Activity Based Modelling with Deep Conditional Generation

Fred Shone*[1] and Tim Hillel[2]

[1]PhD Student, Behaviour and Infrastructure Group (BIG), UCL, UK
[2]Assistant Professor, Behaviour and Infrastructure Group (BIG), UCL, UK

Short summary

Modelling human activity scheduling is a challenging task at the core of activity-based modelling. Existing approaches to activity scheduling are increasingly expensive and slow to develop, and can also produce unrealistically homogenous outputs, failing to model the real diversity in human behaviours. We contribute a novel methodology combining a deep generative model with conditionality, such that the model can be used in an activity modelling or transport simulation based framework. By explicitly and simultaneously modelling variation of observed activity schedules, we better represent real diversity. Our experimental results demonstrate that our approach is both cheaper and faster than existing activity scheduling solutions, whilst still providing closely tailored and high quality outputs.
Keywords: Activity Based Modelling, Choice Modelling, Deep Generative Machine Learning

1 Introduction

We propose a novel approach to modelling activity schedules that combines the qualities of both generative and conditional approaches. We define an activity schedule as a 24-hr long sequence of activities belonging to an individual, with associated start times and durations. We do not consider locations, trips or travel mode choices at this stage.
Activity scheduling is at the core of the motivation for Activity-Based Models (ABMs), driving the consideration of choice interactions across time (Rasouli & Timmermans, 2014). Activity scheduling is also a requirement for simulation-based transport approaches such as *Matsim* (2024).
Activity scheduling can be decomposed into (i) participations - *if an individual chooses to undertake an activity*, and (ii) timing - *when and for how long to undertake the chosen activities*. The participation and timing of multiple activities for an individual creates a high-dimensional object with complex joint distributions.
We consider schedules as having labels, such as socio-economic attributes of the scheduler. These labels influence the activity scheduling process. In some cases this influence is expected to be strong, for example a person *not in employment* will likely not include a *work* activity participation. But in other cases the influence of a label may be weak, for example an *employed* person will probably include a *work* activity, but they might not, perhaps due to sickness or holiday.
The prevailing approach (summarised in Table 1) in applied models is to decompose the scheduling process into series of discrete choices, applied sequentially. To ensure temporal consistency, choices are then combined with rule based scheduling algorithms. We highlight three main critiques of this approach:

1. Sequential choices presume some order of decision making that may be unrealistic.
2. The combination of discrete choices and rules is simplified such that it cannot reproduce the real diversity of observed activity schedules.
3. The complex combination of multiple interacting sub-models and rules is slow and expensive to develop, calibrate and use.

Jointly or simultaneously modelling different discrete choices can allow for more realistic modelling of joint distributions. Pougala et al. (2023) combine activity scheduling (with mode and location choice) into a simultaneous model. Their approach is consistent with existing behavioural theory, but both estimation of the parameters and simulation of schedules are computationally expensive, limiting scalability. Manser et al. (2021) manage to scale the approach to application as part of a

Table 1: Summary of Existing Applied Activity Scheduling Frameworks

Model/Framework	Activity Participation	Activity Timing
TASHA (Miller & Roorda, 2003)	Rules-based	Rules-based
ALBATROSS (Arentze & Timmermans, 2004)	Rules-based	Rules-based
FAMOS (Pendyala et al., 2005)	Nested-logit models	Hazard models
CEMPDAP (Sener et al., 2006)	Nested-logit models	Hazard models
ADAPTS (Auld & Mohammadian, 2009)	Rules-based	Rules-based
DaySim (Bradley et al., 2010)	Multinomial-logit models	Multinomial-logit models
SDS (Khan & Habib, 2023)	Markov Chain Monte Carlo	Rules-based

activity-based demand model, but limit the scope of the simultaneous approach to activity timings only.

Deep generative models have been applied for population synthesis. Borysov et al. (2019) apply a Variational Auto-Encoder (VAE) architecture for population synthesis. They find their approach able to outperform conventional methodologies in high dimensional cases. Kim & Bansal (2023) add to this work, also testing a Generative Adversarial Network. They formalise a feasibility-diversity trade-off. Where high feasibility is the avoidance of infeasible samples and high diversity improves the generation of missing data.

Koushik et al. (2023) use a discriminative model to generate activity schedules for given agent attributes using recurrent neural networks (RNNs). They discretise 24-hr schedules into five minute steps and consider nine different types of activity. This results in 9^{288} possible schedules, a significantly larger problem space than for population synthesis. Evaluation is primarily made through the consideration of conditioned marginal distributions, for example start times by type of activity. They find aggregate realism challenging, particularly the correct representation of infrequently observed activities.

Shone & Hillel (2024) have demonstrated a purely generative approach to activity schedule modelling using deep generative machine learning. This work uses schedule specific versions of the VAE architecture by Kingma & Welling (2013) to approximate the real distribution of schedules. This distribution is then used to generate new schedules, allowing application for data anonymisation and realistic up-sampling of samples.

