Proceedings of the 2023 Winter Simulation Conference
C. G. Corlu, S. R. Hunter, H. Lam, B. S. Onggo, J. Shortle, and B. Biller, eds.

GENERATING POPULATION SYNTHESIS USING A DIFFUSION MODEL

Jaewoong Kang
Young Kim
Muhammad Mu'az Imran
Gi-sun Jung
Yun Bae Kim

Department of Industrial Engineering
Sungkyunkwan University
22066 Seo-bu Street
Suwon, 16419, REPUBLIC OF KOREA

ABSTRACT

Owing to the increase in computing power, large-scale agent-based modeling (ABM) has been increasingly used in various fields. However, a complete and detailed individual population is challenging to obtain because of confidentiality concerns. Thus, modelers must adopt population synthesis to emulate the joint distribution of individual-level attributes of the actual population in the region of interest. Traditional population synthesis methods often exhibit issues regarding scalability and sampling zero. Therefore, this paper presents the use of a deep generative model called the denoising diffusion probabilistic model to generate new samples. Our proposed method uses the characteristics of deep generative model of generation from noise to generate a synthetic population, including sampling zero. In the experimental results, the standardized root mean squared error of our proposed model performed 2.130, which outperformed 2.381 of the deep learning-based population synthesis method, VAE, and 7.620 of the traditional population synthesis method, MCMC.

1 INTRODUCTION

Policy analysis is a complex process that considers various factors such as social and environmental aspects. Agent-based modeling (ABM) has emerged as a powerful tool for assisting policymakers in making informed decisions by narrowing the possible rational decisions that may be impractical and costly to test in the real world. An ABM is a complex system comprising an aggregation of autonomous agents within a controlled environment that independently make decisions based on predefined rules governing their behaviors (Bonabeau 2002; Wurzer et al. 2015). The use of ABM can assist researchers and policymakers in understanding the social heterogeneity across individuals within a population in a geospatial structure regarding the impact of different policies in the short, medium, or long term (Macal and North 2009). ABM can highlight the consequences of a policy design, which helps modelers understand the implicit etiology of the imposed policy on individuals or prognosticate various possible prospective scenarios that may unfold (Hammond 2015).

Additionally, the primary input of an ABM is artificial agents or synthetic populations that represent individuals and households in a specified region of interest for localized analyses. The increasing amount of literature recognizes the importance of population synthesis in ABM in different fields such as epidemiology (Jung et al. 2017), transportation (Aziz et al. 2018), urban development (Liu et al. 2021), energy (Tröndle and Choudhary 2017), social studies (Vidyattama et al. 2013), and disaster management (Saadi et al. 2018). Two types of data are used to generate population synthesis: census data (marginal

information in the contingency table format) and microdata samples (individual information with detailed attributes). However, individual- and household-level data for the entire population in a given region of interest are often impossible to obtain because of privacy and confidentiality concerns. Because of this issue, only a limited percentage of population data is permitted in most countries. Some countries such as Switzerland have made their entire census datasets public (Gulshan et al. 2016).

Population synthesis has traditionally been approached using synthetic reconstruction (SR), combinatorial optimization (CO), simulations-, and sample-free methods (Ye et al. 2017). SR uses information from the census and microdata samples to compute the weights of the joint distribution of the population in a given zone. CO is often used to allocate synthetic individuals to the right combination of households that best fits the marginals. The simulation-based method considers only the microdata sample and derives the joint distribution of all the attributes by approximating the probability for each combination. In most countries where microdata samples are unavailable, the sample-free approach aggregates various census data as inputs to estimate the marginal distributions and/or conditional distributions of partial attributes (Ye et al. 2017). In recent years, deep generative models have emerged, such as the variational autoencoder (VAE) (Borysov et al. 2019) and generative adversarial network (GAN) (Kim and Bansal 2023).

The traditional methods cannot effectively generate a synthetic population with many attributes due to the "curse of dimensionality" that creates the existence of disjoint probability distributions in a latent space. Thus, this case is referred to as a scalability problem. Another widely known challenge in population synthesis studies is the sampling zero issues (Choupani and Mamdoohi 2016; Fournier et al. 2021). Sampling zero can be defined as individuals with feasible attributes who are nonexistent in the original sample but exist in the actual population. Consequently, a model capable of generating individuals with a viable and unique combination of attributes, as in the original sample rather than a direct replication, is required to mimic the actual joint probability distributions of the population in the region. This study attempts to address these challenges.

The main contributions of the paper are as follows:

- We propose a new scalable and robust method for population synthesis based on the denoising diffusion probabilistic model (DDPM) (Ho et al. 2020) derived from deep generative modeling (DGM). To the best of our knowledge, the experimental work presented through the current study is the first to investigate how the DDPM generates a synthetic population.
- Other DGM methods, such as the VAE, fail to provide consistent results, which will be thoroughly discussed in the latter section. We showed that the DDPM converged more consistently, exhibiting less variance when tested with three replicates.

In Section 2, we present the relevant literature review. Section 3 describes the overall flow of population synthesis using a diffusion model. In Section 4, we present two experiments to prove the superiority of the diffusion-model-based synthetic population and the efficiency of covering the sampling zeros problem. Finally, Section 5 presents the conclusions and discusses directions for further research.

