

Activity Sequence Modelling with Deep Generative Models (Sometimes Deep Generative Models Might be All You Need)

Fred Shone*¹ and Tim Hillel²

¹PhD Student, Behaviour and Infrastructure Group (BIG), UCL, UK

²Assistant Professor, Behaviour and Infrastructure Group (BIG), UCL, UK

SHORT SUMMARY

Activity-based models typically structure the scheduling of activities as several sequential choices, e.g. number of activities, primary activity location and duration, secondary activities, etc. Conventional models cannot consider these choices simultaneously - limiting realistic interactions between them. Instead, we apply deep generative models for modelling activity sequence choices simultaneously, allowing full interaction between all dimensions.

We evaluate two different data and model structures. The first uses an image-like representation of activity sequences, the second a text-like representation. We present results demonstrating the practical considerations as well as quality of these models in application. We use a Variational Auto-encoder architecture to provide realistic aggregate as well as dis-aggregate distributions.

Our approach provides an alternative to discrete choice and scheduling based approaches for some applications. Our work also provides behavioural insight into the scheduling process.

Keywords: Activity-based modelling, statistical machine learning, deep generative models.

1 INTRODUCTION

The generation of realistic samples of synthetic activity sequences is a critical component of both Activity-Based Models (ABMs) (Bowman & Ben-Akiva, 2001) and simulation-based transport approaches such as MATSim by W Axhausen et al. (2016), as well as in other domains such as energy and epidemiology. Activity sequence modelling can be decomposed into (i) **participation**, i.e. choosing from the available activities, and (ii) the order and timing of those activities, commonly called **scheduling**. Conventional models, such as ActivitySim by Galli et al. (2009), are unable to model these dimensions simultaneously. Realistic interactions between these choices are therefore limited.

In this paper, we test deep generative models for generating synthetic activity sequences. These models have the potential to learn all real interactions within the scheduling process, by implicitly modelling the different choice dimensions simultaneously. We aim for generated sequences to be **correct** - where correct is defined as (i) being realistic at an individual level and (ii) being representative in aggregate when repeatedly sampled. We define an evaluation framework to assess this correctness. Our work demonstrates learning the distributions of sequences within a population, without additional information, such as income or age distributions.

We share the results of experiments to assist the future development of these models. Our experiments also allow consideration of the nature of activity sequence choice - whether activity sequences are best structured as sequential or simultaneous processes.

We have created the open-source Python software *CAVEAT* (2023) for the development and evaluation of generative activity sequence models.

Activity Modelling

The planning of activity sequences for an individual can be thought of as a complex set of interacting choices. These choices interact with each other, the choices of others, and with the environment. The dominant approach to model activity sequences is with hierarchies of sequential choices, each modelled using discrete choice models. These choices are then combined into coherent sequences using bespoke rules or scheduling algorithms, such as by Manser et al. (2022).

In practice such activity-based models require numerous connected components. They require measurement or estimation of costs for every possible alternative choice. They also require significant simplifications of the choice sets. This simplification restricts their realism, and limits realistic interaction between choices.

Pougala et al. (2023) show an alternative methodology that allows for multiple choice sets to be combined into a single optimisation problem. Allowing simultaneous consideration of traditionally separate choices. The approach samples the combined choices set to make the optimisation feasible. This sampling is computationally expensive, and so does not scale well to millions of agents.

Deep Generative Models

Deep generative models have made headlines for the generation of realistic images and sequences of text. We propose that the approach is also applicable for transport modelling - producing realistic activity sequences - with sensible data and compute requirements.

Deep generative models have already seen some preliminary use in the transport domain - primarily for generating synthetic population attributes (Borysov et al., 2019; Kim & Bansal, 2023). There has also been application of deep generative models to more complex data structures; for example by Choi et al. (2021), for vehicle trajectories. Koushik et al. (2023) generate activity sequences conditional on agent attributes. They find aggregate realism challenging, particularly the correct representation of infrequently observed activities. Their model generates sequences conditionality on demographic attributes, therefore does not learn the underlying distribution of activity sequences in their sample.

2 METHODOLOGY

We consider historic activity sequences sampled from a population, for example from a travel diary survey. We aim to model new sequences, using this sample, that are representative of the whole population.

For this paper we assume representativeness of the historic samples, and so require our model to replicate the distribution of the sample. Future work will add conditionality to the model, so that we can correct for bias in the sample, or use the model to generate samples for a different population.

We use deep generative models for this task. The generative model is trained with the sampled sequences, then is used to generate new sequences that can be used in travel demand models or for simulation. This is classed as self-supervised machine learning.

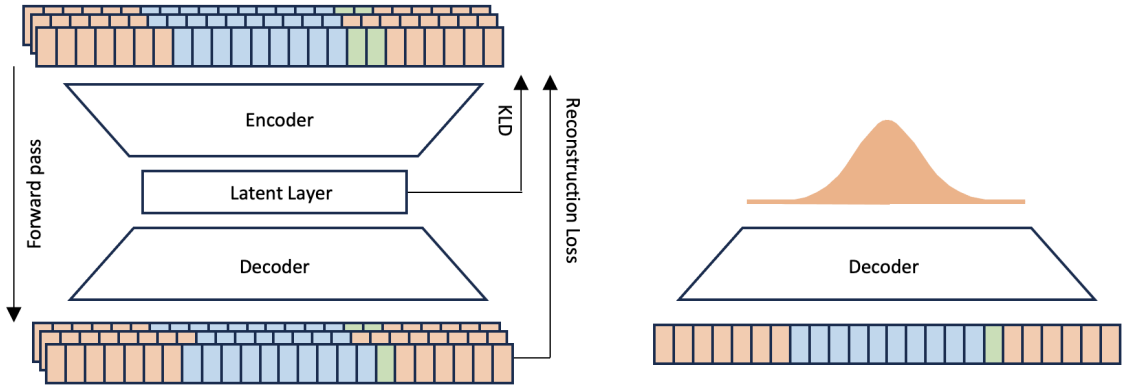
Our models are based around the VAE architecture by Kingma & Welling (2013). During training a VAE learns to map from a “latent space” with a known distribution (typically normal) to the training data distribution. After training, novel sequences are generated by sampling the latent space and applying the learnt mapping. See Figure 1 for an overview of VAEs.

In order to generate synthetic samples, the sample of historic schedules must first be represented with an encoding that can be used by the generative models. We test two different encodings in this paper - (i) image-like, where time is discretised into a fixed number of steps, like pixels in an image, and (ii) text-like, where schedules are encoded as a sequence of activities with associated durations.

