Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

HIERARCHICAL POPULATION SYNTHESIS USING A NEURAL-DIFFERENTIABLE PROGRAMMING APPROACH

Imran Mahmood[1], Anisoara Calinescu[1], and Michael Wooldridge[1]

[1]Dept. of Computer Science, University of Oxford, Oxford, UK

ABSTRACT

Advances in Artificial Intelligence have enabled more accurate and scalable modeling of complex social systems, which depend on realistic, high-resolution population data. We introduce a novel methodology for generating hierarchical synthetic populations using differentiable programming, producing detailed demographic structures essential for simulation and analysis. Existing approaches struggle to model hierarchical population structures and optimize over discrete demographic attributes. Leveraging feed-forward neural networks and Gumbel-Softmax encoding, our approach transforms aggregated census and survey data into continuous, differentiable forms, enabling gradient-based optimization to match target demographics with high fidelity. The framework captures multi-scale population structures, including household composition and socio-economic diversity, with verification via logical rules and validation against census cross tables. A UK case study shows our model closely replicates real-world distributions. This scalable approach provides simulation modelers and analysts with, high-fidelity synthetic populations as input for agent-based simulations of complex societal systems, enabling behavior simulation, intervention evaluation, and demographic analysis.

1 INTRODUCTION

Agent-based models (ABMs) have become prominent tools for representing and analyzing complex systems, as they enable the simulation of individual entities that interact within a structured environment, revealing how local actions contribute to emergent patterns and global phenomena (Crooks et al. 2015; Macal 2016). ABMs are typically composed of three main components: (i) a synthetic population; (ii) a virtual environment; and (iii) a set of rules regulating agent behaviors and interactions (Railsback and Grimm 2019; Taylor 2014; Gilbert 2019). Each of these components plays a vital role in developing a high-quality agent-based model (ABM) capable of simulating complex systems effectively. A well-designed ABM is characterized by its expressiveness, accuracy, scalability, adaptability, and validation against real-world data. Together, these attributes ensure the robustness and reliability of the model. Given the central role of reflecting population structure and variability in ABMs, this paper focuses on the methodology for generating synthetic populations.

1.1 Synthetic Populations

In Agent-based Modeling, synthetic populations play a critical role in modeling individuals, forecasting policy impacts, and understanding shifts in the societal dynamics (Malleson et al. 2022). The need for synthetic populations arises from significant challenges related to data availability, driven by concerns about privacy, security, and government legislation. In many cases, real-world data is either inaccessible or restricted, making it difficult to study population dynamics. Synthetic populations offer a solution by providing detailed and realistic datasets that can simulate social systems without violating privacy or security regulations. As a result, the demand for synthetic data, particularly synthetic populations, has grown in response to these limitations. These populations serve as input data to initialize agents and their attributes within models, ensuring that the simulations accurately reflect the demographic, behavioral, and

socioeconomic characteristics of the real world while maintaining privacy and data integrity. This approach facilitates a realistic abstraction of society in ABMs, ensuring that interactions and outcomes are based on empirically grounded distributions. As a result, simulations are high-fidelity and provide meaningful insights into societal dynamics to help inform interventions, policy decisions, and planning efforts (Jiang et al. 2022; Jordon et al. 2022).

A hierarchical, multi-scale, multi-resolution structured synthetic population is a statistically representative, anonymized collection of entities such as individuals and households that mirrors the structural, behavioral, and demographic characteristics of the real population while maintaining multilevel relationships or hierarchy, such as family units within households (Mahmood et al. 2024). Multi-scale synthetic populations capture demographic variations across different geographic levels, such as national, regional, city, or neighborhood scales. The population characteristics of a city differ from those of a smaller region, e.g., an output area. Multi-resolution populations represent data at different levels of detail, such as grouping individuals by broad age ranges, e.g., children, adults, and elders (low resolution) or specific age ranges, e.g., 5-year age categories (high resolution). Generating synthetic populations that accurately reflect real-world societies presents several challenges. The main challenge of building these populations lies in ensuring their fidelity across various demographic characteristics and maintaining their dynamic nature over time. These populations are typically generated from aggregated census data and survey results, using computational techniques to ensure that diversity and distributions closely align with the actual population (Wu et al. 2022). Census and survey data, often used as source material, are inherently complex, exhibiting hierarchical structures and intricate relationships between variables (Malleson et al. 2022). Moreover, data sparsity can hinder the accurate inference of the complete joint distribution of individual characteristics (Wu et al. 2022). Generating large-scale synthetic populations with high-dimensional data can be computationally expensive (Prédhumeau and Manley 2023).

This paper proposes a neural-differentiable programming method for generating accurate and hierarchically structured synthetic populations. We assess the accuracy and computational efficiency of this method and provide a robust tool for generating input data for agent-based modeling of complex societal systems. To achieve these objectives, we introduce a differentiable programming framework (Blondel and Roulet 2024) that utilizes neural networks and the Gumbel-Softmax encoding technique. This framework seamlessly transforms discrete data (i.e., distributions obtained from census data) into continuous, differentiable forms suitable for synthesis and optimization using gradient-based methods. This in turn, permits the training of neural networks that can produce generated distributions (synthetic populations) that closely match real-world target distributions, such as those implicitly defined by cross-table aggregates of census data. Cross-table aggregates summarize census data by showing how one categorical variable is distributed across the categories of another, helping to analyze relationships between variables. The optimization process is guided by a nested loss function (see section 3.2.2) that aligns the generated output with target distributions. Previous works, (Borysov et al. 2018; Aemmer and MacKenzie 2022), were constrained by relying solely on continuous distributions, which limited the applicability of back-propagation to discrete attributes. In contrast, the reparametrisation trick introduced in (Jang et al. 2016) enables gradient-based optimization over discrete variables. We adopt this technique to effectively model and synthesize discrete population attributes. Our method offers several key advantages:

- **Expressiveness:** It can represent complex relationships and hierarchical structures at any level of granularity and resolution within the population. The nested loss function naturally supports hierarchical relationships among variables (e.g., age within sex or marital status within ethnic groups), enabling structured and interpretable population synthesis.
- **Computational Efficiency:** The use of differentiable programming and gradient-based optimization leads to faster convergence rates. This enables the generation of large-scale synthetic populations without requiring excessive computational resources.