2 RELATED WORKS

2.1 Population Synthesis

Traditionally, population synthesis has been approached using iterative proportional fitting (IPF) as introduced by Beckman et al. 1996. IPF involves two stages: fitting and allocation. In most cases, the availability of census data varies between population attributes based on spatial resolution or statistical area (SA) level. For instance, the joint distribution from the k-way cross-tabulation (CT) of k attributes of a given higher statistical area (SA) level may not be available; however, its marginal distributions can be obtained (seeds). In addition, other census data with lower SA

levels may contain the full CT, describing the population of a small group within the region mentioned above. Subsequently, in the fitting stage of the IPF, the CT of the higher SA level can be fine-tuned based on the information of multiple lower SAs levels using the IPF estimator and seed. Subsequently, the allocation stage generally involves the integration of cell weights into integers, and finally, allocating the individuals to a realistic household composition based on the population distribution (Ye et al. 2016). Although some studies have attempted to solve sampling zero problems using IPF by replacing the zero cell weights with an extremely small positive value (e.g., 0.001), this creates bias and generates synthetic individuals with infeasible combinations of attributes (structural zero) (Ye et al. 2017). Ideally, modelers desire a synthetic population that includes normal samples and sampling zeros, while keeping the structural zeros as few as possible (Kim and Bansal 2023) .

Several studies have been conducted using the Markov-chain Monte Carlo (MCMC) simulation-based method to generate synthetic populations (Farooq et al. 2013; Casati et al. 2015; Gong et al. 2021). The fundamentals of MCMC are based on a stochastic probabilistic framework that draws samples from conditional distributions. The main advantage of MCMC is that it can handle sampling zeros and high-dimensional data; however, it has an increased risk of being trapped in local minima during the initialization phase. In addition, MCMC can generate a synthetic population based on partial conditionals and can still outperform IPF.

Another study used a probabilistic graphical model called the Bayesian network (BN) to generate a joint probability distribution function of a set of attributes selected through a scoring approach (Sun and Erath 2015). They compared the performance of their BN method with that of other known methods, namely, IPF and MCMC. It was found that even when a small percentage of microdata samples were used, BN outperformed the other two methods. However, these two methods outperformed BN when the size of the microdata sample exceeded approximately 40% of the actual population, which was impossible to obtain. The authors extended their prior work by including household structures such as relationship status in a hierarchical manner (Sun et al. 2018).

2.2 Deep Generative Model

Recent trends in deep learning have led to an increasing number of DGM studies on population synthesis. Borysov et al. (2019) pioneered the use of a VAE to generate artificial agents, which have the potential to become a workhorse for large-scale ABM studies with detailed population characteristics. The VAE is built based on neural networks in the encoding and decoding layers and imposes a stochastic sampling procedure prior to the latent space. The primary purpose of imposing Gaussian noise prior to the latent space is to generate a smoother representation of the input data, rather than direct reconstruction. Therefore, the VAE can efficiently address sampling zero problems. The authors compared their proposed methodology with the other traditional generative models, such as MCMC and BN. To assess the superiority of VAE in dealing with the "curse of dimensionality" problem, they divided their experiment into three different cases of varying attribute sizes (4, 21, and 47) and discovered that the traditional generative models outperformed the VAE in a low-dimensional case. As the size of the attributes increased, VAE outperformed the others. One of the limitations of this study is that it does not consider the structural zero problems. Thus, individuals with nonrealistic combinations of attributes exist in the synthetic population.

Garrido et al. (2020) employed an extended version of the GAN, which is another DGM method, namely Wasserstein GAN (WGAN). The intuition behind GAN as a generative model is that the two neural networks (generator and discriminator) compete. The generator attempts to trick the discriminator by synthesizing false data based on the stochastic sampling process or Gaussian noise, and the discriminator attempts to distinguish between genuine and counterfeit data generated by the generator. Thus, the model can be trained to generate a synthetic population close to the actual joint population distributions. The authors compared the WGAN with the VAE and found that the VAE had more structural zeros than the WGAN in both low- and high-dimensional cases by 5% and 44%, respectively.

Another variation of the VAE, Conditional-VAE (CVAE), was employed by Aemmer and MacKenzie (2022) to extend prior work by generating synthetic populations with both individual and household

attributes. Another study attempted to minimize the structural zero problem but maximized the sampling zeros. The authors adopted two metrics to measure the performance of the models in handling sampling and structural zero problems: the feasibility and diversity of the synthetic population (Kim and Bansal 2023).

2.3 Diffusion Model

The diffusion model has received attention as a promising DGM since the release of the Stable Diffusion in 2022, which was developed through research on high-resolution image synthesis. The diffusion model involves two stages of processes: forward and reverse. In the forward process, data are transformed into noise, which gradually converts the complex probability distribution of the data into a simpler distribution that can be analyzed. In the reverse process, noise generates synthetic data and a deep-learning model trains the inverse transformation function to convert the simple probability distribution back into a complex one. As the inverse transformation function is used, the latent vector that extracts information maintains the same size as the dimensions of the data, enabling the storage of relatively more information in the data (Dong and Gao 2021). This enables the high-quality generation of unseen data that could not previously observed.