A purely generative approach does not allow for *conditional* generation, where conditionality is typically used to provide some realistic distribution of schedules *and* labels, such an agents socio-economic attributes. Conditionality is required to model the response of activity scheduling to new scenarios, such as new household locations, increased working from home, or reduced car ownership. Conditionality is also often required for down-stream models, such as for activity location or mode choices. Increasingly, consideration of equity of outcomes across different groups of people is also desired, in which case maintaining the distribution of different scheduling choices across different agent attributes is required.

1. We present an approach using deep generation with a novel joint generative architecture and sampling that is able to combine the advantages of generative modelling within a standard discriminative framework.
2. We show using a comprehensive evaluation framework that our model generates diverse and realistic distributions of activity schedules, while maintaining key distributions with agent labels.
3. We compare our approach to a purely discriminative model and a purely generative model.

We identify three primary benefits of our approach for application; (i) simplicity and speed, (ii) realistic diversity of outputs, and (iii) the potential for more realistic interaction of choice components. We make all experiments reproducible by publishing the open source software Caveat[1].

[1]https://github.com/big-ucl/caveat

Table 2: Summary of sample density estimation and sample quality evaluation

Distribution	Segmentation	Descriptive metric	Distance metric
Density Estimation			
Aggregate			
Participation by time bin	activity	av .probability	EMD
Activity Participation			
Sequence length	-	av. length	EMD
Single participation rate	enum. activity	av. rate	EMD
Pair participation rate	activity pairs	av. rate	EMD
Activity Transitions			
Bi-gram transition rate	activity pairs	av. rate	EMD
Tri-gram transition rate	activity triples	av. rate	EMD
Activity Timing			
Start times	enum. activity	av. time (days)	EMD
End times	enum. activity	av. time (days)	EMD
Durations	enum. activity	av. time (days)	EMD
Start-duration (joint)	activity	av. time (days)	EMD
Sample Quality			
Starts at home	-	probability	EMD
Ends at home	-	probability	EMD
Sequence duration	-	av. time (days)	EMD

2 Methodology

Formal Problem Definition

We denote the observed set of individuals, which we call the *real sample*, as $I = \{1, 2, ..., N_i\}$. Each individual $i \in I$ is associated with an activity schedule x_i and also a set of labels y_i. We define an activity schedule x_i as an ordered sequence of activity types a_{in} with associated durations d_{in}:

$$x_i = [(a_{i1}, d_{i1}), (a_{i2}, d_{i2}), ..., (a_{iN}, d_{iN})] \tag{1}$$

Where n indexes the position in the schedule. The number of activities in each schedule; N may vary, but the total duration should equal the time period T, such that:

$$\sum_{n=1}^{n=N} d_{in} = T. \tag{2}$$

In this work we use a time period T of 24 hours starting and ending at midnight. We aim to estimate the distribution of schedules conditional on labels; $P(\mathbf{X}|\mathbf{Y})$, such that new schedules can be generated by drawing from this distribution.

Evaluation

We refer to schedules generated by a model as a synthetic sample; $\hat{\mathbf{X}}$. In this section we consider the evaluation of the synthetic samples of activity schedules. Where size of the synthetic and real sample are the same, and have the same labels $\mathbf{Y}$.

We use the existing framework by Shone & Hillel (2024) to evaluate synthetic samples. This framework is composed of (i) density estimation, (ii) sample quality, and (iii) creativity. Density estimation is further broken down into; aggregate, participations, transitions and timing. Density estimation and sample quality evaluation metrics are summarised in Table 2.

Creativity is the evaluation of diversity and novelty of synthetic samples. We define homogeneity and conservatism as the opposite of diversity and novelty, for use as distance metrics as described in Table 3.

For *conditional density estimation* we extend on this framework to consider the *joint density estimation* of schedules and labels. We do this by partitioning the synthetic samples into sub-populations based on label categories. Density estimation is then evaluated for each sub-population separately and then combined using averaging, weighted by sub-population size.

Table 3: 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

Table 4: NTS Label Summary

Label	Categories
Gender	{male, female, unknown}
Age	{0-4, 5-10, 11-15, 16-19, 20-29, 30-3,9 40-49, 50-69, 70+}
Car Access	{yes, no, unknown}
Work Status	{employed, education, unemployed}
Income (household)	{highest, high, medium, low, lowest}

Data

For the real sample we extract 37306 schedules from the 2022 UK National Travel Survey (NTS) trip table. We convert this trip data into 24hr-activity schedules using PAM Shone et al. (2024). We simplify the real schedules by removing trips (we maintain the original trip end times) and mapping the activity types to the set {home, work, education, medical, escort, other, visit, shop}. Activity start times, end times, and durations, have a level of precision of one minute.
Corresponding labels for the real sample are extracted from NTS household and individual data tables. We extract the following labels; {gender, age, car access, work status, and household income}. The choice of labels are designed to reflect likely requirements of a transport demand modelling framework. Label categories are summarised in Table 4.