- **Scalability:** The framework can handle a large number of characteristics and batch sizes, allowing for the incorporation of diverse demographic, socioeconomic, and behavioral factors at multiple scales.
- **Extensibility:** New categorical variables can be easily added to the model by extending the nested loss function with additional cross-table aggregates, ensuring flexibility and modularity in model design. The framework supports modeling synthetic populations across different geographical regions or administrative levels (e.g., Output Areas, Wards, Community Councils, Local Authority Districts, Metropolitan Boroughs, Districts, Cities, Counties, Regions, and Countries). In this paper, we use Middle layer Super Output Areas (MSOA) as a geographic unit.

These advantages make our approach particularly well-suited for generating realistic and representative synthetic populations for use in agent-based models and other simulation-based studies of complex societal systems. Our main contributions are as follows:

1. Development of a differentiable programming framework that leverages feedforward neural networks and Gumbel-Softmax encoding to efficiently transform aggregated census and survey data into continuous, differentiable forms for synthetic population generation, resulting in a robust population synthesis tool.
2. Demonstration of the accuracy, scalability, flexibility, and expressiveness of the framework in generating hierarchically structured populations through a case study using UK census data.
3. Verification of the structural integrity of the synthetic population generated using logical rules and validation through goodness-of-fit analysis with census cross tables to assess the precision of the output.

The rest of the paper is structured as follows: Section 2 reviews related work. In Section 3, we describe our proposed method using a neural-differentiable programming approach. Section 4 presents the results of our case study using UK census data, and Section 5 reports conclusions and outlines directions for future work.

2 RELATED WORK

Synthetic population generation is a rapidly evolving field, driven by the growing need for synthetic data and accurate representations of real-world populations. Existing methods can be broadly classified by their underlying techniques and focus, as follows:

2.1 Traditional Methods

Traditional approaches often use iterative proportional fitting (IPF), synthetic reconstruction, or combinatorial optimization. IPF, a foundational technique, iteratively adjusts attribute weights to match marginal distributions (Albiston et al. 2024; Casati et al. 2015; Ponge et al. 2021), but it struggles to capture complex attribute correlations and can lack diversity. Wu et al. (2022) propose an extension of the IPF to estimate health and socioeconomic outcomes in small areas of Great Britain. Combinatorial optimization methods seek attribute combinations that best match target characteristics. Mahmood et al. (2024) present a multi-objective framework for hierarchical synthesis, addressing IPF limitations by incorporating multiple objectives. Antoni et al. (2017) focus on generating individuals and households, while Kim and Lee (2016) use simulated annealing to align synthetic populations with marginal constraints. However, such methods can be computationally expensive at large scales because of the combinatorial search space.

2.2 Machine Learning Methods

Machine learning, and in particular deep generative models (DGMs), has seen growing adoption for modeling complex population structures. Generative Adversarial Networks (GANs) and Variational Auto-encoders (VAEs) have shown promise in producing realistic, diverse populations (Kim and Bansal 2023). GANs use adversarial training between generator and discriminator networks to generate realistic samples, although they often suffer from mode collapse. Neekhra et al. (2023) apply GANs at multiple administrative levels for national-scale synthesis. VAEs, introduced to this domain by (Borysov et al. 2019), provide more stable training by learning compressed latent representations, albeit with reduced diversity compared to GANs. Qian et al. (2023) present Synthcity, a framework supporting multiple ML methods for tabular data synthesis and evaluation across diverse applications. Jeong et al. (2016) introduced a copula-based approach to capture complex dependencies between attributes. Alonso-Betanzos et al. (2021) utilized decision trees for this purpose. Jiang et al. (2021) developed a method incorporating social networks into geographically explicit agent-based models. Dyer et al. (2024) presented a framework aligning synthetic populations with target scenarios in agent-based models. Albiston et al. (2024) proposed a neural network approach, while Rahman and Fatmi (2023) introduced a Bayesian Network and generalized raking technique. Tuccillo et al. (2023) developed UrbanPop, a spatial microsimulation framework for demographic analysis.

Our approach leverages differentiable programming and neural networks to significantly improve the generation of synthetic populations by enhancing accuracy, scalability, and adaptability. Unlike traditional rule-based methods like IPF or combinatorial optimization, which require manual tuning and lack flexibility, our model adapts continuously to new data, reducing computational overhead and integrating auxiliary sources for greater robustness.

3 PROPOSED METHOD

3.1 Differentiable programming

Differentiable programming enhances traditional programming by enabling the computation of gradients through the construction of computational graphs. These graphs represent the flow of data and operations. A forward pass is used to compute the outputs, while a backward pass applies automatic differentiation using the chain rule to propagate gradients through the graph. This enables the use of gradient-based optimization methods, such as those employed to train neural networks, where the gradients guide updates to model parameters to minimize loss functions (Blondel and Roulet 2024).