Data

We use 2021 data extracted from the UK National Travel Survey (NTS) trip table as the sample of historic sequences. We convert this trip data into daily activity sequences using *Population Activity Modeller (PAM)* (2024). We additionally filter the sample of activity sequences to only include those that start and end at a home activity. We do this so that we can create a simple evaluation of the structural quality of synthesised plans. Figure 2 shows example activity sequences from NTS.

This creates approximately 37,000 sequences composed of 137,000 activities. We refer to this data as the *observed* sample of sequences. Output sequences are then called *synthetic* samples. Models



VAE Training Phase. Batches of training sequences are passed through an encoder – decoder structure. The latent layer between represents a statistical distribution. Loss is composed of (i) reconstruction loss and (ii) KL divergence. Reconstruction loss is a measure of how well the full model can reconstruct the training sequences. KL divergence is a measure of how well the latent space matches a target normal distribution.

VAE Inference Phase. Once trained we can think of the decoder as providing a mapping between a normal distribution and the real sequences distribution in the training data. By repeatedly sampling from the targeted latent normal distribution, we can generate a new population of sequences.

Figure 1: Primer to Variational Auto-encoders

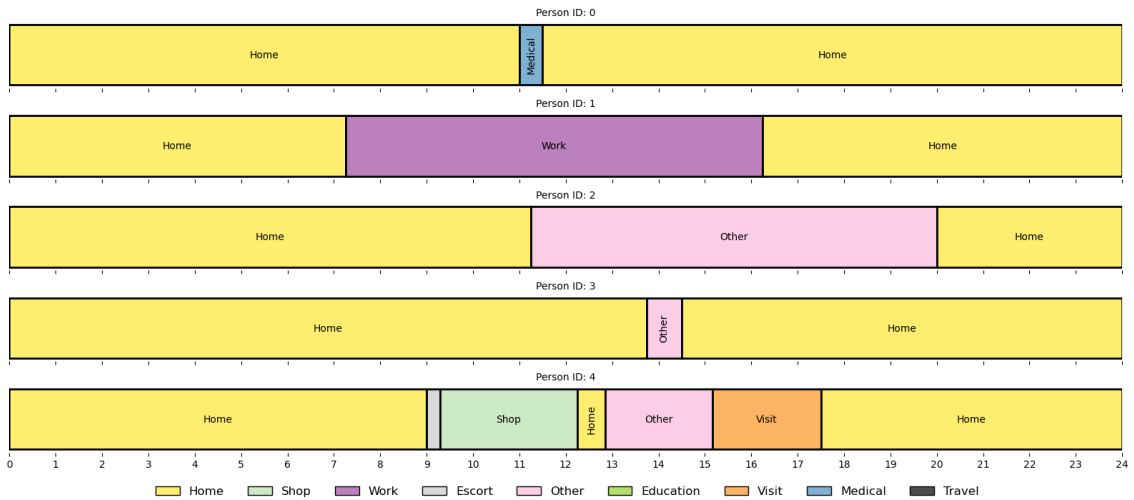


Figure 2: Example input activity-sequences

are trained on 80% of the observed data. We use the remaining 20% for validation during training. For evaluation we generate new synthetic samples of the same size as the observed sample. As this is a generative model, we do not provide a test data set. Instead, we define a framework to evaluate the synthetic samples.

Evaluation

We evaluate synthetic samples of sequences. Where each sequence is a 24-hour chain of activities. Evaluating the quality of outputs from generative models is application specific. We therefore contribute a transport domain framework. Our framework evaluates both the sample **correctness** (Table 1) and **creativity** (Table 2) of the models by comparing their outputs to the observed data.

Correctness

We require output sequences to be individually realistic and our synthetic sample of sequences to be representative in aggregate. We define this as *correctness*. Pougala et al. (2023) verify correctness

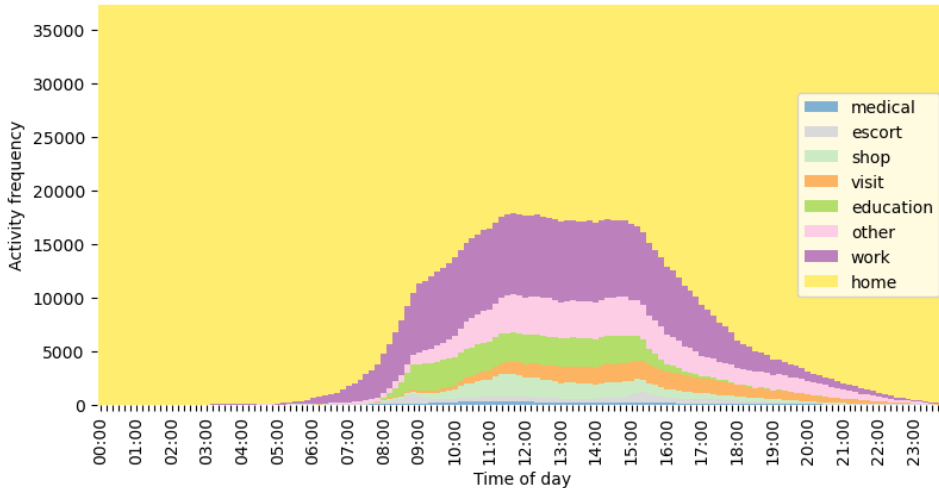


Figure 3: UK NTS binned activity frequency

by visually interrogating sample sequences and by comparing the frequency of aggregate activity participation by time-bin. Figure 3 shows activity frequency of the observed data. VALFRAM by Drchal et al. (2016) provides a more detailed framework of metrics, focused on measuring the difference between various participation and time distributions. We extend on these by refining metrics and adding new distributions.

We evaluate correctness using the following **themes**; (i) structural, (ii) frequency, (iii) participation, (iv) transitions, and (v) timing. These themes are composed of distributions, such as activity start times, activity durations, etc. Each distribution has descriptive and distance metrics for evaluation. Where distance is a measure of difference between synthetic and observed distributions. Distributions are generally dis-aggregated, for example, start times of *work* versus start times of *shop* activities.

We use enumerated activities to distinguish between re-occurrences of the same activity within a sequence. For example, to distinguish between home at the start (*home0*) of day and home again later in the day (*home1*).

We propose EMD (Wasserstein or Earth-Mover Distance) as a general purpose distance metric. With the exception of activity frequencies and participation probabilities where we use MAPE (mean absolute percentage error), clamped below 100%. MAPE highlights poor participation in uncommon activities. In all cases low distances are better.

We average dis-aggregated features and themes as required to make high level evaluations. We measure time-based metrics in days, such that a unit distance for a time feature is equivalent to a missing activity participation or transition.