Diffusion models can be classified into three main types: DDPMs, core-based generative models (SGMs), and stochastic differential equations (Score SDEs). DDPMs use Markov chains to propagate noise and have variational lower-bound objective functions. These properties provide the advantage of combining the characteristics of VAE and MCMC for the efficient optimization of DDPMs. SGMs focus on data sampling using data density scores instead of the log-likelihood of the data distribution during the process of transforming probability distributions. This facilitates accurate predictions, even in regions with sparse data in the data space, while considering the size of the data distribution. Finally, Score-SDEs are an extension of SGMs that use a score function over time instead of computing stepwise scores, enabling an infinite extension of the noise stages and the generation of continuous explicit functions over time. This facilitates the simulation of both data-generation methods in the DDPMs and SGMs. In this study, we performed population synthesis by applying DDPMs, which is the most basic model among the diffusion models.

3 DIFFUSION MODEL-BASED POPULATION SYNTHESIS

The process of population synthesis using the diffusion model is as follows. First, one-dimensional sample data of an individual are preprocessed and transformed into a two-dimensional square matrix, which is required for DDPMs. The diffusion model is then trained using a preprocessed square matrix for the forward and reverse processes. Once the training is complete, the reverse process of the trained model is used to generate the desired number of samples from the noise. After postprocessing, the samples are transformed back into the synthetic population.

3.1 Data Preprocessing

The sample data of individuals, typically used for population synthesis, are one-dimensional vectors, unlike the two-dimensional or three-dimensional image data matrices. Therefore, to perform training using DDPM, the individual format sample data must be converted to the matrix, because the generation part of the diffusion model relies on an image-based U-Net (Dong and Gao 2021). There are two methods of converting the one-dimensional population data vector to an image data format. The first method uses Mel-frequency spectrogram or short-term Fourier transform for high dimensions (more than 40k dimension) or several repetitive signals data. The second method involves aligning individual sample data into a square matrix by adding padding. In this study, we aligned the sample data into a square matrix as the population data vector does not have high dimensions to draw a spectrogram and does not exhibit repetitive signals. The process of data preprocessing is illustrated in Figure 1. To transform each instance of the real population into a vector, the calculation of the square matrix size s is necessary to accommodate all values of the f

features. $s \times s$ sized square matrix of is filled all feature values of a instance in order, and the rest with 0. We used zero padding to convert the sample data vector into a square matrix M_n $(n = 1, \dots, N)$ because it

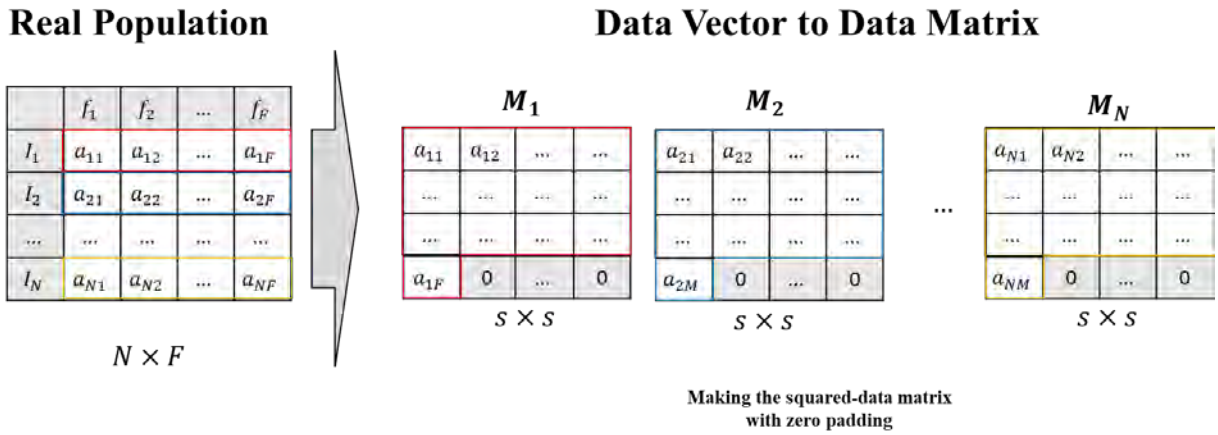

Figure 1: Illustration of real population data preprocessing.

allows the values in certain parts of M_n to be maintained at zero, which can reduce unnecessary computations during the training process.

3.2 Diffusion Model Training

Denoising Diffusion Probabilistic Model (DDPM) uses the propagation of noise through a Markov chain as a method of training, which not only shows a similar effect to models that use VAE multiple times but also has the advantage of being relatively robust in generating results with no loss of information due to the latent vector during the training process, unlike VAE (Yang et al. 2022). Despite the advantage of VAE-based population synthesis models in discovering sampling zero data well, they cannot consistently discover sampling zero due to the inconsistency of VAE. To address this issue, this paper proposes a population synthesis model based on DDPM. To explain the population synthesis model of DDPM, the process can be divided into forward process which involves from the initial data state $[M_n]_0$ to gaussian noise $[M_n]_T$ after T steps and reverse process. DDPM processes are as the following Figure 2.