3.2 Synthetic Population Differentiable Model

The mathematical formulation of a synthetic population is given below. Let $P_{\text{individual}}$ and $P_{\text{household}}$ be the set of individuals and the set of households, respectively, in the synthetic population:

$$\begin{aligned} P_{\text{individual}} &= \{i_1, i_2, \ldots, i_n\} \\ P_{\text{household}} &= \{h_1, h_2, \ldots, h_m\} \end{aligned} \tag{1}$$

Each individual i_k is characterized by a set of demographic attributes $\mathscr{A}_i$, such as age, sex, ethnicity, religion, marital status, and qualification. Each household h_l is characterized by attributes such as household size, composition, ethnicity, and religion of the household reference person:

$$\begin{aligned} \mathscr{A}_i &= \{A_{i1}, A_{i2}, \ldots, A_{im}\} \\ \mathscr{A}_h &= \{A_{h1}, A_{h2}, \ldots, A_{hn}\} \end{aligned} \tag{2}$$

The relationship between individuals and households can be expressed as a mapping $\mathscr{M} : P_{\text{individual}} \rightarrow P_{\text{household}}$

In the context of synthetic population generation, differentiable programming transforms discrete census data into continuous forms, making them amenable to gradient-based optimization techniques. This transformation involves expressing categorical variables and counts as probabilities and expected values, which can then be optimized to match real-world distributions more accurately. This framework provides a robust basis for training a model to generate a synthetic population that accurately reflects complex multi-dimensional joint distributions from real-world demographic data, ensuring that the synthetic data are realistic and useful for simulation and analysis in various applications. We define a feed-forward neural network f parameterized by weights Θ, which takes an input tensor and produces two output tensors $P_{\text{individual}}$ and $P_{\text{household}}$ representing demographic probabilities:

$$P_{\text{individual}},\, P_{\text{household}} = f(X;\Theta) \tag{3}$$

with X as the input, hidden layers comprising linear transformations, batch normalization, ReLU activation, and an output layer applying soft-max to each attribute category. ReLU (Rectified Linear Unit) is an activation function defined as $\text{ReLU}(x) = \max(0,x)$, which introduces non-linearity by zeroing out negative inputs. The output P is structured to include several segments, each representing a probability distribution for a demographic attribute. These attributes are: age, sex, ethnicity, religion, marital status and qualification, and household size, household composition, ethnicity, and religion of the household reference person, and a set of assignments of individuals to the household:

$$\begin{aligned} P_{\text{individual}} &= P_{\text{age}},\, P_{\text{sex}},\, P_{\text{ethnicity}},\, P_{\text{religion}},\, P_{\text{marital status}},\, P_{\text{qualification}} \\ P_{\text{household}} &= P_{\text{size}},\, P_{\text{composition}},\, P_{\text{ethnicity}},\, P_{\text{religion}},\, P_{\text{assignment}}. \end{aligned} \tag{4}$$

Where P_{size} represents the probabilities associated with different household sizes, ensuring that the distribution aligns with demographic data. $P_{\text{composition}}$ is a vector of probabilities corresponding to the categories of household composition as given in Table 1, where C = Children, A = Adults, E = Elders. $P_{\text{ethnicity}}$ denotes the probabilities for the ethnicity categories of the household reference person. P_{religion} indicates the probabilities for the religion categories of the household reference person. $P_{\text{assignment}}$ encapsulates the probabilities of a set of individuals assigned to each household category using an algorithm that deals with the assignments, discussed later in Algorithm 1. Each $P_{\text{attribute}}$ is a softmax probability vector corresponding to the categories of that attribute. A cross table, also known as a contingency table, is a type of table in a matrix format that displays the (multivariate) frequency distribution of the variables. In census data a cross table is a triplet of three selected attributes.

3.2.1 Gumbel-Softmax Encoding

We employ the Gumbel-Softmax encoding, to handle the sampling of discrete distributions and to enable differentiable sampling from discrete categorical variables, which is essential for backpropagation in neural networks:

$$\mathbf{y} \sim \text{softmax}\left(\frac{\log P + G}{\tau}\right) \tag{5}$$

where G are i.i.d Gumbel noise vectors and τ is a temperature parameter controlling the discreteness of the output. Given a categorical distribution parameterized by logits $\mathbf{z} = (z_1, z_2, ..., z_K)$, where K represents the number of classes, the Gumbel-Softmax sample $\mathbf{y} = (y_1, y_2, ..., y_K)$ is computed using Gumbel distribution, Gumble noise and a softmax function (Jang et al. 2016). The probability density function (PDF) of the Gumbel distribution is given by:

$$f(g_i) = \frac{1}{\beta}\exp\left(\frac{g_i-\mu}{\beta} - \exp\left(\frac{g_i-\mu}{\beta}\right)\right) \tag{6}$$

where g_i is a sample from the Gumbel distribution for a given variable (e.g., personal or household attribute), μ is the location parameter, and β is the scale parameter. μ determines where the peak of the

distribution occurs along the primary axis. The scale parameter controls the spread or variability of the distribution. For each attribute, we set $\mu = 0$ and $\beta = 1$. Gumbel noise $\varepsilon = (\varepsilon_1, \varepsilon_2, ..., \varepsilon_K)$ is generated by sampling from the Gumbel distribution.

3.2.2 Aggregation, Nested Loss Function and Optimization

We define a *nested loss function* to jointly optimize over multiple levels of aggregation. Specifically, this loss incorporates errors from both marginal distributions (e.g., Sex, Age) and higher-order cross-tabulations (e.g., Sex by Age, Religion by Sex by Age). This nested structure allows the model to capture fine-grained dependencies between attributes while preserving overall demographic targets. To compare the attributes with the cross tables from census data, we aggregate the individual and household attributes to form distributions that can be compared with the census data distributions. Let C_i represent the cross table from the census data for a specific attribute combination, and let A_i represent the aggregated attributes from the synthetic population. The comparison is done using the Root Mean Squared Error (RMSE):

$$L = \sum_{L_i} \sqrt{\sum (A_i - C_i)^2} \tag{7}$$

where L is the total loss (or nested loss), L_i is the loss of an attribute, A_i are the aggregated values, and C_i is the actual data from the cross table and our objective is to minimize the loss function. Our training objective uses a root mean squared error (RMSE) loss over aggregated counts derived from cross-tabulated population attributes (see Eq. 7). This loss measures the Euclidean distance between predicted and target aggregates and guides the optimization process. However, for evaluation and visualization, we report accuracy scores, computed as:

$$\text{accuracy} = 1 - \frac{|P - T|}{T} \times 100\% \tag{8}$$

where P is the predicted count and T is the target count for a given attribute category.

