimeGrid* model also showed coherent structuring potential but at a higher computational cost, while the other configurations offered trade-offs between expressiveness and training efficiency.

Beyond these methodological results, several operational and academic perspectives emerge, aiming to strengthen the integration of AI-based approaches into tools for generating synthetic populations to assess transport systems.

7.2 Operational Perspectives

Availability for the scientific community

The articles resulting from this thesis are accompanied by commented and documented codes. These resources are already accessible to the scientific community, and in particular to the host research team, which can use them in its own projects. This dissemination aims to facilitate the reuse of the proposed methods and to encourage their adaptation to other study contexts or territorial scales.

Integration into existing tools

The adoption of these new approaches by consulting firms and practitioners requires their implementation within operational tools. The Eqasim open-source software, designed for French data, provides a suitable framework for such integration. Its modular structure allows the methods developed in this thesis to be incorporated seamlessly. This integration will nevertheless require adjustments in terms of data input and output formats, as well as a rigorous software validation process for the developed codes, to ensure robustness and reproducibility of the results.

Integration into public decision support

Beyond supporting experts in the use of these new tools, it is also necessary to develop mediation approaches aimed at citizens. These approaches should help them understand the functioning of such models, their advantages, and their limitations in the context of public decision-making. Fostering this shared understanding would strengthen the transparency and legitimacy of modelling practices, ensuring that simulation-based decision tools serve both policymakers and the broader public.

Beyond these operational perspectives, several academic extensions can be envisioned, both methodological and conceptual, to further strengthen the reliability and scientific scope of the proposed approaches.

7.3 Academic Perspectives

Handling unrealistic individuals

Generative models in Artificial Intelligence can produce individuals who are not present in the training database but who belong to the real population (sampling zero problem). However, they may also generate individuals who do not exist in the real population (structural zero problem). These unrealistic individuals are generally few in number (around 7% for a synthetic population of 1.4 million, with a model trained on 16 500 individuals), and their influence on a transport system evaluation study — which relies on the re-aggregation of generated data — could be negligible. This hypothesis, however, must be verified. If their impact proves to be significant, approaches based on Large Language Models (LLM) could be used to filter or correct such aberrant cases. These models, capable of capturing complex semantic relationships, could automatically identify inconsistencies such as unemployed children or households composed solely of children.

Activity generation

Several modelling approaches and associated methods have been explored in relation to activity generation, but no final results have been achieved. Preliminary experiments combining Conditional VAEs and Transformer-based models (*ActivityGPT*) have shown promising potential for capturing temporal dependencies in daily activity sequences. Future work will focus on completing the training of the model and evaluating the generated plans against the observed mobility data. This will help to identify the optimal combination of modelling and associated methodology that can generate coherent and plausible activity plans at both individual and territorial scales.

Spatialisation of households and activities

In the generation of synthetic populations and activity plans, the spatialisation of households and activities has not been addressed in this thesis. An exciting avenue for future research would be to establish a connection between the latent space of VAE — where the abstract characteristics of individuals and households are represented — and the physical space in which these entities are actually deployed.

Bibliography

Aemmer, Z. et MacKenzie, D. (2022). Generative population synthesis for joint household and individual characteristics. *Computers, Environment and Urban Systems*, 96:101852.

Alabadla, M., Sidi, F., Ishak, I., Ibrahim, H., Affendey, L. S., Che Ani, Z., Jabar, M. A., Bukar, U. A., Devaraj, N. K., Muda, A. S., Tharek, A., Omar, N. et Jaya, M. I. M. (2022). Systematic Review of Using Machine Learning in Imputing Missing Values. *IEEE Access*, 10:44483–44502.

Arentze, T., Timmermans, H. et Hofman, F. (2007). Creating Synthetic Household Populations: Problems and Approach. *Transportation Research Record: Journal of the Transportation Research Board*, 2014(1):85–91.