Creativity

Trivially, repeating schedules from the observed sample would have excellent correctness. However, at large sample sizes, we would suffer from *lumpiness*, i.e. the repetition of sequences. It is desirable for our model to be creative - to synthesise new sequences. We measure creativity as the ability of the model to generate both unique sequences (which we call diversity) and novel sequences (sequences unseen in the training data). We define homogeneity and conservatism as the opposite of diversity and novelty, for use as distance metrics.

Discrete Sequence Model

The **discrete model** uses a discretised representation of time as per Figure 4. We use 10 minute intervals, such that each sequence is composed of 144 steps. The representation can be usefully considered as a one-dimensional image where each pixel represents participation for some fixed time in some activity denoted by the pixel colour.

We use an embedding layer to turn the one-dimensional sequence into a two-dimensional vector

Table 1: Correctness evaluation summary

Feature	Dis-aggregation	Descriptive metric	Distance metric*
Structural			
Starts & ends at home	start/end	probability	EMD
Sequence duration	none	av. time (days)	EMD
Sequence length	none	av. length	EMD
Frequency			
Binned participation	activity	probability	MAPE
Activity Participation (probabilities)			
Single participation prob.	enumerated activity	probability	MAPE
Pair participation prob.	activity pairs	probability	MAPE
Activity Participation (rates)			
Single participation rate	enumerated activity	av. rate	EMD
Pair participation rate	activity pairs	av. rate	EMD
Activity Transitions			
Bi-gram rate	activity pairs	av. rate	EMD
Tri-gram rate	activity triples	av. rate	EMD
Activity Timing			
Start times	enumerated activity	av. time (days)	EMD
End times	enumerated activity	av. time (days)	EMD
Durations	enumerated activity	av. time (days)	EMD
Start-duration (joint)	activity	av. time (days)	EMD

*EMD: Wasserstein Distance, MAPE: Mean Absolute Percentage Error

Table 2: Creativity evaluation summary

Feature	Description	Descriptive metric	Distance metric
Diversity	The probability of a sequence within the synthetic sample being unique.	probability	-
Homogeneity	The probability of a sequence within the synthetic sample not being unique.	-	probability
Novelty	The probability of a sequence not occurring in the observed sample.	probability	-
Conservatism	The probability of a sequence occurring in the observed sample.	-	probability

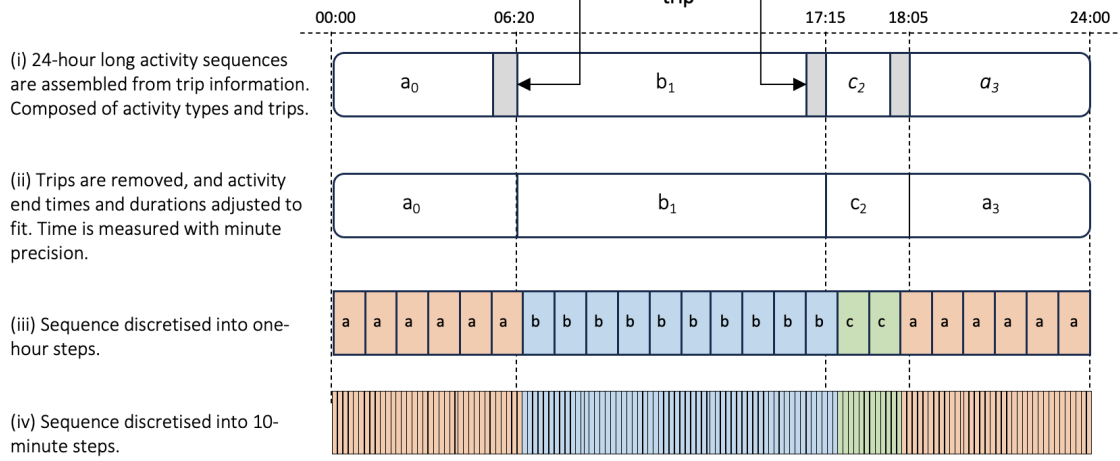


Figure 4: Discretised activity sequence representation

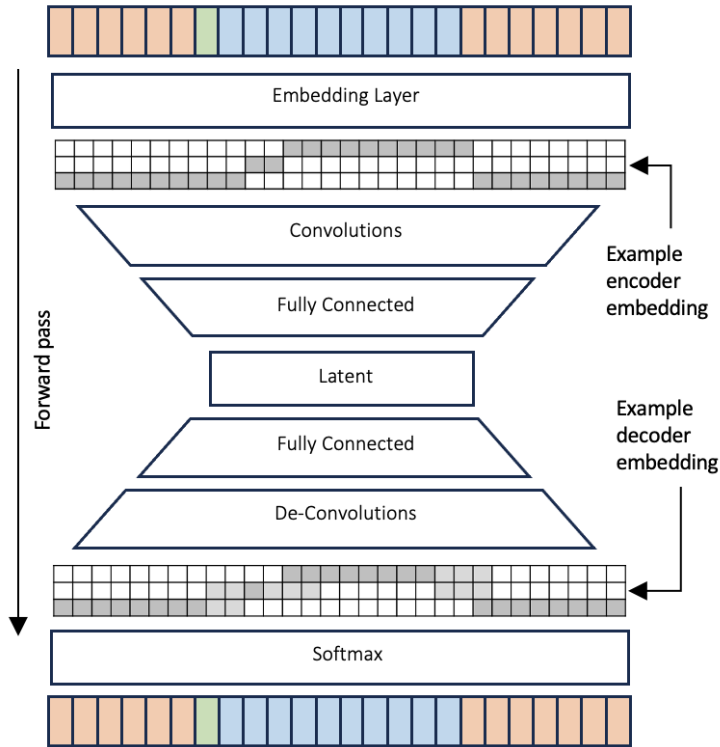


Figure 5: Discrete model VAE structure overview

representation. This is then followed by five layers of 2-dimensional convolutions (128 channels each), followed by a fully connected layer, as per Figure 5. We use a latent layer with six dimensions. The decoder mirrors the encoder, but uses a soft-max layer in place of embedding. The full architecture is available in *CAVEAT* (2023). We use cross-entropy loss, weighted by the inverse of activity total durations in the observed sample. This weighting is important for promoting the generation of infrequent activities.

Sequence Model

The **sequence model** uses a sequence representation as per Figure 6. Activities are represented with two dimensions - type and duration. Sequences are preceded by a special start-of-sequence token and followed by end-of-sequence tokens up to a maximum sequence length of 14. This clips a small number of sequences in our training data.