- m is the total number of members
- Age_i is the age group of the i-th individual, where C = Child, A = Adult, E = Elder
- M_i is the marital status of the i-th individual
- The FM category denotes a male–female married couple, while FC includes all cohabiting couples, regardless of gender, in line with census definitions.

3.2.3 Individuals and Household Generation

Algorithm 1 generates a synthetic population by learning demographic distributions from census data using a feed-forward neural network. A separate network is trained for each demographic attribute, with input tensors $\mathbf{X}_i$ and priors $\mathbf{P}_i$, $\mathbf{H}_i$. Each network comprises linear layers, batch normalization, and ReLU activations. During training, the network predicts the attribute distributions, aggregates them to compare with the cross tables $\mathbf{C}_i$, and updates the parameters through forward and backward propagation. At each epoch, the *GeneratePopulation* function samples synthetic individuals and households using Gumbel-Softmax, forming tensors $\mathbf{P}_{\text{ind}}$ and $\mathbf{P}_{\text{hh}}$. Once generated, individuals are assigned to households based on attribute matching and composition rules, as shown in table 1. Where, m is the total number of members, Age_i denotes the age group of the i-th individual, where C stands for Child, A for Adult, and E for Elder, and M_i represents the marital status of the i-th individual. The FM category refers to a male-female married couple, while FC includes all cohabiting couples, regardless of gender, in line with census definitions.

For each household $h \in \mathbf{P}_{\text{hh}}$, n individuals are sampled from $\mathbf{P}_{\text{ind}}$ such that $n = h_{\text{size}}$. These samples are filtered to match the household's ethnicity and religion, and then evaluated against predefined composition rules which check attributes such as age, sex, and marital status. For example, in a 1FM-1C household,

Algorithm 1 Generation of Individuals and Households

Input: Input tensor $\mathbf{X}_i$, set of attributes $\mathbf{P}_i$, $\mathbf{H}_i$, cross tables $\mathbf{C}_i$
Output: Converged $\mathbf{P}_{\text{ind}}$ and $\mathbf{P}_{\text{hh}}$

Define a feed-forward neural network with a forward pass function $P = f(X;\Theta)$
Initialize the network structure with input nodes, hidden layers, output nodes, linear layers, batch normalization, and ReLU activation
Initialize the weights in the network using $\mathbf{P}_i$ and $\mathbf{H}_i$
▷ **Training Loop**
Initialize neural networks for each demographic attribute
Initialize optimizer and learning rate scheduler
for epoch $= 1$ to maxEpoch **do**
    $(\mathbf{P}_i, \mathbf{H}_i) \leftarrow$ GeneratePopulation(X_i)
    Aggregate probabilities based on $\mathbf{C}_i$
    Perform forward propagation
    Compute total loss $\mathbf{L} \in \mathrm{L_i}$ using aggregated outputs $\mathbf{A}_i$
    Perform backward propagation and compute gradients
    Update network parameters $\theta_{\text{t(i)}}$
end for
▷ Assign individuals to households based on attributes and composition rules
for each household $h \in \mathbf{P}_{\text{hh}}$ **do**
    ▷ Sample individuals as per household size and match attributes
    ▷ Check composition rule R for household compatibility
    Sample n individuals from $\mathbf{P}_{\text{ind}}$ such that:
    $n = h_{\text{size}} \wedge \forall p \in n:\ p_{\text{eth}} = h_{\text{ethnicity}} \wedge p_{\text{rel}} = h_{\text{religion}} \wedge R(p,h)$
    Assign individuals n to household h
end for
return $\mathbf{P}_{\text{ind}}$ and $\mathbf{P}_{\text{hh}}$
▷ Function to generate population tensors via sampling distributions
function GENERATEPOPULATION(X_i)
    Initialize tensors $\mathbf{P}_{\text{ind}}$ and $\mathbf{P}_{\text{hh}}$ using input data
    for each demographic attribute in $\mathbf{P}_i$ and $\mathbf{H}_i$ **do**
        Compute logits using corresponding neural network
        Apply Gumbel-Softmax sampling to convert logits to probabilities
        Stack sampled values into final output tensors
    end for
    return $\mathbf{P}_{\text{ind}}$, $\mathbf{P}_{\text{hh}}$
end function

one married male, one married female (both adults), and one unmarried child are sampled. Once the rule is satisfied, individuals are assigned to the household. Our approach is implemented in PyTorch which uses CUDA devices for GPU acceleration, hence the code is scalable, modular, adaptable and reusable, allowing seamless customization for different datasets and regional contexts. In our experiments, the synthetic population consisted of 7,209 individuals and 3,167 households, matching the size of a real MSOA. The model was trained using a feed-forward neural network with three hidden layers and approximately 120k trainable parameters. Training on this dataset converged within 300 epochs (approximately 45 seconds on a standard laptop CPU), demonstrating good scalability for small to medium-sized populations.