Experiment Design

We test a novel *conditional generative* architecture, which we call **JVAE**. We compare this to a baseline (i) discriminative (*non-generative*) model by Koushik et al. (2023), and (ii) *generative* model by Shone & Hillel (2024). Models design and capability is over-viewed in Table 5.
Our experiments are intended to demonstrate the suitability of the JVAE approach for application in ABM or simulation frameworks, and to allow comparison with (i) a classic non-generative discriminative approach, and (ii) a purely generative approach.

Baseline Discriminative Model

We use a model based on Koushik et al. (2023) as a baseline *discriminative* model. This model uses a discrete schedule encoding and a Recurrent Neural Network architecture (RNN), hence we

Table 5: Experiments Overview

Model Name	Generative	Conditional	Schedule Encoding
Discrete RNN (Baseline)	no	yes	discrete
Continuous RNN VAE	yes	no	continuous
Continuous RNN JVAE	yes	yes	continuous

refer to this model as the **Dicrete RNN**. The model effectively learns the most likely schedule for each observed combination of label categories. This means that there is no variety between schedules with the same labels. Hence, we consider this as a non-generative approach.

Baseline Generative Model

We use a generative model by Shone & Hillel (2024) as a baseline *generative* model. This model is a Variational Auto-Encoder (VAE) and uses a continuous schedule encoding and RNN architecture, hence we refer to this model as the **Continuous RNN VAE** or just **VAE**. The VAE architecture learns a mapping between a known latent distribution and an approximation of the real sample of activity schedules. After training, the model can be used to generate new schedules by sampling from the latent distribution and mapping this to new samples of schedules.

Joint Variational Auto Encoder

In order to provide a generative model with conditional capability we train a VAE to jointly generate paired schedules and labels. The generated schedules can then be sampled based on their associated generated labels to find the conditional probability as per Bayes:

$$P(X|Y) = \frac{P(X,Y)}{P(Y)} \tag{3}$$

By repeatedly generating and sampling from this joint distribution we are able to match a target distribution of labels as required for modelling.

The joint model uses the same continuous schedule encoding and schedule encoder and decoder architectures as the Continuous RNN VAE for generating schedules. It adds a labels encoder and decoder block. Labels and schedule encoder block output vectors are combined by addition before passing into the latent layer. This architecture is summarised in Figure 1. We refer to this model as the **Continuous RNN JointVAE** or **JVAE**.

The schedule encoder block uses a learnt embedding layer for activity types, these are then concatenated back onto activity durations before being passed into the RNN block as inputs. The RNN block uses 5 layers of LSTM units of size 256. Lastly the final RNN hidden states are passed into a feed forward layer (Linear layer with LeakyRELU activation) also of size 256.

The schedule decoder block essentially mirrors the encoder. The latent input is passed through a feed-forwards layer and then the hidden state of the RNN block. Each RNN unit is passed the output of the previous unit as input. RNN outputs are divided into activity type weights which use a soft-max activation and duration weights which uses a sigmoid activation. Activity type inference uses arg-max sampling.

The labels *encoder* block first individually encodes each categorical label using learnt embedding layers. The resulting embedding vectors are combined through addition then passed through a single feed-forward block. The output from the labels encoder is added to the schedule encoder block by addition.

The labels *decoder* block first passes the latent vector into a single shared feed-forward block. The output is then shared between individual feed-forward blocks and soft-max layers for each of the labels. All label embeddings and layers are of size 32, except for the final feed-forward blocks and soft-max layers which are sized as required for the label categorical sizes. Label inference uses arg-max sampling.

The inclusion of labels generation requires the addition of a labels reconstruction loss. We use mean weighted cross-entropy loss, where the weighting is the inverse of label category frequency. During training label weights are normalised by batch such that the total weight is equal to the batch size. This additional label reconstruction loss is weighted by λ.

$$Loss = CrossEntropy_{activities} + MSE_{durations} + \lambda CrossEntropy_{labels} + \beta KLD$$

JVAE Sampling Approach

Synthetic schedules and labels are sampled without replacement to match a target distribution of labels. For this paper we use the real sample of labels as the target, as this is likely the closest requirement in practice. We find that generating a synthetic sample 28 times the target population size is sufficient to match 99.99% of target label combinations. This approach is feasible because the generation process takes negligible time. This massive over-generation of synthetic schedules

