Scaling complex choice models with machine learning

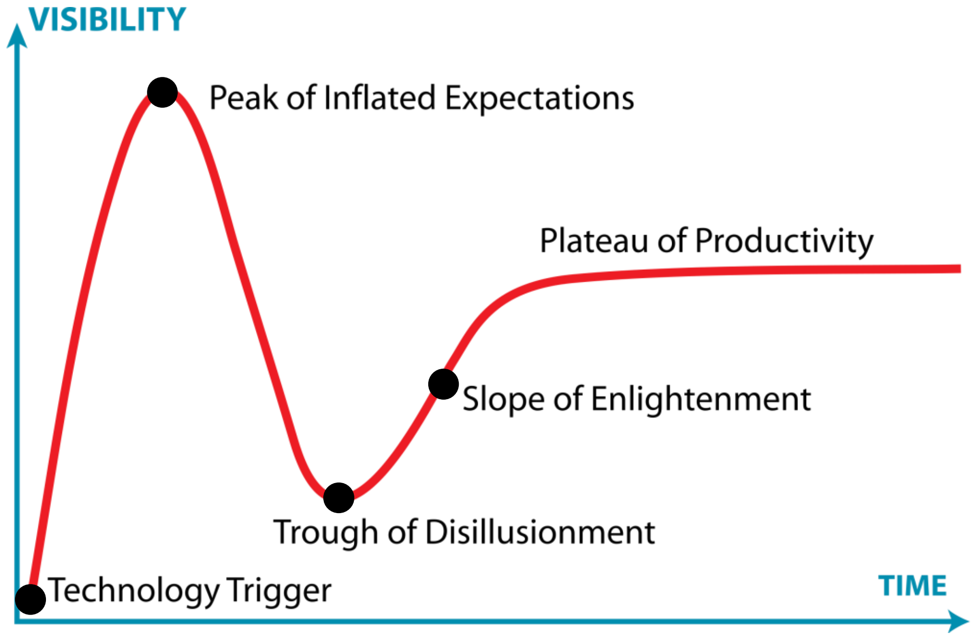
16th workshop on Discrete Choice Models, EPFL

7th June 2024

Tim Hillel
Behaviour & Infrastructure Group (BIG)
Dept. of Civil, Environmental & Geomatic Engineering
University College London
tim.hillel@ucl.ac.uk

big
behaviour & infrastructure group

A brief history of ML for discrete choice...



Gartner Hype Cycle

1. **Seminal ML classification papers**
   - NN, SVM, DT, ensembles
2. **Comparative studies**
   - MNL vs ML with zero-one classification - 99% accuracy!
3. **Establishing common methodologies**
   - *probabilistic models, robust validation*
4. **Hybridisation:**
   1. Extracting behavioural indicators from ML (Martin-Baos et al, 2023; Wang et al, 2020)
   2. Assisted specification of RUM (Ortelli et al, 2021, Hillel et al, 2019)
   3. Utility-based ML (Kim and Bansal, 2023; Han et al, 2022; Wong & Farooq, 2021; Wang et al, 2020; Sifringer et al, 2020)
   - Predominant focus is still MNL...

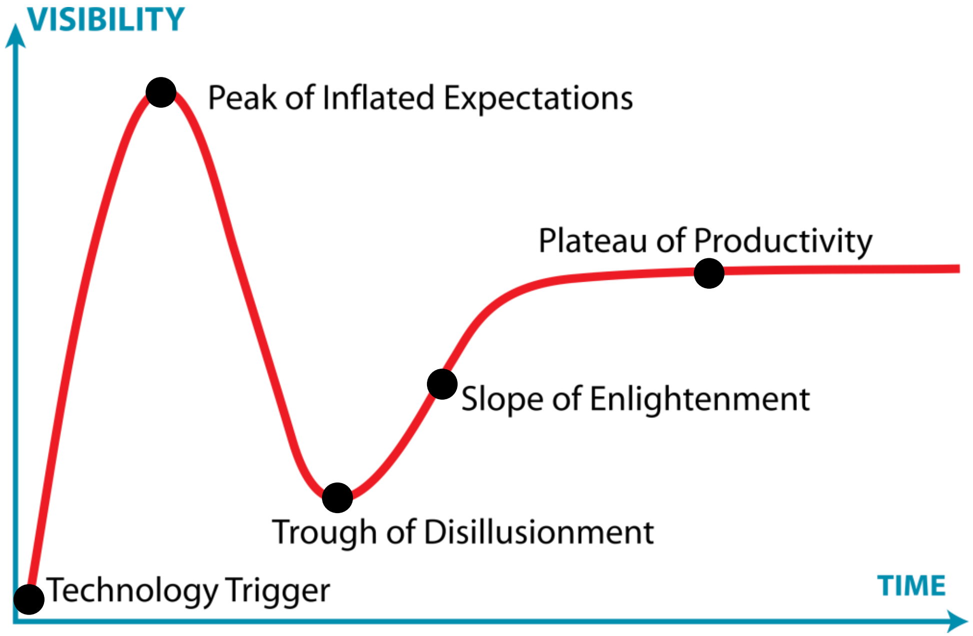
Beyond MNL – state of research

Discrete choice workshop 2024 day 1 – eight talks:

- NL/CNL – 5 talks
- Mixed Logit – 1 talk
- MILP (inc. decompositions) – 2 talks

MNL does not get us far!

A brief history of ML for discrete choice...

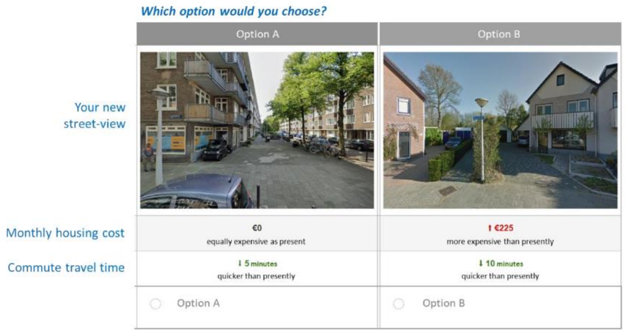


Gartner Hype Cycle

1. **Seminal ML classification papers**
   - NN, SVM, DT, ensembles
2. **Comparative studies**
   - MNL vs ML with zero-one classification - 99% accuracy!
3. **Establishing common methodologies**
   - *probabilistic models, robust validation*
4. **Hybridisation:**
   - Extracting behavioural indicators from ML (Martin-Baos et al, 2023; Wang et al, 2020)
   - Assisted specification of RUM (Ortelli et al, 2021, Hillel et al, 2019)
   - Utility-based ML (Kim and Bansal, 2023; Han et al, 2022; Wong & Farooq, 2021; Wang et al, 2020; Sifringer et al, 2020)
5. **New opportunities...**

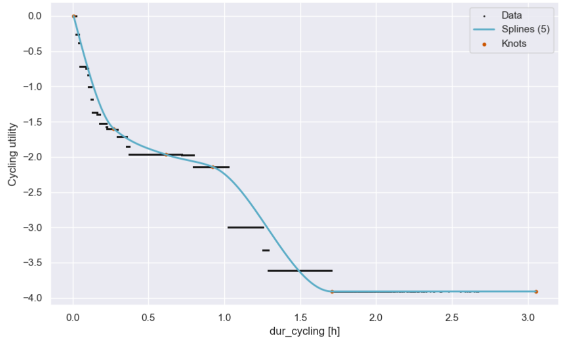
New opportunities with ML

Unstructured data (images/text)



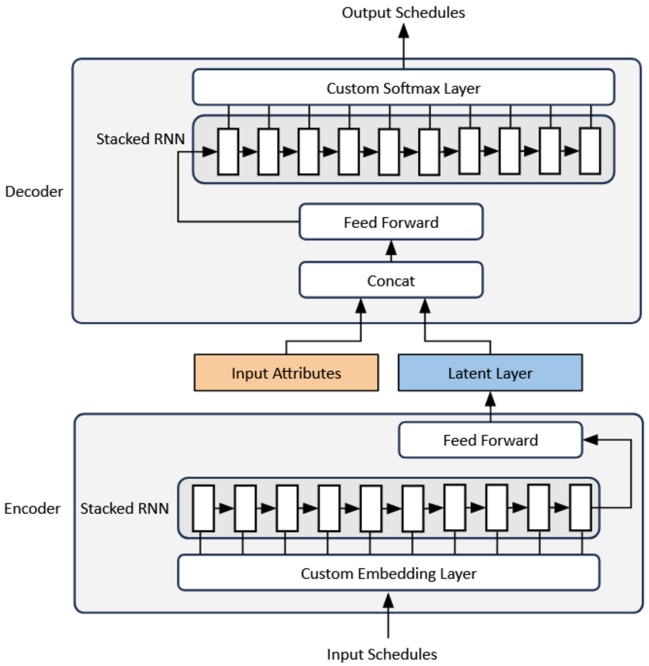
van Cranenburgh, Garrido-Valenzuela (2023)

Non-linear utilities



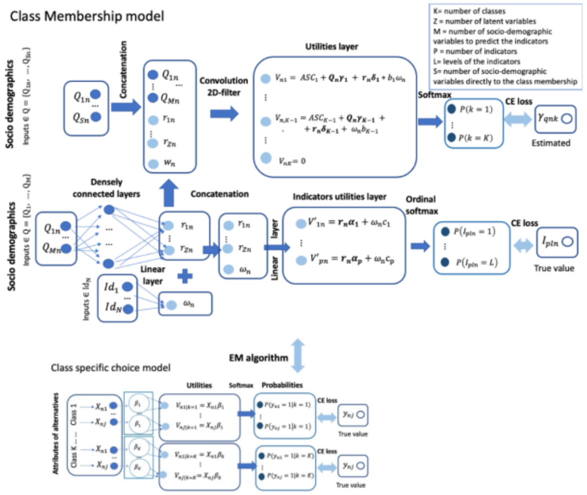
Salvadé & Hillel (2024)

Generative models for complex sequences



Shone & Hillel (2024)

Complex model specifications



Lahoz et al (2023)

Scalability?

Scaling complex choice models with RUMBoost

Nicolas Salvadé
nicolas.salvade.22@ucl.ac.uk

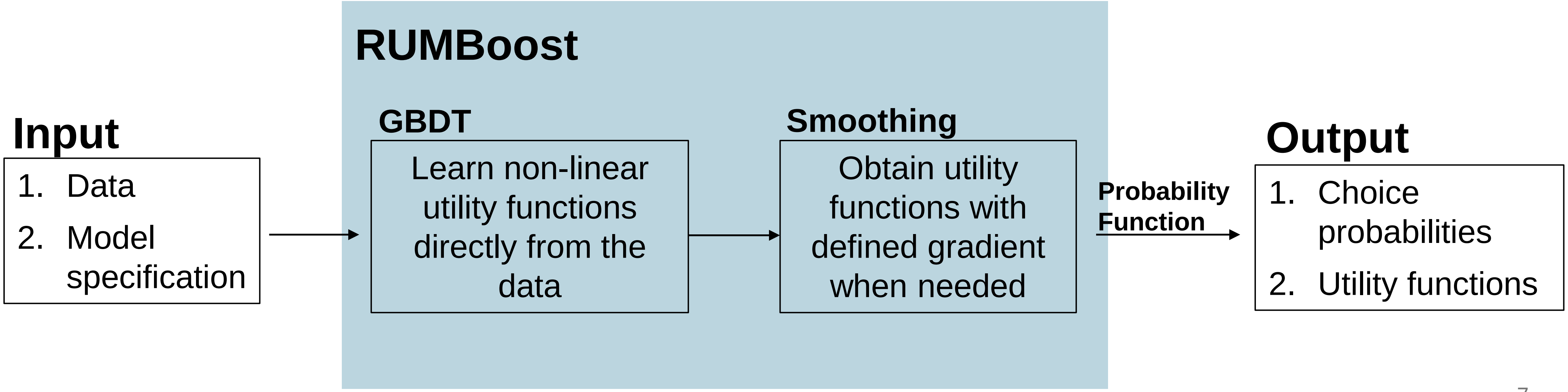
- Technical report:
  - Salvadé, Nicolas, and Tim Hillel. "RUMBoost: Gradient Boosted Random Utility Models." *arXiv preprint arXiv:2401.11954* (2024).