Figure 7 illustrates the structure of the sequence model. We use a custom embedding layer

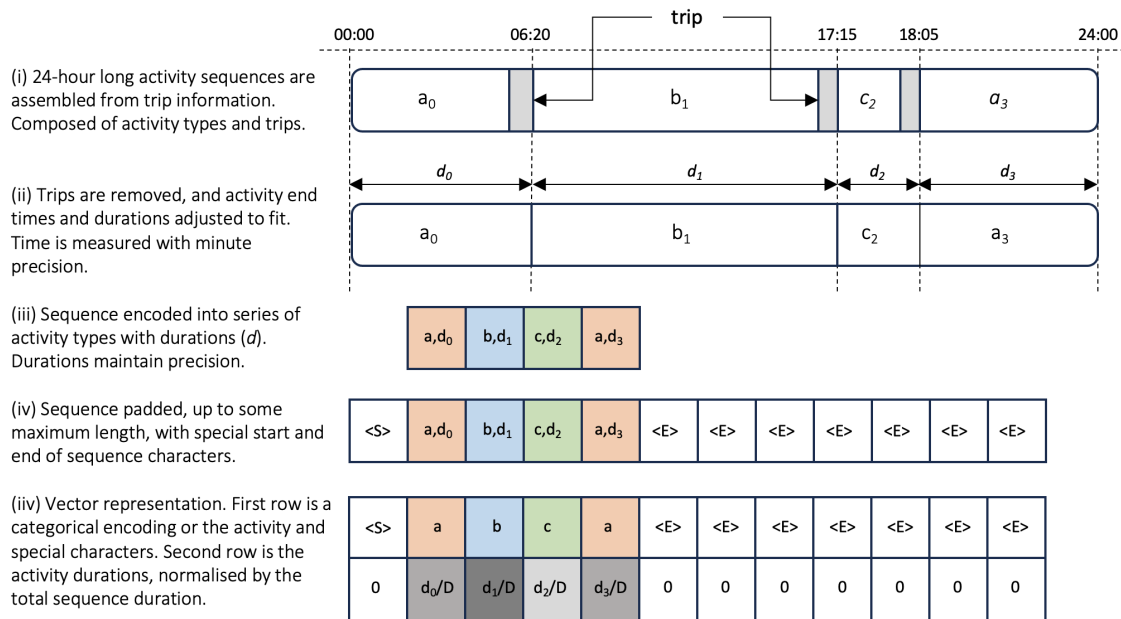


Figure 6: Sequence activity-sequence representation

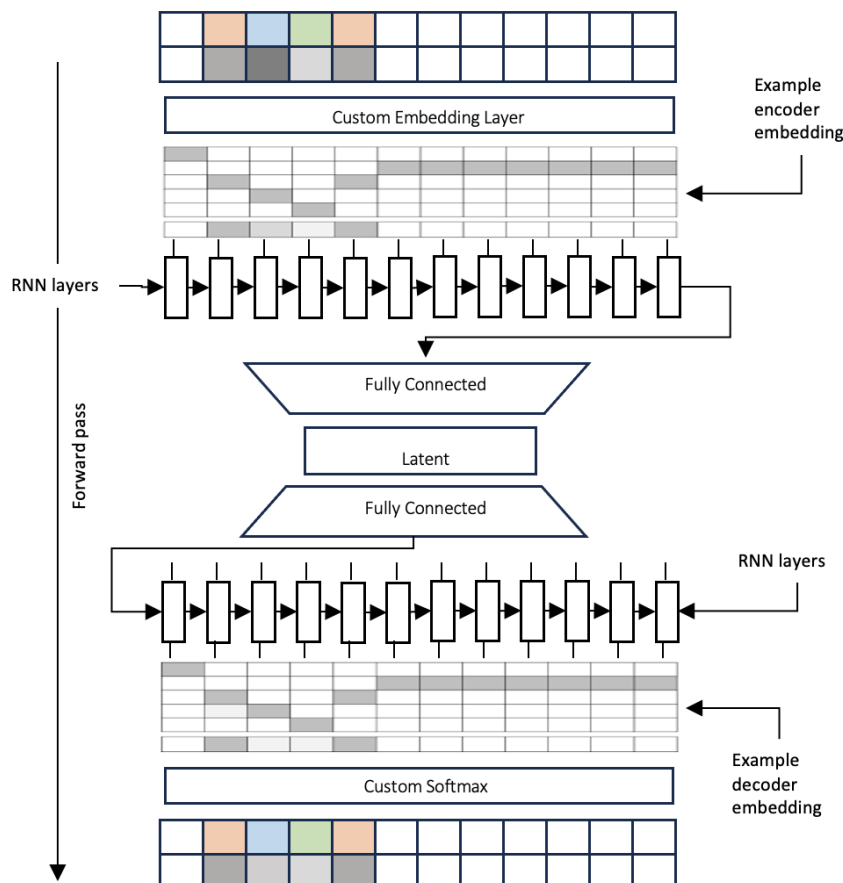


Figure 7: Sequence model VAE overview

that embeds the first dimension (activity type encoding) of the input data into a two-dimensional vector and then concatenates this back to the duration dimension. This is then passed through two stacked recurrent neural network (RNN) layers, each with 256 channels. The hidden states of the final RNN unit are then passed through a fully connected layer. We use Long Short-Term Memory (LSTM) RNN units. We use a latent layer with six dimensions.

The decoder uses a similar structure in reverse. The output from the latent layer is passed via a fully connected layer into the hidden state of the first decoding RNN unit. This first unit is always passed the start of sequence token, further steps then use the output from the previous step. Outputs are passed through a soft-max for activity prediction and sigmoid activation for duration prediction. This structure enforces a highly auto-regressive model, where each activity and duration prediction is made conditional on the previous. The full architecture is available in *CAVEAT* (2023)

We use a combination of cross-entropy loss for the activity prediction and mean squared error for the duration prediction. Cross-entropy is weighted by the inverse of activity frequencies in the observed sample. Training requires both teacher forcing (we use 50%) and masking of the useful part of the input sequence for the loss function.

3 RESULTS AND DISCUSSION

Our models have stochasticity resulting from the training process and from sampling the latent space. We therefore present all evaluation metrics as means and variances from batches of five.

Creativity

Model creativity is described in Table 3. The sequence model is more creative. Both models generate more diversity than the observed sample and generate predominantly novel sequences. But the sequence model generates all unique and all novel sequences. This is partially because of the greater precision of the sequence data representation.