Table 1: Household Composition Structures and Rules

Composition	Description and Rules
1PE	One-person household: Aged 65 and over **Rule:** $m = 1 \wedge \forall i, Age_i \in E$
1PA	One-person household: Aged 18 to 64 **Rule:** $m = 1 \wedge \forall i, Age_i \in A$
1FM-0C	One family only: Married couple: No children **Rule:** $m = 2 \wedge \forall i, Age_i \notin C \wedge M_i =$ 'Married'
1FM-1C	One family only: Married couple: One dependent child **Rule:** $m = 3 \wedge \exists i,j,k \wedge\ (i \neq j \neq k) \wedge Age_i, Age_j \notin C \wedge M_i = M_j =$ **'Married'** $\wedge \forall k, \Rightarrow Age_k \in C$
1FM-nC	One family only: Married couple: Two or more dependent children **Rule:** $m \geq 3 \wedge \exists i,j,k \wedge\ (i \neq j \neq k) \wedge Age_i, Age_j \notin C \wedge M_i = M_j =$ 'Married' $\wedge \forall k, \Rightarrow Age_k \in C$
1FM-nA	One family only: Married couple: All children non-dependent **Rule:** $m \geq 3 \wedge \exists i,j,k \wedge\ (i \neq j \neq k) \wedge Age_i, Age_j \notin C \wedge M_i = M_j =$ 'Married' $\wedge \forall k, \Rightarrow Age_k \in A$
1FC-0C	One family only: Cohabiting couple: No children **Rule:** $m = 2 \wedge \forall i, Age_i \notin C \wedge \forall i, M_i \neq$ 'Married'
1FC-nC	One family only: Cohabiting couple: Two or more dependent children **Rule:** $m \geq 3 \wedge \exists i,j,k \wedge\ (i \neq j \neq k) \wedge Age_i, Age_j \notin C \wedge M_i, M_j \neq$ 'Married' $\wedge \forall k, \Rightarrow Age_k \in C$
1FL-1C	One family only: Lone parent: One dependent child **Rule:** $m = 2 \wedge \exists i, j\ Age_i \notin C \wedge \exists i, M_i \in \{$'Separated','Widowed','Divorced'$\} \wedge \forall j, j \neq i \Rightarrow Age_j \in C)$
1FL-nC	One family only: Lone parent: Two or more dependent children **Rule:** $m \geq 3 \wedge \exists i, j\ Age_i \notin C \wedge \exists i, M_i \in \{$'Separated','Widowed','Divorced'$\} \wedge \forall j, j \neq i \Rightarrow Age_j \in C)$

4 RESULTS AND ANALYSIS

This section illustrates the simulation results of a UK case study, presented to demonstrate our proposed approach. In this case study we obtained census data of Oxford City from the office for national statistics NOMIS, at the scale of Middle layer Super Output Areas (MSOA), and selected one MSOA: E02005941, to generate the synthetic population. The results displayed in Figure 1 (a) illustrate the output distributions of Persons based on selected attributes: Sex, Age, Ethnicity, Religion, Marital status and Qualification. Each generated attribute is displayed in blue and compared with the targeted distribution displayed in red. The accuracy calculated using Root Mean Squared Error (RMSE) is highlighted. Similarly, the output distributions of Households are displayed in Figure 1 (b). All generated persons are assigned to the corresponding households according to the characteristic match and composition rules. Notice, that the ethnicities and religions in the household distributions characterise the household reference persons. The generation results show conformity to the target distributions.

4.1 Verification and Validation

To verify that the structure of the output distributions is correct, i.e., that household sizes and person allocations are consistent with the composition structures, we formalize a rule-based verification approach using a set of logical rules for each household composition type, as shown in Table 1. We executed unit tests on output table rows to assess compliance with these logical rules. Figure 2 reports a 3.3% error rate, showing mismatches in household sizes (red) and incorrect individual-to-household assignments (blue), grouped by household composition type. For validation, we used RMSE to compare the predicted values against actual census data from NOMIS, including person-level cross tables lc1117ew, dc2101ew, dc2107ew, dc1107ew, dc5102ew, and household-level tables dc1201ew, dc1202ew. As there are multiple attributes, we evaluate each combination (triple) of attributes with a cross table available in the UK census data, and calculate the respective RMSE. The overall accuracy of the model is the sum of all RMSE values computed in each iteration. Figure 3 shows that the computed curves (blue bars) and target curves (red bars) are closely aligned. Similarly, Figure 4 illustrates the validation for the generated households. This shows a good fit of the model

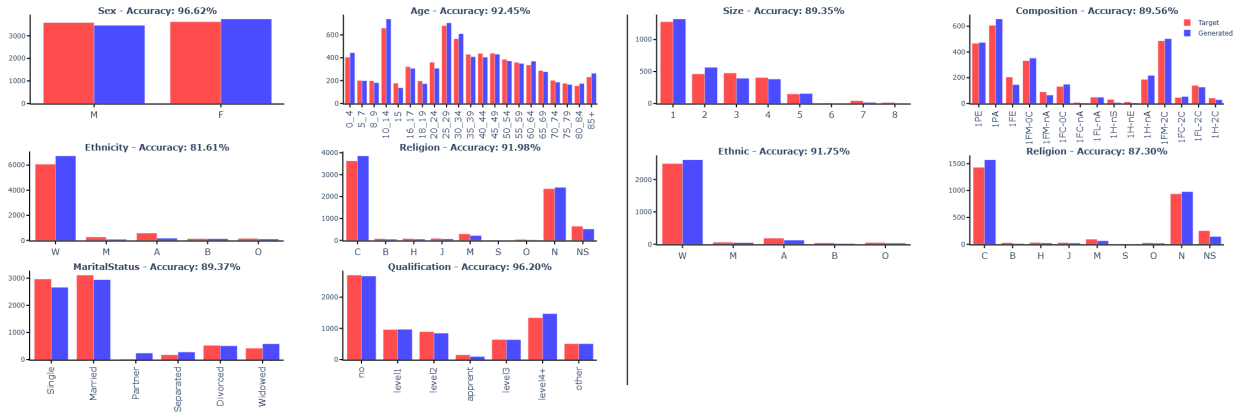

Figure 1: (left) Persons targeted vs computed [Sex, Age, Ethnicity, Religion, Marital Status, Qualification] (right) Household Targeted vs Computed - [Size, Composition, Ethnicity, Religion].

