# COMPUTATION OF MODE-DEPENDENT TRAVEL TIME MATRICES FOR AN AGENT-BASED DEMAND MODEL COMPUTED USING A STANDALONE ACCESSIBILITY TOOL

Daniel Krajzewicz, Alain Schengen

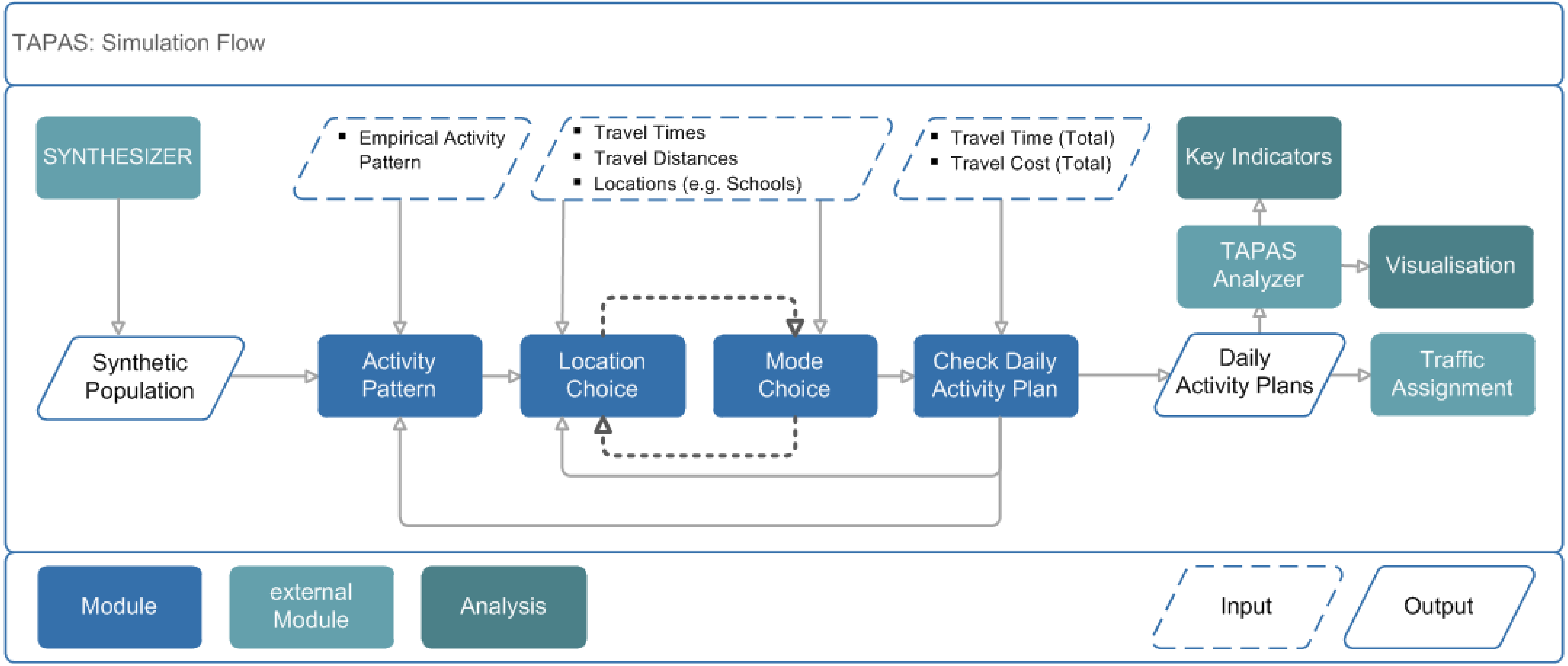
ABMTRANS workshop 2024, 23rd-25th of April 2024

DLR

# The agent-based demand model TAPAS
(Travel-Activity Pattern Simulation)

- TAPAS is a microscopic agent and activity based demand model
  - Every single person is modelled for itself…
  - … has attributes, such as age, gender, employment status (including pupils and students), may have a driver license, a public transport season ticket and/or a bike
  - … is classified into a person group
  - … chooses its daily activities matching the groups‘ behavior
- Peculiarity: Grouping to households
  - Certain mobility options (e.g. the owned car) may be used by different persons along the day
  - The ownership and the access to the option is represented on household level
  - This also counts for the mobility budget
- TAPAS is available as open source (https://github.com/DLR-VF/TAPAS )