- Code available on github/pypi: https://github.com/NicoSlvd/rumboost

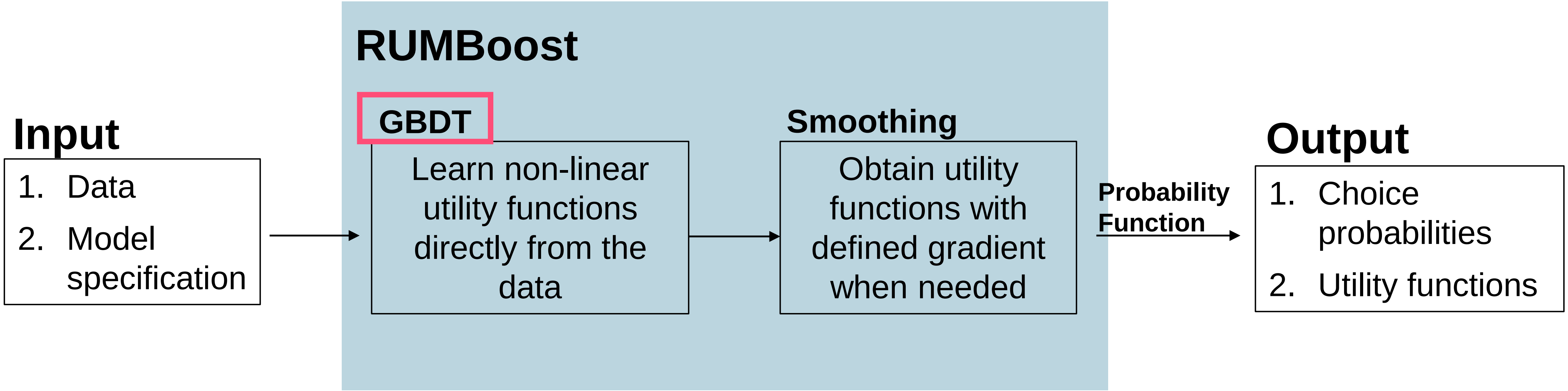
- **See forthcoming presentations at hEART and IATBR**

RUMBoost

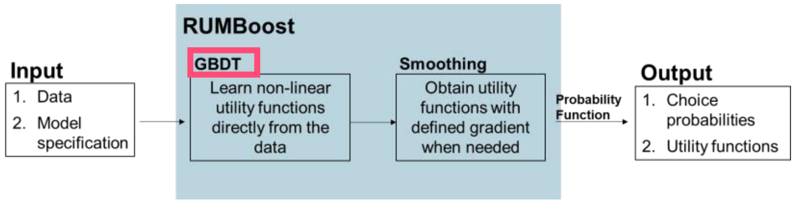
- An **intrinsically interpretable** ML model able to learn **nonlinear utility functions**
- Each parameter in RUM specification replaced with **ensemble** of regression trees
- Ensembles grown to directly optimise cost function – need defined gradient and Hessian
- Smoothing process on key variables to obtain utility functions with defined gradient



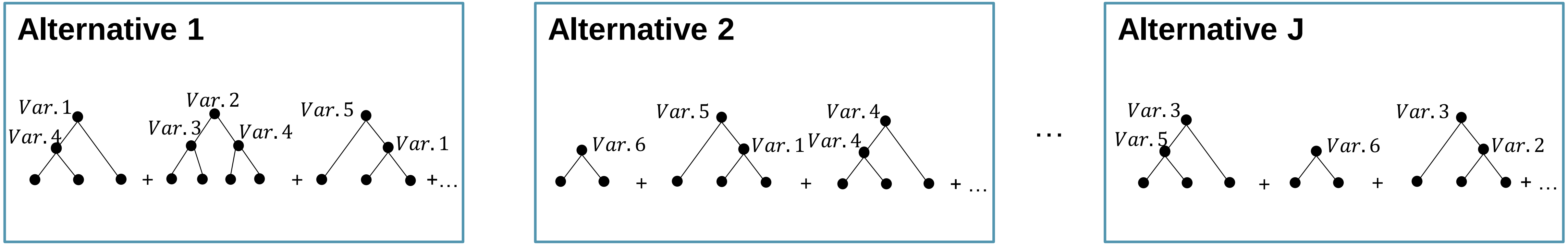
Gradient Boosting Decision Trees



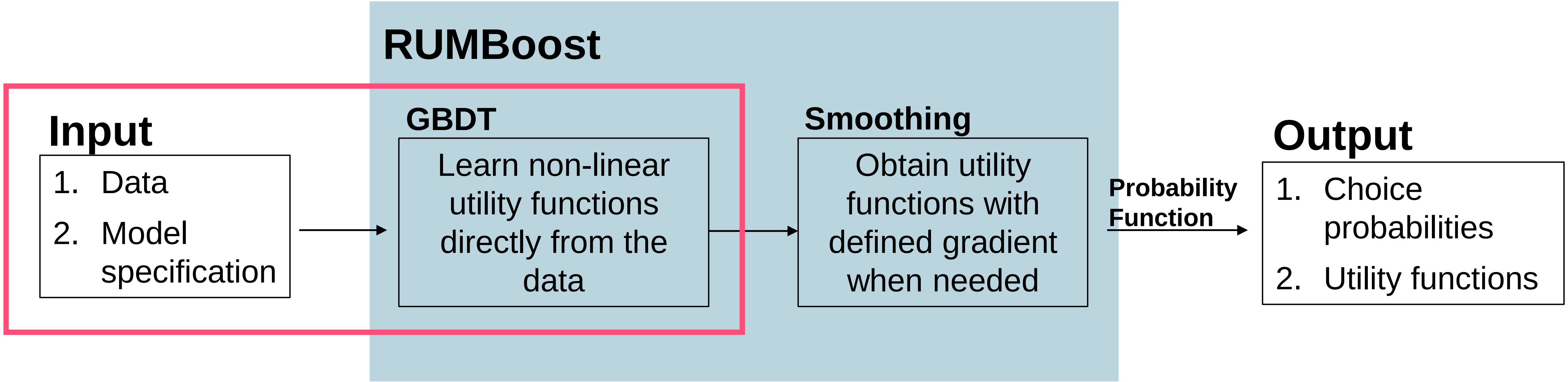
Gradient Boosting Decision Trees



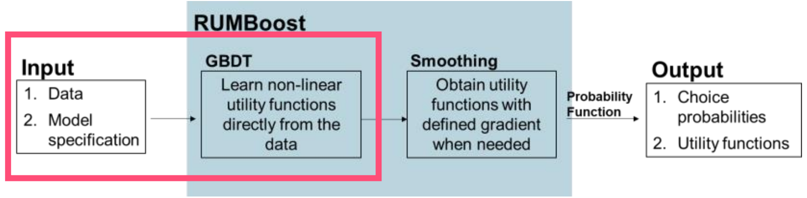
- Multiclass classification - one ensemble of **regression** trees **per alternative**
- At each iteration: add one regression tree of arbitrary depth per ensemble **to directly minimise** the cross-entropy loss (akin to maximum likelihood estimation for MNL)
- Split points optimised across **all variables**
- Leaf values are computed from the sum of gradient over the sum of hessian of all observations at each leaf



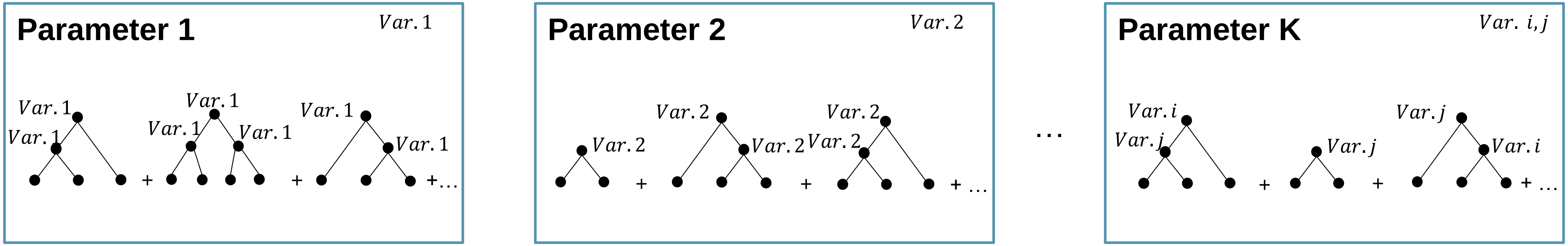
How to make GBDT interpretable?



Gradient Boosted Utility Values (GBUV)

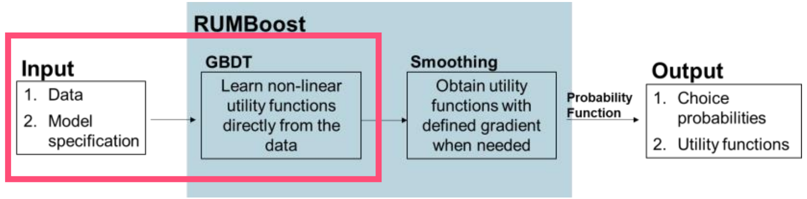


- Replicate RUM utility functions with one ensemble per **parameter** fitted on corresponding variables (constants can be extracted from normalisation of leaf values)
- At each iteration: add one regression tree of arbitrary depth per ensemble to directly minimize any desired cost function (for which gradient and Hessian can be defined)

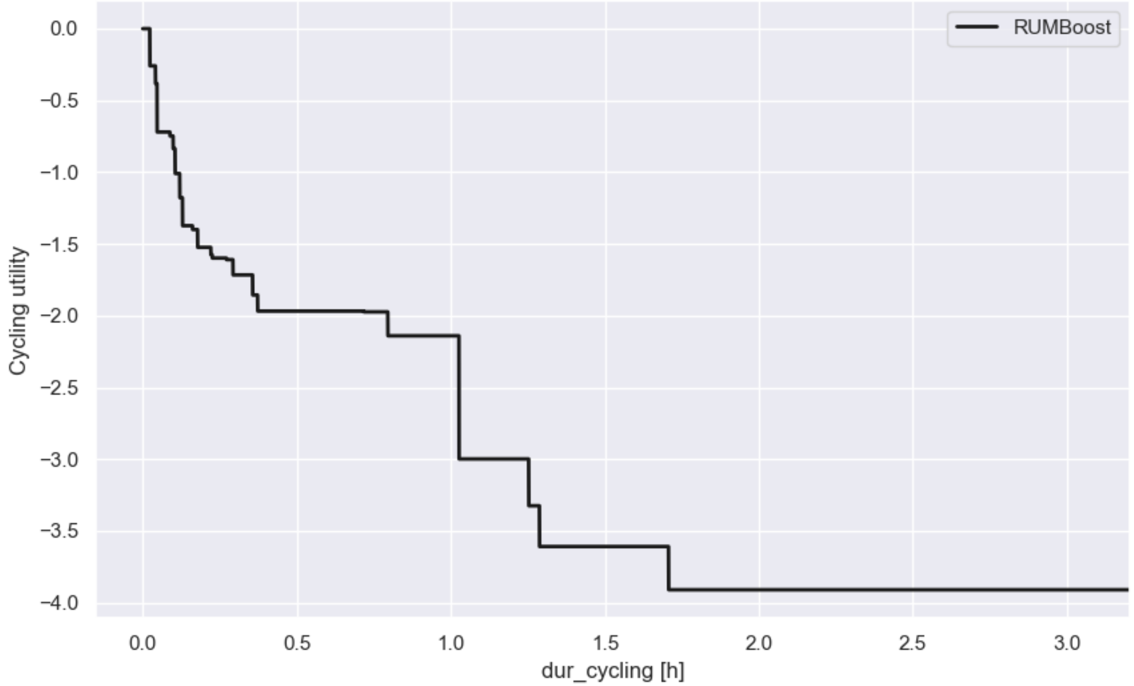


- Parameter-specific variables
- Interpretable utility values
- Monotonicity can be imposed