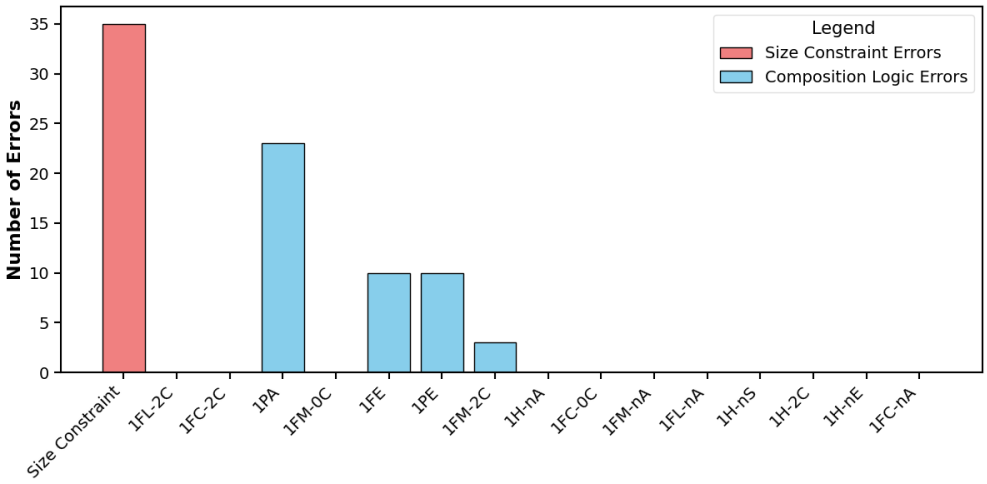

Figure 2: Verification results (errors 3.3%).

in accurately predicting the data. The X-axis of these plots represents the number of keys generated by the product of the attribute fields. E.g., $Sex \times Age = \{\text{Male}, \text{Female}\} \times \{0\text{–}4,\ 5\text{–}7,\ \ldots,\ 80\text{–}84,\ 85+\} = 42$ points

4.1.1 Micro-data Access and Validation Constraints

In the UK, validation of synthetic populations at the micro level is constrained by restricted access to individual-level census data. Publicly available resources such as NOMIS provide only aggregate statistics, while access to micro-data is governed by strict privacy, ethical, and legal controls. Relevant datasets include the UK Longitudinal Study (UKLS), the Secure Research Service (SRS) of the Office for National Statistics, and the Understanding Society dataset (UKHLS). These datasets require secure access, and approved research protocols, and are often only usable in controlled environments. Given these access limitations, aggregate-level validation remains a standard practice in UK-based synthetic population studies (Wu et al. 2022). In this context, our inclusion of a logical rule-based verification framework represents an innovative step toward micro-level structural assessment and logical correctness. Although international datasets such as the US Public Use Microdata Sample (PUMS) provide more accessible micro-data, used

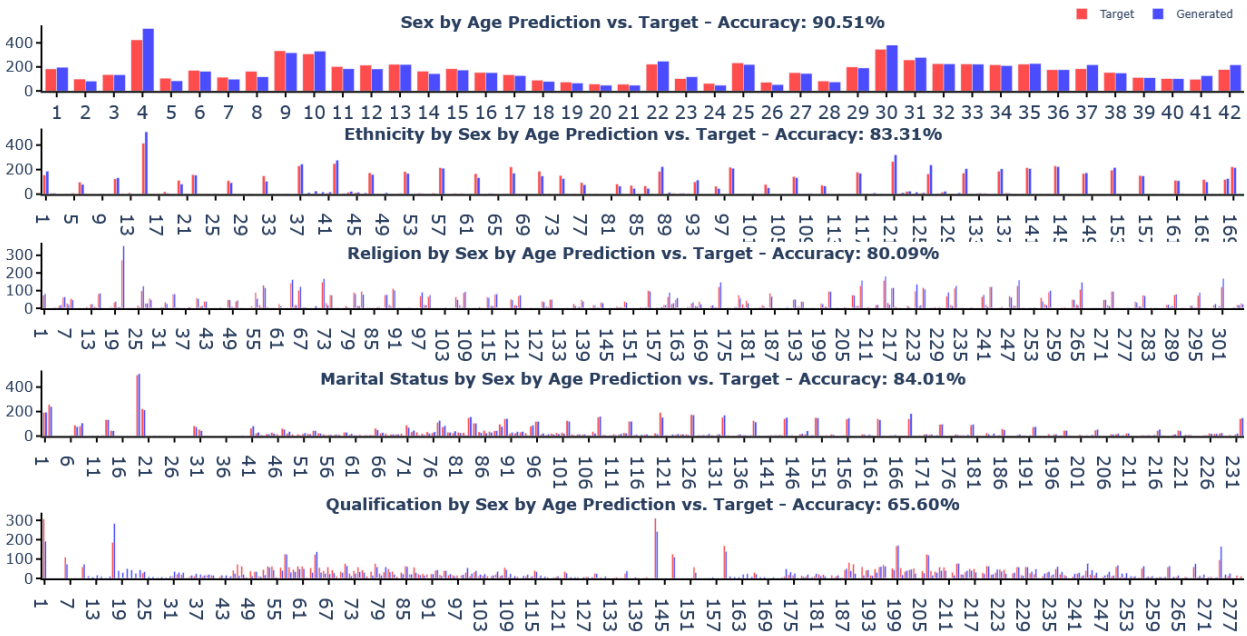

Figure 3: Persons validation using cross-tables.