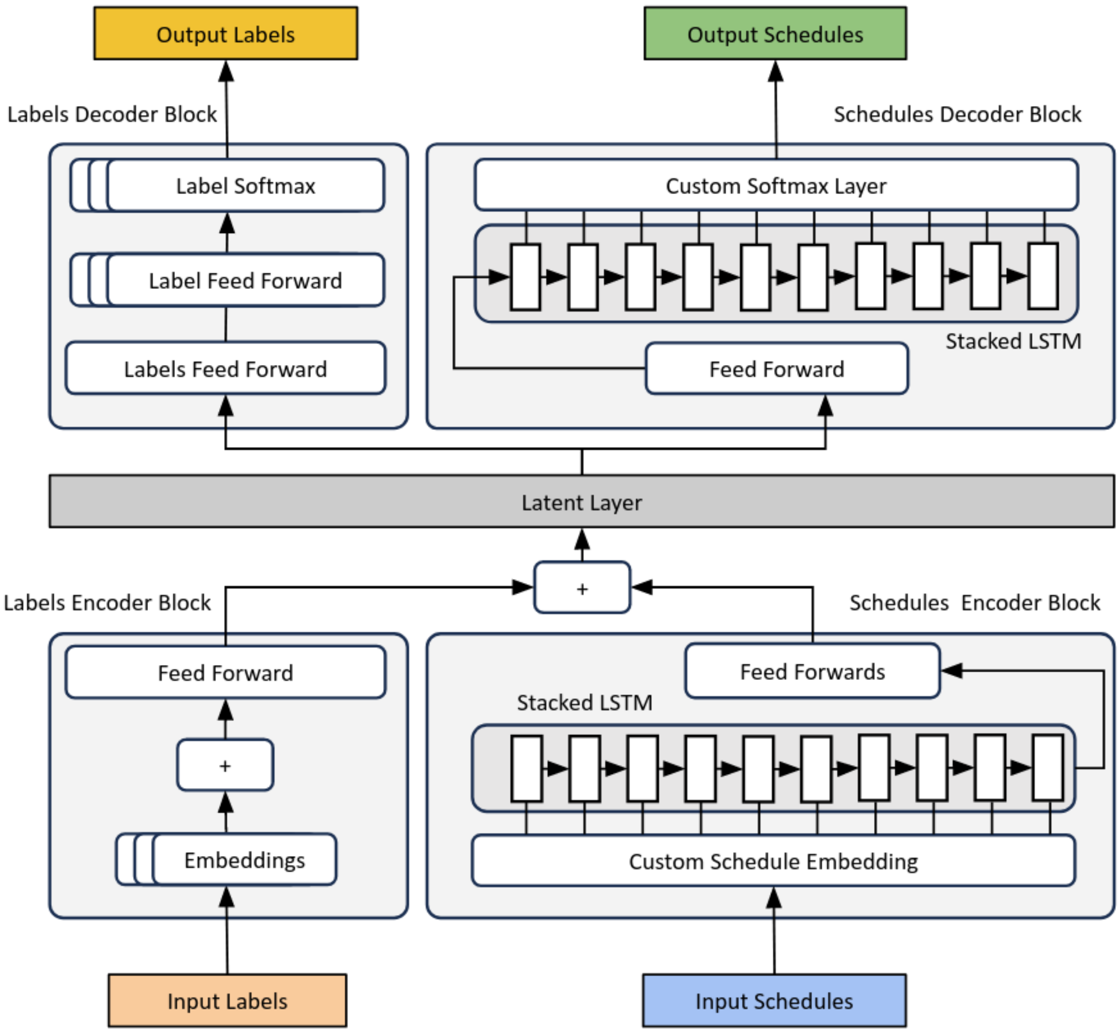

Figure 1: JVAE Architecture

Table 6: Training Overview

Model Name	Learning Rate	Batch Size	Latent Size	RNN Depth	Layer Size	β	λ
Discrete RNN (Baseline)	0.001	1024	-	2	128	-	-
Continuous RNN VAE	0.001	1024	6	5	256	0.01	-
Continuous RNN Joint VAE	0.001	1024	6	5	256	0.01	0.0001

Table 7: Evaluation Summary

	Disc. RNN		Cont. RNN VAE		Cont. RNN JVAE	
	dist*	var.	dist*	var.	dist*	var
Density Estimation						
Aggregate	0.032	0.000	**0.017**	0.000	0.020	0.000
Participations	0.688	0.000	**0.067**	0.000	0.073	0.000
Transitions	0.035	0.000	**0.007**	0.000	**0.007**	0.000
Timing	0.255	0.000	**0.068**	0.000	0.072	0.000
Joint Density Estimation						
Aggregate	0.038	0.000	**0.022**	0.000	0.023	0.000
Participations	0.594	0.000	0.084	0.000	**0.070**	0.000
Transitions	0.049	0.000	0.016	0.000	**0.011**	0.000
Timing	0.261	0.000	**0.075**	0.000	0.076	0.000
Sample Quality	**0.000**	0.000	0.053	0.000	0.050	0.001
Creativity	0.619	0.000	0.003	0.000	**0.002**	0.000

$*$ mean aggregated distances from 5 model runs

is required to ensure the target distribution of labels can be met without duplicating samples and with minimal missing samples.

Models Training and Hyper-parameters

Models are trained on 80% of the real sample data. We use the remaining 20% for validation during training. We train models until validation loss stabilises, typically for around 100 epochs We use Adam for gradient descent. Model hyper-parameters are reported in Table 6. Code and documentation for these models is available in Caveat[2].

