Generative Population Synthesis for Joint Household and Individual Characteristics

Zack Aemmer

A thesis

submitted in partial fulfillment of the

requirements for the degree of

Master of Science in Civil Engineering

University of Washington

2021

Committee:

Don MacKenzie

Cynthia Chen

Program Authorized to Offer Degree:

Civil and Environmental Engineering

University of Washington

Abstract

Generative Population Synthesis for Joint Household and Individual Characteristics

Zack Aemmer

Chair of the Supervisory Committee:

Don MacKenzie

Civil and Environmental Engineering

Household survey collection efforts provide immense value in the fields of transportation and urban planning, where public agencies, private companies, and researchers alike use the data to draw conclusions on populations. However, even the most well-funded surveying agencies rely on sampling methods to estimate the nature of the true population, and microdata from public censuses is frequently aggregated, or limited in volume and detail to protect the privacy of respondents. With growing emphasis on microsimulation to predict population behavior in response to emerging transportation technologies such as electric/autonomous vehicles, or new micromobility and ridesharing services, population synthesis provides a means to scale this socioeconomic microdata into synthetic populations representing much larger areas. Despite their accuracy and widespread adoption, traditional synthesis algorithms for reweighting microdata samples scale poorly with the number of variables and geographic regions being

modeled, and can suffer from non-convergence when smaller sample sizes are used. Several generative models have been proposed to address these shortcomings, but lack key features such as sub-region modeling, and the ability to simultaneously generate both individuals and households. This work proposes a new approach to generating synthetic populations consisting of both individual and household-level variables, that uses a Conditional Variational Autoencoder (CVAE) to learn a distribution of latent variables in the general population, and use them to generate new samples. The accuracy and computational efficiency of this approach are benchmarked against a state of the art open source population synthesizer. In addition, the CVAE model is tested under increasingly minimal training data to determine its ability to generate realistic populations from smaller surveys. Findings indicate that the CVAE model creates more accurate populations, using less time than the traditional synthesizer under small to medium dimensional datasets (4-16 variables). The CVAE also performs well with relatively small (n=100) training data samples, but tends to overfit at lower sample sizes, despite adding additional regularization to the model.

Table of Contents

1. Introduction

1.1 Motivation

The interconnected and global scale of our transportation networks creates immense multipliers for inefficiencies, and seemingly small decisions or influences can create disproportionate impacts on the environment and the welfare of the population. As new and at times unprecedented technologies (e.g. electric and autonomous vehicles, micromobility and ridesharing services) are introduced in various transportation networks, it can be difficult to anticipate their impacts at the system-wide level, without first understanding how they are perceived and valued by individuals and households on a smaller scale. This often necessitates detailed surveys and observations of user behavior at the individual level, followed by microsimulation models to estimate how a particular service or parameter might alter the transportation equilibrium in a given location. Modern advancements in computational power have provided the ability to run these models at both widespread and unprecedentedly disaggregate levels. However, they require highly dimensional input, thus creating an increased demand for depth and quantity of individual and household-level socioeconomic survey data. Tasked with overseeing the development and maintenance of transportation networks, municipal planning organizations (MPOs), departments of transportation (DOTs), academic researchers, and other public agencies must constantly work with limited resources to acquire detailed information about the population of the modeling region in a cost effective way. One method of accomplishing this is through representative sampling and reweighting of household surveys to create datasets that are representative of the true population. Current methods for this population synthesis process are limited in that they require increasingly large sample sizes as the number of

modeled variables increases, and scale poorly with the number of variables and regions being modeled.

1.2 Outline of Research

This work begins with a discussion of the origins and significant developments of algorithms used for population synthesis. This includes an examination of the Iterative Proportional Fitting (IPF) algorithm, and its original application to population synthesis. Shortcomings and improvements to the IPF algorithm are summarized; most importantly the introduction of the Iterative Proportional Updating (IPU) algorithm, which allows for the joint fitting of household and individual characteristics in population synthesis. Then, some of the still-existing problems with IPF/IPU based population synthesis algorithms are discussed, along with the area of open research currently addressing them. Last, the review addresses emerging research in the relatively new field of generative population synthesis, in which predictive techniques are used to generate new samples outside of the original dataset. These methods are capable of creating a truly synthetic population, in which the original microdata samples are not simply reweighted. The general goal of these methods is to seek a new method of population synthesis that is computationally efficient for highly dimensional populations, without sacrificing the accuracy of traditional methods. However, they are still relatively novel, and thus have some shortcomings relative to traditional synthesis, which are also discussed.

In addition to a review of traditional and contemporary population synthesis methods, the fundamental contribution of this work is the development and benchmarking of a new generative population synthesis model, that builds off of previous work using a Variational Autoencoder (VAE) to learn, then sample from, latent variables in a set of socioeconomic microdata. The

model proposed and tested here utilizes a hierarchy containing both a household-generating VAE, and an individual-generating Conditional-VAE (CVAE) to generate a complete population that is representative at both the individual and household level. In the context of traditional synthesizers, this is similar to the functionality provided by the IPU algorithm. Last, its ability to generate an accurate population under increasingly sparse training data is tested. This work concludes with a discussion of what were found to be the benefits and drawbacks of this new form of population synthesis, its most practical use-cases, and the most beneficial avenues for future research.

2. Literature Review

2.1 Traditional Population Synthesis

2.1.1 The Need for Population Synthesis

To predict trends in regional transportation, MPOs often utilize travel demand models that require geospatial socioeconomic data from household travel surveys such as the American Community Survey (ACS). Due to the financial and time infeasibility of collecting data from every individual member of a population, these agencies rely on carefully chosen sampling methods, which in practice allow representative estimates of statistical properties of the population to be ascertained. Thus, when collecting household survey data, important consideration must be given to developing a sampling frame, and corresponding sampling method that collect data which is representative of the population being studied. To these ends, many sampling approaches have been studied and evaluated in the context of transportation planning: the simplest and most common of which are simple random sampling and stratified random sampling (Richardson et al., n.d.). Regardless of the sampling method and frames

chosen, the result of this process is a set of microdata containing socioeconomic and demographic characteristics of the sampled population.

To use the data collected from these household surveys in microsimulation models, the disaggregate microdata must be scaled to represent the complete population, the process of which is called population synthesis. This is accomplished using estimated marginal count data for key variables (e.g. income brackets, mode splits) in the population to generate a complete synthetic population for the base year in which the data was collected, and subsequently expand that population according to forecasted marginal counts of the same key variables in future years (Ortúzar & Willumsen, 2011). Once expanded, the representative set of microdata for current and future years can be used in activity-based forecasting, or other microsimulation models to determine how travel demand in different regions, and among specific populations, might change over time or in response to new modes. This process is phrased as "...generating an artificial population by expanding the disaggregate sample data to mirror known aggregate distributions of household and person variables of interest" (Ortúzar & Willumsen, 2011). In short, population synthesis is a method by which sampled survey data can be expanded to fully represent the larger population from which it was drawn.

2.1.2 Iterative Proportional Fitting Algorithm

The term "Population Synthesis" was first coined by (Beckman et al., 1996) to describe their method for expanding the public-use microdata often published as the result of household surveys to represent an entire population. In it, they utilize the Iterative Proportional Fitting (IPF) algorithm which was first proposed as a generalizable means to weight a contingency table of observations to achieve specific marginal counts (Deming & Stephan, 1940). In the algorithm,

least squares is used to calculate the difference between the marginal counts of variables in the contingency table, and a set of known true values. Then, an iterative procedure is used to weight and reweight individual cells such that this sum of least squares across the marginal counts of each variable are minimized. Beckman et al. (1996) proposed using this algorithm to weight households from the ACS such that the distribution of key "control variables" in the sampled data are matched to the overall marginal distributions provided by an original survey. For example, a survey might collect binned age and income demographics from a population. If only one of these is set as a control variable, its marginal counts will be used to determine weights for both age and income in the synthetic population. There is no limit to the number of control variables which may be used (however this may become computationally intractable), but there must be at least one control variable to use IPF. In this way, control variables will be near-perfectly matched so long as the IPF algorithm converges. Then, households are sampled according to the final weights such that the synthetic population is representative of the true population as described by the marginal count data. Many other variables (which are not included as control variables) may be included in the synthetic population by weighting them according to the weights calculated on the control variables. This procedure was tested by comparing marginal and joint distributions of control and non-control variables in an IPF-generated population, to the naive approach of simple random sampling of individuals from the survey microdata. They observed that the marginal distributions of control variables were met exactly, while other variables and the joint distributions of variables within the synthetic population were favorable. Specifically, they validated the approach on non-control variables by comparing the count of households generated belonging to each household size with those in the true population. Their results are reported in a figure, but the count differs by an average of 7.8

households across each class. This particular synthesis was performed in a single region (Tarrant County, Texas). This algorithm formed the baseline method on which incremental improvements in population synthesis have been made, and remains a core component of most synthesizers.

The adaptation of the IPF algorithm which Beckman et al. (1996) used for their population synthesis relies primarily on an m-way contingency table, in which (m) is the number of classes of all demographic control variables chosen to represent the population. Each cell in the table is calculated as the count of individuals in the study sample corresponding to a single set of joint characteristics of each variable, divided by the total number of observations in the table. Each value in the table thus represents the proportion of the full population which contains a specific combination of characteristics. The second component of the IPF algorithm is a set of marginal counts (T), which describe the total number of individuals in the true population belonging to each class (k) of a given control variable (j). At each iteration of the IPF process, the proportion in each of the cells corresponding to a given marginal is recalculated according to Equation 1. Equation 1 is repeated for each dimension in the m-way contingency table (i.e. for three variables, each a control variable with desired marginal counts, each cell will receive three updates per iteration of the IPF algorithm).

$$P_{new\ j,k} = P_{old\ j,k} * (T_{true\ j,k}/T_{old\ j,k}) \quad (1)$$

This step is then repeated until the cumulative difference between P-old and P-new across the full table is small, indicating convergence. One consideration of this method is that all demographic input variables must be in categorical form, whether by their nature or through binning in the case of continuous variables. In practice, care must be taken to ensure that bin ranges provided in the marginal counts are precisely the same as those in the microdata. Once the algorithm has converged, the final proportion provided in each cell indicates a weight by which

each set of microdata samples corresponding to that cell should be scaled when generating the synthetic population.

Overall, the IPF algorithm provides an extremely accurate means to determine the optimal weights for each input sample, such that every marginal count is met, and the proportions for each cell in the contingency table for the final population are as close as possible to the original sample (IRELAND & KULLBACK, 1968). To accommodate multiple population regions, an additional, identical fitting step can be added to the IPF procedure, which re-balances the weights in each cell according to marginal counts for each variable at each regional level (Beckman et al., 1996). Due to its reliance on reweighting the table of demographics, the complexity of the IPF algorithm scales multiplicatively with the total number of cells in the contingency table, as well as with the number of variables being modeled.

2.1.3 Improvements to the IPF Algorithm for Population Synthesis

There are many population synthesizers which have built off the work of Beckman et al. (1996) through incremental improvements to the synthesis procedure; by adding and replacing steps to either improve the accuracy of the synthetic population, or improving practical functionality of the synthesizer. For example, one paper proposed a two-stage IPF procedure which incorporates additional housing data in a second IPF process, and thus improved its accuracy in recreating the true population (Zhu & Ferreira, 2014). Another incorporated an optimization approach to fitting household weights, which uses multiple metrics to ensure that the synthetic population correctly finds the set of weights corresponding closest to the true population, rather than a set of weights satisfying a local minimum (Zhuge et al., 2017). Other work modified the procedure to allow for marginal counts that are specified at multiple

geographic levels, providing a valuable resource for cases where marginal statistics provided by household surveys are provided at different levels of aggregation (Auld et al., 2009).

Possibly the most noteworthy improvement to the basic IPF algorithm was the development of the Iterative Proportional Updating (IPU) procedure which outlined additional steps to ensure that marginal distributions for both individual and household-level variables were met in the synthetic population (Ye et al., 2009). This solved one limitation of the IPF algorithm, which was previously limited to either generating weights for individuals or households, but not both simultaneously. For some household surveys, individuals are provided with household identifiers allowing them to be grouped, which then allows them to be weighted according to a shared household weight, but does not guarantee that the overall demographics of individual characteristics of the population will be accurate. The IPU algorithm improved the accuracy of the model while removing the need for heuristic methods of joining individuals to households. Later, other work also identified the issue of joint household and individual synthesis, and lamented the additional computational complexity of the IPU algorithm (Pritchard & Miller, 2012). They proposed a synthesizer that uses a sparse list to store the contingency table in a compressed manner, along with a novel technique to fit this table on the marginal count data, immensely improving its computational efficiency. Last, many researchers including Ye et al. (2009) have identified the value in providing a generalizable version of their synthesizer as open source software (PopGen, n.d.). As synthesizers have improved, there have been numerous efforts at creating a fully generalizable software package capable of population synthesis for any region and population. A thorough review of the strengths and weaknesses of various open-source population synthesizers has been documented as well (Müller & Axhausen, 2010).

One notable branch of synthesizers have replaced the IPF/IPU algorithm for finding household weights with a combinatorial optimization approach (Abraham et al., 2012; Bar-Gera et al., n.d.; Lee & Fu, 2011). This method formulates the reweighting process as an optimization problem which can be solved with gradient descent methods. This new approach was demonstrated to be fast and accurate when generating a synthetic population of 33.9 million individuals (Abraham et al., 2012). Additionally, one work tested combinatorial optimization methods against IPF directly, and concluded that while both methods produced statistically accurate representations of the true population, the optimization methods were overall more accurate given similar input data (Ryan et al., 2009). These tests were performed in the relatively low dimensional case (3 variables), and the authors recommend additional testing to verify whether their findings hold for more realistically detailed synthetic populations (8-10 variables). Because optimization based models provide a more generalizable means to meet marginal totals specified at multiple geographic regions, with joint individual/household demographics, without sacrificing accuracy, they have attracted more open source implementations capable of generating populations for any combination of variables and sub-regions. One modern implementation of this combinatorial optimization approach utilizes an entropy maximization objective function, and is programmed in the ActivtySim open source microsimulation framework (ActivitySim — ActivitySim 0.9.7 Documentation, n.d.; Paul et al., 2018). Their implementation of optimization-based population synthesis is perhaps the most robust open source population synthesizer available to date. It is capable of handling simultaneous household and individual balancing, multiple geographies, and performs reasonably well when generating large populations (~3 million individuals). It also has robust documentation and has been adapted

for professional use (Population Synthesizer Development – PopulationSim, n.d.). This model is used as the benchmark for the proposed generative methods in this work.