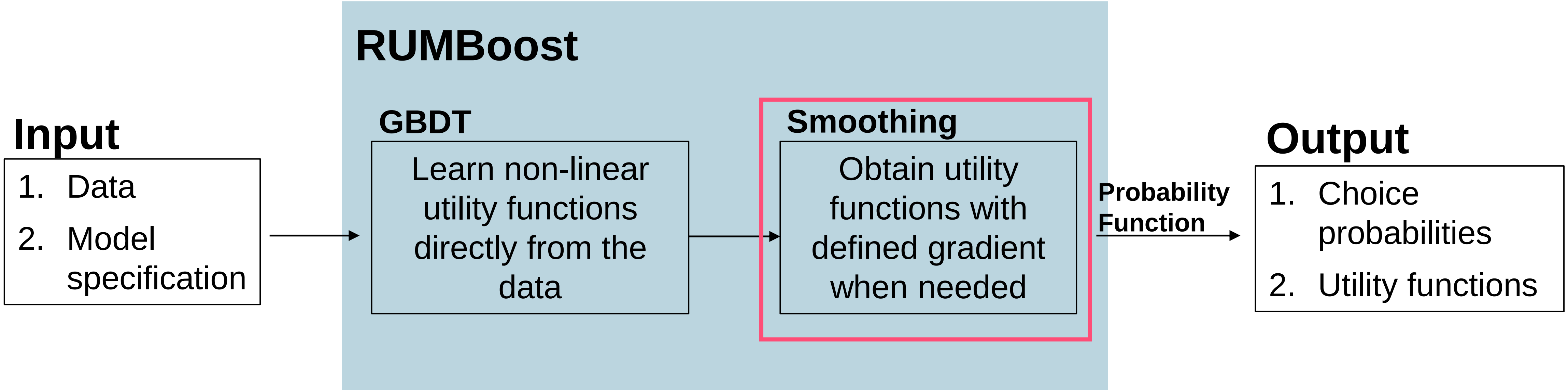
GBUV – Example of utility values

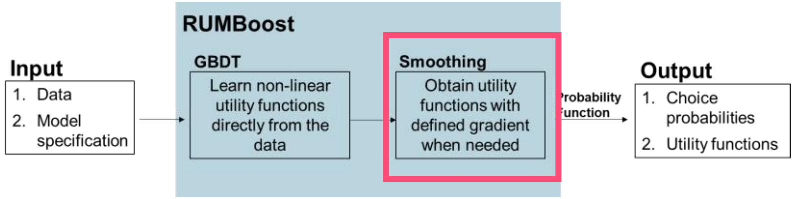


- Cycling travel time (LPMC dataset)
- Piece-wise constant values
  - No defined gradient…
  - …therefore no behavioural indicators



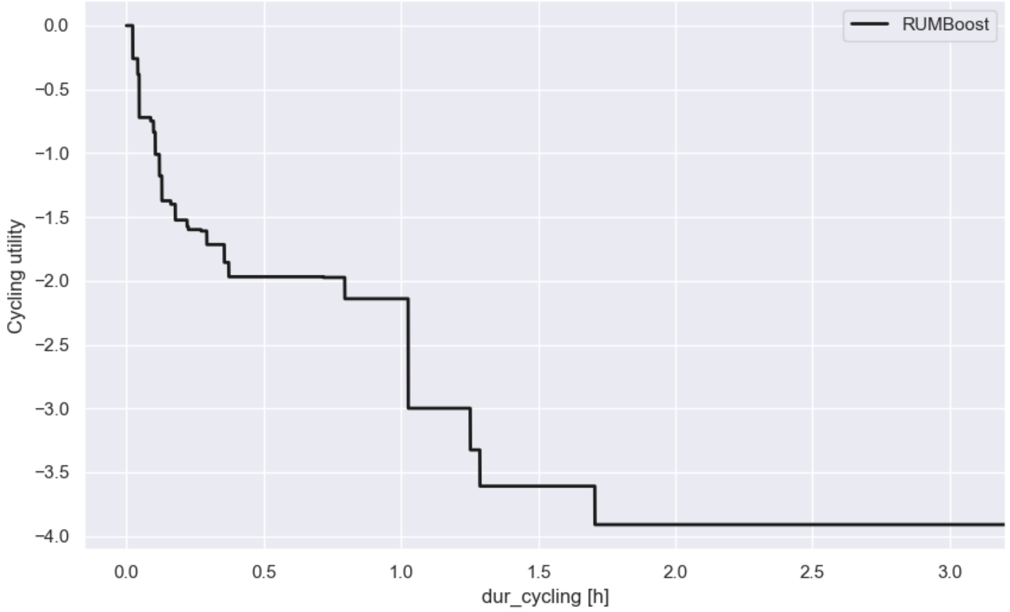
How to smooth GBUV?

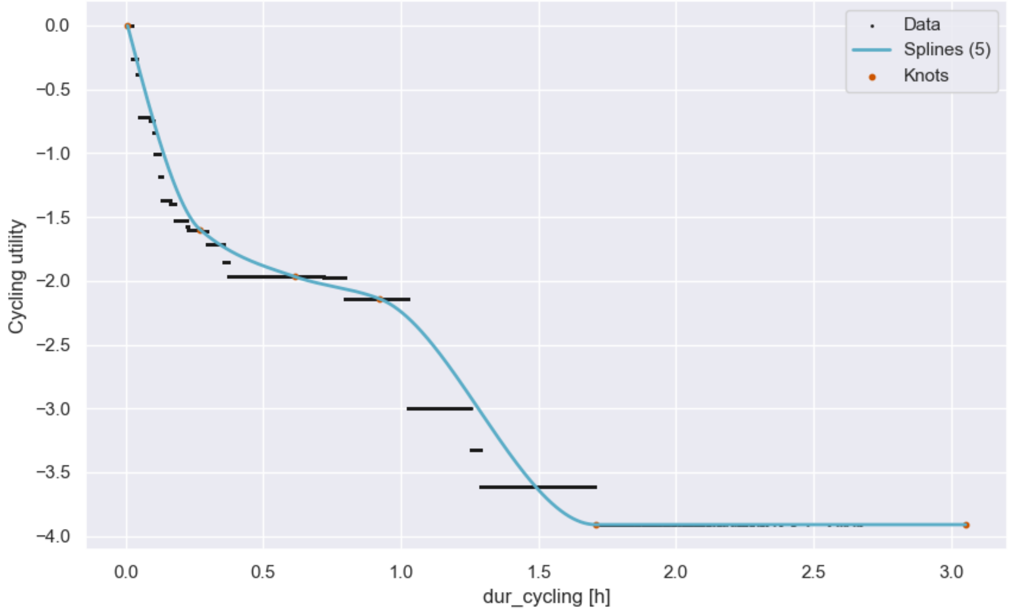




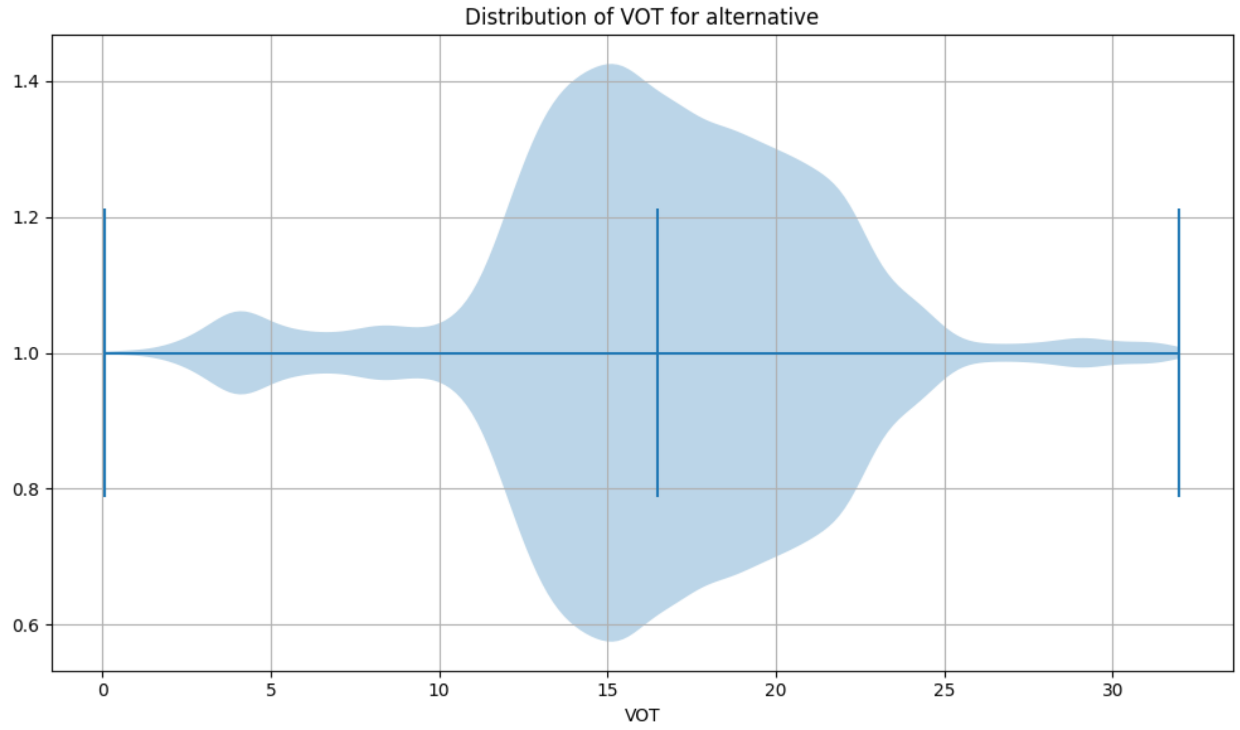
Piecewise Cubic Utility Function (PCUF)