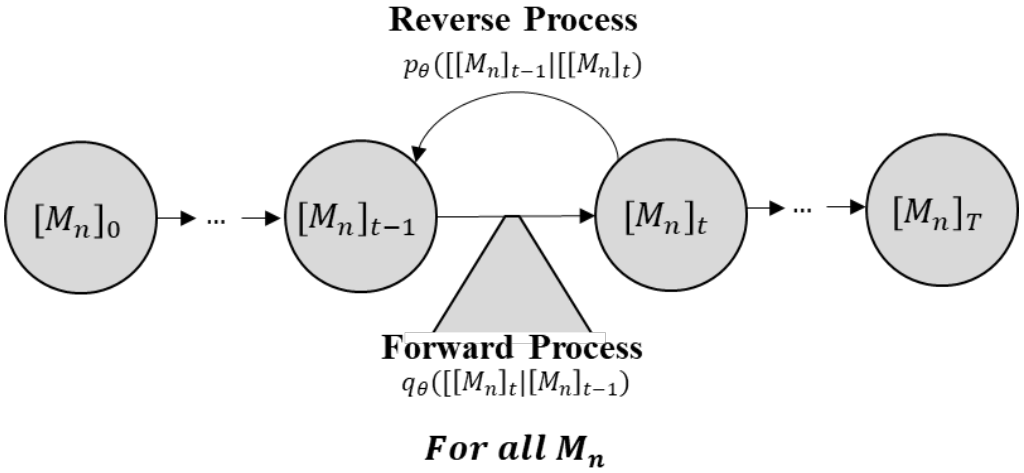

Figure 2: Forward process and reverse process of DDPM using sample data.

In the forward process, the conditional distribution $q([M_n]_t|[M_n]_{t-1})$ is defined and used as the process of transforming Matrix-formed n^{th} individual data at initial state ($[M_n]_0$) into Matrix-formed n^{th} noise at T^{th} state ($[M_n]_T$). The conditional distribution q is as follows.

$$q([M_n]_t|[M_n]_{t-1}) = Normal([M_n]_t; \sqrt{1-\beta_t}[M_n]_{t-1}, \beta_t \mathbf{I})$$

β_t is a hyperparameter that determines how much the state changes from the t^{th} state. β_1 is a value that starts to cause noise in ($[M_n]_0$, so it starts small, and as the t value increases, larger values are used to increase the possibility of gaussian noise. Using the defined conditional distribution q, the forward process that progresses T steps can be expressed as follows.

$$\prod_{t=1}^{T} q([M_n]_t|[M_n]_{t-1}) q([M_n]_0) = q([M_n]_1, \dots, [M_n]_T|[M_n]_0)$$

$$q([M_n]_t|[M_n]_0) = \mathcal{N}([M_n]_t; \sqrt{\overline{\alpha_t}}[M_n]_0, \alpha_t(\mathbf{1} - \overline{\alpha_t})\mathbf{I})$$

$$[M_n]_t = \sqrt{\overline{\alpha_t}}[M_n]_0 + \sqrt{\mathbf{1} - \overline{\alpha_t}}\epsilon$$

where $\alpha_t = 1 - \beta_t \; and \; \overline{\alpha_t} = \prod_{s=1}^{t} \alpha_s$.

In the reverse process, the conditional distribution $p_\theta([M_n]_{t-1}|[M_n]_t)$ that will be discovered through the deep learning model training process is defined through the trainable model θ.

$$p_\theta([M_n]_{t-1}|[M_n]_t) = \mathcal{N}\left([M_n]_{t-1}; \mu_\theta([M_n]_t, t)), \Sigma_\theta([M_n]_t, t))\right)$$

where $p([M_n]_T) = Normal([M_n]_T; 0, \mathbf{I})$.

Therefore, DDPM is trained through the following variational lower bound objective functions in the forward and reverse processes.

$$L = L_T + L_{1:t} + L_0$$

where $L_T = D_{KL}(q([M_n]_T|[M_n]_0||p_\theta([M_n]_T), L_0 = -\,logp_\theta([M_n]_0|[M_n]_1))$, and
$L_{1:t} = \sum_{t=1}^{T-1} D_{KL}\left(q([M_n]_{t-1}|[M_n]_t, [M_n]_0||p_\theta\left([M_n]_{t-1}|[M_n]_t\right)\right)$ which are the loss functions of each step from 0 to T.

3.3 Post-processing using Census Data

The data generated by the reverse process of DDPM include sampling zeros as well as structural zeros. Structural zeros refer to infeasible sample data where values appear in combinations of features that cannot exist in real world but cannot be classified as an outlier, such as the data wherein children under 10 years of age have children or where the number of family members is negative. A structural zero arises when the probability of generating a sample by the generative model becomes greater than zero and the DDPM itself does not have the means to discard it (Yang et al. 2022). Therefore, this study proposes a rule-based postprocessing method that utilizes constraints generated from the census (or statistical data) of real data to remove structural zeros from the generated results.