2.1.4 Issues with IPF/IPU Population Synthesis

There are several weaknesses of the unmodified IPF/IPU algorithm. The first is the issue of "sampling zeros" in which certain joint combinations of variables in the true population are not captured in the sampled data (Choupani & Mamdoohi, 2016; Mohri & Roark, 2005). Thus, these algorithms fail to converge when they are unable to weight sampled households in a way that meets the marginal control counts. Several approaches to addressing this issue using IPF have been identified by prior research (Guo & Bhat, 2007). First, the algorithm can simply be terminated after a specified number of iterations, which will produce a synthetic population which is as accurate as possible, but fails to meet the marginal control variable counts. Alternatively, the zero-cells can be given a small value such as 0.1, which will allow for convergence but introduce bias in the synthetic population. Last, the joint distribution of population variables can be specified at a more aggregate resolution, in the hopes that there will be a less sparse contingency table generated from the microdata, and all control variable counts will be able to be met. Although each of these methods allows for the algorithm to converge, they do not address the underlying problem; any marginal total that does not have supporting microdata will not be represented in the final synthetic population. As the number of modeled variables increases, or the quantity of microdata available decreases, this issue will become more prominent.

Another weakness of IPF/IPU based synthesis is its relatively slow computation time as the number of regions and dimensionality of the population increase (Borysov et al., 2019;

Moreno & Moeckel, 2018). Despite improvements in computational power, IPF/IPU and combinatorial optimization approaches scale poorly, and sacrifices must be made in the level of detail present in the synthetic population. In IPF synthesis, the poor scaling is due to the contingency table which must be fit to each sample and marginal total. This problem is multiplicative in IPU synthesis, wherein the algorithm requires multiple steps to fit both household and individual characteristics. Research on combinatorial optimization solutions is somewhat mixed, with some work showing that these approaches increase the overall computation time (Ryan et al., 2009), while others find them both faster and more efficient than IPF/IPU (Abraham et al., 2012; Lee & Fu, 2011). Ultimately, to create significant improvements on the computational efficiency of algorithms for population synthesis, new techniques must be developed which do not rely on reweighting individual samples.

2.2 Generative Population Synthesis

2.2.1 Prior Generative Models

"Generative" statistical models utilize all of the variables available in a set of data to model their joint probability distribution, and use it to make probabilistic predictions based on known information for a set of observations (Jebara, 2004). This is in contrast to discriminative models, which instead focus on learning conditional probabilities to make predictions based on predefined relationships between input variables, and the desired classification scheme (Jebara, 2004). These terms are somewhat loosely defined, and most models exist somewhere on a spectrum between the two. An example of a truly generative model would be a Bayesian network, in which the joint distribution of variables in a dataset are modeled as a directed graph structure, where information on one or more variables informs the statistical likelihood of others.

On the other hand, a Support Vector Machine would be discriminative, in the sense that it models a decision boundary based on input variables that allows it to directly classify new data points. The subtle difference between the two is that one learns the complete variable-space and makes the most likely prediction based on known inputs, while the other learns a heuristic method for making predictions based on specific combinations of inputs. In population synthesis, reweighting models (IPF/IPU/Combinatorial Optimization) as discussed previously would be closer to discriminative in that they do not directly learn the joint distribution of input variables to generate a new population. Rather, conditional on the set of input microdata and marginal counts, they use either the IPF algorithm or a gradient descent method to find a set of weights for the existing samples that produces the marginal counts desired. A generative population synthesizer, on the other hand, might examine a set of individual microdata and learn to model the joint distribution of socioeconomic variables in them, then draw new individuals probabilistically from that distribution.

Several generative model structures have already been tested to develop synthetic populations. These approaches can accomplish similar results to reweighting methods without using marginal counts, but they are less developed, and in most cases missing key features of population synthesizers such as modeling multiple regions or performing joint individual/household synthesis. Furthermore, they can introduce the issue of “structural zeros" in which new individuals are generated who have joint combinations of characteristics that are not present in the true population. One generative synthesis approach is the aforementioned Bayesian network, which models the joint distribution of socioeconomic variables as a probabilistic graph network (Joubert & de Waal, 2020; Sun & Erath, 2015). This model frames population synthesis as an effort to maintain the joint distribution of socioeconomic variables, rather than the marginal

totals, in practice generating a population with the relationships between variables in the data based on their probabilistic combinations. Others have taken advantage of the ability of generative models to create new individuals with combinations of socioeconomic variables that are rare in the sampled population; directly addressing the problem of “sampling zeros" while still retaining IPF for the synthesis (Garrido et al., 2019). Another generative approach uses Markov Chain Monte Carlo (MCMC) simulation (Farooq et al., 2013). This method also emphasizes reconstruction of the joint distribution by generating individuals from the distribution of sampled data using a Gibbs sampling method. Both of these methods were found to have computational efficiency advantages over the traditional IPF algorithm, given that neither requires fitting of the contingency table, which scales poorly with the number of variables included in the population. Last, there has been work using the Variational Autoencoder (VAE) model, which was first proposed as a scalable, unsupervised means to learn distributions of latent variables in a dataset (Kingma & Welling, 2014), and has most frequently been applied to tasks related to image generation using a convolutional architecture (Cai et al., 2019; Razavi et al., 2019). It has since been adapted for population synthesis in one paper by using socioeconomic data as inputs (Borysov et al., 2019).

Generative models have the potential to address one of the primary shortcomings of traditional population synthesis which is sampling zeros. One of the benefits of generative models is that they do not directly reweight the sampled population, instead drawing new individuals based on learned properties of the sampled data. This allows them to draw individuals who are statistically representative of the population being studied, and may exist in the marginal totals for the true population, but have joint characteristics that are not present in the survey microdata. This would most likely include individuals from smaller, traditionally

under-sampled groups. From the lens of socioeconomic data, this might include people with disabilities, racial minorities, or commuters with uncommon work schedules. These groups are frequently underserved by existing transportation networks, and generative models could provide a means to better understand their commuting behavior, and plan accordingly. Generative models also have the potential to address the poor scalability of traditional synthesizers as the number of modeled variables increases. Due to the fact that they are trained using various methods, and their training time can be dependent on hyperparameters and training methods used, it is somewhat difficult to say definitively whether generative models are faster than reweighting methods. However, many of the aforementioned works have benchmarked generative techniques and empirically found them to be faster than traditional synthesis (Farooq et al., 2013; Pritchard & Miller, 2012; Sun & Erath, 2015). Last, it is worth mentioning that in the absence of sampling zeros, IPF/IPU synthesizers will always converge on a solution. Generative and combinatorial optimization based methods rely on gradient descent algorithms to search for minima and are not guaranteed to converge on the global solution.

Although there are several approaches to using generative models for population synthesis, this work focuses on implementing and improving the VAE. Prior work has found VAEs to scale more effectively than other generative approaches, and to create more accurate synthetic populations in cases with more than a handful of input variables (Borysov et al., 2019). The VAE model is also relatively easy to implement and train using open source deep learning libraries, which can aid with consistency between implementations by different works, and findings more practically useful for replication (Team, n.d.). Last and most importantly, VAEs are a topic of open research in image generation from which there have been many findings which may be useful for applications in population synthesis. For example, this work proposes

and tests a CVAE hierarchical model to incorporate household characteristics when generating individuals; this model was originally proposed for different purposes by the same author as the original VAE (Kingma et al., 2014). The remainder of this section discusses some of the basic use-cases for VAEs in these other fields, and how they are relatable to population synthesis.

2.2.2 VAEs and Image Generation

VAE models have proven exceptionally successful in the field of image recognition and generation (Cai et al., 2019; Higgins et al., 2016). They are a strong candidate for population synthesis because the nature of the tasks that they have been successful at in image generation (expand a sample while maintaining the relationships between features in the sample), are similar to the task of a population synthesizer. The fundamental concept relies on using unsupervised learning to determine latent variables present in the training data, and sample from these latent variables to generate new data (Kingma & Welling, 2014). A more complete outline of the exact VAE structure and generation process is described in Section 3.1. Because they can incorporate many different types of artificial neural network (ANN) architectures, VAEs are also easily adaptable to different contexts and datasets. Simultaneously, because the ANN architecture is constrained to the encoder/decoder/latent variable relationships defined in the VAE, the results they produce are more interpretable than those of a typical ANN.

In the context of image generation, VAEs are typically designed with a convolutional architecture, in which a kernel is used to look over windowed sets of neighboring pixel values and learn relationships between pixels (O'Shea & Nash, 2015). They are capable of learning how distinct features in the pixel data such as individual pixel darkness, lines, and other shapes form

latent variables such as skin color or gender. Notably, the latent space is continuous, meaning that each latent variable will exist on a spectrum (Figure 1).

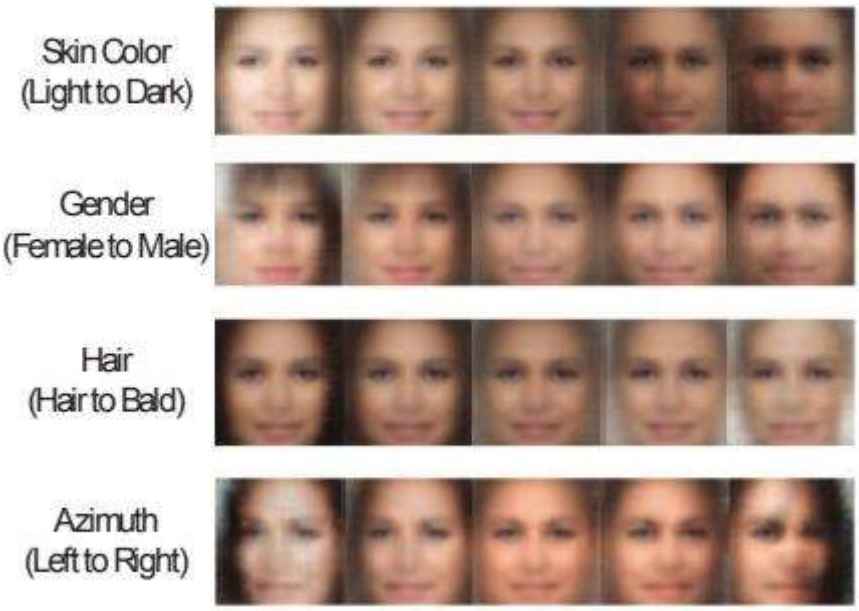


Figure 1: Latent variables visualized by a VAE in an image generation context (Uncovering and Mitigating Algorithmic Bias through Learned Latent Structure | Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, n.d.).

A great deal of research has been performed on developing new methods to force the latent structure in VAEs to represent certain variables (Burgess et al., 2018; Chen et al., 2019; Higgins et al., 2016). In image recognition, one benefit of this has been that it allows for generating images of individual faces with under-sampled features, and using them to de-bias facial recognition models based on limited datasets (Bao et al., 2017; Uncovering and Mitigating Algorithmic Bias through Learned Latent Structure | Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, n.d.). In the context of population synthesis, this could both be seen as a solution to the sampling zeros problem, as well as an opportunity to generate more data from under-sampled population groups for use in microsimulation. Another method

used to encourage specific structure in the latent space of a VAE by its original creator is the Conditional-VAE (CVAE) model (Kingma et al., 2014). This model incorporates additional "conditional" variables into the VAE during training and generation, allowing the user to provide information on what type of sample should be generated (e.g. training the CVAE on a set of animal pictures, while supplying the type of the animal as a conditional variable, then generating new images of a specific animal by imputing that type of animal as the conditional variable repeatedly).

One issue for VAEs in image generation is that the use of the continuous latent spectrum creates "noise" in the generated images which is not present in the training data (i.e. the issue of structural zeros where samples are generated that do not exist in the true population). One method of addressing this has been the introduction of generative adversarial networks (GANs) which use an entirely different, less interpretable structure than the VAE, but can produce higher quality images (Creswell et al., 2018; Goodfellow et al., 2014). GANs are typically used in place of VAEs where sharp images are desired, but require careful tuning and copious amounts of training data. In a population synthesis context, GANs have yet to be researched, but may provide additional accuracy to the synthetic population at the cost of training time and overall model simplicity.

2.2.3 VAE/CVAEs in Population Synthesis and Transportation Modeling

Ultimately, comparisons between VAE models used for image generation and population synthesis are abstract, due to their respective input data and applications being somewhat conceptually far apart. In the field of transportation, VAEs have been relatively unexplored as a method of combining behavioral and socioeconomic variables, or increasing under-sampled

groups in socioeconomic survey data. The most notable study examined the feasibility of the algorithm, but did not validate their synthetic population against population data outside of the training data, or otherwise put it to use in travel demand modeling (Borysov et al., 2019). Furthermore, only the most basic VAE structure was used, and more complex generative models such as the CVAE, GANs or Recurrent Neural Networks (RNNs) could be utilized to accommodate individuals and households, yield more accurate results, or to incorporate time series data such as housing market shifts respectively.

This prior work did benchmark several of the aforementioned generative models (Bayesian Network, Gibbs Sampler) against a VAE-based population synthesizer, and found it to be the most computationally efficient model, with overall similar results in terms of accuracy between the models. As dimensionality of the synthetic population was increased, the VAE began to perform better than other methods (~20 variables) eventually outperforming them in all but univariate distributions (the Gibbs sampler performed best for low-dimensional synthesis with 0-4 variables, and the Bayesian network for mid-dimensional synthesis with 4-21 variables). Unfortunately, traditional IPF/IPU and combinatorial optimization approaches were not tested. Separately, additional work has been performed using a CVAE model to develop a large, cross-sectional dataset of different socioeconomic archetypes, by imputing fixed conditional variables (Borysov & Rich, 2020). Specifically, conditional variables indicating psychosocial preferences (willingness to change behavior), were input to the CVAE, and it was used to generate a large dataset from which the cross-section of individual characteristics in each conditional group could be examined. Although it is not discussed in the paper, CVAE models may provide a convenient method to expand samples from small, or traditionally under-sampled populations in household survey data.