3 Results and discussion

Table 7 presents summary evaluation metrics for density estimation, conditional density estimation, sample quality and creativity. Lower distances are better. The models have stochastics resulting from the training process and from sampling the latent space. We therefore present all evaluation metrics as means from five model runs with varying seeds.

Impact of a Generative Approach

The generative approaches by the VAE and JVAE models significantly improve evaluation of density estimations and creativity. Figures 2, 3, and 4 illustrate the quality of density estimation for activity aggregate frequency, participations and transitions, and times respectively. The VAE and JVAE are both almost perfectly creative, while the Discrete RNN is limited to variation only within the input labels. The non-generative Discrete RNN has perfect sample quality, partly due to it's discrete encoding ensuring correct total durations.
The difference between regular and joint density estimation for each model are minimal and the performance of the generative models clearly superior. This is because of the massive variance or diversity of real schedules. Individuals with similar or the same labels can have very different schedules. From the models point of view, the non-conditional or *random* variance of schedules dominates the conditional variance from the labels. Alternately we can say that schedules are

[2]https://github.com/big-ucl/caveat

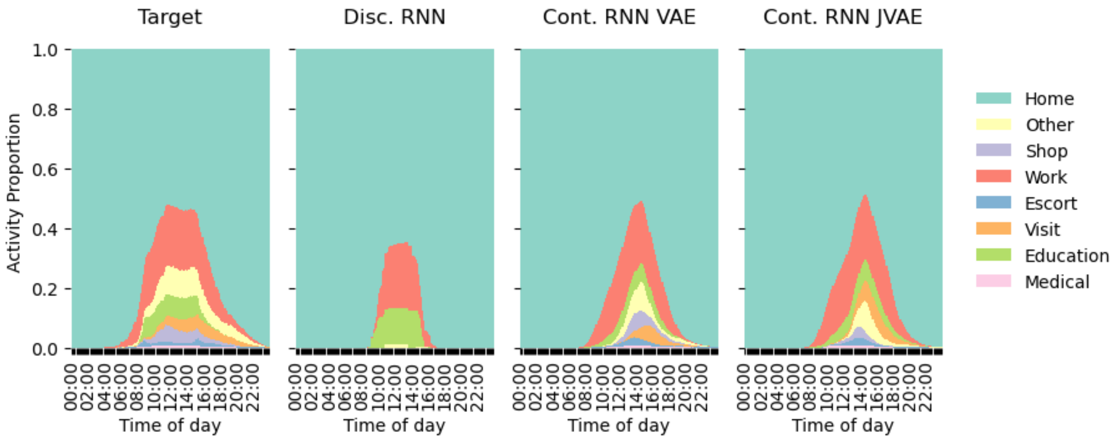

Figure 2: Aggregate Frequencies

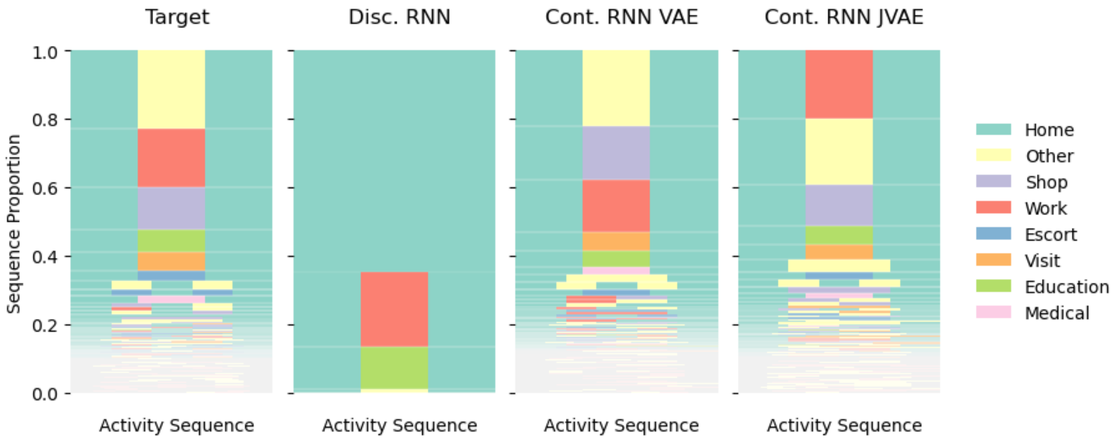

Figure 3: Activity Sequences

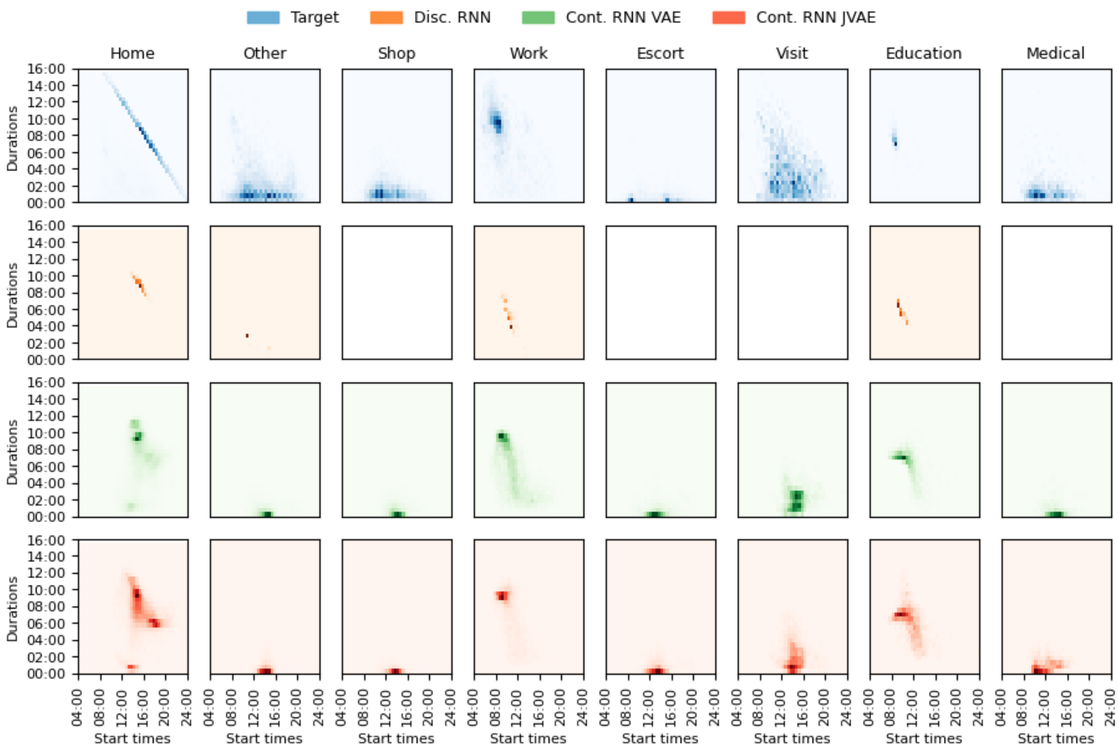

Figure 4: Activity Start Times and Durations

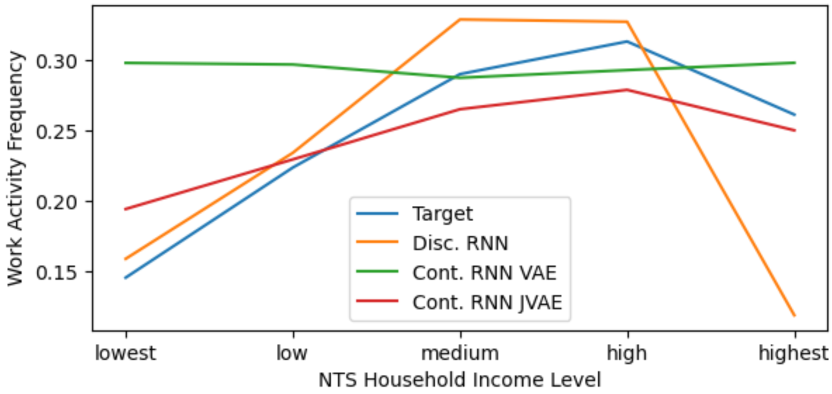

Figure 5: Work Activity Participation Rates by Household Income

only 'weakly' conditional on the labels. Adding a generative capability explicitly allows for the full variance to be modelled.

Impact of Conditional Generation

The Continuous RNN VAE has no conditional capability but still outperforms the Discrete RNN (which can be considered as 'purely' conditional) at joint density estimation. This is due to the poor performance of the Discrete RNN at regular density estimation. Our evaluation does not separate regular density estimation from the evaluation of joint density estimation.

The JVAE architecture adds a conditional capability to the VAE generative capability, this improves conditional density estimation compared to the regular VAE. However, because of the dominance of non-conditional variance, the difference between non-conditional and joint evaluations for all models is small. Similarly, the influence of adding conditionality to the generative approach, on both sample quality and creativity is minor. To get a more useful evaluation of conditional capability we therefore consider two example joint distributions in more detail; (ii) work participation conditional on income, and (ii) schedule sequences conditional on employment status.

Figure 5 shows the relationships between work activity participations and household incomes for the target NTS and synthetic samples. The target distribution shows increasing participation in work with increased income, with a drop at the highest level. The VAE has no conditional capacity and therefore does not synthesise this pattern. The JVAE is closest to the target distribution, clearly capturing the trend and similar levels of participation.