The features that primarily cause structural zeros are numerical values recorded as continuous or discrete. This is because it is impossible to use the upper and lower bounds of the numerical values in the DDPM. Information on the upper and lower bounds of the numerical values can be easily found in the

census data. Therefore, bins for numerical features that collect only nonstructural zero data were created from the census data. The bins for m^{th} numerical feature ($Bin(f_m)$) are expressed as follows.

$$Bin(f_m) = \{(l_{f_{m,k}}, u_{f_{m,k}}) | \forall k, l_{f_{m,k}} = \frac{\max(f_m) - \min(f_m)}{K} * (k-1), u_{f_{m,k}} = \frac{\max(f_m) - \min(f_m)}{K} * k\}$$

where k is the hyperparameter for the number of bin components, and $\max(f_m)$ and $\min(f_m)$ are obtained from the census data rather than the sample data.

The algorithm of the DDPM-based population synthesis model that performs post processing from the data pre-processing process to the bins generated by the census data is as follows.

Algorithm 1 DDPM-based Synthetic Population Method

Input :
Real Population P_R
Output :
Synthetic Population P_S
Option :
Total number of synthetic population SP
Matrix form data of n^{th} instance M_n
Total Number of Features in Real Data F

1. # Data preprocessing
2. **For** i **in range**(len(P_R)):
3. s = 0
4. While ($s^2 < F$):
5. s = s + 1
6. $I_n = I_n + zeros(s^2 - F)$
7. $M_n = I_n.reshape(s, s)$
8. **End For**
9. # DDPM Model training
10. **For** M_n **in range**(len(I_n)):
11. $[M_n]_0 \sim q([M_{n \in N}]_0)$
12. $t \sim Uniform(\{1, \dots, T\})$
13. $\epsilon \sim \mathcal{N}(0, I)$
14. Optimize on Loss $\nabla_{\theta} ||\epsilon - \epsilon_{\theta}(\sqrt{\overline{\alpha_t}}[M_n]_0 + \sqrt{1 - \overline{\alpha_t}}\epsilon, t)||^2$
15. **End For**
16. # Sample Generation
17. $[M_n]_T \sim \mathcal{N}(0, I)$
18. **For** n **in range**(SP):
19. **For** t **in** [T,…,1]:
20. $z \sim \mathcal{N}(0, I)$
21. Generate $[M_n]_{t-1} = \frac{1}{\sqrt{\alpha_t}}([M_n]_{t-1} - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha_t}}} z_{\theta}([M_n]_t, t) + \sigma_t z$
22. **End For**
23. # Post Processing
23. **If** $[M_n]_0$ **not in** $Bin(f_m)$ for all m
24. **GOTO LINE 19**
25. $O_n = [M_n]_0$
26. **End For**

4 EXPERIMENTS AND PERFORMANCE EVALUATION

In this paper, we conducted experiments by increasing the number of samples in the training data to demonstrate the superiority of the Diffusion model-based population synthesis method. The code used in this paper is available at https://github.com/Ninekrad/PopulationSynthesis. Due to the higher training cost during the sampling process in the Diffusion model compared to other models, we did not perform optimal hyperparameter tuning as mentioned by Cao et al. (2022). Instead, we used the default hyperparameters provided in the Diffusion model example from Keras (https://keras.io/examples/generative/ddpm/) for training. To increase the probability of transforming data into simple gaussian noise, the β_t parameter, which adds noise to the data during the Forward Process, was gradually increased from 1e-4 to 0.02. The total step of the process was set to 1000. Among the hyperparameters required for deep learning training, the epoch was set to 300, the learning rate to 0.003, and the optimization algorithm for the loss function was set to AdamW. Diffusion model training was performed on Google Colab, using the A100 GPU.

4.1 Data Description

In this study, experiments were conducted based on the 2% microdata sample (A-type data consisting of a 2% detailed sample of the total population of Korea) from the Korea Statistics (KOSTAT) Department. The 2% microdata sample contained data on 927,843 individuals and comprised the features listed in Table 1.

Table 1: Summary of population features of the 2% microdata samples.

#	Name	Type	Number of values	Description
1	PlCode	Categorical	17	Code for Province
2	PsCode	Categorical	Varies according to PlCode.	Code for city in Province
3	Sex	Categorical	2	(1) Male; (2) Female
4	Age	Numerical (integer)	86	(0-84) Individual's age; (85) Age 85+
5	P1	Categorical	3	Type of commuting: (0) does not commute; (1) commute within the residential area; (2) commute outside the residential area
6	P2	Categorical	18	Commuting place: (0) same as the residential area; the rest of the values are the same as PlCode.
7-16	T1-T10	Indicator	2 for each	Type of transportation for commuting: subway, car, and shuttle bus
17	TH	Numerical (continuous)	-	Commuting time (hours)
18	TM	Numerical (continuous)	-	Commuting time (minutes)

4.2 Experimental Results

To compare the performance of the population synthesis model using DDPM with other representative population synthesis methods, such as MCMC and VAE. MCMC is a representative generative method

used in large-scale population synthesis models (Farooq et al. 2013; Casati et al. 2015; Gong et al. 2021), and VAE is the first deep learning model-based population generation method (Borysov et al. 2019).