VAEs and other generative population synthesis methods are still in a nascent state relative to traditional population synthesis. Despite the benefits of these models, they lack key features such as the ability to easily model multiple sub-regions, or forecast future populations for use in activity based modeling. One primary benefit of traditional methods is that they are capable of generating joint synthetic populations of individuals and households, by incorporating them into the balancing process when using the IPU algorithm. However, current generative models do not incorporate the joint individual-household structure of most household surveys into their synthesized populations (Fabrice Yaméogo et al., 2021; Fournier et al., 2021). In this work, the primary contribution is a generative VAE model which is capable of synthesizing simultaneous household and individual data, that does not rely on heuristics or post-synthesis grouping to place individuals in representative households. To accomplish this, a model using the CVAE structure is proposed to generate individuals conditional on a set of synthesized household variables. This model is then tested against state of the art combinatorial optimization-based population synthesis methods. An overview of the model and its detailed specification is discussed further in Section 3.1.

2.2.4 Potential Use of VAE/CVAEs to Synthesize from a Small Microdata Samples

VAEs, and other generative models use different methods to learn the joint probability distribution of input variables, to generate new individuals from those variable relationships. This is to say that because generative models do not rely on reweighting the existing microdata samples, they are able to generate individuals who are “out of sample” and thus they are not susceptible to the sampling zeros problem. This trait may be particularly useful in cases where only a small amount of microdata is available. On a sparse dataset, it is highly likely if not

certain that traditional synthesis methods will not be able to converge, given their reliance on reweighting the sampled data to meet the marginal counts. This issue is multiplicative if the sample is highly dimensional, because it creates a contingency table with many zero-cells, which may either lead to the sampling zeros problem (in which a marginal count cannot be met because there are no samples for it) or overweighted samples (in which a small number of samples for a rare subset of the population are inflated to represent a much larger group). VAEs in particular have performed exceedingly well on highly dimensional population synthesis, when trained with a relatively large dataset. In the case of a small microdata sample with many socioeconomic variables, they are untested.

If a VAE is able to learn relationships between input variables well enough to generate an accurate latent space based on sparse training data, sampling from the VAE may allow the reconstruction of a much larger dataset than the original input, with statistically representative combinations of the input variables. Given the proclivity of machine learning approaches to overfitting the training dataset, there is some concern that when using a smaller input dataset it may be difficult to generalize the model to an entire population. The question of how to create models that generalize on small datasets is an open area of research (Olson et al., 2018). However, the unique structure of the VAE model with its latent space sampling lends itself well to natural regularization (and even over-regularization (Takahashi et al., 2019)), which provides some amount of resistance to overfitting (O'Malley et al., 2019; Shen et al., 2020). Other studies have found VAEs useful in augmenting or drawing latent variables from small datasets (Chadebec & Allassonnière, 2021; Mannam & Kazemi, 2020; Wang et al., 2020). In the realm of population synthesis, this could make VAE models a valuable research tool for examining the effects of new variables not typically sampled in household surveys, that must be expanded to

regional or national levels for microsimulation. This has potential to be particularly useful in situations where a relatively small, but rich dataset is available (e.g. a psychometric survey of individual preferences related to travel behavior), but must be scaled up for simulation on a macroeconomic (e.g. a national choice model) scale. To determine the extent to which the CVAE model is capable of this, its ability to generate a population similar to the complete testing dataset, under increasingly sparse training data, is tested and documented in Section 4.2.

3. Methodology

3.1 VAE/CVAE Model Structure

The general VAE structure is shown below in Figure 2. It consists of an encoder neural network attached to a sampling layer of latent distributions, which feeds into a mirrored decoder. During training, the encoder attempts to learn a transformation of the input variables into a set of latent distributions, represented by a set of means (μ) and variances (σ). Using Equation 2, a sample (Z) for each latent variable (D) is then drawn from the distributions, and provided as the input to the decoder, which during training aims to reconstruct the original input variables.

$$Z_D = \mu_D + \sigma_D \odot \epsilon, \ \epsilon = N(0,1) \quad (2)$$

In this way, the model is essentially learning to perform compression and decompression of the input data by mapping it to a continuous, compressed latent space. Ideally, this is done without losing information in the original sample, so that it can be reconstructed as close to the original as possible by the decoder. To accomplish this, two loss terms are used to form the objective function on which the model is trained. The first is the reconstruction loss (Equation 2) which

quantifies the difference between the input vector (t) and the generated output from the decoder (s) for each class of a given variable (k) across (n) classes, using the categorical cross-entropy equation. Prior to this, the softmax function is used to normalize the output from the decoder into a set of probabilities for each class (j) of a given variable (i). Then, the categorical cross-entropy is summed across each variable (k) for the total number of variables in the synthetic population (m). Equation 3 thus shows the loss calculation for a single reconstructed individual against the input values from the training data.

$$\text{Reconstruction Loss} = -\sum_{k=1}^{m}\sum_{i=1}^{n} t_i \log(\text{softmax}(s)_i),\ \text{softmax}(s)_i = \frac{e^{s_i}}{\sum_{j=1}^{n} e^{s_j}} \quad (3)$$

When using the VAE for population synthesis, mixed continuous and categorical socioeconomic variables might be desired to represent a household or individual, and thus the reconstruction loss might require separate calculations to account for each of these variable types. Because categorical cross-entropy is not intended to analyze continuous variables, a separate loss function must be incorporated for continuous variables in addition to the categorical cross-entropy term for the categorical variables. During initial testing of the model, a separate mean squared error term was used to determine reconstruction loss for continuous variables. This also required balancing the continuous and categorical reconstruction loss terms; introducing additional hyperparameters in the model. While developing this model, much worse performance was observed in the reconstruction when using a mean squared error term to quantify the loss for continuous variables, something which was also observed in previous work (Borysov et al., 2019). Instead, continuous variables were binned into quantiles, which allowed for the categorical cross-entropy loss term to be used across all reconstructed variables.

$$\text{KLD Loss} = -\frac{1}{2}\sum_{i=1}^{D}(1 + \log(\sigma_i) - {\mu_i}^2 - \sigma_i) \qquad (4)$$

The second loss term used is the Kullback-Leibler Divergence (KLD), which is used to quantify the difference between two probability distributions. In this case, the distribution of each latent variable is compared to a normal distribution, which trains the model to create a normally distributed latent space. This aids the model in drawing new samples of these variables during the synthesis process. When applied to a normal distribution, the KLD loss term for (D) latent variables can be written as Equation 4 shown above.

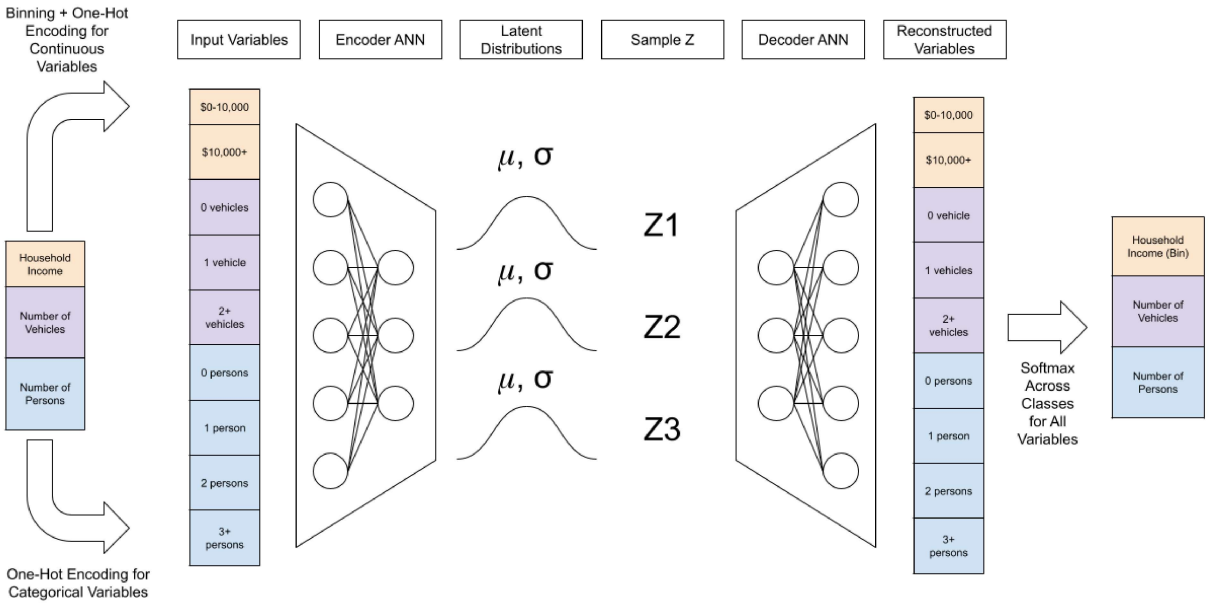


Figure 2: Variational Autoencoder structure used for synthesis of socioeconomic variables. Three input variables (1 continuous, 2 categorical) and 3 latent variables are shown.

Once the VAE has been trained, it can be used to generate new, statistically representative samples from the original dataset. To generate new samples using this model, a set of random normal samples (one for each latent variable) are fed into the decoder. In the context

of population synthesis, this means that a generated individual will have a random normal set of latent traits as dictated by the learned latent variables. The assumption here, is that during training the KLD loss term forces each latent variable in the model to follow its own independent, univariate normal distribution. If the model has been trained correctly, this will ensure that over many samples being drawn, an accurate reconstruction of the original population will be made. If the model has not learned to normalize the latent variables, it will produce strange combinations of traits. If the model has learned to normalize the latent variables, but not reconstruct the data, it will tend to produce samples from the mean value in the training data. The balance between these outcomes is achieved with a weight on the KLD loss term which was tuned using a grid search optimization. This phenomenon is documented in Section 4.3, and the model training process is documented in Section 3.2.

One shortcoming of the VAE model is that it cannot easily synthesize the individuals comprising a household without the use of additional heuristics. The IPF/IPU algorithms accomplish this by weighting samples directly from the input data, which for most household surveys includes identifiers joining individuals to their households. To address this problem while maintaining the benefits of generative models, a dual-VAE/CVAE model structure is proposed that first generates a population of synthetic households, then incorporates the variables from those households into the synthesis process for individuals. This structure is summarized in Figure 3 below.

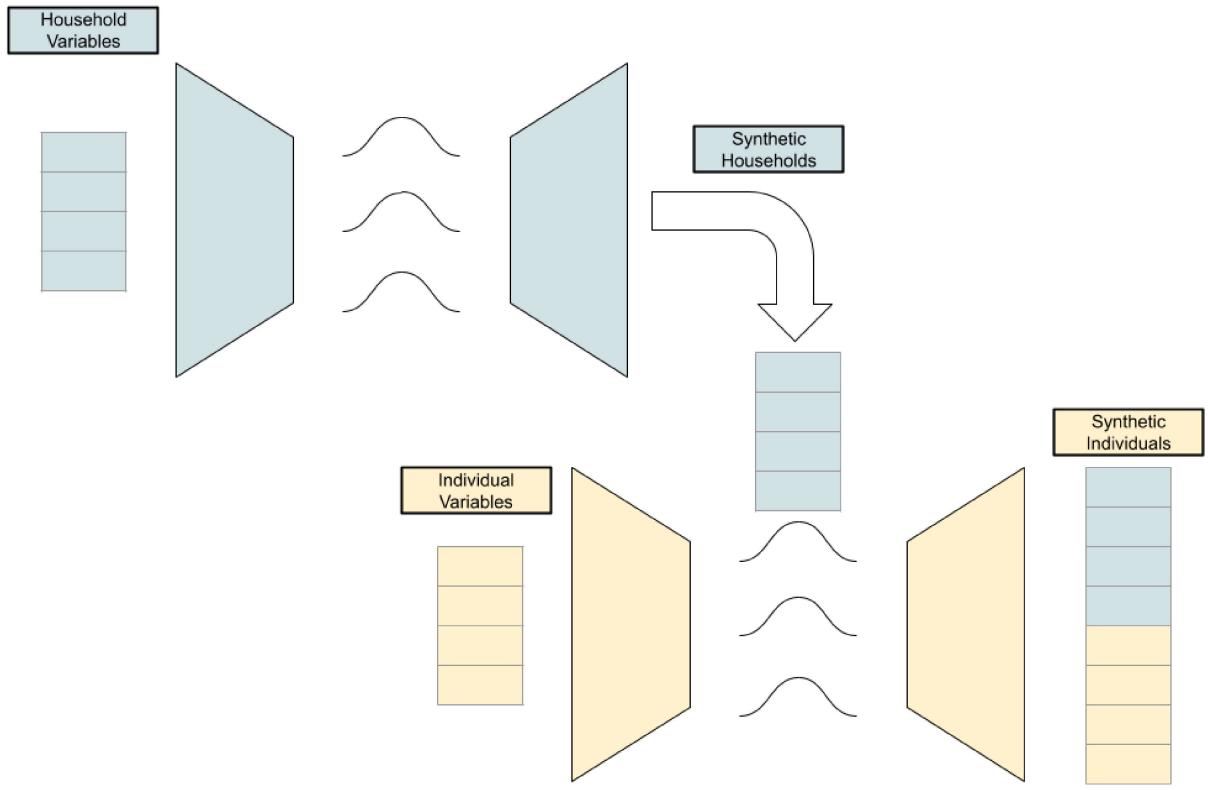


Figure 3: Individuals are generated for each synthetic household by passing household variables alongside latent samples to the decoder. During training, the Individual CVAE is fed both household and individual variables, to enable the decoder to learn the relationship between them.