Barr, A. A., Quan, J., Guo, E. et Sezgin, E. (2025). Large language models generating synthetic clinical datasets: a feasibility and comparative analysis with real-world perioperative data. *Frontiers in Artificial Intelligence*, 8:1533508.

Beckman, R. J., Baggerly, K. A. et McKay, M. D. (1996). Creating synthetic baseline populations. *Transportation Research Part A: Policy and Practice*, 30(6):415–429.

Borisov, V., Sessler, K., Leemann, T., Pawelczyk, M. et Kasneci, G. (2022). Language Models are Realistic Tabular Data Generators. Version Number: 2.

Borysov, S. S., Rich, J. et Pereira, F. C. (2018). Population Synthesis Meets Deep Generative Modelling. page 3.

Borysov, S. S., Rich, J. et Pereira, F. C. (2019a). How to generate micro-agents? A deep generative modeling approach to population synthesis. *Transportation Research Part C: Emerging Technologies*, 106:73–97.

Borysov, S. S., Rich, J. et Pereira, F. C. (2019b). Scalable Population Synthesis with Deep Generative Modeling. *Transportation Research Part C: Emerging Technologies*, 106: 73–97.

Bouzouina, L., Baraklianos, I., Bonnel, P. et Aissaoui, H. (2021). Renters vs owners: The impact of accessibility on residential location choice. Evidence from Lyon urban area,

France (1999–2013). *Transport Policy*, 109:72–84.

Brückmann, G., Willibald, F. et Blanco, V. (2021). Battery Electric Vehicle adoption in regions without strong policies. *Transportation Research Part D: Transport and Environment*, 90:102615. Publisher: Elsevier.

Casati, D., Müller, K., Fourie, P. J., Erath, A. et Axhausen, K. W. (2015). Synthetic Population Generation by Combining a Hierarchical, Simulation-Based Approach with Reweighting by Generalized Raking. *Transportation Research Record: Journal of the Transportation Research Board*, 2493(1):107–116.

Chapuis, K., Taillandier, P. et Drogoul, A. (2022). Generation of Synthetic Populations in Social Simulations: A Review of Methods and Practices. *Journal of Artificial Societies and Social Simulation*, 25(2):6.

Daudé, E., Chapuis, K., Taillandier, P., Tranouez, P., Caron, C., Drogoul, A., Gaudou, B., Rey-Coyrehourq, S., Saval, A. et Zucker, J.-D. (2019). ESCAPE: Exploring by Simulation Cities Awareness on Population Evacuation. *In Proceedings of the 16th International Conference on Information Systems for Crisis Response And Management*, Valencia, Spain.

Delhoum, Y., Belaroussi, R., Dupin, F. et Zargayouna, M. (2021). Modeling Activity-Time to Build Realistic Plannings in Population Synthesis in a Suburban Area. *Applied Sciences*, 11(16):7654.

Delhoum, Y., Cardin, O., Nouiri, M. et Harzallah, M. (2024). Machine learning-assisted microscopic public transportation simulation: Two coupling strategies. *Simulation Modelling Practice and Theory*, 137:103019.

Deming, W. E. et Stephan, F. F. (1940). On a Least Squares Adjustment of a Sampled Frequency Table When the Expected Marginal Totals are Known. *The Annals of Mathematical Statistics*, 11(4):427–444.

Deville, J., Särndal, C. et Sautory, O. (1993). Generalized raking procedures in survey sampling. *Journal of the American Statistical Association*, 88:1013–1020.

Donders, A. R. T., Van Der Heijden, G. J., Stijnen, T. et Moons, K. G. (2006). Review: A gentle introduction to imputation of missing values. *Journal of Clinical Epidemiology*, 59(10):1087–1091.

D'orazio, M., Di Zio, M. et Scanu, M. (2006). *Statistical matching: Theory and practice*. John Wiley & Sons.