Table 3: Creativity

	observed	discrete model*		sequence model*		metric
		mean	variance	mean	variance	
diversity	0.550	0.598	0.004	1.000	0.000	probability unique
novelty	-	0.923	0.000	1.000	0.000	probability unseen

*mean and variance of metrics from 5 runs of each model.

Correctness

Both models generate feasible sequences as shown by the examples in Figures 8 and 9. Table 4 summarises the full correctness evaluations of both models. The sequence model generally outperforms the discrete model (lower distances to observed) but performance is often quite similar. Variance is low, showing good model training stability.

Structure

The discrete data representation ensures correct sequence total duration. Structural descriptions in Table 5 show that the sequence models do not generate correct sequence durations. We suggest

Table 4: Correctness evaluation themes summary

	discrete model*		sequence model*		metric
	mean	variance	mean	variance	
structure	0.268	0.000	0.075	0.001	EMD
frequency	0.126	0.003	0.039	0.000	MAPE
participation prob.	0.564	0.001	0.296	0.001	MAPE
participation rate	0.019	0.000	0.007	0.000	EMD
transitions	0.015	0.000	0.006	0.000	EMD
timing	0.059	0.000	0.037	0.000	EMD

*mean and variance of correctness evaluation metrics from 5 runs of each model.

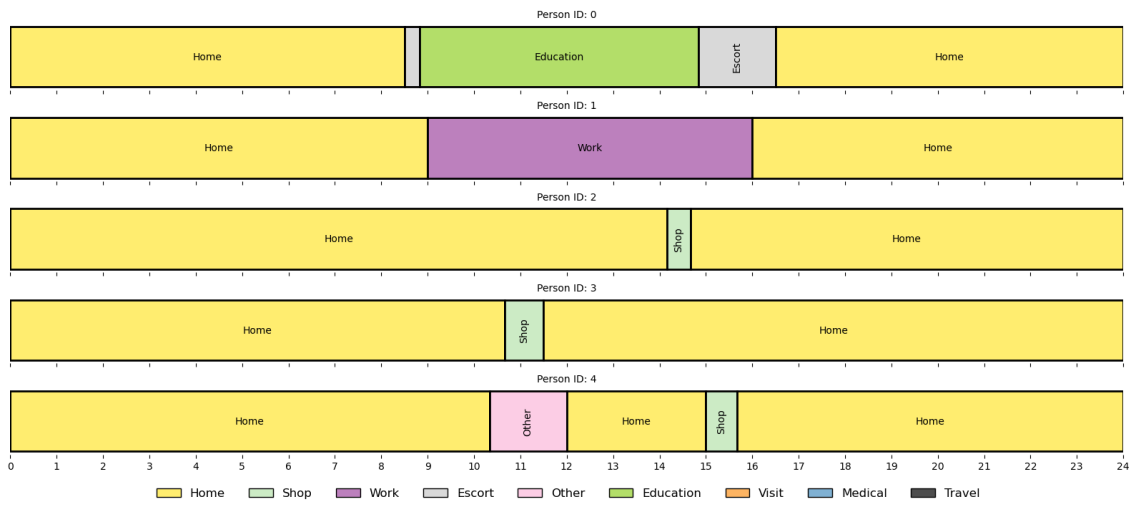


Figure 8: Discrete model - example synthetic activity sequences

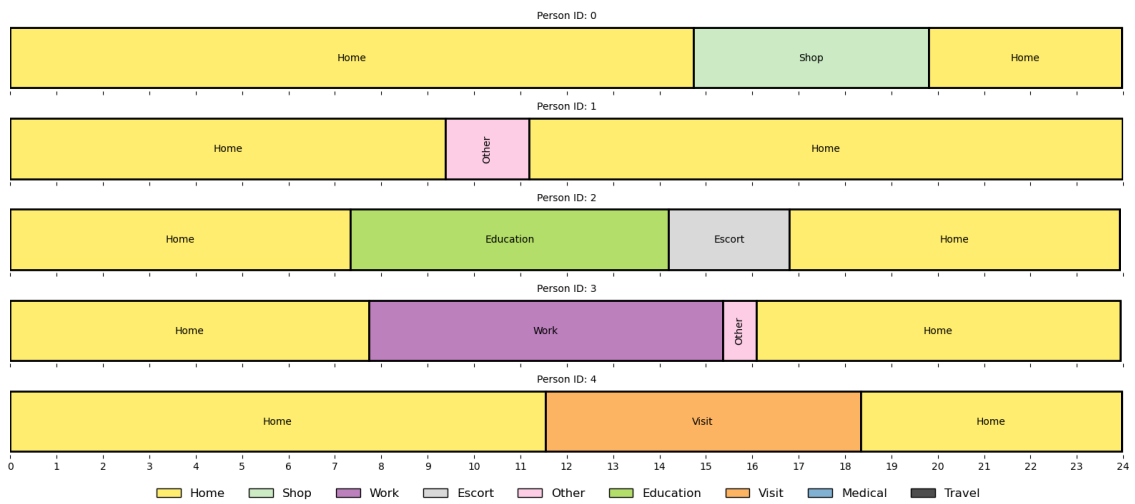


Figure 9: Sequence model - example synthetic activity sequences

Table 5: Structural

	observed	metric	discrete model*			sequence model*		
			mean	var.	EMD**	mean	var.	EMD**
av. total duration	1.000	days	1.000	0.000	0.000	0.995	0.000	0.025
first act home	1.000	prob.	1.000	0.000	0.000	1.000	0.000	0.000
last act home	1.000	prob.	1.000	0.000	0.000	0.996	0.000	0.004
av. seq. length	3.680	count	3.990	0.065	0.804	3.566	0.001	0.198

*mean and variance of structural evaluation metric from 5 runs of each model.