This model requires separate training of the Household (VAE), and Individual (CVAE) models, then combination of their outputs during synthesis. This means that when an individual is generated from the distribution of latent variables in the Individual CVAE, their final attributes are conditional on the characteristics of the household they are being generated to fill. This reduces the need of combining individuals into households post-synthesis, for which most existing methods are ad hoc (Ye et al., 2009). Importantly, this structure necessitates a training dataset in which individuals are already matched to households, as both person and household level characteristics are used to train the Individual CVAE. Fortunately, this is common in most household surveys such as the public use microdata samples (PUMS) provided by the ACS.

This contribution is novel in that there is currently no established generative method for synthesizing both households and their comprising individuals, and it is valuable in that this is a core component of modern population synthesizers, and highly informative in the realm of transportation modeling. The primary benefit of using a CVAE over a VAE is to add a level of control over what types of samples are drawn during the generation process. In population synthesis, this allows samples to be generated from the Individual CVAE which are conditional on a set of household variables previously generated by the Household VAE. Currently, in population synthesis for transportation modeling, this added control has only been leveraged to generate large, cross-sectional datasets of synthetic individuals belonging to certain sub-populations (Borysov & Rich, 2020).

3.2 Model Development

The process of designing neural network architectures and selecting hyperparameters varies widely based on the domain and task at hand. For this work, a relatively basic model was selected. To optimize the hyperparameters, a grid search optimization was performed. Prior work informed the range of values to test, and ultimately there were some small variations from previous VAE models (shown in Table 1). This is not surprising, given that the prior focus was on testing computational efficiency across generative models, and involved a much higher dimensional space, which was stated to be computationally infeasible for IPF based synthesis (Borysov et al., 2019).

Table 1: List of Hyperparameters Tested During Grid Search Optimization

Model	Number	Number	Number	Number	Batch	Beta	Gradient	Node

	of Layers	of Nodes	of Latent Variables	of Training Epochs	Size		Descent Algorithm	Activations
CVAE	1, 2, 3	8, 16, 32, 64	2, 3, 4, 6, 8	1000	64, 128, 256, 512, 1024	.01, .05, .1, 1	Adam	Tanh
(Borysov et al., 2019)	1, 2, 3	25,50, 100	5, 10, 25	100	64	.01, .05, .1, .5, 1, 10, 100	RMSProp	Tanh

In the lower dimensional case tested here, the number of latent variables was reduced accordingly (so as to maintain the bottleneck structure of the compression to latent space). Additionally, better results were found using the Adam optimization algorithm which separates learning rates for each parameter (Kingma & Ba, 2017). For node activations in both the encoder and decoder, the Tanh function was used. The output layer of the decoder uses a softmax function which normalizes the output to a probability across each output class. Given the dual VAE (household) and CVAE (individual) architecture, separate grid searches were performed for each half of the model.

To quantify the accuracy of a synthetic population across model runs, a metric is needed to compare the distributions of socioeconomic variables of the generated population to that of a separate testing dataset. Standardized Root Mean Squared Error (SRMSE) was used to evaluate the similarity of marginal and joint distributions of each population generated from a set of hyperparameters to the PUMS data. Previous research has also used the Standardized Root Mean Squared Error metric for this purpose (Borysov et al., 2019; Hu et al., 2018; Pritchard & Miller, 2012). The calculation for this metric is shown in Equation 5 below. Continuous variables are binned, and bin frequencies (π) are calculated across each bin for continuous variables, and across each class for categorical variables. This is repeated for the bivariate distributions of each

possible combination of variables. The root mean squared error between the distribution of variables in the true population (π) and the synthetic population ($\hat{\pi}$) is then calculated by comparing the frequencies of each univariate and bivariate bin, and finally standardized by the mean frequency per bin of the true population.

$$\mathrm{SRMSE}(\hat{\pi}, \pi) = \frac{\sqrt{\sum (\hat{\pi} - \pi)^2 / N_b}}{\sum \pi / N_b} \qquad (5)$$

This approach to quantifying the accuracy of the synthetic joint distribution is similar to the least squares approach used in the original IPF algorithm (Deming & Stephan, 1940). One potential weakness is that it relies on Euclidean distance between two observations, and thus will penalize more heavily cells with common population traits that likely contain larger overall magnitudes, relative to less common traits which will be smaller and more likely to carry small differences between the original and synthetic populations. On one hand, this ensures that the metric effectively captures the largest portion of a given population. On the other hand, it might under-penalize models that are less accurate on niche subsets of the target population. An alternative metric that might mitigate this to some degree would be Standardized Mean Absolute Error (SMAE), which would simply calculate the mean difference between synthetic and true population frequencies across each bin. This would have a linear penalty increase with the magnitude of the bin, with the downside being that it may decrease the overall performance of the model.

Each combination of hyperparameters was ranked according to 1) SRMSE of the univariate distribution 2) SRMSE of the bivariate distribution and 3) training + generation time

for the synthetic population. Figure 4 displays an example comparison of the hyperparameter sets which scored highest in these three metrics for the CVAE model, while Table 2 shows a summary of the final selected hyperparameters for both the VAE and CVAE models.

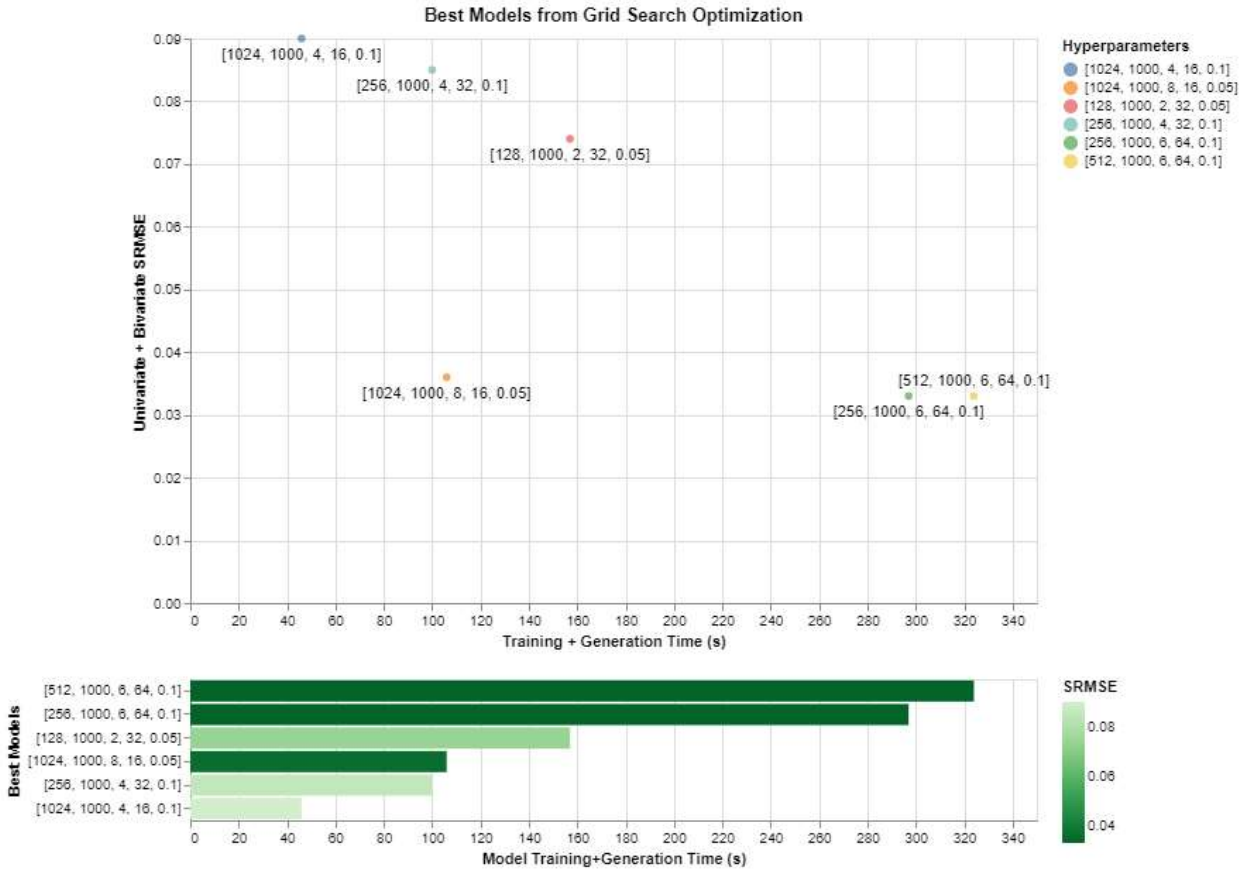


Figure 4: Results of the grid search optimization for models with 2 layers, hyperparameters are listed as [Batch Size, Epochs, Latent Dimensions, Nodes per Layer, Beta]. Some models are significantly faster than others, with little sacrifice to SRMSE. In general, the individual model takes much longer to train due to more input variables and higher dimensionality of the latent space.

The results of the grid search were relatively consistent for most models with a tradeoff between model complexity (more nodes, higher number of latent dimensions) and training time. Because the number of epochs were fixed, the batch size played a large role in training time for the model. During each epoch, the model splits the training data randomly into batches and

propagates each batch through the network to estimate the current gradient, then takes a step. Each iteration through the complete training data is a single epoch. Using smaller batch sizes slows down the training and creates more steps, but can help the model find a better solution. The CVAE model in Figure 4 which was most efficient used 8 latent variables to minimize the amount of compression in the model, thus accomplishing a low SRMSE, while using a relatively small number of nodes and large batch size to keep its training time low. The Beta hyperparameter controls the weight of the KLD loss during training, and can play a disproportionately large role in the accuracy of the model. The results of varying Beta are discussed in Section 4.3.

Table 2: Selected Hyperparameters for VAE/CVAE Models

	Number of Layers	Number of Nodes	Number of Latent Variables	Number of Training Epochs	Batch Size	Beta	Gradient Descent Algorithm	Node Activations
VAE (Household)	2	16, 16	4	1000	1024	0.1	Adam	Tanh
CVAE (Person)	2	16, 16	8	1000	1024	.05	Adam	Tanh

3.3 Data Sources

The primary dataset used as the microdata inputs for both the CVAE and traditional synthesizers is drawn from the 2017 Public Use Microdata Samples (PUMS) data provided by the American Community Survey (ACS). These samples contain anonymized responses to the census with socioeconomic and demographic characteristics included, and individual records can be joined into households based on shared IDs. To protect the privacy of respondents, location

data is not provided for individuals. Rather, they are aggregated to Public Use Microdata Areas (PUMAs) which are designed by the ACS to be drawn along census tract lines and contain approximately 100,000 individuals each. For each PUMA, 5% of the census responses are available in the PUMS. This means that for a given PUMA, there are approximately 5,000 responses available in the data. In this study, microdata is limited to samples collected from Washington state. For the traditional synthesis, marginal count data is also required for control variables at each geographic level specified. In this case, only a single geographic level is tested (Washington state). The marginal count data was drawn from several socioeconomic tables in the ACS. All data was downloaded at the census tract level, then aggregated to a single region. Additionally, a "geographic crosswalk" is required when using IPF based population synthesizers, which maps microdata areas to the regions at which marginal counts are provided. In this case, since only a single region is synthesized, this file simply maps each PUMA to a single shared region.

In all testing cases, several variables from the PUMS were used as both control variables in the traditional population synthesis, and input training variables in the VAE/CVAE. A mix of continuous and categorical variables were drawn from both household and person-level datasets, so as to test the ability of the generative model to recreate complete households rather than pure individuals which would otherwise be grouped heuristically. The variables used in this analysis and their code in the PUMS are listed in Table 3 below:

Table 3: Variables taken from PUMS

Variable Type	Continuous Variables (binned)	Categorical Variables
Household	• Household Income (HINCP)	• Number of Persons (NP) • Number of Vehicles

		(VEH) • Number of Building Units (BLD) • Internet Access (ACCESS)
Person	• Personal Income (PINCP) • Age (AGEP) • Commute Time (JWMNP)	• Gender (SEX) • School Completed (SCHL) • Marital Status (MAR) • Work Sector (COW) • Ambulatory Difficulty (DPHY) • Vision Difficulty (DEYE) • Cognitive Difficulty (DREM) • Primary Race Code (RAC1P)

3.4 Testing Details

To examine the efficacy of the VAE/CVAE generative model against traditional synthesis methods, benchmarks of accuracy and speed against the Popsim open-source population synthesizer were tested. Additionally, to determine whether generative synthesis methods could be used to recreate a rich dataset from sparsely sampled individuals, synthesis was performed on increasingly sparse training data and degradation to the accuracy of the generated population examined. The results of these analyses are summarized in Sections 4.1.1 and 4.1.2 respectively. The software used in this study to benchmark traditional synthesizers is a modern, open source, combinatorial optimization-based population synthesizer called Popsim, which was built to be part of the ActivitySim microsimulation framework (ActivitySim — ActivitySim 0.9.7 Documentation, n.d.; Paul et al., 2018). Popsim is a synthesizer capable of matching both household and individual control variables at multiple geographic levels. For the purposes of this study, only one overarching geographic region is used. This limits the complexity of the analysis, and again reveals one shortcoming of generative synthesis methods:

they cannot easily account for multiple geographies without using separate models and datasets for each region. In future work, conditional variables could be added which are representative of a given region (e.g. the median household income, or the count of households containing three individuals). This would allow the CVAE to generate a new individual who is conditional on both characteristics of the household, as well as characteristics of a particular region. Because the model is trained directly on the socioeconomic relationships in the microdata, the most disaggregate area of analysis this would allow is the area at which the microdata samples were collected. In this case, that would be the PUMA-level.