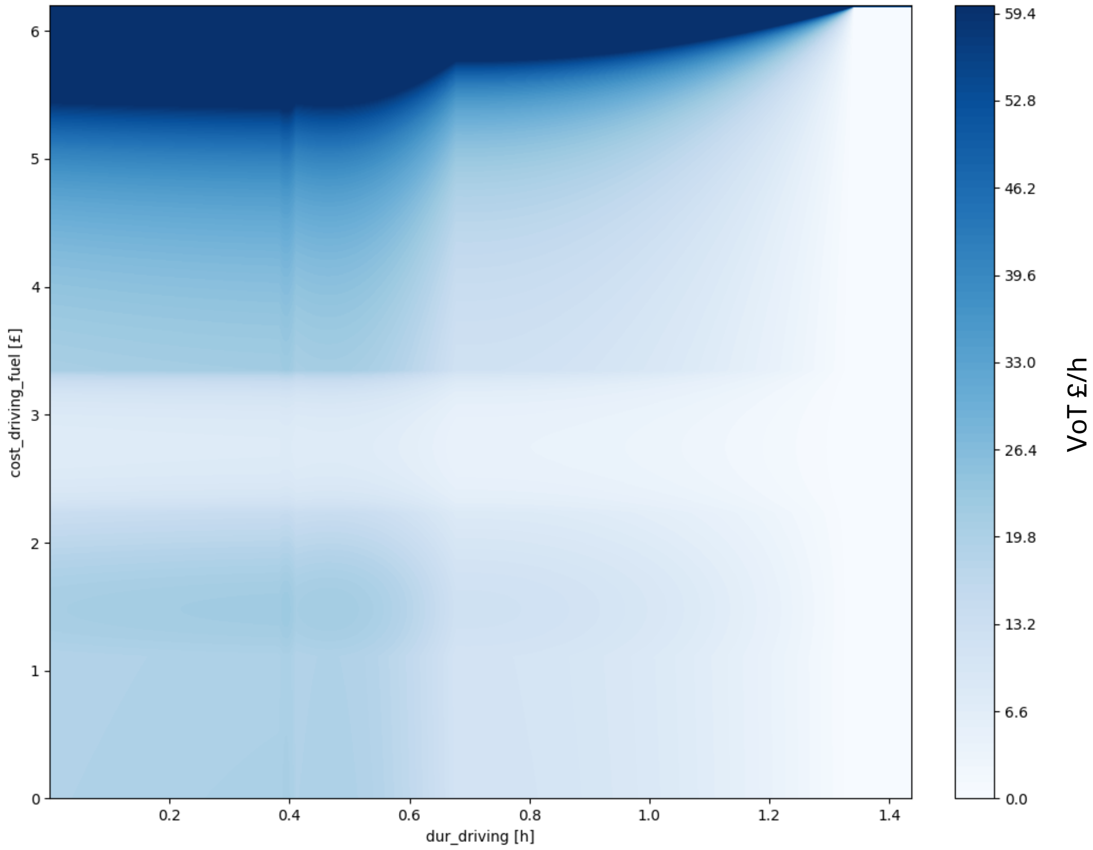
- **Interpolation** of GBUV with monotonic Hermite splines (Fritsch and Butland, 1984)



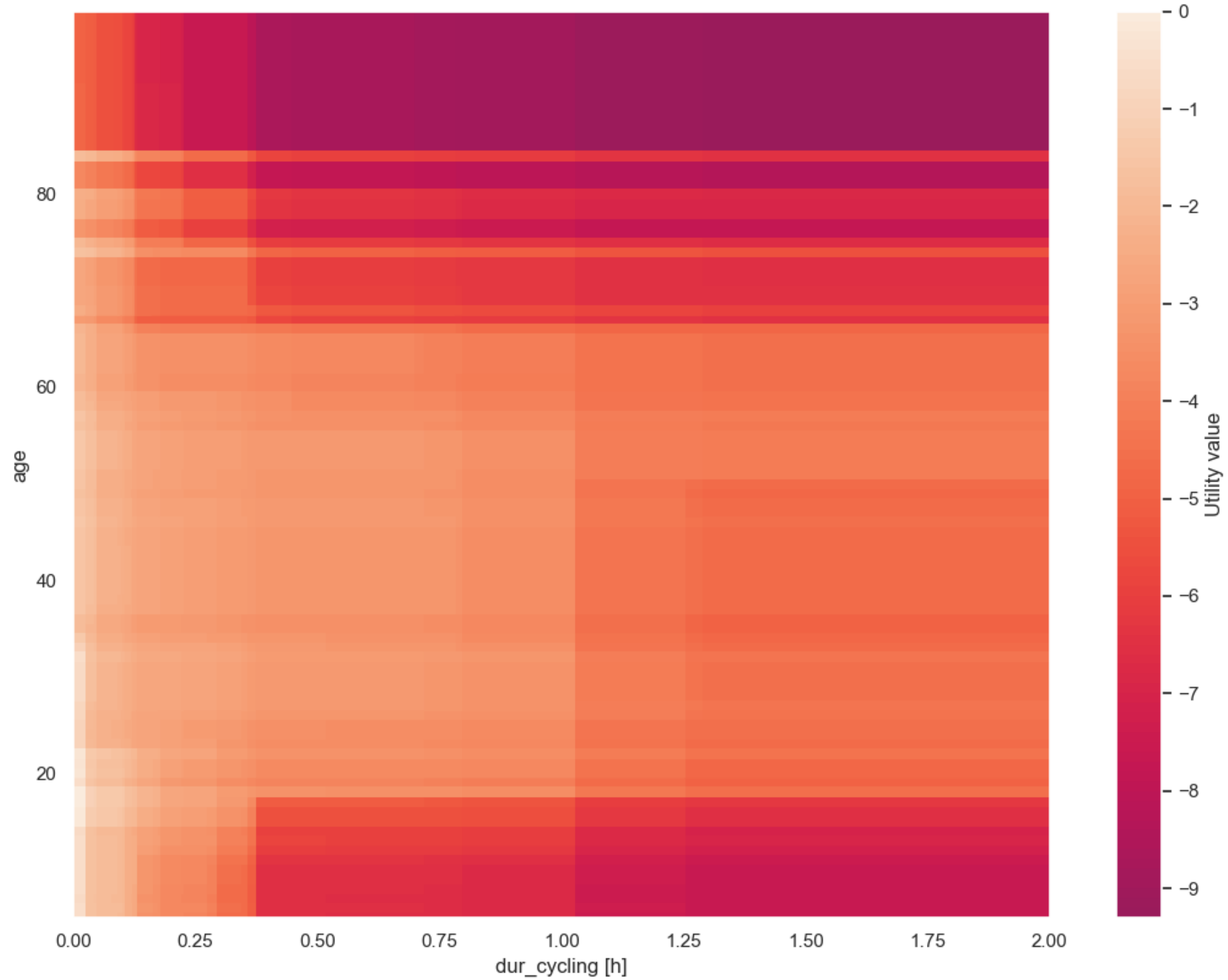


VoT


Distribution of VOT for alternative
1.4
1.2
1.0
0.8
0.6
0
5
10
15
20
25
30
VOT


cost_driving_fuel [£]
dur_driving [h]
0
1
2
3
4
5
6
0.2
0.4
0.6
0.8
1.0
1.2
1.4
59.4
52.8
46.2
39.6
33.0
26.4
19.8
13.2
6.6
0.0
VoT £/h

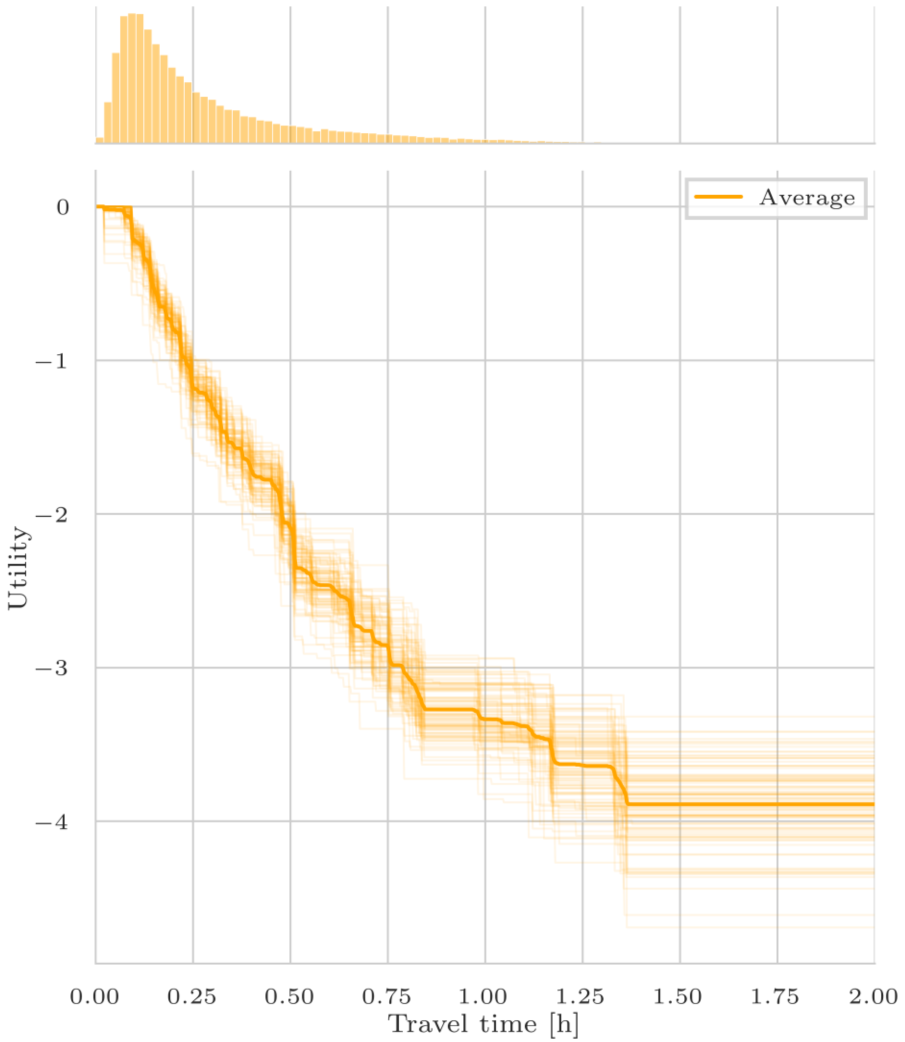
Attribute interaction



- Variables can be arbitrarily interacted within parameter ensembles

- Shown here: cycling duration (monotonic) with age

Bootstrapping



- Model does not fully converge – do not obtain confidence intervals

- Can be estimated empirically using bootstrapping

Extension to complex model specifications

Input

1. Data
2. Model specification

RUMBoost

GBDT

Learn non-linear utility functions directly from the data

Smoothing

Obtain utility functions with defined gradient when needed

Probability Function

Output

1. Choice probabilities
2. Utility functions

Assumption on the error term:

Independent and identically distributed → MNL

Correlation within alternatives → NL, CNL

Additional parameters (μ and α) as well as nested structure are optimised with **scipy.minimize**

Case study – LPMC dataset

- Case study on a mode choice dataset (appr. 80000 observations and 4 alternatives)
- Nests are on **motorised** modes (public transport and driving) (NL and CNL) and **flexible** modes (walking, cycling and driving) (CNL)

	NL	CNL	RUMBoost-NL	RUMBoost-CNL
$\mu_{motorised}$	1.391	2.025	1.167	1.821
$\mu_{flexible}$	-	1.000	-	1.000
$\alpha_{driving,motorised}$	-	0.467	-	0.364