We expect *employment status* labels to have a strong influence on schedules. For example we expect people in education to have more education activities. Figure 6 shows the target NTS schedule sequences split by the *employed*, *student*, and *unemployed* sub-populations. Figure 7 show sequences for the JVAE model. We can see sensible and realistic changes in sequences within each sub-population, similar to the target NTS. For comparison the Discrete RNN synthetic sequences are shown in Figure 8. Sequences are clearly conditional on employment status, but they are dissimilar to the target NTS and have minimal diversity. The continuous RNN VAE is not shown - it has no conditional capability so produces similar sequences within each sub-population.

4 Conclusions

We show that a deep generative approach to modelling activity schedules significantly improves on an existing discriminative modelling approach. Our generative approach allows the rapid generation of diverse and high quality schedules. This is likely because of the high diversity of real activity schedules, making them unsuited for discriminative approaches. We demonstrate results suitable for synthetic schedule generation, for building predictive models, and for up-sampling or anonymising existing data. Our approach allows for rapid model development, training, evaluation and application.

We demonstrate that our novel JVAE approach provides conditional density estimation, via both (i) a quantitative evaluation framework, and (ii) the consideration of example joint distributions. Our results show that the JVAE suitable for application in activity-based modelling frameworks

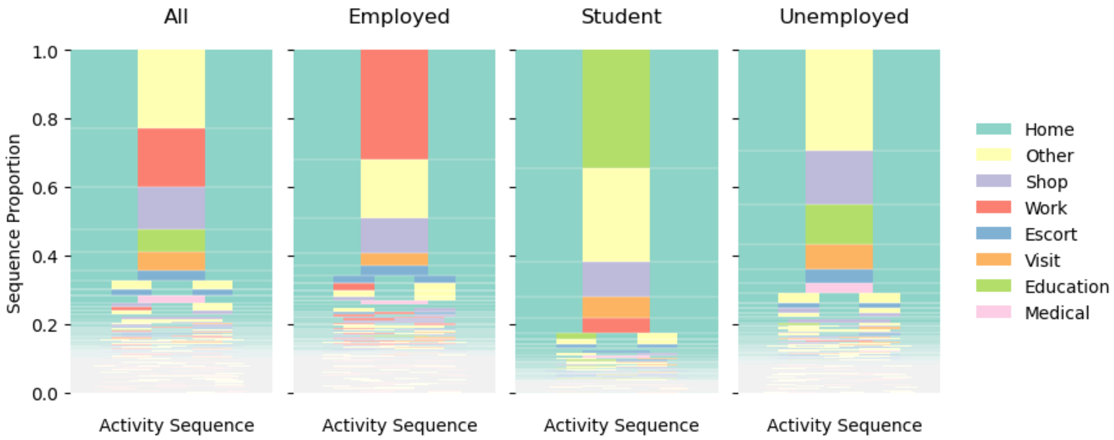

Figure 6: Target NTS Schedules Sequences by Employment Status

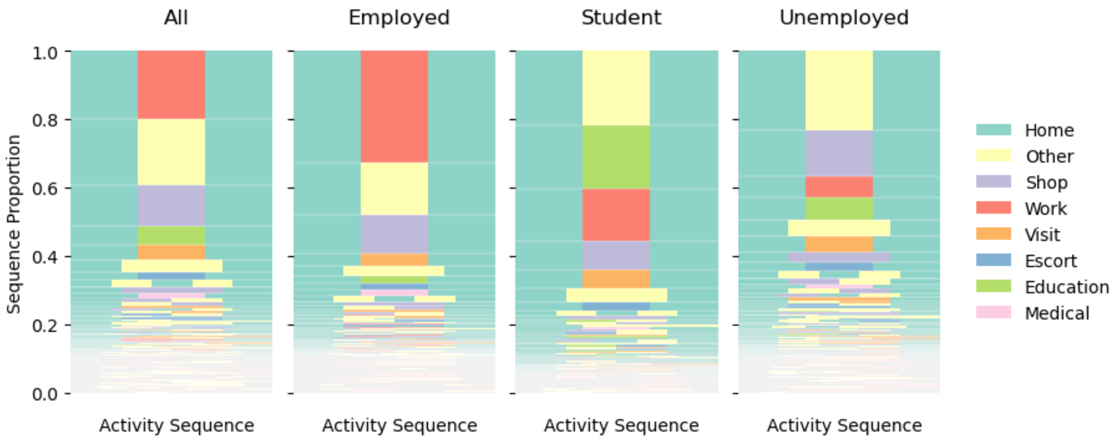

Figure 7: JVAE Synthetic Sequences by Employment Status

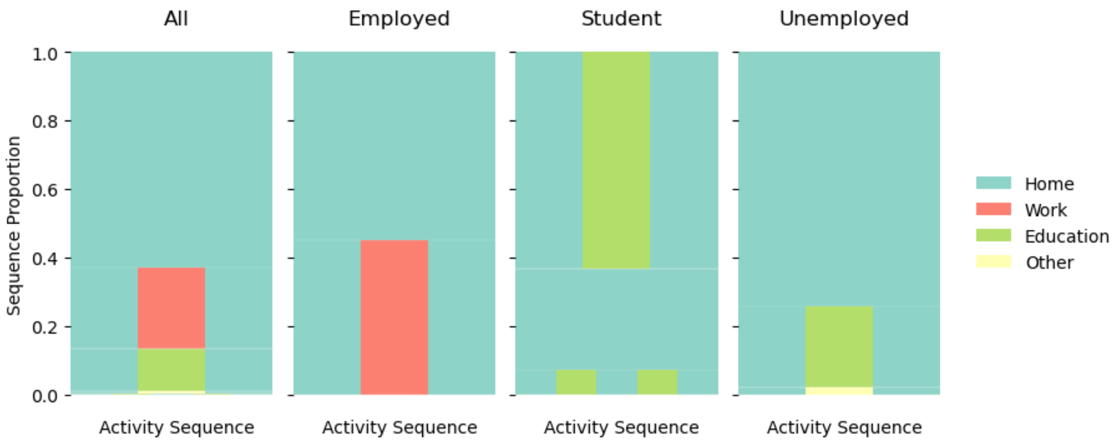

Figure 8: Discrete RNN Synthetic Sequences by Employment Status

and simulations.

Incorporating a deep generative approach takes some control away from the modeller. However, it greatly improves the capacity of the model to learn complex distributions. A key finding is the dominance of non-conditional ("random") variation in observed schedules. The deep generative approach allows explicit modelling of this variance in a single model, better capturing variation and generating realistic model outputs.

Our approach is very fast - models can be rapidly developed, trained in minutes, and massive populations generated in seconds. We share the developed models and provide a framework for extensive model and hyper-parameter exploration using Caveat[3].

Acknowledgements

This work is funded by the UK EPSRC DTP.

References

Arentze, T. A., & Timmermans, H. J. (2004). A learning-based transportation oriented simulation system. *Transportation Research Part B: Methodological*, *38*(7), 613-633. doi: https://doi.org/10.1016/j.trb.2002.10.001