4. Results

4.1 CVAE Performance Benchmarking

To first analyze whether the CVAE model was worth pursuing, accuracy comparisons using SRMSE are drawn between the CVAE and Popsim models. This is intended to provide insight as to whether the CVAE generative model is even capable of matching the performance of a traditional synthesizer when it comes to creating an accurate synthetic population. Once a comparison of accuracy has been determined, it is contextualized against the primary purported benefit of generative models, which is their scalability as the number of modeled variables increases. Previous research has compared the accuracy and scalability of generative models against one another, but has not directly benchmarked VAEs against traditional synthesis. The results may indicate whether there is a tradeoff to using generative models (e.g. the model scales better with the number of modeled variables, but sacrifices accuracy in the synthetic population), or if it provides any benefits at all (i.e. whether there are any accuracy or scalability benefits to speak of).

4.1.1 Accuracy of the Marginal and Joint Distributions

After training the CVAE and PopSim models on the complete PUMS dataset for Washington state (~150,000 individuals) and generating a test synthetic population consisting of 100,000 individuals, SRMSE was calculated for both synthetic populations (Figure 5). The results indicate that the CVAE outperforms the traditional population synthesizer in this metric, with a 7.1% reduction in univariate SRMSE, and 2.3% reduction in bivariate SRMSE. It is worth noting that the minimum value for SRMSE is zero, which would indicate a perfect match between cell frequencies in the synthetic and target populations.

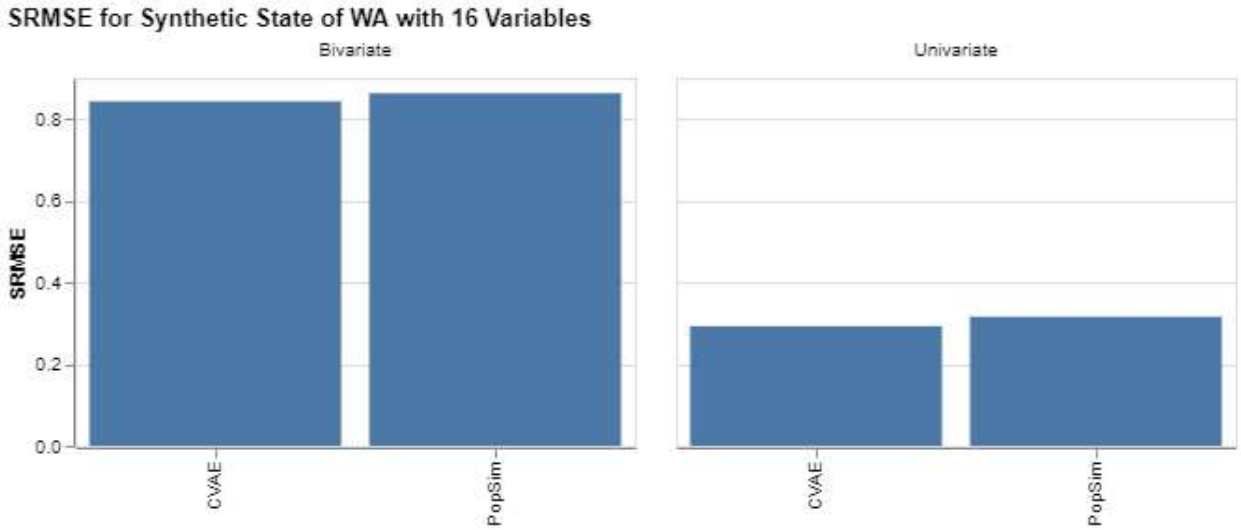


Figure 5: SRMSE comparison of CVAE, PopSim, and sampled PUMS data.

Although SRMSE provides a summary of the accuracy of the generated population, it is still useful to compare the univariate distributions of each variable in the synthetic population as a check against their overall accuracy. Figure 6 shows the empirical cumulative distributions of

each variable in the synthetic populations generated by the CVAE and traditional synthesizer, as well as the original distribution from the PUMS data.

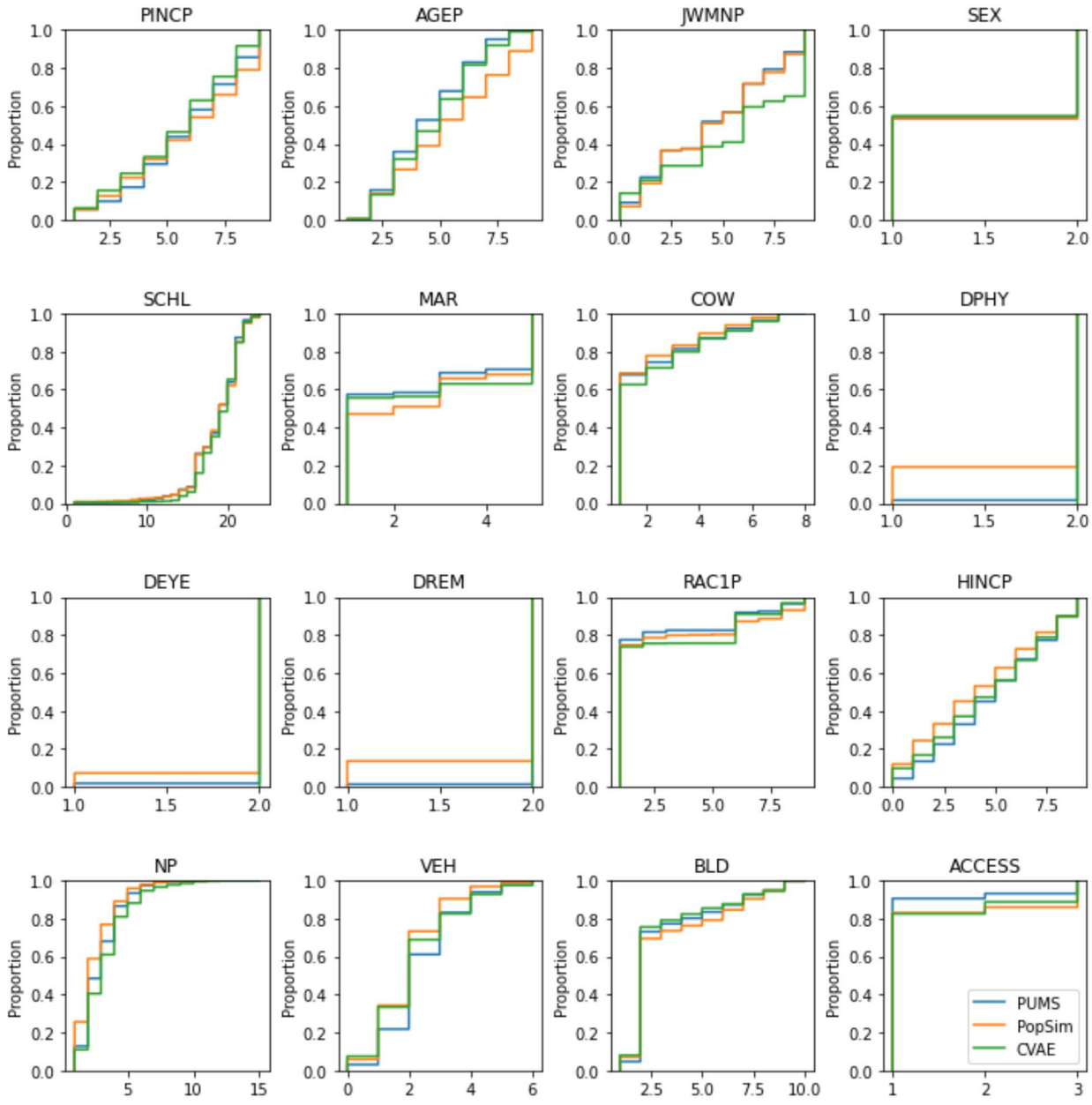


Figure 6: Empirical cumulative distribution of all input variables in the PUMS data, and synthetic populations generated by PopSim and the CVAE model.

In most cases, both the CVAE and PopSim perform reasonably well at reconstructing the distributions found in the original PUMS data. In some binary variables where one category is

much less likely than the other (e.g. DPHY, DEYE, DREM) the CVAE performs poorly; it tends to always assume the more common option. This could be the result of undertraining, or lack of complexity in the model, since information can be lost during the compression process. PopSim is also less accurate on these variables, although it performs better than the CVAE. In a few other multi-class cases (e.g. MAR, BLD, NP) the CVAE is more accurate, while for at least one (RAC1P) it fails to reconstruct the distribution. It is difficult to say what drives this accuracy; however, in all cases both methods are capable of reconstructing the overall shape of the univariate distributions.

4.1.2 Computational Efficiency

One of the potential benefits of generative synthesis methods is their computational efficiency relative to traditional IPF synthesizers. As the synthetic population increases in dimensionality, the benefits of this efficiency become more pronounced. Beyond a certain point, it is infeasible to use IPF for generating a synthetic population, however combinatorial optimization methods may have potential to help. Time to convergence is dependent on both the number of regions, and dimensionality of the population being generated. Prior research has compared the efficiency of different generative models in the extremely high-dimensional (50+ variables) case (Borysov et al., 2019). However, there may be situations where it is desirable to rapidly generate populations in the low- to mid-level dimensional cases, where both generative and traditional synthesis are viable. This overlapping region is compared by using the CVAE and PopSim models to generate a synthetic population for the state of Washington (~3 million households). The PopSim synthesizer (along with most modern IPF/IPU synthesizers) is capable of generating a population composed of multiple geographical sub-regions, but for the sake of

fair comparison it is only used at the state level. This is because IPF/IPU based synthesizers scale with the number of regions being modeled, and thus their computational performance would be diminished relative to the CVAE, which is currently limited to generating populations at a single geographic region. The results of this time trial are shown in Figure 7 below:

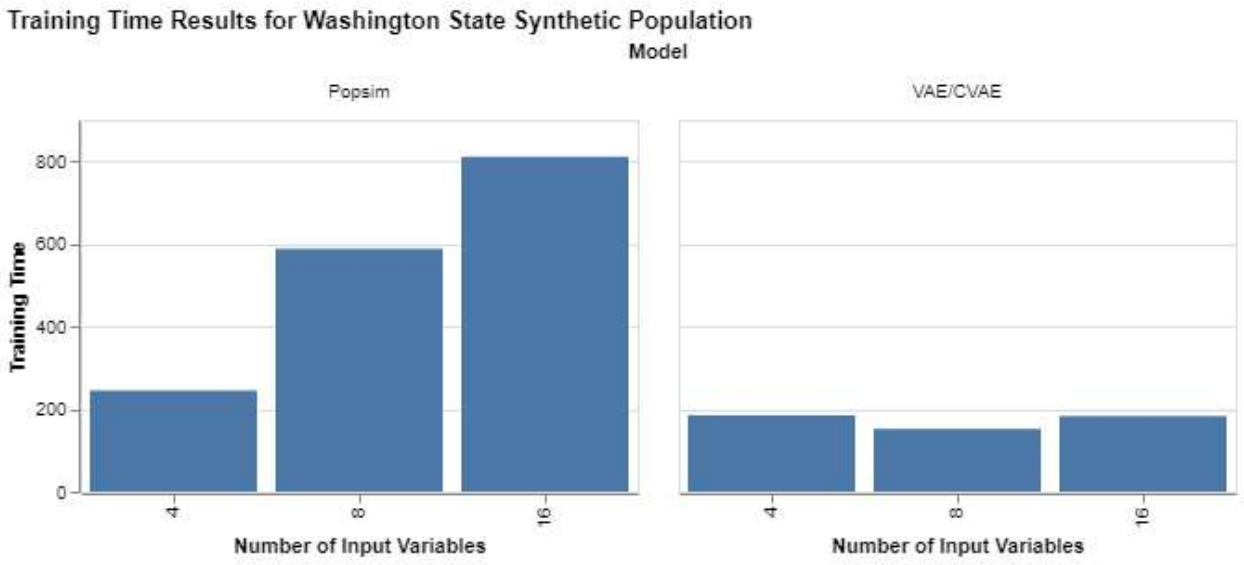


Figure 7: Training times under an increasing number of input variables for traditional synthesizer (convergence) and combined VAE/CVAE models (1000 epochs).

The CVAE model outperformed the traditional synthesizer by essentially maintaining a static training time regardless of the number of input variables. There is a small amount of random variation in training times between runs of the CVAE, and these fluctuations outweigh any of the effects of increasing the input variables on this scale. The traditional synthesizer, on the other hand, increases convergence time logarithmically with the number of training variables. Overall, the Popsim synthesizer performs quite well, and the VAE/CVAE even better.

4.2 CVAE Under Sparse Training Data

One of the untested potential benefits of generative models is in generating synthetic populations when only sparse microdata samples are available, as in theory they can learn the joint distribution of input variables, and generate new, representative individuals who are out-of-sample. This is something that traditional synthesizers struggle with, because in any case where a marginal count cannot be met using the sampled data, issues with non-convergence arise. This becomes more and more likely as the quantity of microdata decreases, or the number of modeled variables increases. With the exception of public agency administered household surveys, it is unrealistic to recreate a population from 100,000 samples. In research or smaller scale applications it is more likely that 1000, 100, or even only a handful of samples might be collected. To determine whether the CVAE model is capable of accurately expanding these samples into a reconstructed population, tests were performed on decreasing amounts of training data randomly sampled from the PUMS data. Each time, a synthetic population of 100,000 individuals was generated after training, and compared to a separate random sample of 100,000 individuals from the PUMS data. The accuracy of the reconstructed synthetic population was once again quantified using univariate and bivariate SRMSE (Figure 8).

Figure 8: Univariate and Bivariate SRMSE of synthetic population generated by CVAE under decreasing training data.

Both univariate and bivariate SRMSE were found to improve rapidly as the quantity of training data increased until approximately 100 training samples were used; after which diminishing returns set in. The univariate distributions of all variables in the synthetic populations, as well as the PUMS data are shown below in Figure 9. In most cases, the CVAE is effective in reconstructing the population. The exception to this are the populations generated on 1 and 10 training samples, which are (unsurprisingly) far off. These distributions provide context to the rapid increase in reconstruction accuracy in Figure 8 as 1-100 samples are used, and the subsequent diminishing returns of adding more training samples. Most distributions track the PUMS data well with some amount of random variation around the true distribution.