Therefore, they serve as suitable benchmarks to demonstrate the superiority of our proposed model. To assess the similarity between the synthesized populations and the actual populations, we compared them using the standardized root mean square error (SRMSE) measure. Additionally, we measured the shortest distance between the synthesized populations and the actual populations to demonstrate the feasibility of the synthesized results. SRMSE is evaluated the difference between the predicted and actual values based on the joint probability distribution handled in each study (Borysov et al. 2019; Kim et al. 2022; Kim and Bansal 2023). A lower SRMSE indicates a higher level of similarity.

$$RMSE(\pi,\ \hat{\pi}) = \frac{RMSE(\pi, \hat{\pi})}{\bar{\pi}} = \frac{\sqrt{\sum_{(k,k`)}\left(\pi_{(k,k`)} - \hat{\pi}_{(k,k`)}\right)^2 / N_b}}{\sum_{(k,k`)} \pi_{(k,k`)} / N_b}$$

where π and $\hat{\pi}$ are the categorical distributions of real population and the synthesized data respectively. N_b is the total number of category combinations, which is calculated The number of cases for all combinations of different features k, k.

The experimental results are summarized in Table 2. As shown in Table 2, the distribution of the data generated by our proposed model is closer to the distribution of the real population sample data than the data generated by the MCMC. In the experimental results of each model, the best results were indicated in bold font depending on the number of samples. This demonstrates that our proposed model performs equally well in population synthesis as it does as an image generation model. Additionally, in numerous cases, the average marginal RMSE and bivariate RMSE of our proposed model were comparable to or superior to the marginal RMSE and bivariate RMSE of VAE. Average and standard deviation between the nearest real population instance and the synthetic population were evaluated slightly higher than MCMC for both our proposed model and the VAE model, because DGM models generate the case of sampling zero. This indicates that although the diversity of the distribution of data generated by our proposed model is relatively lower than VAE, our proposed model is relatively feasible because it can generate real population-like results.

Table 2: Evaluation results of synthesized data with models and the number of samples.

Model		**# Samples**	**Marg. SRMSE**	**Bivar. SRMSE**	μ_{R-S}	σ_{R-S}
Traditional Model	**MCMC**	10000	7.536	16.524	**0.1215**	**0.1057**
		100000	7.594	16.538	**0.1286**	0.1083
		500000	7.683	17.253	**0.1247**	**0.1073**
		1000000	7.666	17.685	**0.1263**	0.1075
		Average	7.620	17.000	**0.1253**	**0.1072**
Deep Learning-Based Model	**VAE**	10000	**2.066**	**4.553**	0.2256	0.1258
		100000	2.213	4.952	0.2172	0.1158
		500000	2.992	5.912	0.2022	0.1178
		1000000	2.252	5.121	0.2219	0.1485
		Average	2.381	5.135	0.2167	0.1270
	Our Approach	10000	2.249	4.826	0.1816	0.1077
		100000	**1.904**	**4.235**	0.1770	**0.1051**
		500000	**2.214**	**4.752**	0.2215	0.1261
		1000000	**2.154**	**4.532**	0.1810	**0.0918**
		Average	**2.130**	**4.586**	0.1903	0.1076

The superior DDPM was also evaluated for the presence of sampling zero in the generated data through postprocessing. In this experiment, only age was used as a constraint, and the ratio of synthetic population data generated by the DDPM to the actual population sample data was compared and can be visualized in Figure 3. Under the category of age (upper left of Figure 3), it was confirmed that data were generated even from the data distribution that was not used for training (right panel). Although the SRMSE with unused population sample data was not significantly measured, considering that generating data was impossible with methods other than the existing DGM method, the results were considered sufficiently significant.