Dumont, M., Carletti, T. et Cornélis, E. (2017). Population synthétique: un outil pour une analyse spatiale fine des besoins futurs en soins de santé. *In Vieillissement et entraide*, volume 6, pages 55–74. Presses Universitaires de Namur (PUN), Namur.

Département-Loire-Atlantique (2015). Enquête déplacements (2015) en Loire-Atlantique. Place: https://data.loire-atlantique.fr/explore/dataset/224400028_enquete-deplacements-en-loire-atlantique/information/.

Fabrice Yaméogo, B., Gastineau, P., Hankach, P. et Vandanjon, P.-O. (2021). Comparing methods for generating a two-layered synthetic population. *Transportation research record*, 2675(1):136–147. Publisher: SAGE Publications Sage CA: Los Angeles, CA.

Farooq, B., Bierlaire, M., Hurtubia, R. et Flötteröd, G. (2013). Simulation based population synthesis. *Transportation Research Part B: Methodological*, 58:243–263.

Fienberg, S. E. (1970). An Iterative Procedure for Estimation in Contingency Tables. *The Annals of Mathematical Statistics*, 41(3):907–917.

Galassi Luquezi, L., Le Bescond, V., Aumond, P., Gastineau, P. et Can, A. (2025). Including everyday mobility with agent-based models in the assessment of critical areas of noise exposure. *Sustainable Cities and Society*, 130:106551.

Garrido, S., Borysov, S. S., Pereira, F. C. et Rich, J. (2020). Prediction of rare feature combinations in population synthesis: Application of deep generative modelling. *Transportation Research Part C: Emerging Technologies*, 120:102787.

Gastineau, P., Can, A., Yaméogo, B. F., Luquezi, L. G., Hankach, P., Vandanjon, P.-O. et Bescond, V. L. (2023). Modeling Exposure to Mobility-Related Pollution: Review and Key Challenges. *Transportation Research Record*, 0(0):03611981231208907. _eprint: https://doi.org/10.1177/03611981231208907.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. et Bengio, Y. (2014). Generative Adversarial Networks.

Gómez-Marín, C. G., Mosquera-Tobón, J. D. et Serna-Urán, C. A. (2023). Microsimulation Calibration Integrating Synthetic Population Generation and Complex Interaction Clusters to Evaluate COVID-19 Spread. *In* Zapata-Cortes, J. A., Sánchez-Ramírez, C., Alor-Hernández, G. et García-Alcaraz, J. L., éditeurs : *Handbook on Decision Making: Volume 3: Trends and Challenges in Intelligent Decision Support Systems*, pages 419–437. Springer International Publishing, Cham.

Hankach, P., Bescond, V. L., Gastineau, P., Vandanjon, P.-O., Can, A. et Aumond, P. (2024). Individual-level activity-based modeling and indicators for assessing construction sites noise exposure in urban areas. *Sustainable Cities and Society*, 101:105188.

Harland, K., Heppenstall, A., Smith, D. et Birkin, M. (2012). Creating Realistic Synthetic Populations at Varying Spatial Scales: A Comparative Critique of Population Synthesis Techniques. *Journal of Artificial Societies and Social Simulation*, 15(1):1.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S. et Lerchner, A. (2017). beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. *In International Conference on Learning Representations*.

Ho, J., Jain, A. et Abbeel, P. (2020). Denoising Diffusion Probabilistic Models. Version Number: 2.

Horni, A., Nagel, K. et Axhausen, K. W. (2016). Introducing MATSim. *In The Multi-Agent Transport Simulation MATSim*, pages 3–8. Ubiquity Press.

Hörl, S. et Balac, M. (2021a). Introducing the eqasim pipeline: From raw data to agent-based transport simulation. *Procedia Computer Science*, 184:712–719.

Hörl, S. et Balac, M. (2021b). Synthetic population and travel demand for Paris and Île-de-France based on open and publicly available data. *Transportation Research Part C: Emerging Technologies*, 130:103291.