**mean EMD of structural evaluation distributions from 5 runs of each model.

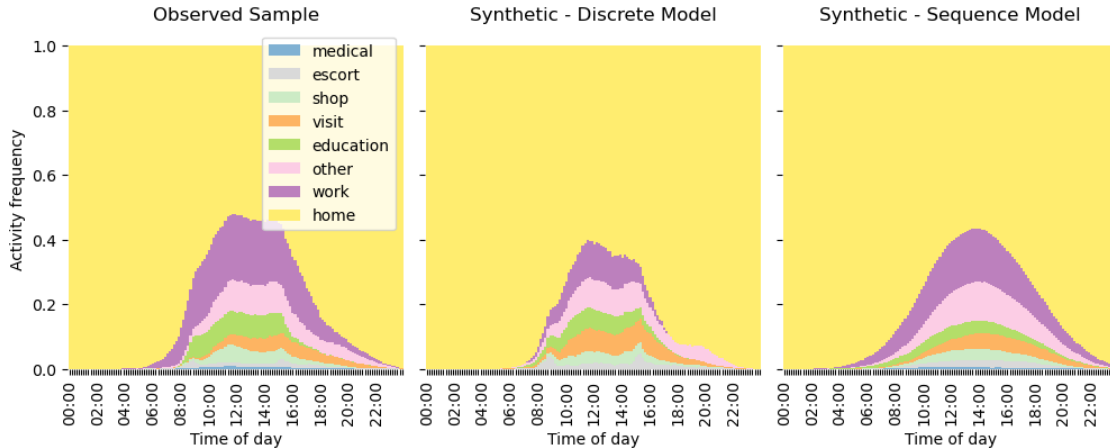


Figure 10: Model activity frequencies

these discrepancies can be corrected in post-processing. The sequence model omits *home* as a final activity around 0.4% of the time. The discrete model over-predicts the number of activities by 0.3 on average. The sequence model does better, under predicting the number of activities by 0.1 on average.

Frequency

Figure 10 compares the activity frequency of the synthetic samples with the observed. This figure can be thought of as describing the aggregate of what activities are taking place during the day. The discrete model clearly under predicts work activity frequency but does a better job following the detailed shape of the observed for all activities. The sequence model appears to smooth the distributions.

Participation

Table 6 shows participation probabilities (the probability of each activity type occurring in a sequence). The sequence model consistently outperforms the discrete model for both common and uncommon activities. The discrete model almost entirely fails to generate the uncommon medical activity. But over-generates, the also uncommon, visit and education activities. Suggesting that this is not a simple class imbalance issue.

Transitions

Table 7 shows the average transition rates of the 20 most frequent bi-grams (transitions from one activity to another). Compared to probabilities, using rates allows the consideration of multiple occurrences within a single sequence. The sequence model clearly out-performs the discrete model. The discrete model fails to generate some transitions, such as *home* \rightarrow *medical*.

Figure 11 compares the frequencies of four pattern types. Each row represents a distinct sequence, such as *home* \rightarrow *work* \rightarrow *home*. The depth of the row represents it's frequency within the sample.

Table 6: Participation probabilities

	observed	discrete model*			sequence model*		
		mean	var.	MAPE**	mean	var.	MAPE**
home	1.000	1.000	0.000	0.000	1.000	0.000	0.000
other	0.394	0.373	0.004	0.130	0.384	0.001	0.055
work	0.237	0.127	0.002	0.723	0.261	0.000	0.086
shop	0.233	0.312	0.003	0.239	0.230	0.000	0.057
escort	0.140	0.208	0.002	0.313	0.156	0.000	0.110
visit	0.109	0.224	0.001	0.503	0.137	0.001	0.191
education	0.075	0.098	0.001	0.230	0.066	0.000	0.210
medical	0.037	0.000 [†]	0.000	1.000	0.045	0.000	0.158

*mean and variance of participation probability from 5 runs of model.

**mean MAPE from 5 runs of model

[†]4 of the 5 discrete models fail to generate any medical activities.

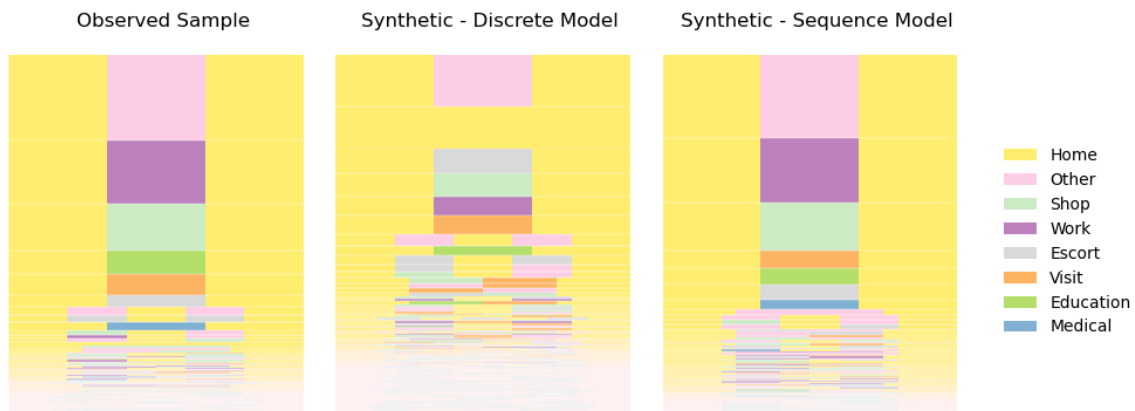


Figure 11: Models activity sequence pattern comparison

The most common tours and the density of tour lengths are all reasonably consistent, especially for the sequence model. The discrete model produces a large increase in completely home-based tours (from 1% in the observed to around 10%).

Timing

Tables 8 and 9 show average start-times and durations for the 20 most frequent enumerated activities. Performance is mixed across both models with no clearly superior model. Figure 12 provides a visual comparison of the distribution densities of activity start times and durations. We see that the sequence model mostly fails to generate the bimodal escort activity distribution and performs poorly at distributing the start times and durations of work and education activities. Suggesting that the discrete model performs better at the *timing* theme. This is reflective of the better distribution of activity frequencies in Figure 10.

Practicalities

We train all models until validation loss stabilises. We find that the models do not over-fit (either in terms of the loss function or evaluation metrics) when trained for longer. With a modern computer with GPU, the models presented take around ten minutes to train and can be sampled in negligible time to create millions of sequences.

We provide *CAVEAT* (2023) to enable extensive model and hyper-parameter exploration. We ultimately find both models perform similarly across a broad range of model structures, sizes and hyper-parameters.