LPMC – benchmarks

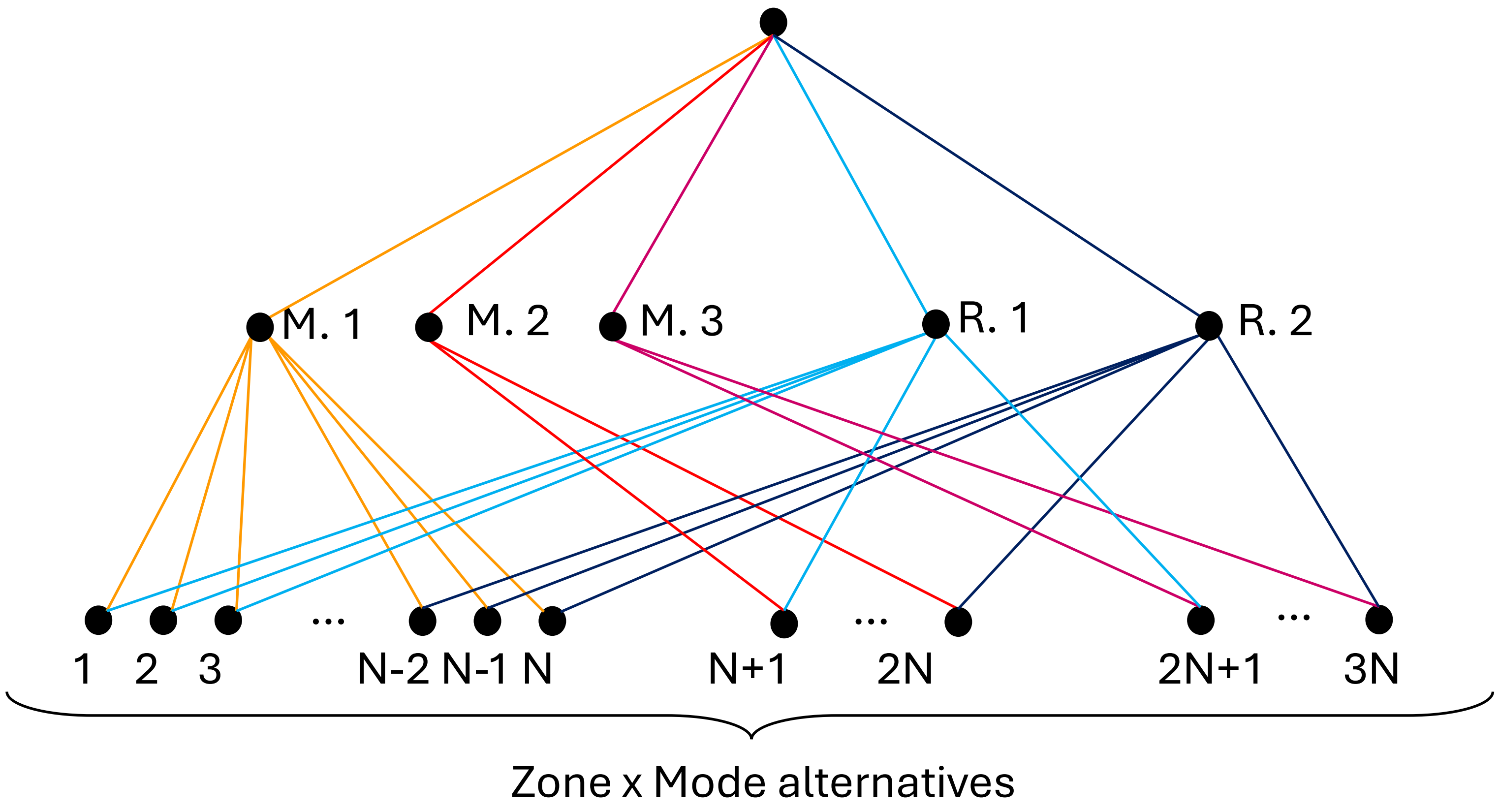
Models	Metrics	LPMC 5 fold CV	LPMC Holdout test set
MNL	CEL	0.6913	0.7085
	Comp. Time [s]	242.14	-
NL	CEL	0.6921	0.7091
	Comp. Time [s]	1067.04	-
CNL	CEL	0.6908	0.7070
	Comp. Time [s]	5120.01	-
RUMBoost-GBUV	CEL	0.6570	0.6737
	Comp. Time [s]	6.48	-
RUMBoost-PCUF	CEL	0.6479*	0.6730
	Comp. Time [s]	712.48*	-
RUMBoost-NL	CEL	0.6568	0.6731
	Comp. Time [s]	48.53	-
RUMBoost-CNL	CEL	0.6546	**0.6716**
	Comp. Time [s]	183.91	-

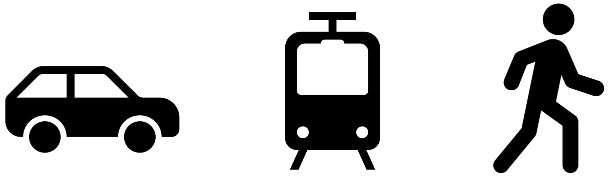
*Not with CV

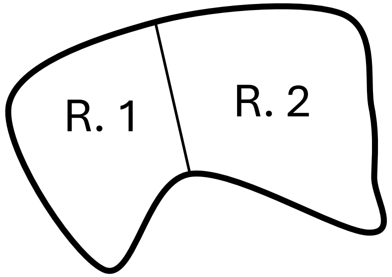
- All RUMBoost models outperform their relative RUM while being **20 to 40** times faster
- No loss of interpretability (only of formal significance testing)
- RUMBoost models would scale well to harder problems

Complex model – mode & location choice

Nests for each transport mode and region







Behavioural assumptions

1. Mode and location choices are inherently **linked**
2. The **utility** derived by the choice of a location, with a transportation mode, depends on the travel time and some measures of the attractiveness of the zone
3. **Activity** choice is given and impacts the utility function
4. Alternatives with the same **transportation mode** are correlated
5. Alternatives are **spatially** correlated

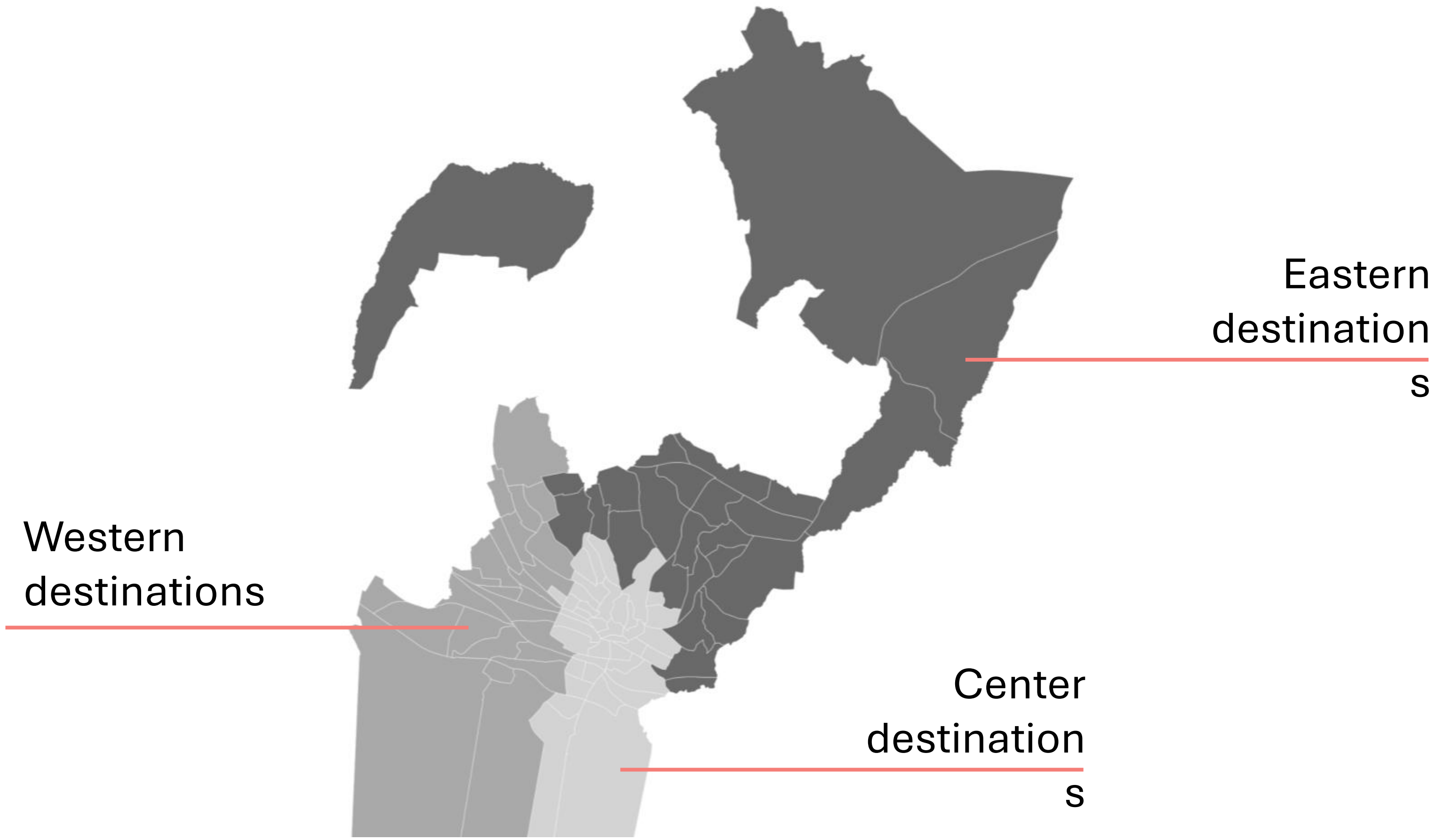
Case study - Lausanne

- MTMC: **trip diary** dataset collected by the Swiss government (2017)
  - Using only zones from **Lausanne** (88 zones) and trips with destination in Lausanne (about 3500 trips)

- **Zones** defined by the Swiss government

- Zone-to-zone **travel time** and **attractivity** measures (job density, population density) provided by SBB

Group of destinations



Case study – Lausanne model

- Trips with destination to Lausanne only (about 3000 observations)

$$V_{mln} = ASC_m + \beta_{cost,m}\mathrm{COST}_{lm} + \beta_{TT,m}\mathrm{TT}_{lm} + \sum \beta_a (a_n \mathrm{JOBDENSITY}_l + a_n \mathrm{POPDENSITY}_l)$$

- 88 zones in Lausanne and 3 transportation modes (car, pt, soft modes) – 264 alternatives

- 2*88 (cost) + 3*88 (travel time) + 6*88 (zone attractiveness) = 968 features!

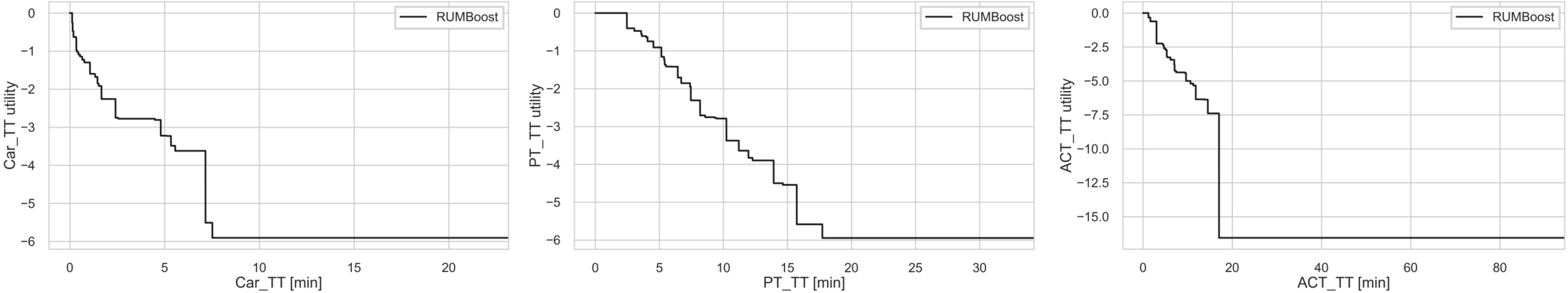
Lausanne study – estimation results

	CNL	RUMBoost-CNL
mu_{east}	1.04	1.15
mu_{west}	1.04	1.03
mu_{center}	1	1.05
mu_{car}	1.39	1.36
mu_{pt}	2.01	1.99
mu_{act}	1	1.09
Value found by:	Estimation	Hyperparameter search with 100 iterations

Models	Metrics	MTMC - Lausanne 5 fold CV	MTMC - Lausanne Holdout test set
CNL	CEL	-	4.77
	Comp. Time [min]	-	5760+ (4 days+)
RUMBoost-CNL	CEL	4.72	4.73
	Comp. Time [min]	933	2.5

New approach implemented to estimate nesting parameters μ, α directly

Lausanne study – GBUV



Case study – nationwide model

- Full observations from the MTMC 2015 dataset (about 180000 observations, 147000 in the training set)

- 23895 alternatives (7965 zones and 3 transportation modes)

- 3*7965 (travel time per mode) + 6*7965 (zone attractiveness per activity) + 5 constants (mode and zone type) = 71690 features per choice situation (cost omitted)

- Impossible to estimate with conventional CNL (as is…!)