Auld, J., & Mohammadian, A. (2009). Framework for the development of the agent-based dynamic activity planning and travel scheduling (adapts) model. *Transportation Letters*, *1*(3), 245–255.

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

Bradley, M., Bowman, J. L., & Griesenbeck, B. (2010). Sacsim: An applied activity-based model system with fine-level spatial and temporal resolution. *Journal of Choice Modelling*, *3*(1), 5–31.

Khan, N. A., & Habib, M. A. (2023). Microsimulation of activity generation, activity scheduling and shared travel choices within an activity-based travel demand modelling system. *Travel Behaviour and Society*, *32*, 100590. Retrieved from `https://www.sciencedirect.com/science/article/pii/S2214367X23000418` doi: https://doi.org/10.1016/j.tbs.2023.100590

Kim, E.-J., & Bansal, AP., Prateek. (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. (2021). Resolving temporal scheduling conflicts in activity-based modelling.. Retrieved from `https://api.semanticscholar.org/CorpusID:246389483`

Matsim. (2024). https://github.com/matsim-org/matsim-libs.

Miller, E. J., & Roorda, M. J. (2003). Prototype model of household activity-travel scheduling. *Transportation Research Record*, *1831*(1), 114–121.

Pendyala, R. M., Kitamura, R., Kikuchi, A., Yamamoto, T., & Fujii, S. (2005). Florida activity mobility simulator: overview and preliminary validation results. *Transportation Research Record*, *1921*(1), 123–130.

[3]https://github.com/big-ucl/caveat

Pougala, J., Hillel, T., & Bierlaire, M. (2023). Oasis: Optimisation-based activity scheduling with integrated simultaneous choice dimensions. *Transportation Research Part C: Emerging Technologies*, *155*, 104291. doi: https://doi.org/10.1016/j.trc.2023.104291

Rasouli, S., & Timmermans, H. (2014). Activity-based models of travel demand: promises, progress and prospects. *International Journal of Urban Sciences*, *18*(1), 31–60. Retrieved from `https://doi.org/10.1080/12265934.2013.835118` doi: 10.1080/12265934.2013.835118

Sener, I. N., Bhat, C. R., Copperman, R., Srinivasan, S., Guo, J. Y., Pinjari, A., & Eluru, N. (2006). *Activity-based travel-demand analysis for metropolitan areas in texas: Cemdap models, framework, software architecture and application results* (Tech. Rep.). Midwest Regional University Transportation Center.

Shone, F., Chatziioannou, T., Pickering, B., Kozlowska, K., & Fitzmaurice, M. (2024, April). PAM: Population Activity Modeller. *Journal of Open Source Software*, *9*(96), 6097. doi: 10.21105/joss.06097

Shone, F., & Hillel, T. (2024). *Activity sequence modelling with deep generative models.* https://transp-or.epfl.ch/heart/2024/abstracts/hEART$_{2}024_{p}aper_{0}145.pdf$. (18.12.2024)