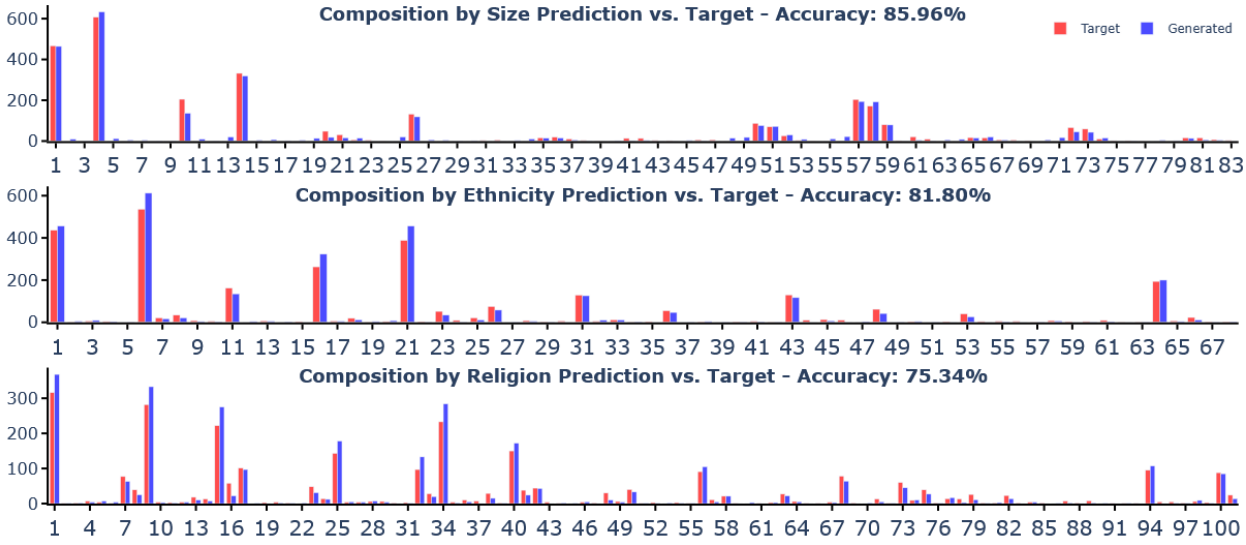

Figure 4: Households validation using cross-tables.

by (Qian et al. 2024) for the synthesis of households, they are not directly applicable to UK-based models due to differences in demographic structure and household typologies. Future work will aim to integrate secure-access UK micro-data to support deeper structural validation, where permitted.

5 CONCLUSION

This paper presents a novel differentiable programming framework for generating highly accurate and scalable hierarchical synthetic populations, specifically designed for modeling complex societal systems. By leveraging feedforward neural networks and Gumbel-Softmax encoding, our methodology transforms

aggregated census and survey data into continuous, differentiable forms, enabling the generation of synthetic populations that accurately reflect real-world demographic and socio-economic structures. The framework captures the intricate hierarchical relationships within populations, ensuring that these representations are realistic and adaptable across different scales. The key contributions of this work include the development of a scalable and flexible synthetic population generation tool, which significantly improves the representational accuracy required for complex societal system simulations. Our approach integrates an in-loop verification and validation process, ensuring that the synthetic populations adhere to logical consistency and align with empirical data. The case study using UK census data demonstrates the method's robustness, providing synthetic populations that closely match real-world distributions across various demographic and socio-economic dimensions. By improving the accuracy and scalability of synthetic population generation, our framework provides researchers and policymakers with a powerful tool for simulating complex societal dynamics. In future work, we plan to extend the framework to incorporate additional features in synthesizing populations such as activities and movements, and to explore its applicability across diverse geographic regions. By continuing to refine and scale our synthetic population generation method, we aim to further enhance its utility in modeling the dynamic complex societal systems.

ACKNOWLEDGMENTS

This research was supported by a UKRI AI World Leading Researcher Fellowship awarded to Wooldridge (grant EP/W002949/1). M.Wooldridge and A. Calinescu acknowledge funding from Trustworthy AI - Integrating Learning, Optimization and Reasoning (TAILOR) (https://tailor-network.eu/), a project funded by European Union Horizon2020 research and innovation program under Grant Agreement 952215.

REFERENCES

Aemmer, Z., and D. MacKenzie. 2022. "Generative Population Synthesis for Joint Household and Individual Characteristics". *Computers, Environment and Urban Systems* 96.

Albiston, G., T. Osman, and D. Brown. 2024. "A Neural Network Approach for Population Synthesis". *Simulation* 100(8):823–847.

Alonso-Betanzos, A., B. Guijarro-Berdiñas, A. Rodríguez-Arias, and N. Sánchez-Maroño. 2021. "Generating a Synthetic Population of Agents Through Decision Trees and Socio Demographic Data". In *International Work-Conference on Artificial Neural Networks, 16–18 June 2021, Madeira, Portugal*, 128–140.

Antoni, J.-p., G. Vuidel, and O. Klein. 2017. "Generating a Located Synthetic Population of Individuals, Households, and Dwellings". *Luxembourg Institute of Socio-Economic Research (LISER)*.

Blondel, M., and V. Roulet. 2024. "The elements of differentiable programming". *arXiv 2403.14606*.

Borysov, S. S., J. Rich, and F. C. Pereira. 2018. "Scalable population synthesis with deep generative modeling". *arXiv 1808.06910*.

Borysov, S. S., J. Rich, and F. C. Pereira. 2019. "How to Generate Micro-Agents? A Deep Generative Modeling Approach to Population Synthesis". *Transportation Research Part C: Emerging Technologies* 106:73–97.

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

Crooks, A., N. Malleson, E. Manley, and A. Heppenstall. 2015. "Agent-Based Modeling and Geographical Information Systems". *Geocomputation: A Practical Primer. SAGE Publications Ltd, Thousand Oaks, CA*.

Dyer, J., A. Quera-Bofarull, N. Bishop, J. D. Farmer, A. Calinescu, and M. Wooldridge. 2024. "Population Synthesis as Scenario Generation for Simulation-Based Planning under Uncertainty". In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '24, 490–498. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Gilbert, N. 2019. *Agent-based models*. Thousand Oaks, CA: Sage Publications.