Ilahi, A. et Axhausen, K. W. (2019). Integrating Bayesian network and generalized raking for population synthesis in Greater Jakarta. *Regional Studies, Regional Science*, 6(1):623–636.

Ilaty, A., Shirazi, H. et Homayouni, H. (2025). SynLLM: A Comparative Analysis of Large Language Models for Medical Tabular Synthetic Data Generation via Prompt Engineering. Version Number: 1.

Ireland, C. et Kullback, S. (1968). Contingency tables with given marginals. *Biometrika*, 55 1:179–88.

Johnsen, M., Brandt, O., Garrido, S. et Pereira, F. (2022). Population synthesis for urban resident modeling using deep generative models. *Neural Computing and Applications*, 34(6):4677–4692.

Joubert, J. W. et De Waal, A. (2020). Activity-based travel demand generation using Bayesian networks. *Transportation Research Part C: Emerging Technologies*, 120:102804.

Kim, E.-J. et Bansal, P. (2022). A Deep Generative Model for Feasible and Diverse Population Synthesis.

Kingma, D. P. et Welling, M. (2013). Auto-Encoding Variational Bayes.

Kingma, D. P. et Welling, M. (2019). An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392.

Kotelnikov, A., Baranchuk, D., Rubachev, I. et Babenko, A. (2023). TabDDPM: Modelling tabular data with diffusion models. *In* Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S. et Scarlett, J., éditeurs : *Proceedings of the 40th International Conference on Machine Learning*, volume 202 de *Proceedings of Machine Learning Research*, pages 17564–17579. PMLR.

Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243.

Lee, D.-H. et Fu, Y. (2011). Cross-Entropy Optimization Model for Population Synthesis in Activity-Based Microsimulation Models. *Transportation Research Record: Journal of the Transportation Research Board*, 2255(1):20–27.

Li, Z., Huang, Q., Yang, L., Shi, J., Yang, Z., Stein, N. v., Bäck, T. et Leeuwen, M. v. (2025). Diffusion Models for Tabular Data: Challenges, Current Progress, and Future Directions. arXiv:2502.17119 [cs].

Lim, P. P. et Gargett, D. (2013). Population synthesis for travel demand forecasting.

Liu, F., Janssens, D., Cui, J., Wets, G. et Cools, M. (2015). Characterizing activity sequences using profile Hidden Markov Models. *Expert Systems with Applications*, 42(13): 5705–5722.

Lovelace, R. et Ballas, D. (2013). 'Truncate, replicate, sample': A method for creating integer weights for spatial microsimulation. *Computers, Environment and Urban Systems*, 41:1–11.

Lovelace, R., Birkin, M., Ballas, D. et Van Leeuwen, E. (2015). Evaluating the Performance of Iterative Proportional Fitting for Spatial Microsimulation: New Tests for an Established Technique. *Journal of Artificial Societies and Social Simulation*, 18(2):21.

Ma, L. et Srinivasan, S. (2015). Synthetic Population Generation with Multilevel Controls: A Fitness-Based Synthesis Approach and Validations: A fitness-based approach for population synthesis. *Computer-Aided Civil and Infrastructure Engineering*, 30(2):135–150.

Markey, M. K., Tourassi, G. D., Margolis, M. et DeLong, D. M. (2006). Impact of missing data in evaluating artificial neural networks trained on complete data. *Computers in Biology and Medicine*, 36(5):516–525.

Müller, K. et Axhausen, K. W. (2010). Population synthesis for microsimulation: State of the art. *Arbeitsberichte Verkehrs-und Raumplanung*, 638. Publisher: IVT, ETH Zurich.

Müller, K. et Axhausen, K. W. (2011). Hierarchical IPF: Generating a synthetic population for Switzerland. Medium: application/pdf,Online-Datei Publisher: ETH Zurich.