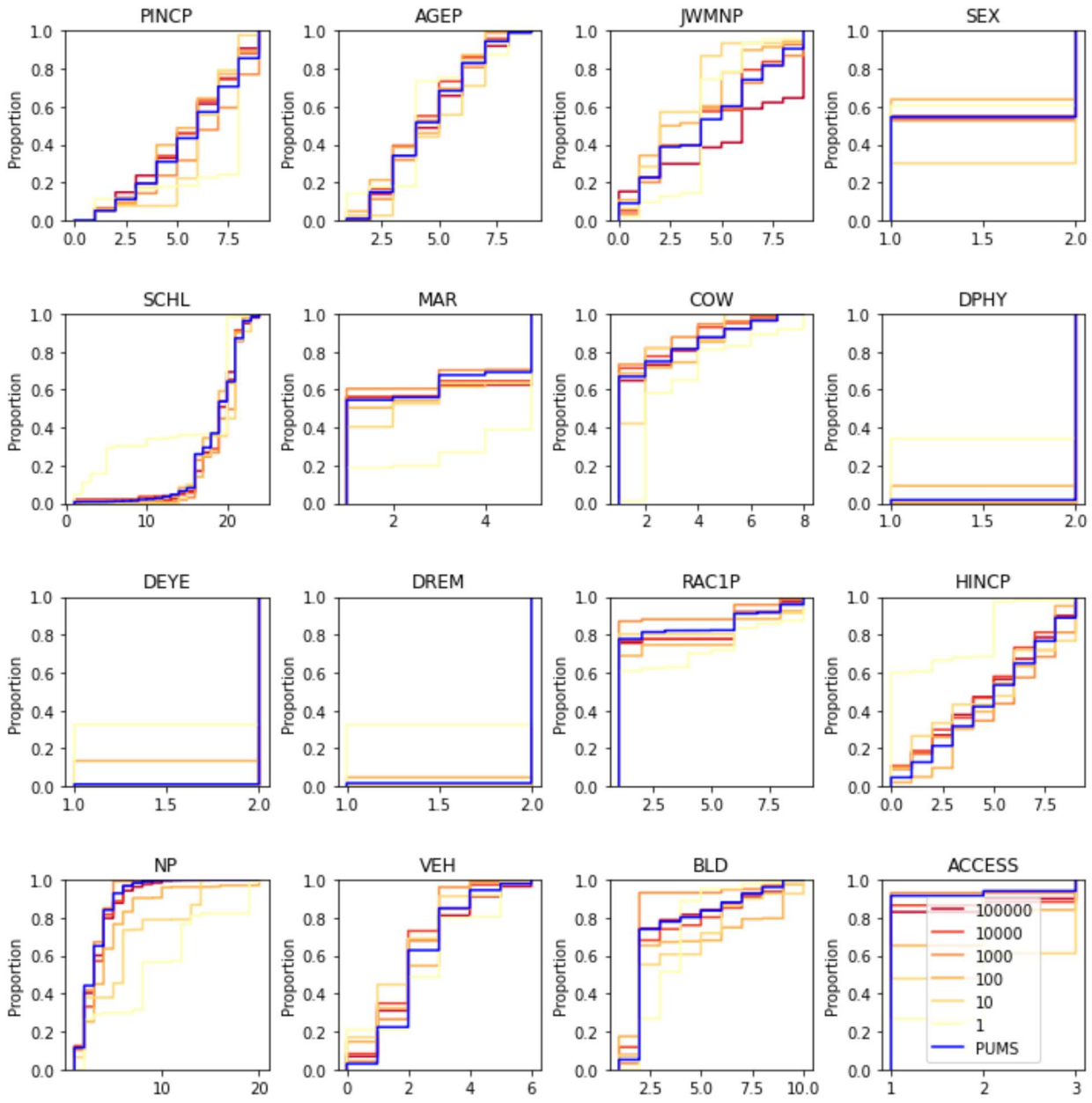


Figure 9: Distribution of each input variable in the synthetic population generated by the CVAE under decreasing training data.

While training on these smaller datasets, one trend that arose was the failure of the CVAE to avoid overfitting. Figure 10 provides a summary of the evidence for this, across three model runs for 10, 100, and 1000 samples respectively. In this figure, each line indicates one of the loss terms across the 1000 training epochs, with the red line indicating validation loss. The validation

loss is calculated periodically during the training process by inputting a set of "holdout" test data to the model and calculating the loss terms against it. As the model learns from the training data, all loss terms would be expected to decrease. When the model begins to overfit on the training data, the Reconstruction and KLD losses would be expected to decrease, while the validation loss begins to increase. This indicates that the model is improving its loss on the training data through adjustments that do not generalize to other (non-training) data points. This is evidenced in Figure 10, where the validation curve begins to trend upwards later and later as more training data is introduced. This is because with a small amount of training data, the complexity of the dataset is low enough that it is easier for the model to fit each individual sample than learn the complex relationship between the input variables. Also known as the bias-variance tradeoff; the model is essentially learning to fit the noise of each individual point in the training data, rather than a generalizable relationship between the points. In the context of the VAE, the model can predict exact values by mapping to latent variables which have near-zero variance, thus allowing it to output specific values during training. This of course makes it generalize poorly, as during the generation process random normal samples are fed into a decoder expecting precise inputs. At approximately 1000 samples, overfitting may still occur on the training data, but it no longer negatively affects the validation loss.

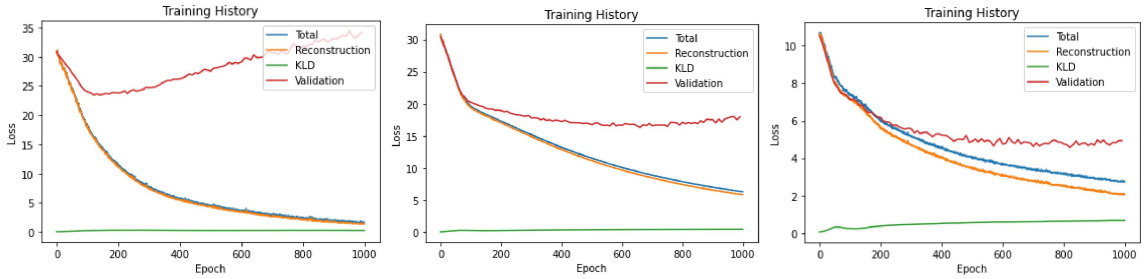


Figure 10: When training using smaller datasets, the model begins to overfit despite the natural regularization. Training losses for 10, 100, and 1000 samples are shown above.

VAEs in general are fairly resistant to overfitting due to the random sampling performed in the sampling layer, which causes the same input into the encoder to produce slightly different output values across different runs. This makes it more difficult for the model to fit precisely to the variance in the training data, because there will be some element of unpredictable, additional variance added by the random sampling. That being said, the aforementioned issue of a sampling layer which produces latent distributions with a variance of zero can sidestep this regularization. This is seen in the case where the drastically reduced quantity of training data allows the model to accomplish extremely low reconstruction loss by learning noise in the training dataset. To help prevent this, early stopping (~200 epochs) and L1 regularization were used in the CVAE when training on the smallest datasets. In addition, batch sizes for the model were decreased proportionally to the number of training samples being used, so as to not train on a larger batch size than the number of training samples. Overall, as shown in Figure 9, as few as 10 samples can be used to generate a reasonably accurate picture of the true population. Variables that suffer most in the synthesis are binary variables with relatively low occurrence rates, for which the CVAE tends to assume the most common class. This is likely due to undertraining the model. To explain further; there are two possible outcomes for these variables, with one outcome having extremely high likelihood (99.9%) in both the true population and in the training data. In terms of reconstruction loss, it is quite simple for the model to find a local minimum by always assuming the more common outcome, but difficult to find the global minimum by predicting the rare cases of these variables. This is likely exasperated by the fact that these particular variables represent traits that are not strongly dependent on the other socioeconomic variables in the

analysis (e.g. physical, hearing, and cognitive disability), and thus when compressed to latent space are the first information lost in the compression process. If the model were trained longer, and on a larger dataset that contains more examples of individuals with the rare outcome for these variables, it might perform better at synthesizing them. Although not tested for this purpose here, CVAEs have been used to address this issue in other fields by generating a wide range of new training samples for rare input data to de-bias facial recognition software (Uncovering and Mitigating Algorithmic Bias through Learned Latent Structure | Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, n.d.).

4.3 Effect of Beta on the Latent Space and Synthetic Population

One additional phenomenon observed during the CVAE model development process was the effect of the Beta (KLD loss weight) hyperparameter on the distribution of latent variables, and accuracy of the generated synthetic population. During model development and testing this term was found to have a significant effect on the accuracy of the generative model, something which has also been observed in the image recognition space (Higgins et al., 2016). Because this phenomenon is undocumented in generative synthesis, and minimally documented in the general VAE literature, it creates a potential pitfall for anyone attempting to recreate these results. Thus, a discussion of the effects of altering Beta on the synthetic population and latent space is included here.

The Beta hyperparameter is used as a multiplier for the KLD loss term, shown in Equation 6 for the total loss on a single training data sample below:

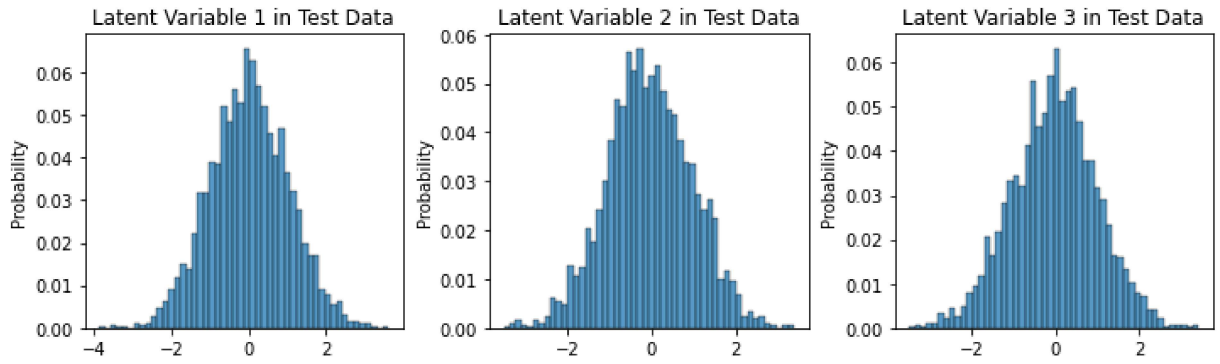
$$\text{Total Loss} = -\sum_{k=1}^{m}\sum_{i=1}^{n} t_i \log(\text{softmax}(s)_i) + \beta * -\frac{1}{2}\sum_{i=1}^{D}(1 + \log(\sigma_i) - {\mu_i}^2 - \sigma_i) \quad (6)$$

In this case, a single training sample represents either a single individual, or a single household sent through the VAE/CVAE model and compared to its reconstruction. The first term is the reconstruction loss (Equation 3), and the second term is the KLD loss (Equation 4). The KLD loss is multiplied by the Beta hyperparameter. The Beta hyperparameter thus exerts "pressure" on the model during the training process to construct latent distributions which approximate a normal distribution as closely as possible, according to the Kullback-Leibler Divergence. When the Beta hyperparameter is high, the KLD loss term will have a larger impact on the gradient, and the model will favor learning network weights which minimize this loss at a cost to reconstruction loss. This means that during training the model will prefer to maintain normal latent variables, even if they make the reconstructed socioeconomic outputs less accurate to the inputs. In this work, the latent variables remain "entangled" in that they are not trained to represent independent factors in the dataset (e.g. those shown in Figure 1). The original proposition for the Beta hyperparameter specified a set of "disentangled" or conditionally independent data generative factors (z) which were kept close to known prior distributions using the loss in Equation 6, which is the KLD loss for the real (p) and estimated (q) latent distributions.

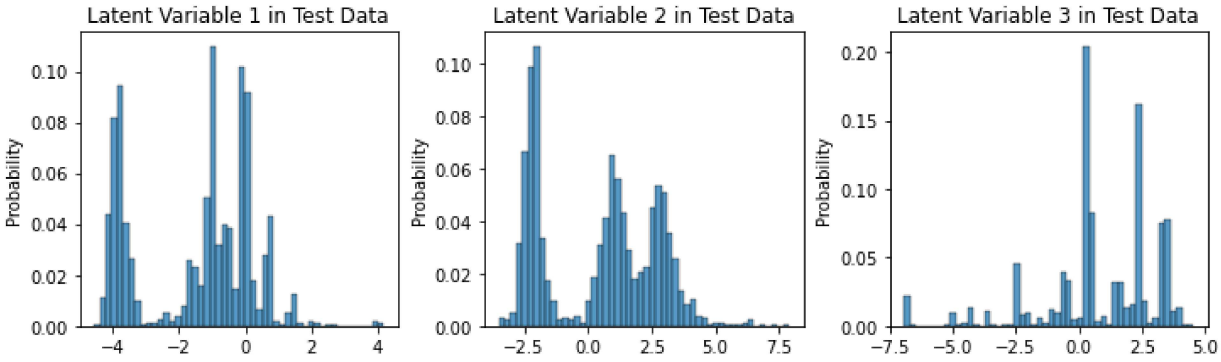
$$\beta * \mathrm{KLD}(q_\Phi z|x)||p_\theta(z) \qquad (6)$$

Below, the results of training and generating synthetic populations using two extreme values of Beta are shown. Figure 11 shows the distribution of the sampled Z values gathered from feeding a holdout test set through the encoder, after training the model on the training data

(i.e. it shows the values of the test data mapped to the latent space). This allows the visualization of the latent distributions, which are otherwise assumed normal by the encoder and are described as only a mean and variance. In the case where Beta is high, the latent variables take a well-formed normal distribution, as would be expected if the KLD loss was weighted higher during training. In the case where Beta is low, the latent variables become poorly-formed distributions, with many spikes and little coherence.



(a)



(b)

Figure 11: Distribution of three latent variables given a Beta of 5.0 (a) and .0005 (b). In one case, the model achieves normal latent distributions, in the other, they are skewed.