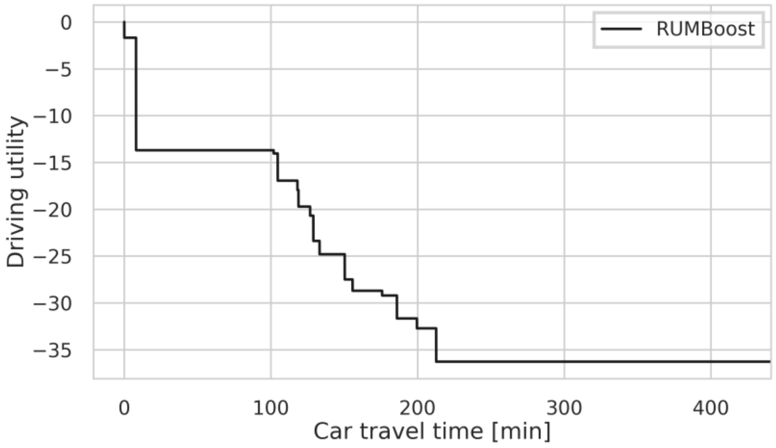
Engineering detail:

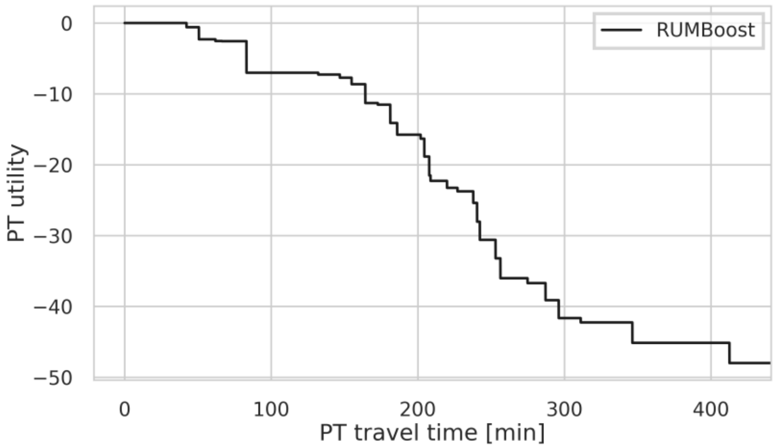
- Gradient and hessian are computed on the GPU using pytorch
- Batch estimation to avoid memory errors (2000 observations)
- Nesting parameters (μ, α) are minimized at each boosting round, using SLSQP (scipy.minimize)

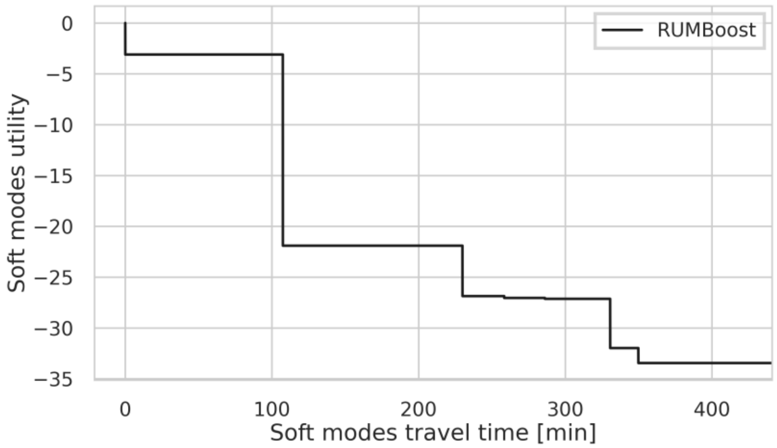
Nationwide model – Estimation results

MTMC - Switzerland	RUMBoost-CNL
CEL (holdout test set)	8.20
CEL (train set)	8.12
Comp. time [h]	4.9
N. boosting rounds	2220 (30 per batch)
$\mathrm{ASC}_{\mathrm{car}}$	-1.35
$\mathrm{ASC}_{\mathrm{PT}}$	-4.85
$\mathrm{ASC}_{\mathrm{s.\ modes}}$	0 (normalised)
$\mu_{\mathrm{swiss\ german}}$	1.03
$\mu_{\mathrm{swiss\ french}}$	1.15
$\mu_{\mathrm{swiss\ italian}}$	1.15
μ_{car}	1.00
μ_{pt}	1.15
$\mu_{\mathrm{soft\ modes}}$	1.15

Nationwide model – GBUV

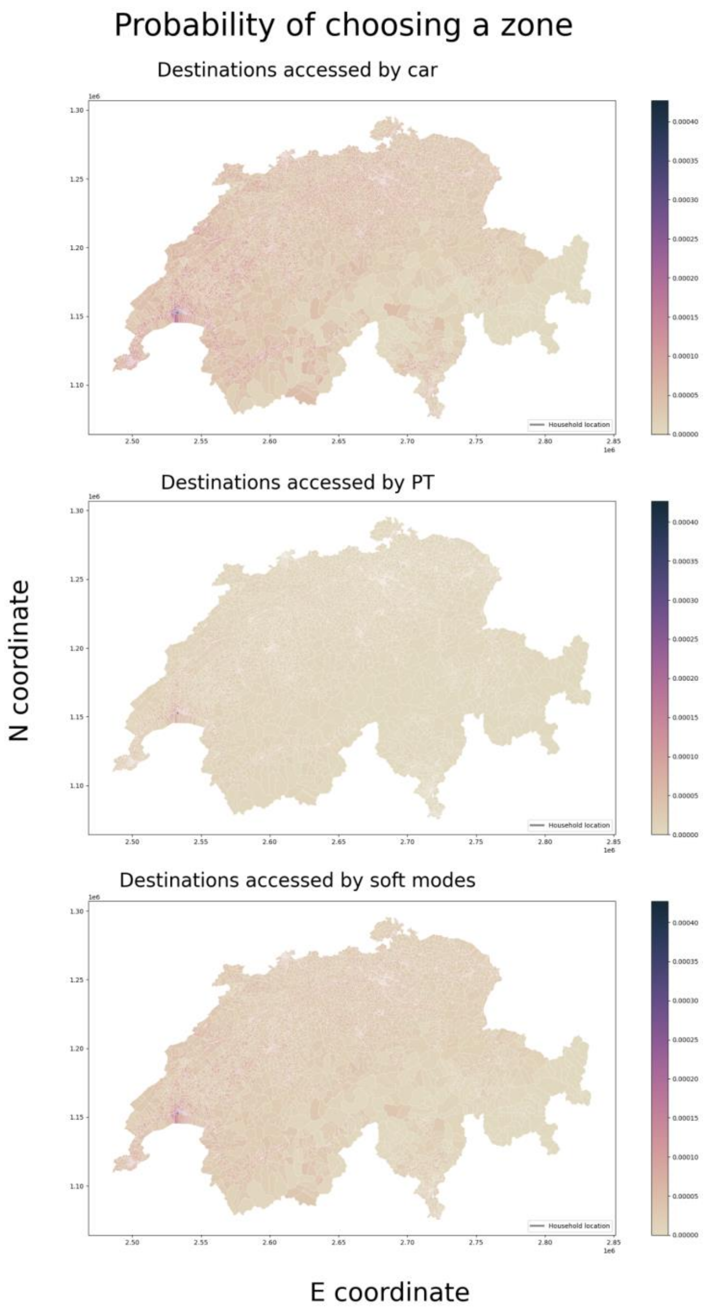






Nationwide model – choice probabilities

Work trip for Lausanne resident:



Daily activity scheduling with Caveat

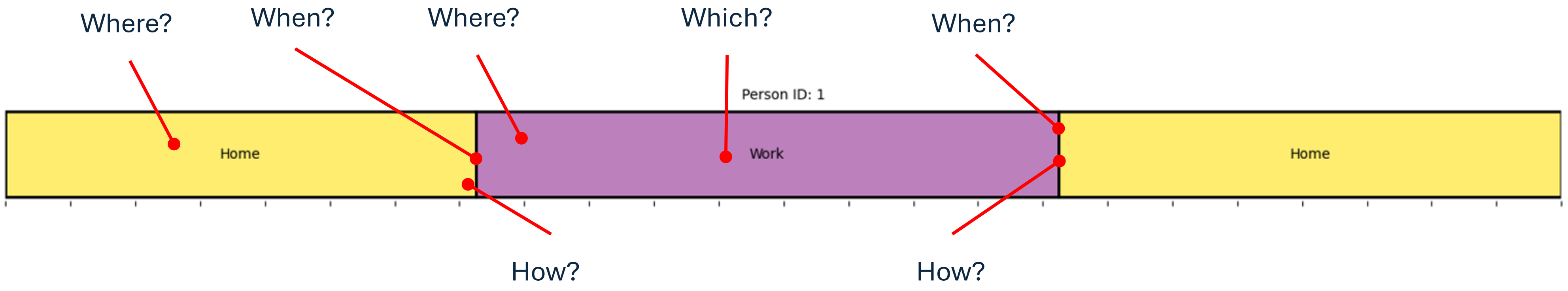
Fred Shone
frederick.shone.17@ucl.ac.uk

- Code available on github/pypi: https://github.com/fredshone/caveat

- **See forthcoming presentations at hEART and MUM**

Why is Activity-based Modelling hard?

- An individual’s activity sequence is result of multiple different choices, with no clear order of dependency, even for simplest case:



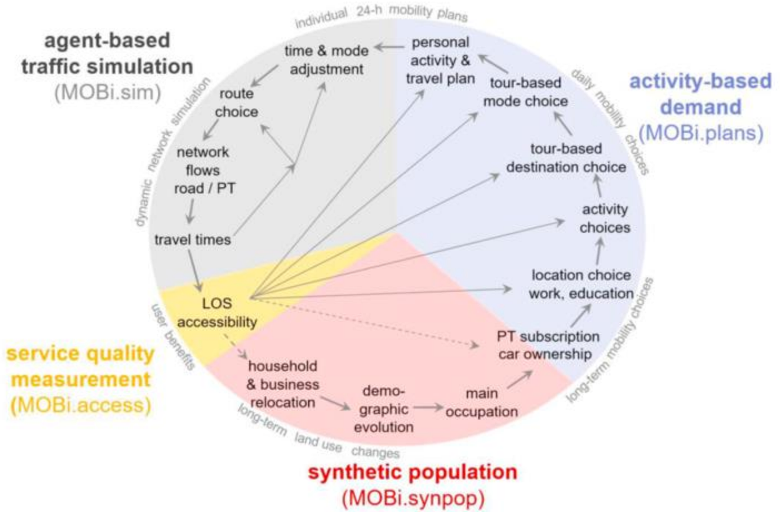
The Status Quo

Existing approaches are complex, requiring many interacting discrete choices

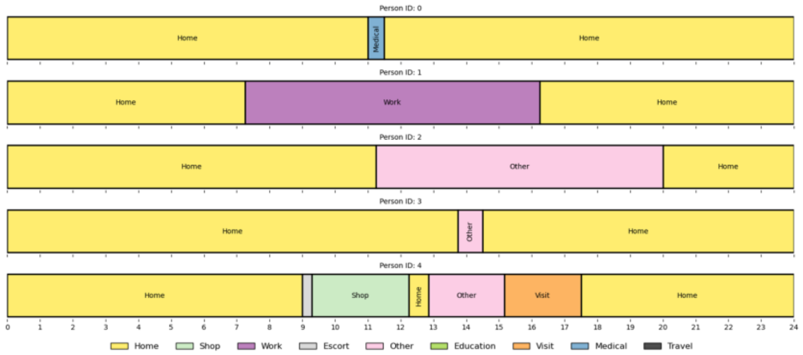
Results in models that are either:

- Expensive to develop and use, or
- Lacking realistic diversity of outputs

Efforts to combine multiple activity scheduling choices simultaneously, such as **OASIS**, are computationally challenging – both for estimation and simulation



SBB Activity Modelling Framework



Example schedules from London Travel Demand Survey

Deep Generative Models for activity scheduling

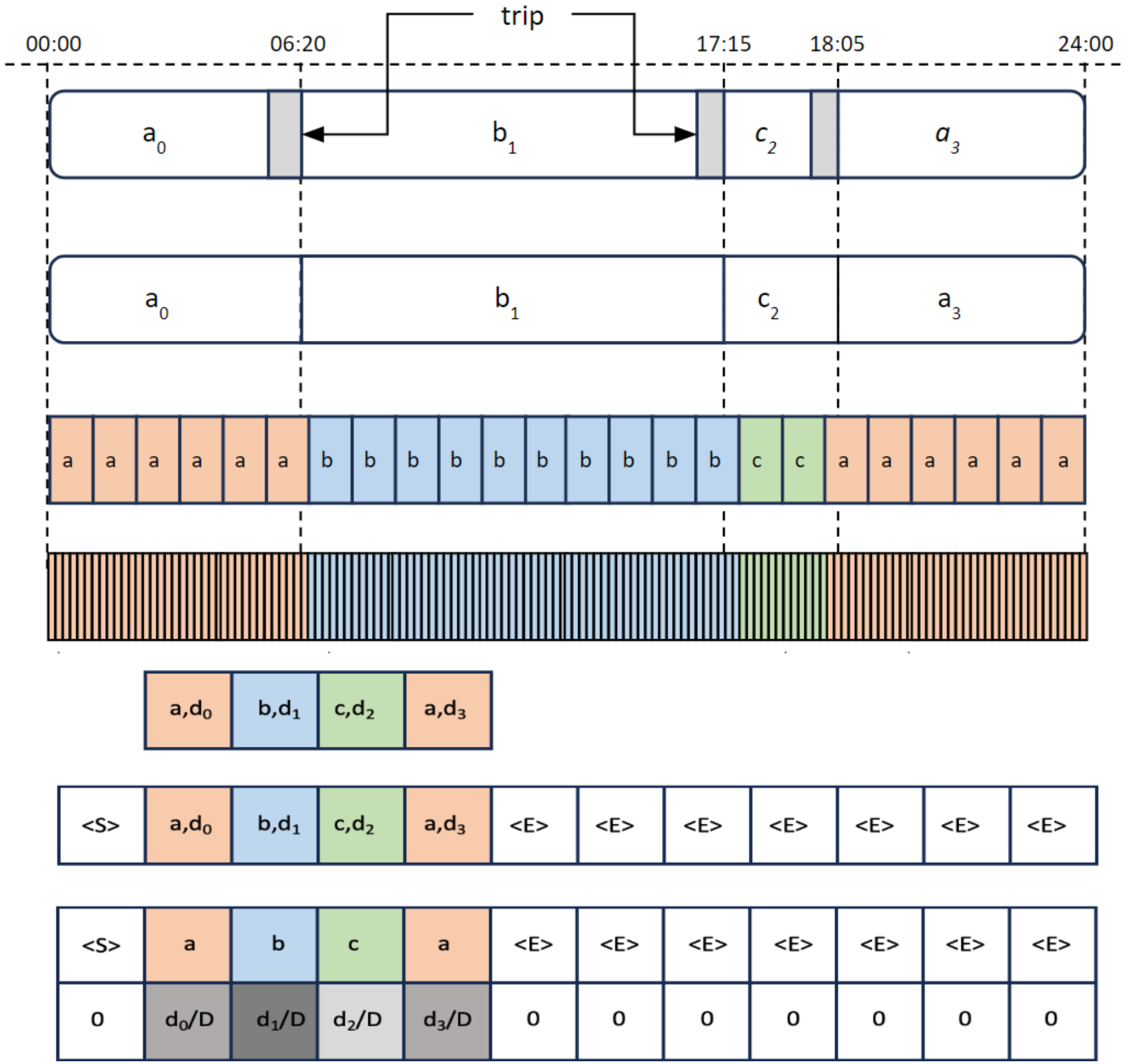
- Learn to model observed distribution of historic data using Conditional Variational Auto-Encoders (CVAEs)
  - Map from a known random distribution (typically Gaussian) to observed activity schedules
  - Conditional on agent attributes such as location, age, gender, etc
  - New synthetic schedules can then be sampled efficiently by drawing from latent space

- Possible applications:
  - Anonymisation/obfuscation of historic data
  - Resampling for bias correction and simple forecasting
  - Simulation through up-sampling for realistic and diverse populations

- Three key technical contributions
  1. Novel variable length sequence encoding of activity schedules, evaluated against fixed length image-like encoding
  2. CVAE architectures derived from language (sequence) models
  3. Domain specific evaluation framework

- Case study:
  - 40,000 activity schedules from the UK National Travel Survey

Contribution 1: Encoding



1. Recorded schedule

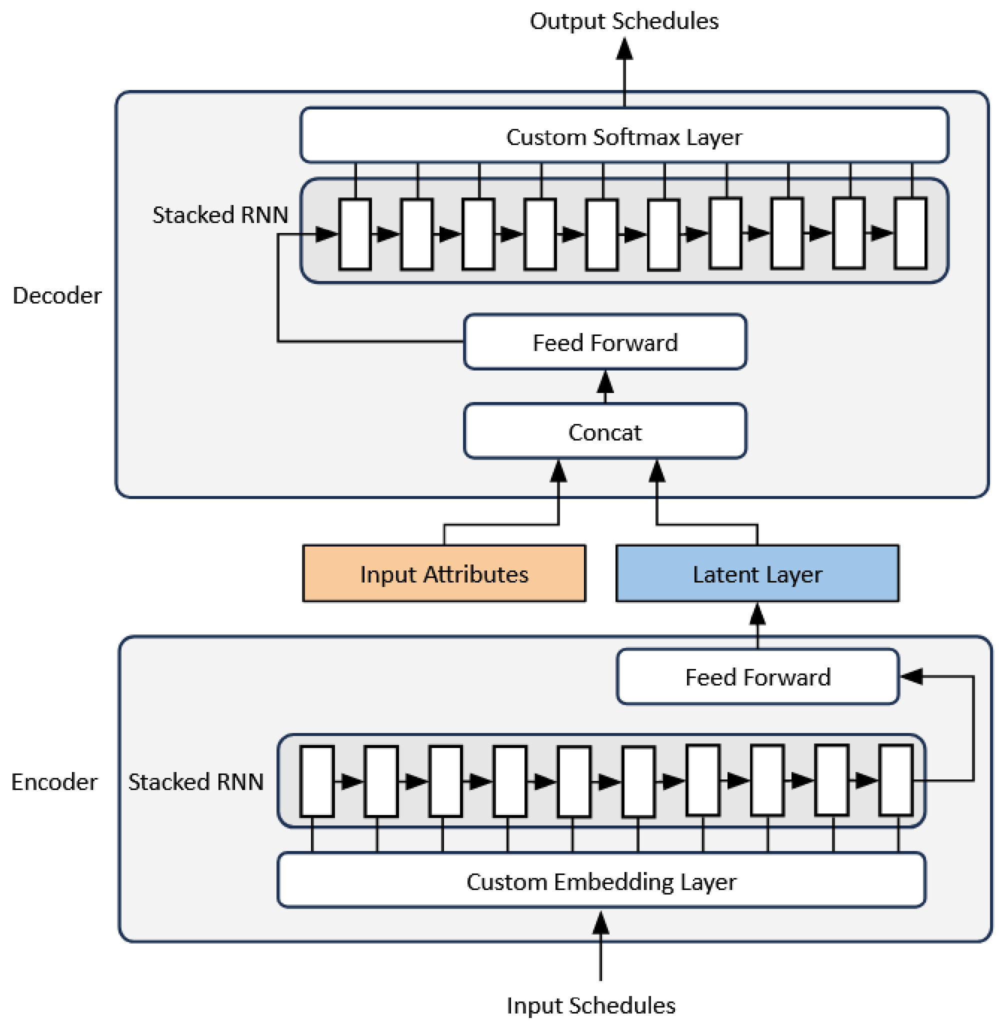
2. Travel incorporated into previous activity

3. Image-like encoding – fixed length sequence with 10-minute resolution

4. Sentence-like encoding – variable length sequence with ordering and associated duration for each activity

https://github.com/fredshone/caveat

Contribution 2: CVAE Architecture



https://github.com/fredshone/caveat

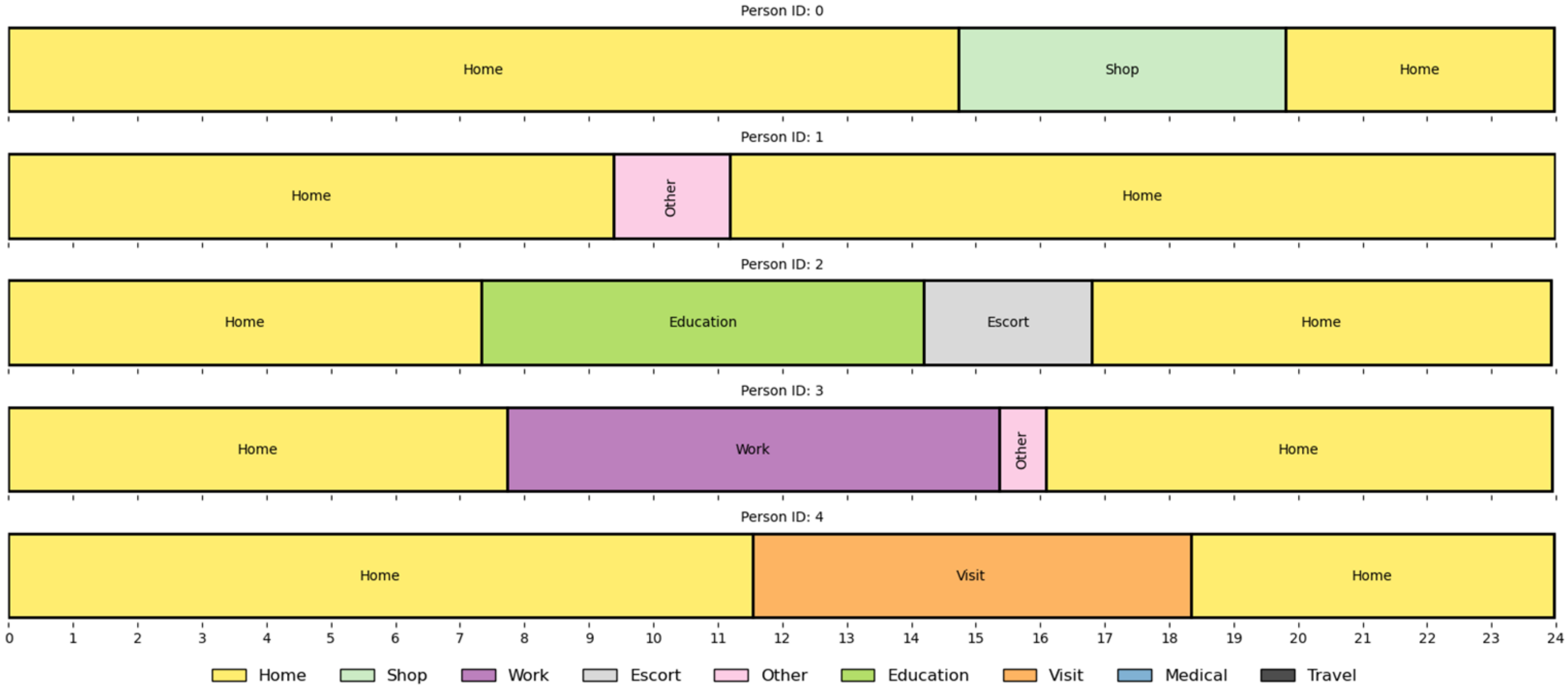
Contribution 3: Evaluation

- Correctness
  - Unlike traditional models, generative approaches cannot be evaluated via a withheld test dataset
  - Consider a model generating text or images - how do we measure how good the synthetic test, or image, or activity schedule is?
  - We provide an evaluation framework that measures the distance between key distributions in the observed and synthetic populations, such as participations, orderings and timings.
- Creativity
  - It is also desirable for our models to be creative.
  - We therefore include evaluation of the models ability to generate diverse and novel activity schedules