Müller, K. et Axhausen, K. W. (2012). Multi-level fitting algorithms for population synthesis. *Arbeitsberichte Verkehrs-und Raumplanung*, 821.

Namazi-Rad, M.-R., Tanton, R., Steel, D., Mokhtarian, P. et Das, S. (2017). An unconstrained statistical matching algorithm for combining individual and household level geo-specific census and survey data. *Computers, Environment and Urban Systems*, 63:3–14.

Plaut, P. O. (2006). The intra-household choices regarding commuting and housing. *Transportation Research Part A: Policy and Practice*, 40(7):561–571.

Potdar, K., S., T. et D., C. (2017). A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers. *International Journal of Computer Applications*, 175(4):7–9.

Pritchard, D. R. et Miller, E. J. (2012). Advances in population synthesis: fitting many attributes per agent and fitting to household and person margins simultaneously. *Transportation*, 39(3):685–704.

Qian, X., Gangwal, U., Dong, S. et Davidson, R. (2024). A Deep Generative Framework for Joint Households and Individuals Population Synthesis.

Rahman, M. N. et Fatmi, M. R. (2023). Population Synthesis Accommodating Heterogeneity: A Bayesian Network and Generalized Raking Technique. *Transportation Research Record: Journal of the Transportation Research Board*, 2677(6):41–57.

Rouchier, J. (2024). L'éthique des modèles de simulation : une question d'interprétation ? *L'Année sociologique*, 74(2):495–527. Publisher: Presses Universitaires de France.

Saadi, I., Mustafa, A., Teller, J. et Cools, M. (2016a). Forecasting travel behavior using Markov Chains-based approaches. *Transportation Research Part C: Emerging Technologies*, 69:402–417.

Saadi, I., Mustafa, A., Teller, J., Farooq, B. et Cools, M. (2016b). Hidden Markov Model-based population synthesis. *Transportation Research Part B: Methodological*, 90:1–21.

Sai, A. B., Mohankumar, A. K. et Khapra, M. M. (2020). A Survey of Evaluation Metrics Used for NLG Systems. arXiv:2008.12009 [cs].

Sallard, A. et Balać, M. (2023). Travel demand generation using Bayesian Networks: an application to Switzerland. *Procedia Computer Science*, 220:267–274.

Sané, A. R., Belaroussi, R., Hankach, P. et Vandanjon, P.-O. (2025a). Population synthesis with deep generative model: a joint household-individual approach. *Computational Urban Science*, 5(1):34.

Sané, A. R., Vandanjon, P.-O., Belaroussi, R. et Hankach, P. (2025b). A comprehensive investigation of variational auto-encoders for population synthesis. *Journal of Computational Social Science*, 8(1):13.

Sharma, S., Sharma, S. et Athaiya, A. (2020). ACTIVATION FUNCTIONS IN NEURAL NETWORKS. *International Journal of Engineering Applied Sciences and Technology*, 04(12):310–316.

Shaw, P., Uszkoreit, J. et Vaswani, A. (2018). Self-Attention with Relative Position Representations. arXiv:1803.02155 [cs].

Shi, J., Xu, M., Hua, H., Zhang, H., Ermon, S. et Leskovec, J. (2025). TabDiff: a Mixed-type Diffusion Model for Tabular Data Generation. arXiv:2410.20626 [cs].

Shone, F. et Hillel, T. (2025). Synthesising Activity Participations and Scheduling with Deep Generative Machine Learning. arXiv:2501.10221 [cs].

Smolyak, D., Bjarnadóttir, M. V., Crowley, K. et Agarwal, R. (2024). Large language models and synthetic health data: progress and prospects. *JAMIA Open*, 7(4):ooae114.

Sohn, K., Yan, X. et Lee, H. (2015). Learning structured output representation using deep conditional generative models. *In Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pages 3483–3491, Cambridge, MA, USA. MIT Press. event-place: Montreal, Canada.