Of course, if the latent distributions are well-formed, it might seem that the variables in the synthetic population will be accurate, but this is not the case as shown in the distributions of variables in the synthetic population produced from the Household VAE with a high Beta value, as shown in Figure 12 below:

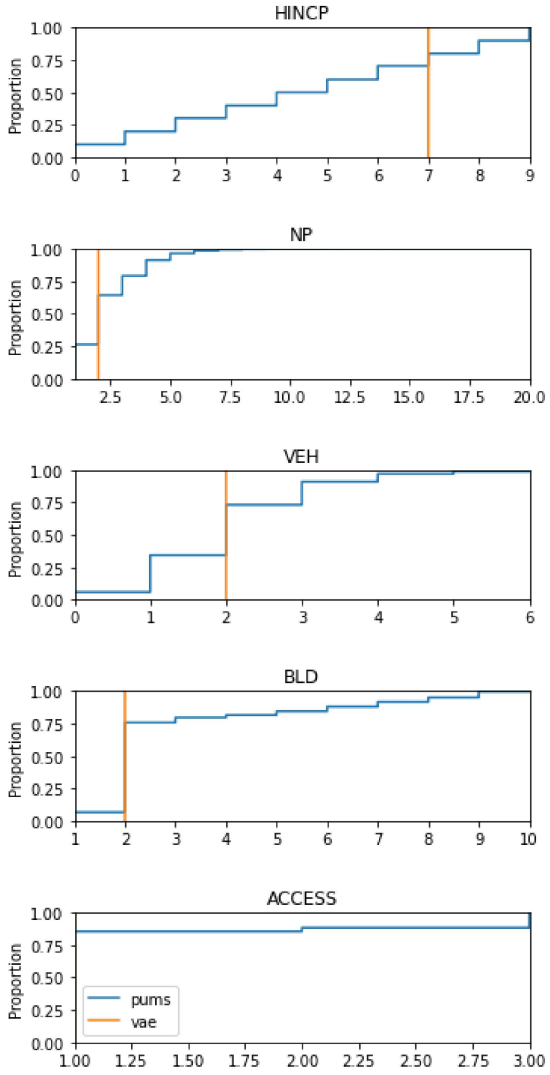


Figure 12: Due to the overly high Beta value, the model has not effectively learned to reconstruct the input distributions, instead focusing on the latent variables. More often than not, it tends to predict the average class for a given variable.

In this case, the Beta chosen is too high, and the model does not effectively learn to reconstruct the input variables. It emphasizes normally distributed latent variables at a cost to the reconstruction accuracy. When the synthetic population is drawn from normally distributed latent variables, they accurately represent the model's learned structure, but they produce meaningless results. A balance must be struck between achieving a KLD loss that pushes the model to form normally distributed latent variables, while still mapping those variables to accurate outputs at the decoder. Similarly poor results (although slightly more realistic) are generated from the model trained with an extremely low Beta value. In a sense, this model is overfitting the latent space: It has learned to effectively map the input data to latent distributions that are not necessarily normal, then sample those distributions to create precise reconstructions of the training data. However, when random normal samples are fed into the decoder in the process of generating new households, the model cannot generalize and produces the incorrect distributions shown below in Figure 13:

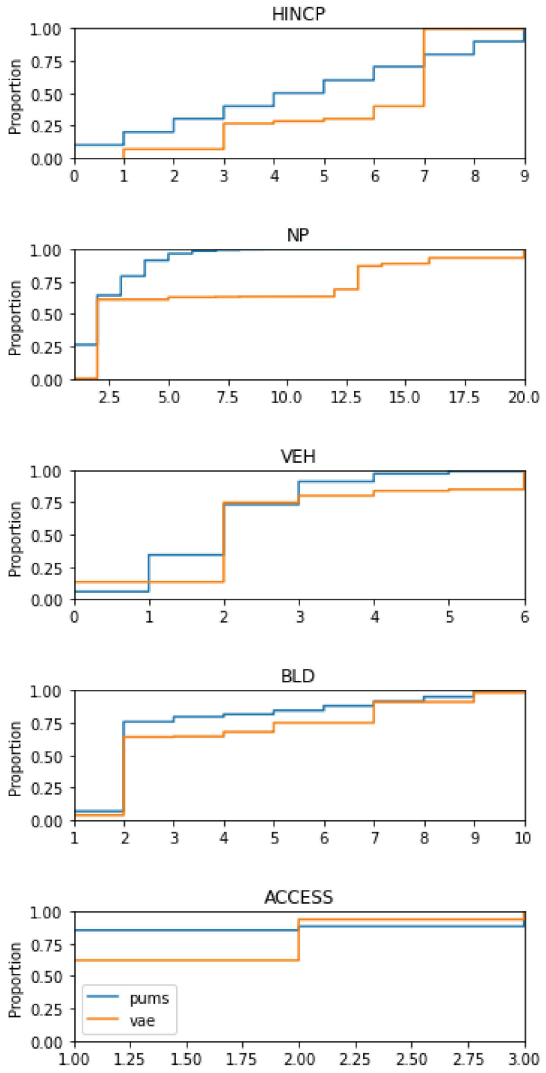


Figure 13: Although the model has somewhat learned to reconstruct the input data, the latent variables are not normally distributed, so treating them as normal and sampling from them leads to outputs with skewed distributions.

Ultimately, a balance must be struck between forcing the latent space to be normally distributed and allowing the model to learn accurate reconstructions. The originally proposed structure for VAEs did not include a weight on the KLD loss, and so essentially used a value of 1.0. In the image data space, the large impact of this hyperparameter on both the quality of the output, and stability of training has since been identified, making it fundamental to the VAE model (Higgins

et al., 2016). In this work, the grid search optimization found a value of .05 to 0.1 was found to accomplish the best SRMSE results for both the household and individual models. Other work has successfully incorporated Beta as a trainable model parameter, allowing the model to optimize its own weights for reconstruction and KLD losses (Asperti & Trentin, 2020). Although more complex to implement, using this method to determine Beta may have the potential to increase the accuracy of the CVAE model further than a simple grid search. In general, the best approach found as a result of this research is to maximize the KLD loss term, up until the point at which the generated data begins to suffer. This achieves the most normally distributed latent space possible, without sacrificing the model's generalizability. This trains the decoder to correctly interpret random normal samples (that are drawn separately from the model during the synthesis process to represent the latent variables).

5. Discussion

5.1 Advantages of the CVAE

The primary advantage of using a VAE/CVAE, or any other generative model in place of traditional population synthesis methods, is in its ability to rapidly scale the dimensionality of the population. In this analysis, it was found that regardless of the number of control variables, the CVAE outperformed the population synthesizer in terms of synthesis time. In general, the training time for the CVAE model was tied to hyperparameters of the model (e.g. number of training epochs, batch size, etc.) whereas the training time for the traditional population synthesizer was tied to the number of control variables. On one hand, it is possible to overcome this with the traditional synthesizer by using a larger number of input variables than control variables, and using the weights determined by the smaller number of control variables to scale

all of the variables. However, this is not necessarily a direct comparison to the CVAE, given that the CVAE will always incorporate all of the input variables (i.e. every variable in the CVAE model is always a control variable). While it might seem that these scalability benefits come at a cost to accuracy, the results of this work found the CVAE to be more accurate in reconstructing the target population than the Popsim model, as measured by the SRMSE. This test was performed by generating a synthetic population consisting of 100,000 individuals, and comparing it to a random sample of 100,000 individuals from the PUMS data for Washington state. A more accurate synthetic population should provide better estimates of the behavior of the population being studied. This result indicates that the CVAE model may be able to compete with traditional synthesizers in the future, despite currently lacking some of the features (e.g. geographic sub-regions, modeling future-year populations) of these more established models.

Also, the CVAE model is capable of generating new, out-of-sample individuals, which are still representative of the target population. Because the KLD loss term enforces normally distributed, continuous latent variables, the random samples drawn from this space do not directly represent an individual in the training sample. This is in contrast to traditional synthesizers, which reweight samples from the microdata to represent a larger population. The primary benefit of this is to avoid the sampling zeros problem, in which the microdata does not contain individuals belonging to certain niche groups with rare combinations of input variables, and therefore cannot represent those groups in the final sample. In traditional synthesis, this has the potential to leave out individuals with disabilities, uncommon work schedules, and any other population minority. This is particularly true when working with small, highly dimensional microdata samples. When using a generative synthesizer, any combination of input variables is possible, as sampling from the learned continuous latent space can lead to samples that are

statistically representative, yet not found anywhere in the original sample. This may contribute to the findings which showed the CVAEs ability to accurately reconstruct the target population under sparse training data: By learning the probabilistic relationships between the input variables, it was able to extrapolate those relationships to individuals which exist in the true population, but were not found in the training data.

5.2 Disadvantages of the CVAE

There are also a few disadvantages of the CVAE model. First, it is not guaranteed to match the marginal distributions of control variables perfectly. The accuracy of the model when generating new populations is dependent on how effectively it has been trained, and how well it is able to compress the input variables to latent space without losing information (i.e. how lossless is the compression). The random sampling of latent variables aids the generative model in preventing overfitting, however, in this analysis overfitting was still found to be an issue when using smaller training datasets (~40 samples or less). This could be due to simply not having enough information in such a small sample to fully understand the relationships between the input variables, or it could be a failure of the model and addressed through careful regularization techniques and additional hyperparameter tuning. Regardless of these concerns, the model was tested empirically against a traditional synthesizer, and outperformed it in terms of SRMSE.

There is also the issue of the CVAE requiring two models (two training cycles, two grid searches, etc.) to generate complete households with individuals. On one hand, this adds burden to the training and hyperparameter selection process. On the other, this did not cause any undue issues during this work, as the only real difference between the two are changes in the number of input and output variables fed to/from the encoder and decoder. This is analogous to the

additional rebalancing step introduced by the IPU algorithm over IPF synthesis. While there is a cost to balancing at both the household and individual levels, the benefit is a more realistic population that is grouped together in statistically representative households, similar to most household survey microdata. Ultimately, when performing population synthesis, one must weigh whether this information is useful in the final population, and whether the cost of additional model complexity and training time is worthwhile.

Last, the fundamental flaw of the CVAE and other generative models is that they do not currently have a way to match individuals to specific regions or geographic areas within the sample. Most if not all traditional synthesizers are capable of generating weights such that the marginal control variable counts are met at multiple geographic levels (e.g. state, census tract, and block group levels). However, this does require marginal count data for every control variable at each geographic level being modeled. One possible solution to this issue is to include that marginal regional data as conditional variables in the CVAE in the same way that the household variables have been used here. This would provide the CVAE with information about the distribution of certain variables in each given geography while generating new individuals. Alternatively, heuristic methods may be devised to split up the population among constituent geographies after it has been generated. Although possible, it is unlikely to be worthwhile to train separate models for each geographic region for more disaggregated regions. There is a time efficiency benefit to training with the CVAE over traditional synthesis, but the findings of this work suggest it would be quickly outweighed by the training costs unless only a handful of geographies were included.

6. Conclusions

6.1 Summary

The CVAE performs well where small datasets must be expanded to represent a full population. It can especially be useful when this sample is highly dimensional, and rapid-testing is desired due to the computational efficiency relative to traditional synthesizers. Given that the CVAE and other generative methods also require no additional data about the population (whereas traditional synthesizers require painstakingly organized marginal counts) it seems particularly practical in academic or other low-resource, rapid-prototyping applications. In testing, the CVAE scaled more effectively with the dimensionality of the synthetic population than the traditional synthesizer (26% improvement in time using 16 input variables), and was more accurate in both univariate (7.1% improvement) and bivariate (2.3% improvement) SRMSE.

These advantages may not hold in the case where an unrepresentative sample is used as input microdata for the model. In the case where representative marginal counts are available, traditional synthesis methods will be able to scale the unrepresentative microdata appropriately to match the true population, while the CVAE model will fail. The reason that the CVAE fails is because it relies on learning the probabilistic relationships between the input variables. If the sample contains biased relationships, the model will learn biased relationships. In the case where marginal counts are unavailable, and the distribution of variables in the true population cannot be estimated other than through the microdata, then the previously stated advantages and disadvantages of each model hold. An example scenario where the CVAE might be useful would be for a dataset consisting of detailed behavioral characteristics of a small number of individuals that must be scaled to assess regional adoption of a new transportation mode. There is unlikely to

be marginal count data available for such non-census related variables, so neither method will recreate the true distribution perfectly. Essentially, the traditional synthesizer will perform the same as resampling the microdata. However, the CVAE will be able to iterate faster, and generate new samples based on the relationships it learns between the surveyed variables (which for behavioral variables may also be highly dependent and thus benefit from a compressed latent representation). While the marginal statistics of the synthetic population in this case may end up skewed, there will be no issues with convergence, and a more diverse set of individuals will be generated, as it will not consist of a small number of reweighted samples.

Despite some advantages, it is unlikely that generative synthesis will replace traditional population synthesizers as the synthesis tool of choice for household surveys performed by public agencies or other large organizations. Not only are the computational efficiency advantages negligible when the model is only run once every few years, but these agencies must nearly always apply the synthesizer at multiple sub-regions, for which there is currently no practical approach when using generative methods. One approach to this may be to incorporate additional conditional variables into the CVAE model, which would provide information on the marginal totals for each sub-region. Given that the CVAE was found to scale well with increased input dimensions, this approach would be unlikely to affect its computational efficiency advantage (as opposed to building separate models for each sub-region). If the marginal count data (or even median values) for specific variables of interest were available at sub-region levels, these could be implemented into the CVAE model during training and generation as conditional variables. Thus, any individual generated from the CVAE synthesizer would be conditional on both a set of household-related characteristics, as well as region-related characteristics. This has the potential to address the primary missing feature of VAE/CVAE population synthesizers.

6.2 Contributions of this Research

This work contributes to generative population synthesis literature by proposing and testing a method of using the CVAE model to generate synthetic populations which contain representative households consisting of individuals in a computationally efficient way. This is analogous to the IPU algorithm which improved upon traditional IPF methods in population synthesis by incorporating a rebalancing step to ensure that both household and individual marginal totals were met. In this case, a VAE/CVAE structure is proposed which first generates households, then incorporates their variables as conditional inputs to the CVAE which reconstructs the population of individuals.