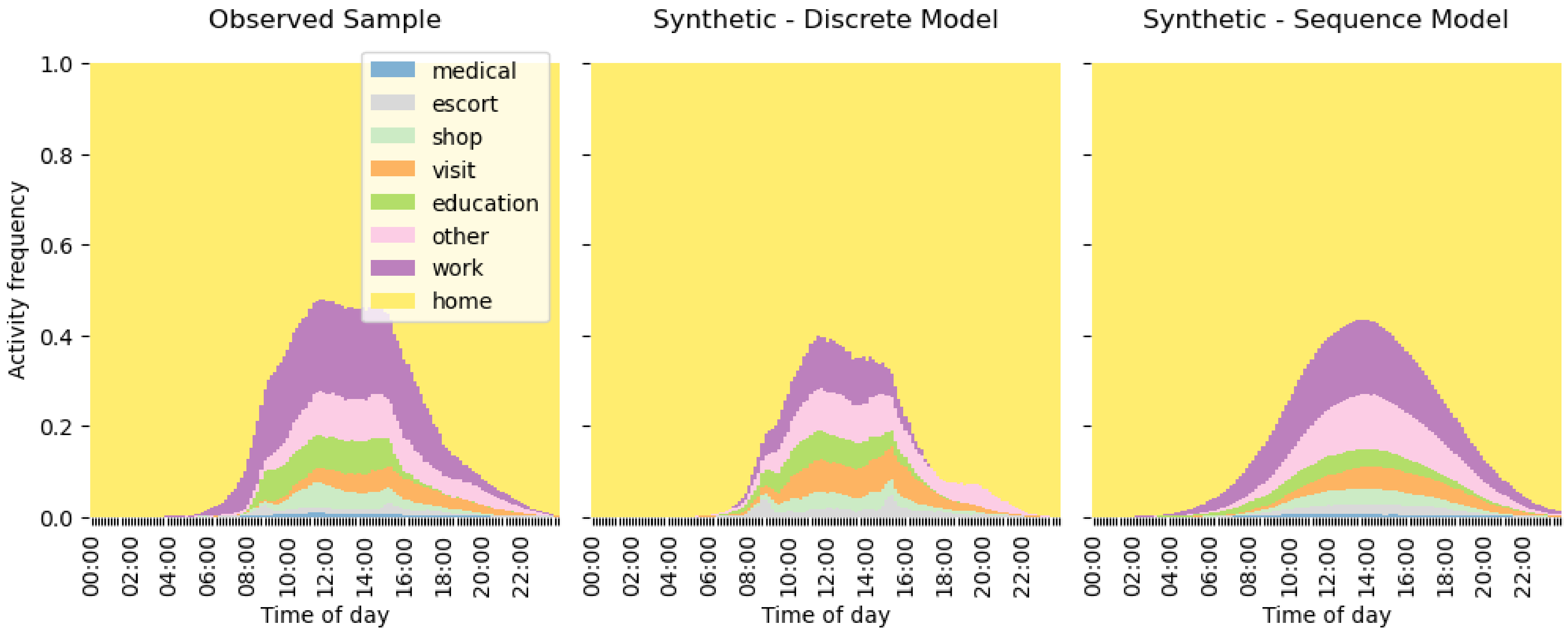
https://github.com/fredshone/caveat

big

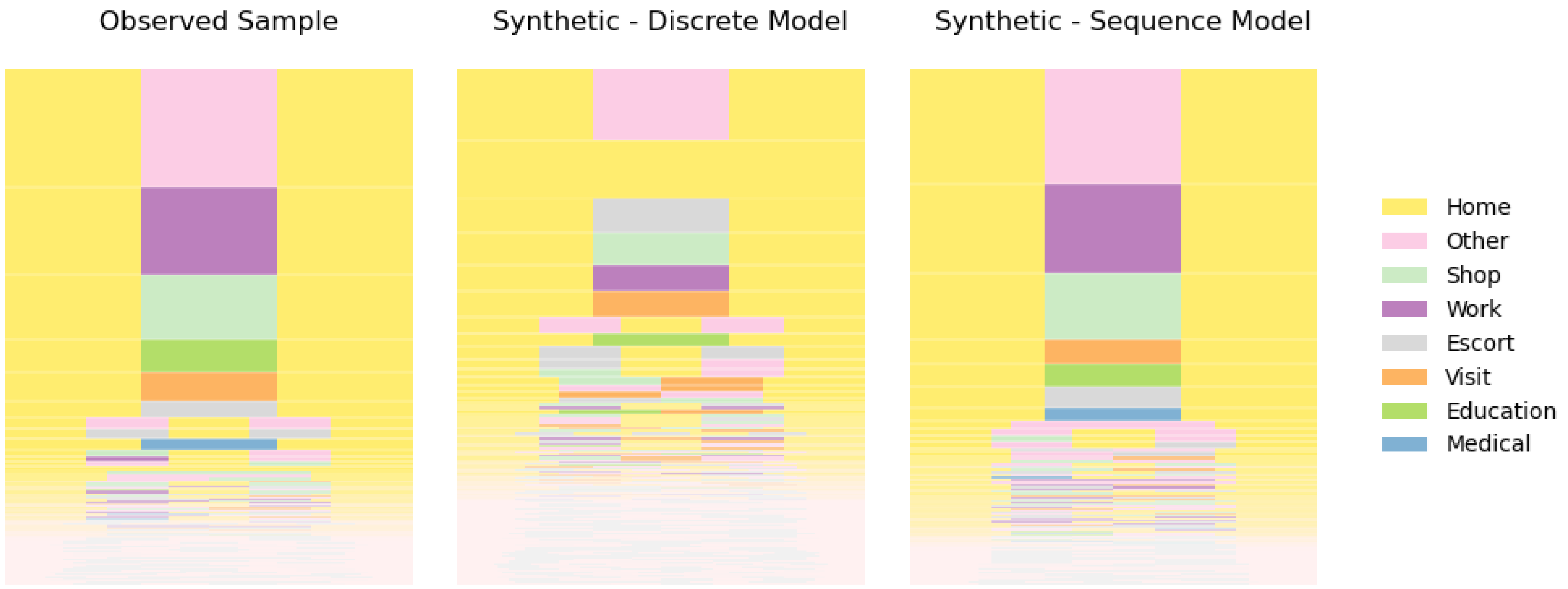
Example output



Evaluation - Aggregate Activity Histograms

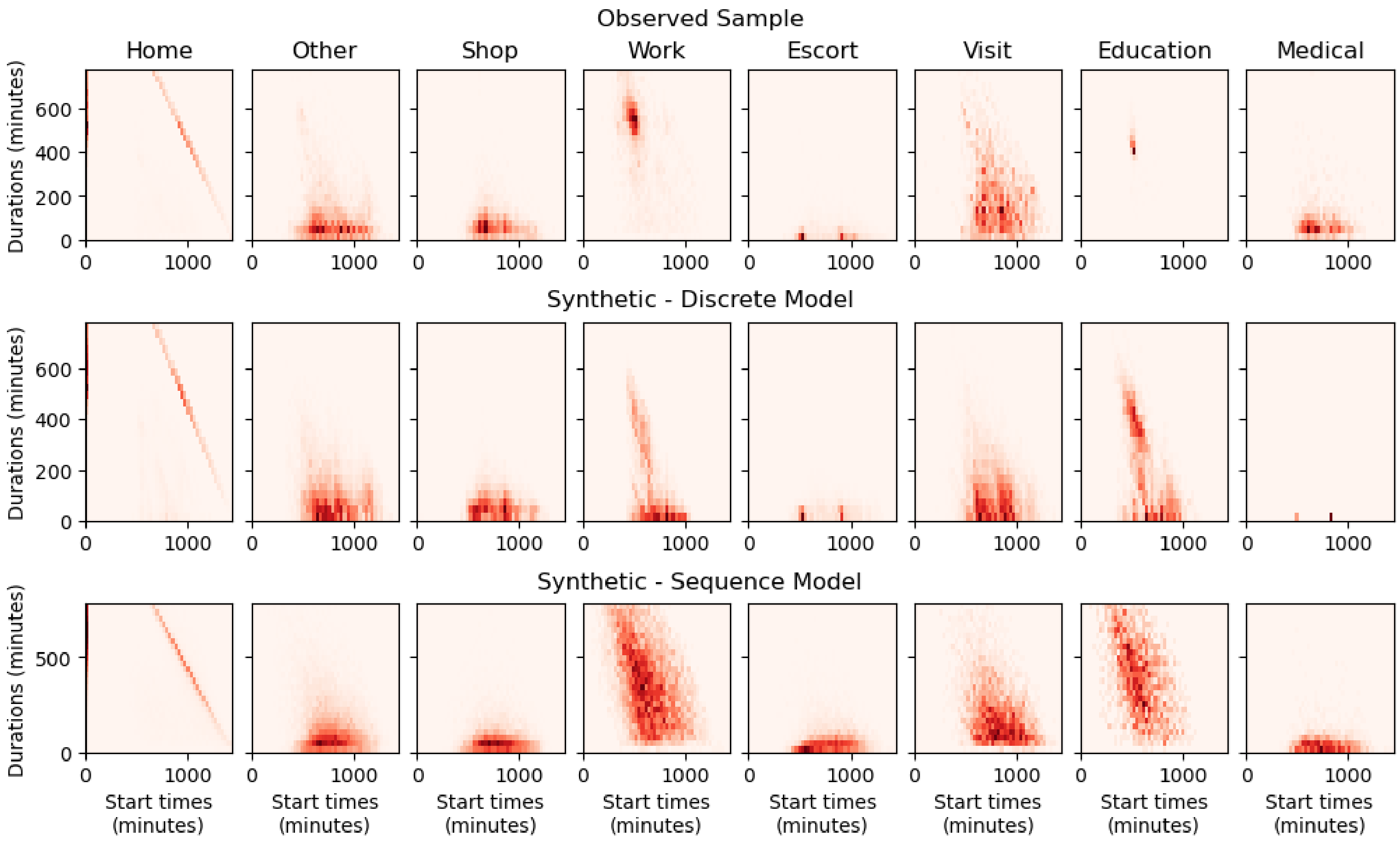


Evaluation - Tour Structures



Evaluation - Activity Times and Durations

Evaluation - Joint Activity Starts and Durations



Scalability

- Able to learn from single year of UK travel survey data
    - Can be extended to alternative data sources - mobile and GPS
- Requires GPU, but **extremely** efficient
    - On a modern GPU (~1k GBP), trains in ~20 minutes and can generate new populations near instantaneously.
- The model will likely scale easily to:
    - More data and larger populations
    - More - and more complex - choices, such as locations and trip mode
    - Longer (multi-day) sequences
    - Household activity sequences
- Can be incorporated into existing agent-based simulations models, replacing numerous discrete models (primary participation, secondary participation, tour type) and scheduling algorithms with a single step.

Thank you



Nicolas Salvadé
nicolas.salvade.22@ucl.ac.uk

Fred Shone
frederick.shone.17@ucl.ac.uk

Tim Hillel
tim.hillel@ucl.ac.uk

big
behaviour &
infrastructure group

https://tinyurl.com/big-ucl