Sun, L. et Erath, A. (2015). A Bayesian network approach for population synthesis. *Transportation Research Part C: Emerging Technologies*, 61:49–62.

Sun, L., Erath, A. et Cai, M. (2018). A hierarchical mixture modeling framework for population synthesis. *Transportation Research Part B: Methodological*, 114:199–212. Publisher: Elsevier.

Sun, Z. et Zacharias, J. (2021). Do housing tenure and public transport provision matter in automobile use in bedroom suburban communities? Evidence from Beijing. *Journal of Housing and the Built Environment*, 36(1):241–262. Publisher: Springer.

Templ, M., Meindl, B., Kowarik, A. et Dupriez, O. (2017). Simulation of Synthetic Complex Data: The *R* Package **simPop**. *Journal of Statistical Software*, 79(10).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. et Polosukhin, I. (2017). Attention Is All You Need. arXiv:1706.03762 [cs].

Voas, D. et Williamson, P. (2000). An evaluation of the combinatorial optimisation approach to the creation of synthetic microdata. *International Journal of Population Geography*, 6:349–366.

Wang, D. et Cao, X. (2017). Impacts of the built environment on activity-travel behavior: Are there differences between public and private housing residents in Hong Kong? *Transportation Research Part A: Policy and Practice*, 103:25–35.

Wang, X., Yan, X., Zhao, X. et Cao, Z. (2022). Identifying latent shared mobility preference segments in low-income communities: Ride-hailing, fixed-route bus, and mobility-on-demand transit. *Travel Behaviour and Society*, 26:134–142.

Williamson, P., Birkin, M. et Rees, P. H. (1998). The Estimation of Population Microdata by Using Data from Small Area Statistics and Samples of Anonymised Records. *Environment and Planning A: Economy and Space*, 30(5):785–816.

Xu, S., Lee, C.-T., Sharma, M., Yousuf, R. B., Muralidhar, N. et Ramakrishnan, N. (2025). Why LLMs Are Bad at Synthetic Table Generation (and what to do about it). arXiv:2406.14541 [cs].

Yameogo, B. F., Vandanjon, P.-O., Gastineau, P. et Hankach, P. (2021). Generating a Two-Layered Synthetic Population for French Municipalities: Results and Evaluation of Four Synthetic Reconstruction Methods. *Journal of Artificial Societies and Social Simulation*, 24(2):5.

Ye, P., Tian, B., Lv, Y., Li, Q. et Wang, F.-Y. (2022). On Iterative Proportional Updating: Limitations and Improvements for General Population Synthesis. *IEEE Transactions on Cybernetics*, 52(3):1726–1735.

Ye, P.-j., Wang, X., Chen, C., Lin, Y.-t. et Wang, F.-y. (2016). Hybrid Agent Modeling in Population Simulation: Current Approaches and Future Directions. *Journal of Artificial Societies and Social Simulation*, 19(1):12.

Ye, X., Konduri, K., Pendyala, R. M., Sana, B. et Waddell, P. (2009). Methodology to match distributions of both household and person attributes in generation of synthetic population.

Zhang, D., Cao, J., Feygin, S., Tang, D., Shen, Z.-J. et Pozdnoukhov, A. (2019). Connected population synthesis for transportation simulation. *Transportation Research Part C: Emerging Technologies*, 103:1–16.

Zhu, Y. et Ferreira, J. (2014). Synthetic Population Generation at Disaggregated Spatial Scales for Land Use and Transportation Microsimulation. *Transportation Research Record: Journal of the Transportation Research Board*, 2429(1):168–177.