Jang, E., S. Gu, and B. Poole. 2016. "Categorical Reparameterization with Gumbel-Softmax". *arXiv,1611.01144*.

Jeong, B., W. Lee, D.-S. Kim, and H. Shin. 2016. "Copula-Based Approach to Synthetic Population Generation". *PLoS ONE* 11(8) https://doi.org/https://doi.org/10.1371/journal.pone.0159496.

Jiang, N., A. T. Crooks, H. Kavak, A. Burger, and W. G. Kennedy. 2022. "A Method to Create a Synthetic Population with Social Networks for Geographically-Explicit Agent-Based Models". *Computational Urban Science* 2(1):7 https://doi.org/10.1007/s43762-022-00034-1.

Jiang, N., H. Kavak, W. G. Kennedy, and A. T. Crooks. 2021. "Generation of Reusable Synthetic Population and Social Networks for Agent-Based Modeling". In *2021 Annual Modeling and Simulation Conference (ANNSIM)*, 1–12. IEEE.

Jordon, J., L. Szpruch, F. Houssiau, M. Bottarelli, G. Cherubin, C. Maple, *et al*. 2022. "Synthetic Data–What, Why and How?". *arXiv, 2205.03257*.

Kim, E.-J., and P. Bansal. 2023. "A Deep Generative Model for Feasible and Diverse Population Synthesis". *Transportation Research Part C: Emerging Technologies* 148.

Kim, J., and S. Lee. 2016. "A Simulated Annealing Algorithm for the Creation of Synthetic Population in Activity-Based Travel Demand Model". *KSCE Journal of Civil Engineering* 20:2513–2523.

Macal, C. M. 2016. "Everything You Need to Know About Agent-Based modeling and Simulation". *Journal of Simulation* 10:144–156.

Mahmood, I., N. Bishop, A. Calinescu, M. Wooldridge, and I. Zachos. 2024. "A Multi-Objective Combinatorial Optimisation Framework for Large Scale Hierarchical Population Synthesis". *arXiv 2407.03180*.

Malleson, N., M. Birkin, D. Birks, J. Ge, A. Heppenstall, E. Manley, *et al*. 2022. "Agent-Based Modeling for Urban Analytics: State of the Art and Challenges". *AI Communications* 35(4):393–406.

Neekhra, B., K. Kapoor, and D. Gupta. 2023. "Synthpop++: A Hybrid Framework for Generating A Country-Scale Synthetic Population". *arXiv 2304.12284*.

Ponge, J., M. Enbergs, M. Schüngel, B. Hellingrath, A. Karch, and S. Ludwig. 2021. "Generating Synthetic Populations Based on German Census Data". In *2021 Proceedings of the Winter Simulation Conference (WSC),* https://doi.org/https://doi.org/10.1109/WSC52266.2021.9715369.

Prédhumeau, M., and E. Manley. 2023. "A Synthetic Population for Agent-Based Modeling in Canada". *Scientific Data* 10(1).

Qian, X., U. Gangwal, S. Dong, and R. Davidson. 2024. "A Deep Generative Framework for Joint Households and Individuals Population Synthesis". *arXiv 2407.01643*.

Qian, Z., B.-C. Cebere, and M. van der Schaar. 2023. "Synthcity: Facilitating Innovative Use Cases of Synthetic Data in Different Data Modalities". *arXiv 2301.07573*.

Rahman, M. N., and M. R. Fatmi. 2023. "Population Synthesis Accommodating Heterogeneity: A Bayesian Network and Generalized Raking Technique". *Transportation Research Record* 2677(6):41–57 https://doi.org/10.1177/03611981221144289.

Railsback, S. F., and V. Grimm. 2019. *Agent-Based and Individual-Based Modeling: A Practical Introduction*. Princeton: Princeton University Press.

Taylor, S. 2014. *Agent-Based Modeling And Simulation*. London: Palgrave Macmillan.

Tuccillo, J., R. Stewart, A. Rose, N. Trombley, J. Moehl, N. Nagle *et al*. 2023. "UrbanPop: A Spatial Microsimulation Framework for Exploring Demographic Influences on Human Dynamics". *Applied Geography,* 151 https://doi.org/10.1016/j.apgeog.2022.102844.

Wu, G., A. Heppenstall, P. Meier, R. Purshouse, and N. Lomax. 2022. "A Synthetic Population Dataset for Estimating Small Area Health and Socio-Economic Outcomes in Great Britain". *Scientific Data* 9(1):19.

AUTHOR BIOGRAPHIES

IMRAN MAHMOOD is a Senior Postdoc Researcher at the Department of Computer Science, University of Oxford. His research interests include agent-based modeling, and Synthetic population generation. His email address is imran.hashmi@cs.ox.ac.uk and website is https://www.cs.ox.ac.uk/people/imran.hashmi/. He can be contacted for the code and input data.

ANISOARA CALINESCU is an Associate Professor at the Department of Computer Science, University of Oxford. Her main research interests are in modeling and reasoning about complex networked systems; the theory, practice, and methodology of agent-based modeling, including model calibration and validation, and using models for predictions. Her e-mail address is ani.calinescu@cs.ox.ac.uk and website is https://www.cs.ox.ac.uk/people/ani.calinescu/.

MICHAEL WOOLDRIDGE is the Ashall Professor of the Foundations of AI at the University of Oxford, with over 30 years of research experience and more than 450 publications, including nine books translated into eight languages. He is a Fellow of ACM, AAAI, and EurAI, a member of Academia Europaea, and President Elect of AAAI. He previously served as President of EurAI and IJCAI, and is currently co-editor-in-chief of the Artificial Intelligence journal. His awards include the BCS Lovelace Medal (2020), AAAI's Outstanding Educator Award (2021), and EurAI's Distinguished Service Award (2023). His email address is michael.wooldridge@cs.ox.ac.uk and website is https://www.cs.ox.ac.uk/people/michael.wooldridge/.