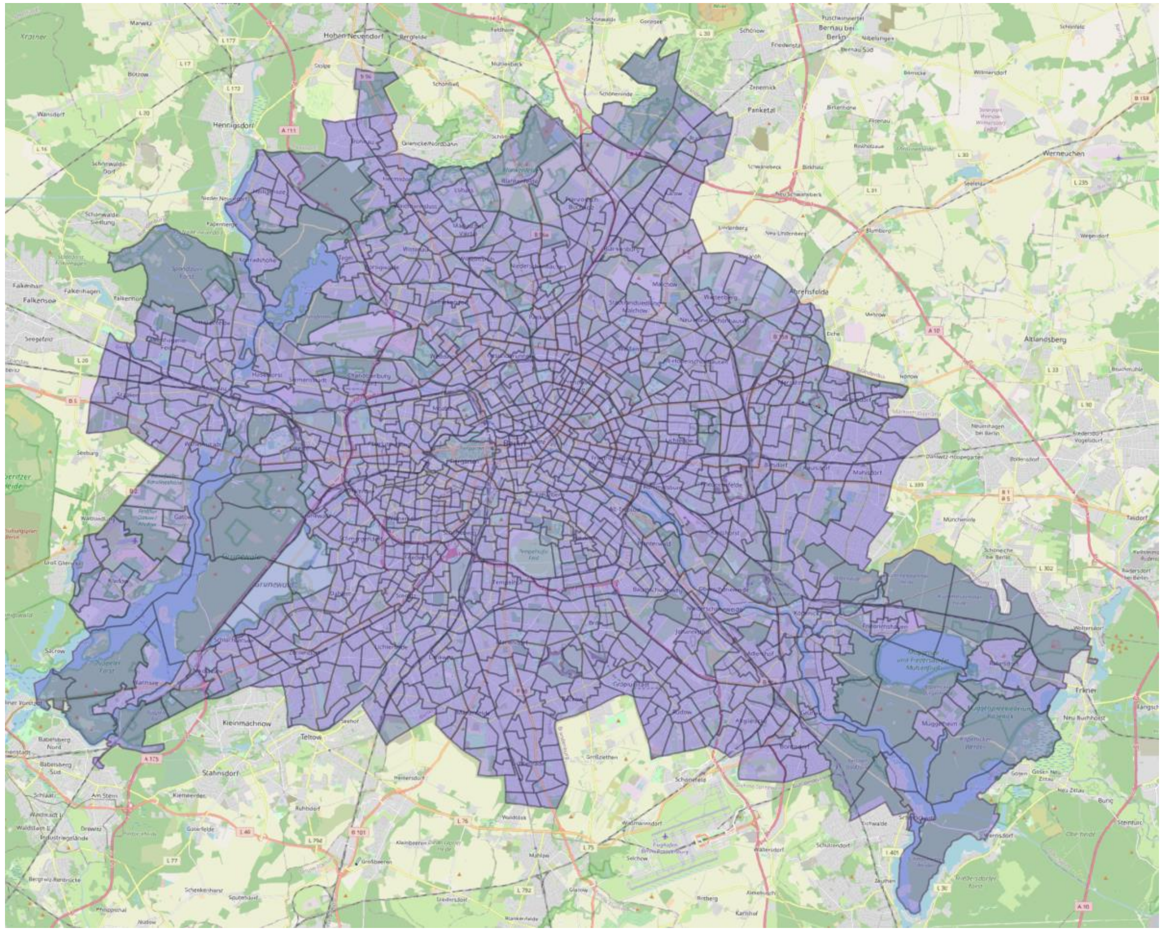

# The agent-based demand model TAPAS
## Workflow



# The agent-based demand model TAPAS
## Mode Choice in TAPAS

- Multiple mode choice model implementations:
  - Decision trees
  - MNL
  - EKT
- MNL is mostly used currently
- Influenced by attributes of the mode, the trip, and the person
  - costs
  - travel time
  - trip purpose
  - person‘s age
  - driver license
  - car availability
  - public transport season ticket