We find no obvious trade-offs in evaluation. Models are able to be optimised both in terms of

Table 7: Bi-gram rates (transitions)

	observed	discrete model*			sequence model*		
		mean	var.	EMD**	mean	var.	EMD**
other → home	0.414	0.448	0.005	0.072	0.347	0.001	0.067
home → other	0.414	0.460	0.004	0.074	0.357	0.001	0.057
home → work	0.229	0.152	0.003	0.135	0.227	0.000	0.013
work → home	0.225	0.154	0.003	0.128	0.228	0.000	0.016
shop → home	0.224	0.339	0.009	0.116	0.201	0.001	0.034
home → shop	0.213	0.321	0.007	0.107	0.196	0.000	0.023
home → escort	0.178	0.279	0.005	0.115	0.133	0.001	0.060
escort → home	0.169	0.256	0.005	0.100	0.118	0.000	0.051
visit → home	0.103	0.184	0.000	0.082	0.115	0.001	0.025
home → visit	0.094	0.164	0.000	0.070	0.090	0.000	0.015
home → education	0.073	0.093	0.002	0.033	0.061	0.000	0.014
education → home	0.070	0.087	0.001	0.026	0.056	0.000	0.014
home → medical	0.035	0.000	0.000	0.035	0.041	0.000	0.007
medical → home	0.031	0.000	0.000	0.031	0.035	0.000	0.005
other → other	0.019	0.000	0.000	0.019	0.035	0.000	0.017
escort → escort	0.019	0.000	0.000	0.019	0.019	0.000	0.005
home → home	0.015	0.000	0.000	0.015	0.011	0.000	0.006
shop → shop	0.015	0.000	0.000	0.015	0.022	0.000	0.010
escort → work	0.011	0.005	0.000	0.007	0.017	0.000	0.008
other → shop	0.010	0.013	0.000	0.010	0.015	0.000	0.005

*average rate mean and variance from 5 runs of each model.

**mean EMD from 5 runs of model

Table 8: Average activity start times (days)

	observed	discrete model*			sequence model*		
		mean	var.	EMD**	mean	var.	EMD**
home0	0.000	0.000	0.000	0.000	0.000	0.000	0.000
home1	0.619	0.573	0.000	0.047	0.656	0.000	0.037
other0	0.532	0.535	0.000	0.014	0.526	0.000	0.017
work0	0.366	0.455	0.001	0.091	0.419	0.000	0.064
shop0	0.524	0.516	0.000	0.015	0.537	0.000	0.022
home2	0.700	0.689	0.000	0.019	0.749	0.002	0.051
escort0	0.484	0.502	0.000	0.023	0.525	0.001	0.048
visit0	0.552	0.537	0.000	0.023	0.559	0.000	0.016
education0	0.363	0.431	0.000	0.074	0.379	0.000	0.074
escort1	0.626	0.633	0.000	0.022	0.627	0.000	0.022
other1	0.637	0.625	0.000	0.020	0.614	0.000	0.027
home3	0.745	0.712	0.000	0.034	0.800	0.003	0.066
medical0	0.505	0.103	0.054	0.419	0.516	0.002	0.033
shop1	0.580	0.632	0.000	0.053	0.601	0.000	0.028
work1	0.557	0.530	0.000	0.037	0.576	0.000	0.024
escort2	0.631	0.677	0.000	0.054	0.644	0.000	0.026
other2	0.677	0.643	0.000	0.036	0.654	0.001	0.032
visit1	0.645	0.590	0.000	0.054	0.645	0.000	0.019
home4	0.776	0.727	0.000	0.049	0.340	0.218	0.503
escort3	0.669	0.698	0.001	0.047	0.646	0.001	0.043

*average start time mean and variance from 5 runs of each model.

**mean EMD from 5 runs of each model.

Table 9: Activity durations (days)

	observed	discrete model*			sequence model*		
		mean	var.	EMD	mean	var.	EMD
home0	0.458	0.540	0.000	0.082	0.475	0.000	0.029
home1	0.307	0.290	0.000	0.022	0.302	0.000	0.014
other0	0.079	0.072	0.000	0.014	0.094	0.000	0.017
work0	0.339	0.106	0.001	0.233	0.282	0.000	0.068
shop0	0.051	0.045	0.000	0.009	0.062	0.000	0.014
home2	0.244	0.241	0.000	0.016	0.244	0.001	0.032
escort0	0.030	0.028	0.000	0.010	0.056	0.000	0.027
visit0	0.136	0.085	0.000	0.052	0.139	0.000	0.010
education0	0.293	0.152	0.000	0.141	0.280	0.001	0.081
escort1	0.020	0.029	0.000	0.009	0.053	0.000	0.033
other1	0.060	0.056	0.000	0.007	0.106	0.001	0.047
home3	0.215	0.221	0.000	0.014	0.201	0.007	0.064
medical0	0.053	0.001	0.000	0.052	0.061	0.000	0.020
shop1	0.040	0.037	0.000	0.008	0.067	0.001	0.028
work1	0.125	0.053	0.000	0.072	0.155	0.001	0.033
escort2	0.022	0.028	0.000	0.007	0.071	0.000	0.049
other2	0.046	0.049	0.000	0.007	0.102	0.001	0.056
visit1	0.087	0.058	0.000	0.029	0.128	0.001	0.042
home4	0.166	0.213	0.000	0.048	0.083	0.023	0.159
escort3	0.022	0.033	0.000	0.016	0.079	0.001	0.058

*average duration mean and variance from 5 runs of each model.

*mean EMD from 5 runs of each model.

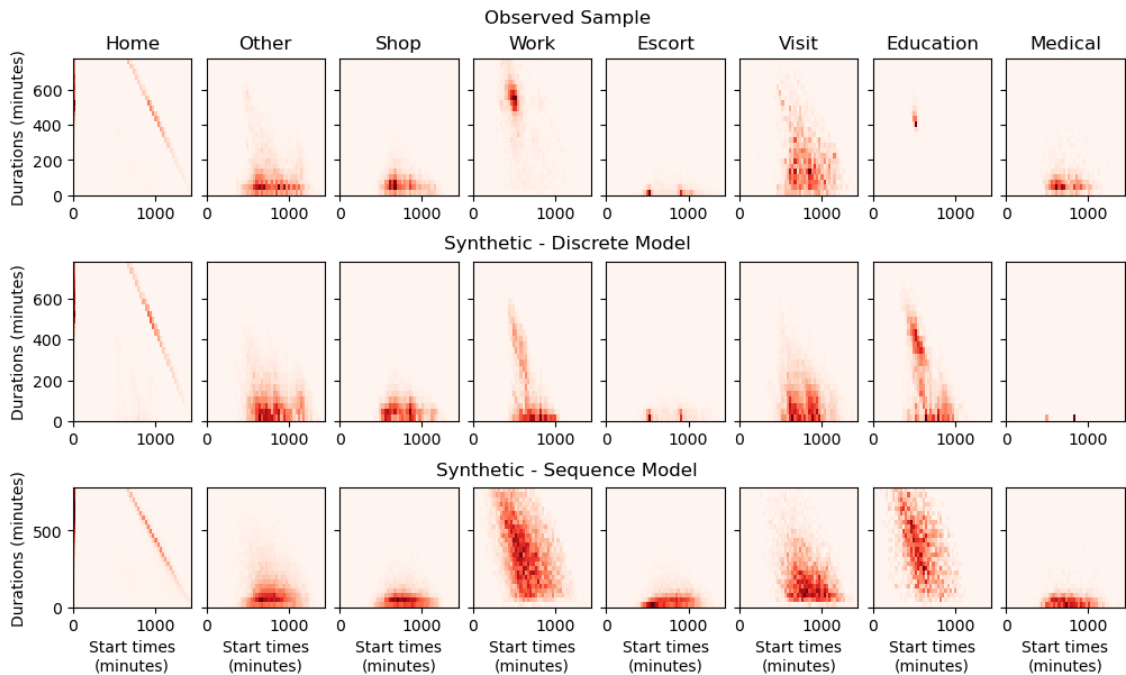


Figure 12: Activity start-Time and duration joint distribution densities

creativity and correctness.