Université Gustave Eiffel

PARIS EST SUP — ÉCOLE DOCTORALE
Ville, Transports et Territoires

Title : Deep Generative Models for Synthetic Population Modeling in Mobility Research

Keywords : synthetic population, deep generative models, variational autoencoders, hybrid modeling, households, activity plans, microsimulation, AI for urban mobility

Abstract :

This thesis investigates the use of recent artificial intelligence methods—particularly deep generative models—for the generation of synthetic populations and activity plans. Its overarching goal is to reproduce realistic and statistically coherent representations of individuals, households, and their daily behaviors, as required in urban mobility and transport microsimulation frameworks. The research begins with a comprehensive review of synthetic population generation methods, ranging from traditional statistical reweighting and combinatorial optimization techniques to probabilistic, deep, and large language models (LLMs). This review highlights the trade-offs between robustness and flexibility and identifies hybrid models as a promising research direction. Building on this foundation, Variational Autoencoders (VAEs) are introduced as a core methodology for generating high-dimensional mixed data. Empirical experiments demonstrate their capacity to synthesize large, coherent populations from small samples, effectively addressing the issue of sampling zeros while preserving statistical consistency. The approach is then extended to a two-layered framework (*SVAE-Pop2* and *MVAE-Pop2*) that jointly models households and their members, capturing intra-household dependencies crucial for realistic behavioral simulations. To further enhance representativeness, a hybrid *VAE+IPF* model is proposed, combining the VAE's flexibility in learning complex distributions with the Iterative Proportional Fitting procedure to enforce external marginal constraints. This hybridization significantly improves both marginal and joint consistency while reducing structural zeros. Finally, the thesis explores the generation of daily activity plans through two complementary deep architectures: a Conditional VAE (CVAE) and a Transformer-based model termed *ActivityGPT*. Preliminary analyses show that sequential representations combined with GPT architectures hold strong potential for capturing temporal dependencies in daily mobility behaviors. Overall, the thesis contributes a coherent methodological and conceptual framework for integrating deep generative models into synthetic population and activity modeling.

Titre : Modèles génératifs profonds pour la modélisation de populations synthétiques en recherche sur la mobilité

Mots-clés : population synthétique, modèles génératifs profonds, autoencodeurs variationnels, modélisation hybride, ménages, plans d'activités, microsimulation, intelligence artificielle pour la mobilité urbaine

Résumé :

Cette thèse explore l'application des approches récentes de l'intelligence artificielle, et plus particulièrement des modèles génératifs profonds, à la génération de populations et de plans d'activités synthétiques. L'objectif général est de reproduire des représentations réalistes et statistiquement cohérentes des individus, des ménages et de leurs comportements quotidiens, indispensables à la modélisation urbaine et à la microsimulation des transports. Le travail débute par une revue comparative des principales méthodes de génération de populations synthétiques, allant des approches classiques de rééchantillonnage et d'optimisation combinatoire aux modèles probabilistes, profonds et fondés sur les grands modèles de langage (LLM). Cette analyse met en évidence les compromis entre robustesse et flexibilité, et souligne le potentiel des approches hybrides combinant la rigueur statistique des méthodes traditionnelles et la puissance expressive des modèles modernes. Sur cette base, les Autoencodeurs Variationnels (VAE) sont introduits comme cadre méthodologique central pour la génération de données mixtes de grande dimension. Les résultats empiriques démontrent leur capacité à produire des populations cohérentes à partir d'échantillons réduits, tout en limitant le problème des zéros d'échantillonnage. L'approche est ensuite étendue à un cadre hiérarchique à deux niveaux (*SVAE-Pop2* et *MVAE-Pop2*), permettant la génération conjointe des ménages et de leurs membres et la prise en compte des dépendances intra-ménages. Pour renforcer la représentativité statistique, un modèle hybride *VAE+IPF* est proposé, combinant la flexibilité du VAE à la rigueur du procédé d'ajustement itératif des marges (IPF). Enfin, la thèse explore la génération de plans d'activités quotidiens à l'aide de deux architectures complémentaires : un VAE conditionnel (CVAE) et un modèle de type Transformer, *ActivityGPT*. Les résultats préliminaires confirment le potentiel de ces modèles pour capturer les dépendances temporelles des comportements de mobilité.