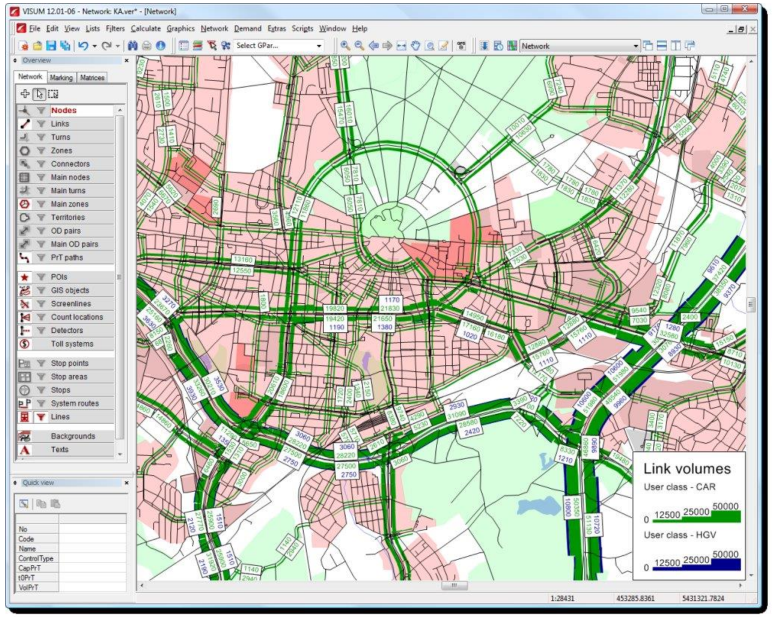
# **Study area**: Berlin



# Travel Time and Distance Matrices
## Initial Computation

- … travel time and distance matrices are a crucial input

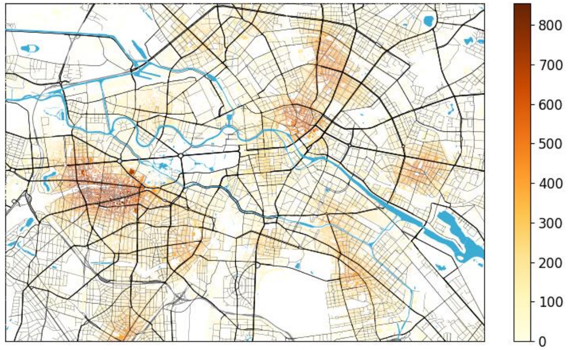
- Initially, we used PTV VISUM (PTV AG) to compute them
- This has some drawbacks
  - Some manual steps are required as VISUM relies on its graphical user interface (at least we did not script it)
  - VISUM is a commercial application – could not be used by some TAPAS users




PTV VISUM screen shot
Source: PTV AG

# Travel Time and Distance Matrices
## New computation method

- The new approach uses the tool „UrMoAC“ (Urban Mobility Accessibility Computer)
  - Open source (https://github.com/DLR-VF/UrMoAC)
  - Run on the command line → scriptable, so no manual interaction needed
  - Relatively fast
  - Loads network travel times from simulations or FCD
- In the following, the results of the new approach are put against the ones obtained using the initial approach
  - Please note that we used different networks for walking and using a bike (VISUM: Here, UrMoAC: OSM)
  - … because OSM is what we always use, currently

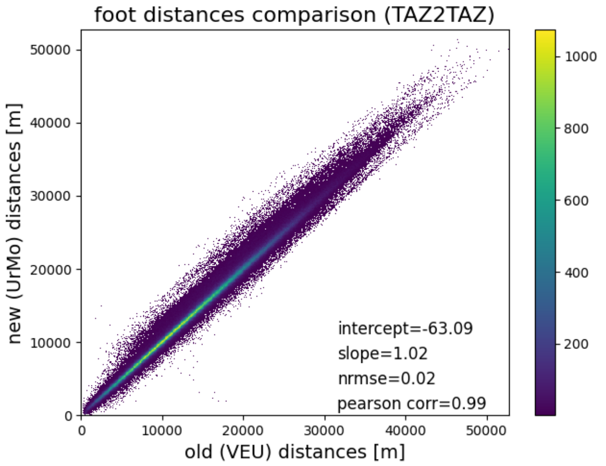


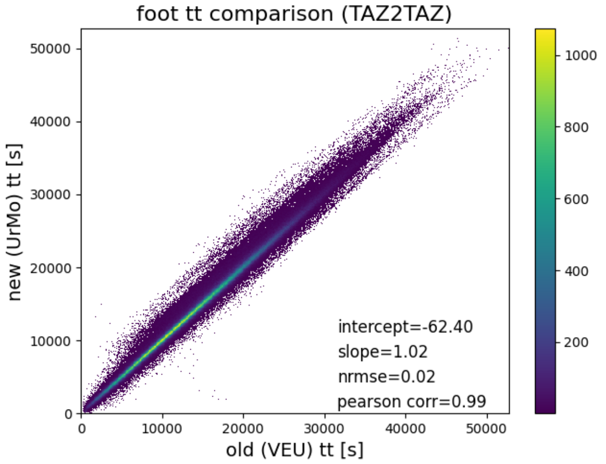

Number of shopping facilities accessible by walking in 15 minutes (Berlin)


Access time of the next public transport halt by walking in seconds (Berlin Adlershof)