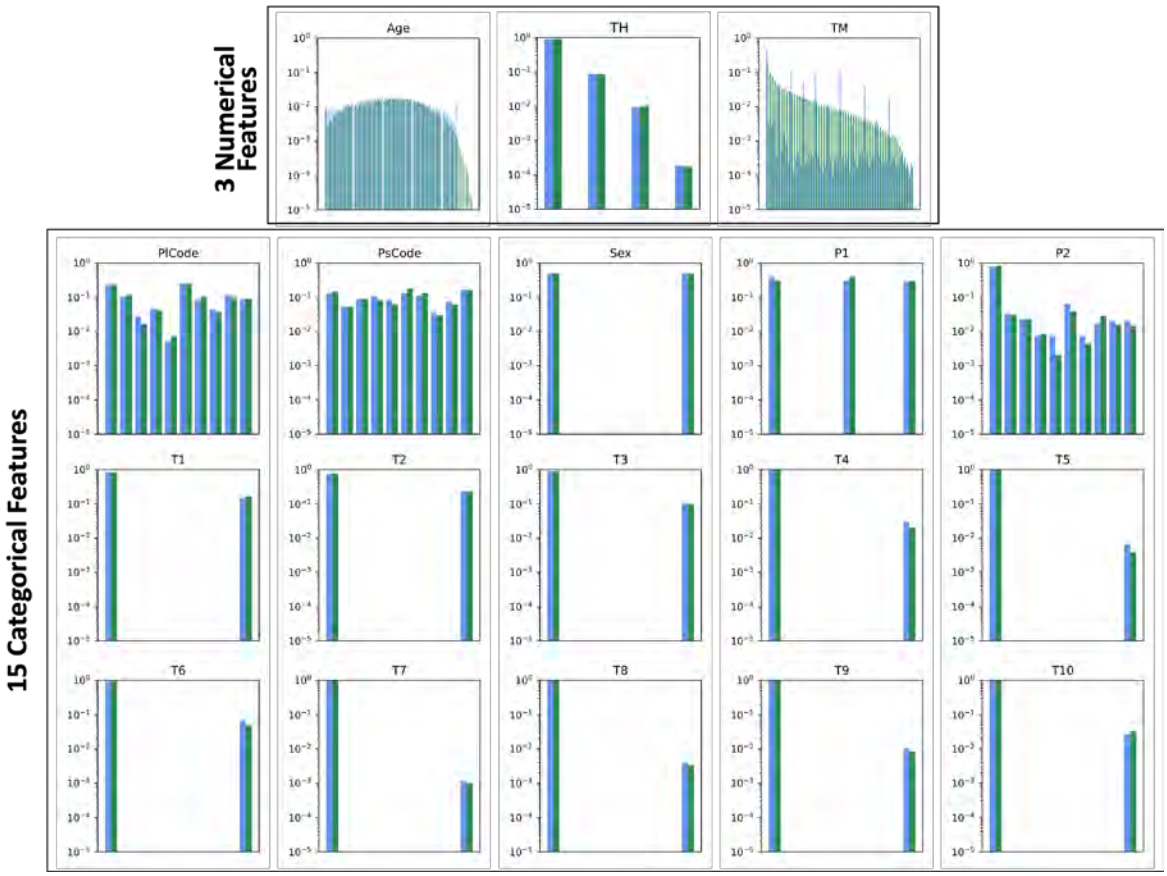

Figure 3: Comparison of the ratio between Real data (blue) and Synthesized data using Diffusion Model (green).

5 CONCLUSION AND FUTURE WORKS

In this study, we implemented the diffusion model, known as the most advanced deep learning method in the field of imaging, for population synthesis. We confirmed that by utilizing the excellent unobserved data generation capability of the diffusion model, we can partially address the sampling zero problem, and that the exceptional feature generation capability of deep learning can also address the scalability issue. Furthermore, this study demonstrated that deep learning models that perform well in specific fields can provide meaningful results in other generative tasks. Although, this study has some limitations. First, to solve the sampling zero problem, bins were used to eliminate structural zeros; however, a large amount of prior information related to the population is required to create the bins. In this study, only significant information related to age was obtained from the census, which resulted in a limited number of possible bins. Second, only individual-level population synthesis was attempted; household-level population synthesis was not performed. Using household features with various values in the feature engineering

process for deep generative models increases the computational complexity, inhibiting the accuracy of the learning process of the model. Therefore, future research will focus on developing data preprocessing techniques that use household features to create a complete population synthesis model. Additionally, because the performance of the DDPM is comparable to that of the VAE, we plan to train a population synthesis model using score-SGD, which can be used irrespective of the feature type, to develop a generalized population synthesis model.

REFERENCES

Aziz, H. A., B. H. Park, A. Morton, R. N. Stewart, M. Hilliard, and M. Maness. 2018. "A High Resolution Agent-Based Model to Support Walk-Bicycle Infrastructure Investment Decisions: A Case Study with New York City". *Transportation Research Part C: Emerging Technologies* 86:280-299.

Bonabeau, E. 2002. "Agent-Based Modeling: Methods and Techniques for Simulating Human Systems". *Proceedings of the National Academy of Sciences* 99 (suppl_3):7280-7287.

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.

Choupani, A.-A., and A. R. Mamdoohi. 2016. "Population Synthesis Using Iterative Proportional Fitting (Ipf): A Review and Future Research". *Transportation Research Procedia* 17:223-233.

Dong, Y., and C. Gao. 2021. "Elbd: Efficient Score Algorithm for Feature Selection on Latent Variables of Vae". *arXiv e-prints*:arXiv: 2111.08493.

Farooq, B., M. Bierlaire, R. Hurtubia, and G. Flötteröd. 2013. "Simulation Based Population Synthesis". *Transportation Research Part B: Methodological* 58:243-263.

Fournier, N., E. Christofa, A. P. Akkinepally, and C. L. Azevedo. 2021. "Integrated Population Synthesis and Workplace Assignment Using an Efficient Optimization-Based Person-Household Matching Method". *Transportation* 48:1061-1087.

Gong, S., I. Saadi, J. Teller, and M. Cools. 2021. "Validation of MCMC-Based Travel Simulation Framework Using Mobile Phone Data". *Frontiers in Future Transportation* 2:660929.

Gulshan, V., L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros, R. Kim, R. Raman, P. C. Nelson, J. L. Mega, and D. R. Webster. 2016. "Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs". *JAMA* 316 (22):2402-2410.