Discussion - to discretise or not?

The discrete model breaks sequences into many parts, requiring many correct predictions to create sensible activities and sequences. This is reflected in the weaker evaluation of participation and transitions compared to the sequence model. But overall the discrete model performs reasonably well. In particular it performs well at the timing of activities.

The sequence model structure encourages auto-regressive behaviour, in which activities and durations are generated sequentially, with the previous activity and duration providing context. All information from further in the past, the future, or about the time of day, must be held in the hidden state of the RNN units.

Our intuition is that discretisation provides clear context of time of day at each inference. This allows it to perform well at timing of activities that are structured around time of day (such as escort), but not so well at activities that may be more influenced by preceding activities, such as work. In comparison, the sequence model architecture has weaker context at inference about the time of day, but excellent context at inference about the preceding activities.

Part of the success of the convolutional architecture is from being able to stack many layers, allowing the model to maintain a direct relationship between relatively high and low-level structure. This could be thought of as truly simultaneous modelling. However this stacking comes at a computational cost, which may be limiting if scaling up to longer durations or greater precision.

The sequence model performs particularly well at activity participation, with no obvious class imbalance for uncommon activities. The model representation could be considered as a more efficient encoding. It is trivially more creative by allowing greater precision than the discretised representation. However this comes at the cost of additional complexity, in particular an additional hyper-parameter controlling the weighting of activity versus duration in the reconstruction loss.

4 CONCLUSIONS

We show that our VAE-based models are able to perform well at generating large synthetic populations of activity sequences that are both realistic individually and representative in aggregate. This is achieved without the help of population-attribute information, such as the distribution of car-ownership or age.

We test two models, representing alternative ways of considering the process of activity sequencing. The discrete model, uses an image-like approach, where time is invariant and choices are made in a top-down manner. The sequence model, uses a text-like approach, where each activity is of variable length and choices are made sequentially. We find the sequence model to be better performing and expect it to scale well to larger problems. However, we find the discrete model to be simpler to implement and also perform well, particularly at activity timing. We find the models perform differently at different activity types, suggesting the decision making processes for different activities varies.

Both models are able to generate millions of sequences in seconds. They have minor stochastics and are not overly sensitive to parameterisation. This makes them a practical choice for activity modelling.

Further Research

It is likely that better encoder/decoders are possible. We are particularly interested in the use of different loss functions and model structures such as transformers. We have focused on the VAE architecture, but other generative models may perform better. A flow-based model approach could be considered to reduce model hyper-parameters.

For application, the current work is limited to where activity sequence distributions can be based on observed data. This might be acceptable for modelling very near-future demand or for applications where sequence choices are assumed exogenous. Another application is for anonymisation of activity sequence data.

To make our work more applicable, we will look to incorporate conditionality to the VAE structure, such that it can be used to infer a synthetic sequence given some variables, such as person or location attributes. We also intend to increase the complexity of the activity sequences, extending to multiple days, incorporating additional activity types, trips, and additional attributes such as location or mode choice.

REFERENCES

- Borysov, S. S., Rich, J., & Pereira, F. C. (2019, September). Scalable Population Synthesis with Deep Generative Modeling. *Transportation Research Part C: Emerging Technologies*, 106, 73–97. doi: 10.1016/j.trc.2019.07.006
- Bowman, J. L., & Ben-Akiva, M. E. (2001, January). Activity-based disaggregate travel demand model system with activity schedules. *Transportation Research Part A: Policy and Practice*, 35(1), 1–28. doi: 10.1016/S0965-8564(99)00043-9
- Caveat*. (2023, December). <https://github.com/fredshone/caveat>.
- Choi, S., Kim, J., & Yeo, H. (2021, July). TrajGAIL: Generating urban vehicle trajectories using generative adversarial imitation learning. *Transportation Research Part C: Emerging Technologies*, 128, 103091. doi: 10.1016/j.trc.2021.103091
- Drchal, J., Čertický, M., & Jakob, M. (2016). VALFRAM: Validation Framework for Activity-Based Models. *Journal of Artificial Societies and Social Simulation*, 19(3), 5.
- Galli, E., Cuéllar, L., Eidenbenz, S., Ewers, M., Mniszewski, S., & Teuscher, C. (2009, March). ActivitySim: Large-scale agent-based activity generation for infrastructure simulation. In *Proceedings of the 2009 Spring Simulation Multiconference* (pp. 1–9). San Diego, CA, USA: Society for Computer Simulation International.
- Kim, E.-J., & Bansal, P. (2023, March). A deep generative model for feasible and diverse population synthesis. *Transportation Research Part C: Emerging Technologies*, 148, 104053. doi: 10.1016/j.trc.2023.104053
- Kingma, D. P., & Welling, M. (2013, December). *Auto-Encoding Variational Bayes* (No. arXiv:1312.6114). arXiv. doi: 10.48550/arXiv.1312.6114
- Koushik, A., Manoj, M., Nezamuddin, N., & Prathosh, AP. (2023, August). Activity Schedule Modeling Using Machine Learning. *Transportation Research Record*, 2677(8), 1–23. doi: 10.1177/03611981231155426
- Manser, P., Haering, T., Hillel, T., Pougala, J., Krueger, R., & Bierlaire, M. (2022, October). Estimating flexibility preferences to resolve temporal scheduling conflicts in activity-based modelling. *Transportation*. doi: <https://doi.org/10.1007/s11116-022-10330-8>
- Population activity modeller (PAM)*. (2024). <https://github.com/arup-group/pam>.
- Pougala, J., Hillel, T., & Bierlaire, M. (2023, October). OASIS: Optimisation-based Activity Scheduling with Integrated Simultaneous choice dimensions. *Transportation Research Part C: Emerging Technologies*, 155, 104291. doi: 10.1016/j.trc.2023.104291
- W Axhausen, K., Horni, A., & Nagel, K. (2016). *The multi-agent transport simulation matsim*. Ubiquity Press.