Additionally, the proposed CVAE model was tested against a state of the art implementation of a traditional population synthesizer. Tests on convergence and training times, as well as SRMSE were performed to quantify the advantages and disadvantages of either model in generating a synthetic population. Previous work has examined generative and traditional synthesis separately for low and high dimensional populations, but did not address the overlapping area where either model may be valid. During testing, it became apparent that the CVAE had potential to perform well as a synthesizer for small, highly dimensional training datasets.

6.3 Opportunities for Future Work

Perhaps the most immediate value in using the CVAE for practical population synthesis tasks would be to develop and test a method of incorporating regional control variables such that geographic subregions can be modeled. Currently, the CVAE can only generate synthetic

populations from which the training data was sampled, and multiple models and training sets would be needed for multiple geographies. Alternatively, a geographic ID variable such as zip code or census tract could be incorporated into the dataset itself, and the model trained to predict where each individual is based. In this work the PUMS data was used, which does not contain a geographic identifier lower than the PUMA, which is relatively aggregate. Overall, incorporating a way to model multiple geographies would allow a generative model to more closely replace the tasks performed by traditional synthesizers.

Additional generative models may also be explored. In the realm of generative modeling, VAEs are relatively dated compared to newer models such as Generative Adversarial Networks (GANs) and Recurrent Neural Networks (RNNs). The former has essentially surpassed the ability of VAEs in the field of image generation; they are frequently able to create new images that are indistinguishable from real ones to the human eye. Applied to population synthesis, a GAN model may achieve higher SRMSE accuracy, and more realistic households (e.g. ones where all individual incomes add exactly to the household income). On the other hand, RNNs are typically applied to time series data, and could be used to generate predictions given a previous sequence of states. In population synthesis, RNNs could then provide an opportunity to generate trip chains for synthetic individuals, or socioeconomic population shifts over time for an entire region, potentially allowing for the prediction of travel demand, and locations of regional growth centers, among other useful tasks.

7. References

Abraham, J. E., Stefan, K. J., & Hunt, J. D. (2012). Population Synthesis Using Combinatorial Optimization at Multiple Levels (No. 12–3383). Article 12–3383. Transportation Research Board 91st Annual MeetingTransportation Research Board. https://trid.trb.org/view/1130260

ActivitySim—ActivitySim 0.9.7 documentation. (n.d.). Retrieved April 12, 2021, from https://activitysim.github.io/activitysim/

Asperti, A., & Trentin, M. (2020). Balancing reconstruction error and Kullback-Leibler divergence in Variational Autoencoders. ArXiv:2002.07514 [Cs]. http://arxiv.org/abs/2002.07514

Auld, J. A., Mohammadian, A. (Kouros), & Wies, K. (2009). Population Synthesis with Subregion-Level Control Variable Aggregation. Journal of Transportation Engineering, 135(9), 632–639. https://doi.org/10.1061/(ASCE)TE.1943-5436.0000040

Bao, J., Chen, D., Wen, F., Li, H., & Hua, G. (2017). CVAE-GAN: Fine-Grained Image Generation Through Asymmetric Training. 2745–2754. https://openaccess.thecvf.com/content_iccv_2017/html/Bao_CVAE-GAN_Fine-Grained_Image_ICCV_2017_paper.html

Bar-Gera, H., Konduri, K., Sana, B., Ye, X., & Pendyala, R. M. (n.d.). Estimating Survey Weights with Multiple Constraints Using Entropy Optimization Methods.

Beckman, R. J., Baggerly, K. A., & McKay, M. D. (1996). Creating synthetic baseline populations. Transportation Research Part A: Policy and Practice, 30(6), 415–429. https://doi.org/10.1016/0965-8564(96)00004-3

Borysov, S. S., & Rich, J. (2020). Introducing synthetic pseudo panels: Application to transport behaviour dynamics. Transportation. https://doi.org/10.1007/s11116-020-10137-5

Borysov, S. S., Rich, J., & Pereira, F. C. (2019). Scalable Population Synthesis with Deep Generative Modeling. Transportation Research Part C: Emerging Technologies, 106, 73–97. https://doi.org/10.1016/j.trc.2019.07.006

Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., & Lerchner, A. (2018). Understanding disentangling in β-VAE. ArXiv:1804.03599 [Cs, Stat]. http://arxiv.org/abs/1804.03599

Cai, L., Gao, H., & Ji, S. (2019). Multi-Stage Variational Auto-Encoders for Coarse-to-Fine Image Generation. In Proceedings of the 2019 SIAM International Conference on Data Mining (SDM) (Vol. 1–0, pp. 630–638). Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9781611975673.71

Chadebec, C., & Allassonnière, S. (2021). Data Generation in Low Sample Size Setting Using Manifold Sampling and a Geometry-Aware VAE. ArXiv:2103.13751 [Cs, Stat]. http://arxiv.org/abs/2103.13751

Chen, R. T. Q., Li, X., Grosse, R., & Duvenaud, D. (2019). Isolating Sources of Disentanglement in Variational Autoencoders. ArXiv:1802.04942 [Cs, Stat]. http://arxiv.org/abs/1802.04942

Choupani, A.-A., & Mamdoohi, A. R. (2016). Population Synthesis Using Iterative Proportional Fitting (IPF): A Review and Future Research. Transportation Research Procedia, 17, 223–233. https://doi.org/10.1016/j.trpro.2016.11.078

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. (2018). Generative Adversarial Networks: An Overview. IEEE Signal Processing Magazine,

35(1), 53–65. https://doi.org/10.1109/MSP.2017.2765202
Deming, W. E., & 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. https://doi.org/10.1214/aoms/1177731829
Fabrice Yaméogo, B., Gastineau, P., Hankach, P., & Vandanjon, P.-O. (2021). Comparing Methods for Generating a Two-Layered Synthetic Population. Transportation Research Record, 2675(1), 136–147. https://doi.org/10.1177/0361198120964734
Farooq, B., Bierlaire, M., Hurtubia, R., & Flötteröd, G. (2013). Simulation based population synthesis. Transportation Research Part B: Methodological, 58, 243–263. https://doi.org/10.1016/j.trb.2013.09.012
Fournier, N., Christofa, E., Akkinepally, A. P., & Azevedo, C. L. (2021). Integrated population synthesis and workplace assignment using an efficient optimization-based person-household matching method. Transportation, 48(2), 1061–1087. https://doi.org/10.1007/s11116-020-10090-3
Garrido, S., Borysov, S. S., Pereira, F. C., & Rich, J. (2019). Prediction of rare feature combinations in population synthesis: Application of deep generative modelling. ArXiv:1909.07689 [Cs, Stat]. http://arxiv.org/abs/1909.07689
Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Networks. ArXiv:1406.2661 [Cs, Stat]. http://arxiv.org/abs/1406.2661
Guo, J. Y., & Bhat, C. R. (2007). Population Synthesis for Microsimulating Travel Behavior. Transportation Research Record, 2014(1), 92–101. https://doi.org/10.3141/2014-12
Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., & Lerchner, A. (2016). beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. https://openreview.net/forum?id=Sy2fzU9gl
Hu, J., Reiter, J. P., & Wang, Q. (2018). Dirichlet Process Mixture Models for Modeling and Generating Synthetic Versions of Nested Categorical Data. Bayesian Analysis, 13(1), 183–200. https://doi.org/10.1214/16-BA1047
IRELAND, C. T., & KULLBACK, S. (1968). Contingency tables with given marginals. Biometrika, 55(1), 179–188. https://doi.org/10.1093/biomet/55.1.179
Jebara, T. (2004). Generative Versus Discriminative Learning. In T. Jebara (Ed.), Machine Learning: Discriminative and Generative (pp. 17–60). Springer US. https://doi.org/10.1007/978-1-4419-9011-2_2
Joubert, J. W., & de Waal, A. (2020). Activity-based travel demand generation using Bayesian networks. Transportation Research Part C: Emerging Technologies, 120, 102804. https://doi.org/10.1016/j.trc.2020.102804
Kingma, D. P., & Ba, J. (2017). Adam: A Method for Stochastic Optimization. ArXiv:1412.6980 [Cs]. http://arxiv.org/abs/1412.6980
Kingma, D. P., Rezende, D. J., Mohamed, S., & Welling, M. (2014). Semi-supervised learning with deep generative models. Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, 3581–3589.
Kingma, D. P., & Welling, M. (2014). Auto-Encoding Variational Bayes. ArXiv:1312.6114 [Cs, Stat]. http://arxiv.org/abs/1312.6114
Lee, D.-H., & Fu, Y. (2011). Cross-Entropy Optimization Model for Population Synthesis in Activity-Based Microsimulation Models. Transportation Research Record, 2255(1), 20–27. https://doi.org/10.3141/2255-03

Mannam, V., & Kazemi, A. (2020). Performance Analysis of Semi-supervised Learning in the Small-data Regime using VAEs. ArXiv:2002.12164 [Cs, Eess, Stat]. http://arxiv.org/abs/2002.12164

Mohri, M., & Roark, B. (2005). Structural zeros versus sampling zeros. Oregon Health & Science University, Portland, OR, USA.

Moreno, A. T., & Moeckel, R. (2018). Population Synthesis Handling Three Geographical Resolutions. ISPRS International Journal of Geo-Information, 7(5), 174. https://doi.org/10.3390/ijgi7050174

Müller, K., & Axhausen, K. W. (2010). Population synthesis for microsimulation: State of the art. Arbeitsberichte Verkehrs- Und Raumplanung, 638. https://doi.org/10.3929/ethz-a-006127782

Olson, M., Wyner, A., & Berk, R. (2018). Modern Neural Networks Generalize on Small Data Sets. Advances in Neural Information Processing Systems, 31. https://papers.nips.cc/paper/2018/hash/fface8385abbf94b4593a0ed53a0c70f-Abstract.html

O'Malley, D., Golden, J. K., & Vesselinov, V. V. (2019). Learning to regularize with a variational autoencoder for hydrologic inverse analysis. ArXiv:1906.02401 [Physics]. http://arxiv.org/abs/1906.02401

Ortúzar, J. de D., & Willumsen, L. G. (2011). Modelling Transport. John Wiley & Sons.

O'Shea, K., & Nash, R. (2015). An Introduction to Convolutional Neural Networks. ArXiv:1511.08458 [Cs]. http://arxiv.org/abs/1511.08458

Paul, B. M., Doyle, J., Stabler, B., Freedman, J., & Bettinardi, A. (2018). Multi-level Population Synthesis Using Entropy Maximization-Based Simultaneous List Balancing (No. 18–03886). Article 18–03886. Transportation Research Board 97th Annual MeetingTransportation Research Board. https://trid.trb.org/view/1496005

PopGen. (n.d.). MARG - Mobility Analytics Research Group. Retrieved April 12, 2021, from https://www.mobilityanalytics.org/popgen.html

Population Synthesizer Development – PopulationSim. (n.d.). RSG. Retrieved May 30, 2021, from https://rsginc.com/project/population-synthesizer-development/

Pritchard, D. R., & 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. https://doi.org/10.1007/s11116-011-9367-4

Razavi, A., Oord, A. van den, & Vinyals, O. (2019). Generating Diverse High-Resolution Images with VQ-VAE. https://openreview.net/forum?id=ryeBN88Ku4

Richardson, A. J., Ampt, E. S., & Meyburg, A. H. (n.d.). Survey Methods for Transport Planning. 475.

Ryan, J., Maoh, H., & Kanaroglou, P. (2009). Population Synthesis: Comparing the Major Techniques Using a Small, Complete Population of Firms. Geographical Analysis, 41(2), 181–203. https://doi.org/10.1111/j.1538-4632.2009.00750.x

Shen, X., Liu, B., Zhou, Y., Zhao, J., & Liu, M. (2020). Remote sensing image captioning via Variational Autoencoder and Reinforcement Learning. Knowledge-Based Systems, 203, 105920. https://doi.org/10.1016/j.knosys.2020.105920

Sun, L., & Erath, A. (2015). A Bayesian network approach for population synthesis. Transportation Research Part C: Emerging Technologies, 61, 49–62. https://doi.org/10.1016/j.trc.2015.10.010

Takahashi, H., Iwata, T., Yamanaka, Y., Yamada, M., & Yagi, S. (2019). Variational

Autoencoder with Implicit Optimal Priors. Proceedings of the AAAI Conference on Artificial Intelligence, 33(01), 5066–5073. https://doi.org/10.1609/aaai.v33i01.33015066
Team, K. (n.d.). Keras documentation: Variational AutoEncoder. Retrieved May 30, 2021, from https://keras.io/examples/generative/vae/
Uncovering and Mitigating Algorithmic Bias through Learned Latent Structure | Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society. (n.d.). Retrieved May 27, 2021, from https://dl.acm.org/doi/abs/10.1145/3306618.3314243
Wang, Q., Meng, F., & Breckon, T. P. (2020). Data Augmentation with norm-VAE for Unsupervised Domain Adaptation. ArXiv:2012.00848 [Cs]. http://arxiv.org/abs/2012.00848
Ye, X., Konduri, K. C., Pendyala, R. M., Sana, B., & Waddell, P. (2009). Methodology to Match Distributions of Both Household and Person Attributes in Generation of Synthetic Populations (No. 09–2096). Article 09–2096. Transportation Research Board 88th Annual MeetingTransportation Research Board. https://trid.trb.org/view/881554
Zhu, Y., & Ferreira, J. (2014). Synthetic Population Generation at Disaggregated Spatial Scales for Land Use and Transportation Microsimulation. Transportation Research Record, 2429(1), 168–177. https://doi.org/10.3141/2429-18
Zhuge, C., Li, X., Ku, C.-A., Gao, J., & Zhang, H. (2017). A heuristic-based population synthesis method for micro-simulation in transportation. KSCE Journal of Civil Engineering, 21(6), 2373–2383. https://doi.org/10.1007/s12205-016-0704-1