Hammond, R. A. 2015. "Considerations and Best Practices in Agent-Based Modeling to Inform Policy". In *Assessing the Use of Agent-Based Models for Tobacco Regulation*. National Academies Press (US).

Ho, J., A. Jain, and P. Abbeel. 2020. "Denoising Diffusion Probabilistic Models". *Advances in Neural Information Processing Systems* 33:6840-6851.

Jung, H. J., G. S. Jung, Y. Kim, N. T. Khan, Y. H. Kim, Y. B. Kim, and J. S. Park. 2017. "Development and Appplication of Agent-Bsed Disease Spread Simulation Model: The Case of Suwon, Korea". In *Proceedings of the 2017 Winter Simulation Conference*, edited by Victor W.K. Chan, Andrea D'Ambrogio, Gregory Zacharewicz, Navonil Mustafee, Gabriel Wainer, and Ernest H. Page, 2810-2820. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Kim, E.-J., and P. Bansal. 2023. "A Deep Generative Model for Feasible and Diverse Population Synthesis". *Transportation Research Part C: Emerging Technologies* 148:104053.

Kim, E.-J., D.-K. Kim, and K. Sohn. 2022. "Imputing Qualitative Attributes for Trip Chains Extracted from Smart Card Data Using a Conditional Generative Adversarial Network". *Transportation Research Part C: Emerging Technologies* 137:103616.

Liu, J., X. Ma, Y. Zhu, J. Li, Z. He, and S. Ye. 2021. "Generating and Visualizing Spatially Disaggregated Synthetic Population Using a Web-Based Geospatial Service". *Sustainability* 13 (3):1587.

Macal, C. M., and M. J. North. 2009. "Agent-Based Modeling and Simulation". In *Proceedings of the 2009 Winter Simulation Conference*, edited by Ann Dunkin, and Ricki G. Ingalls, 86–98. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Saadi, I., A. Mustafa, J. Teller, and M. Cools. 2018. "Investigating the Impact of River Floods on Travel Demand Based on an Agent-Based Modeling Approach: The Case of Liège, Belgium". *Transport Policy* 67:102-110.

Sun, L., and A. Erath. 2015. "A Bayesian Network Approach for Population Synthesis". *Transportation Research Part C: Emerging Technologies* 61:49-62.

Sun, L., A. Erath, and M. Cai. 2018. "A Hierarchical Mixture Modeling Framework for Population Synthesis". *Transportation Research Part B: Methodological* 114:199-212.

Tröndle, T., and R. Choudhary. 2017. "Occupancy Based Thermal Energy Modelling in the Urban Residential Sector". *WIT Transactions on Ecology and the Environment* 224:31-44.

Vidyattama, Y., R. Miranti, J. McNamara, R. Tanton, and A. Harding. 2013. "The Challenges of Combining Two Databases in Small-Area Estimation: An Example Using Spatial Microsimulation of Child Poverty". *Environment and Planning A* 45 (2):344-361.

Wurzer, G., K. Kowarik, and H. Reschreiter. 2015. *Agent-Based Modeling and Simulation in Archaeology*: Springer

Yang, L., Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, Y. Shao, W. Zhang, B. Cui, and M.-H. Yang. 2022. "Diffusion Models: A Comprehensive Survey of Methods and Applications". *arXiv preprint arXiv:2209.00796*.

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

Ye, P., X. Hu, Y. Yuan, and F. Y. Wang. 2017. "Population Synthesis Based on Joint Distribution Inference without Disaggregate Samples". *Journal of Artificial Societies and Social Simulation* 20 (4).

AUTHOR BIOGRAPHIES

JAEWOONG KANG is a Ph.D. student in Department of Industrial Engineering, Sungkyunkwan University. His research interests are deep learning, machine learning in healthcare and popluation synthesis. His email address is kjw1727@skku.edu.

YOUNG KIM is a Ph.D. student in Department of Industrial Engineering, Sungkyunkwan University. He is currently interested in simulation-based modeling, forecasting method, statistical analysis and population synthesis. His email address is lmjlguard@gmail.com.

MUHAMMAD MU'AZ IMRAN is a Ph.D. student in the Faculty of Integrated Technologies and the Department of Industrial Engineering at Sungkyunkwan University. He completed his undergraduate in Systems Engineering with a Manufacturing major at the Faculty of Integrated Technologies, Universiti Brunei Darussalam. His research interests are big data analysis, metal additive manufacturing, machine learning, and population synthesis. His email address is muazimran27@gmail.com.

GI-SUN JUNG is a Postdoctoral researcher in Dept. of Industrial Engineering, Sungkyunkwan University. He received a Ph.D. in Industrial Engineering from Sungkyunkwan University and BS in System Management Engineering from Sungkyunkwan University. He is interested in demand forecasting, modeling and simulation methodology, data analytics based on stochastic process and machine learning. His email address gsjung09@naver.com.

YUN BAE KIM is a Professor with the Department of Industrial Engineering, Sungkyunkwan University. He received the MS degree from the University of Florida, and the Ph.D. degree from Rensselaer Polytechnic Institute. His current research interests are demand forecasting, simulation methodology, high tech market analysis and scheduling. His email address is kimyb@skku.edu.