# Results Comparison
## Walking

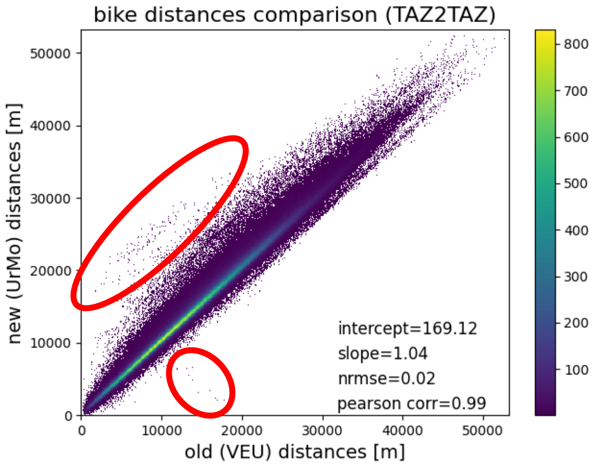
- Direct comparison (as on following slides)
  - x-axis: old values (VISUM)
  - y-axis: new values (UrMoAC)
  - top: distances, bottom: travel times
- distance=travel time as walking is performed with 3.6 km/h
- Only minor differences
  - Neglectable, with no major effects
  - Differences due to using different road networks

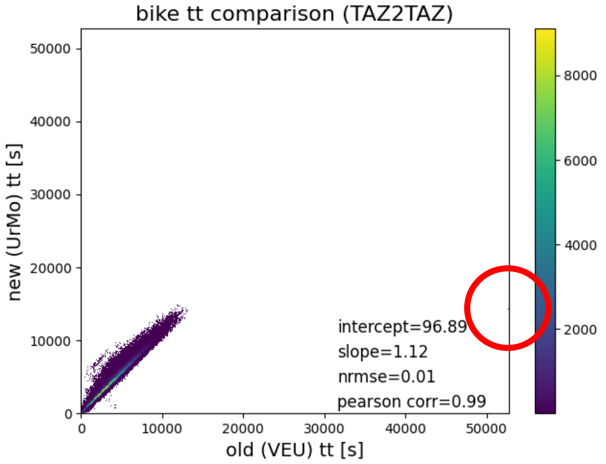


# Results Comparison
## Bicycling

- Biggest difference: outlier in travel times (bottom image)
  - Reason: a connection between an origin and a destination was marked as being invalid (marked using a red circle in bottom image)
  - Besides this artifact, travel times match to a high degree
- Higher distances in the new computation (large top ellipsoid, top image)
  - located in a single TVZ in Grunewald, Berlin, an area dominated by a forest
  - even though more bike infrastructure is given in the OSM data set, its velocity is low, leading to the travel time increase
- Lower travel distances when using the new approach (smaller bottom ellipsoid in top image)
  - located in a single TVZ, as well
  - Here, the Navteq network contains more bicycle infrastructure what yields in longer ways

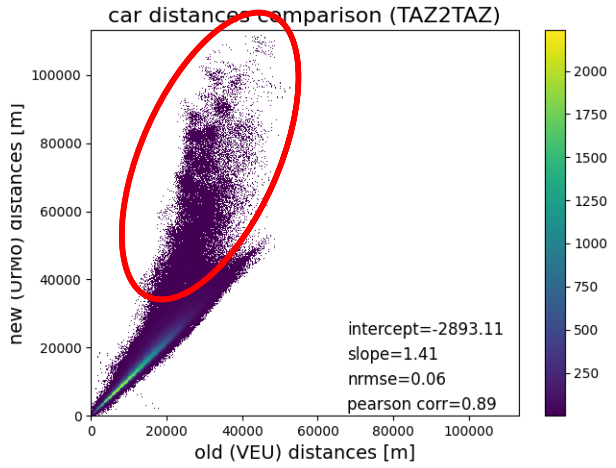


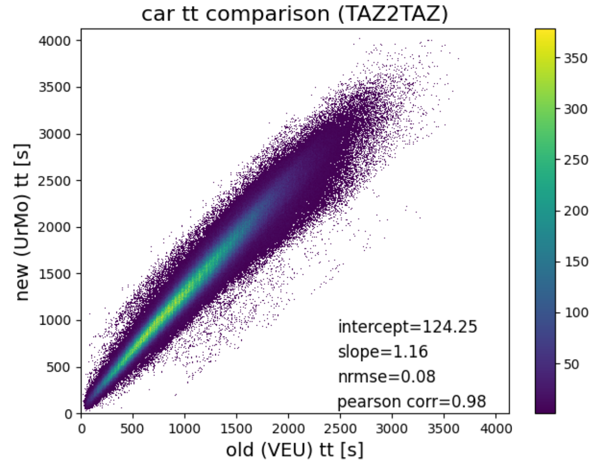



# Results Comparison
## Motorized Individual Transport

- Difference in the distances is clearly visible (top)
  - Reason: inclusion of the outer highway ring around the city of Berlin, missing in the initial computation runs
  - Vehicles using it to avoid crossing the city need to pass longer distances, yet at a higher speed
  - As a result, the travel times stay similar to the initial computation
  - This explanation is supported by the fact that the probability of a TVZ being included in this set as origin or destination increase with an increasing distance from the city center

→ Neglectable, with no major effects
  - Differences due to using different road networks



# Results Comparison
## Public Transport (1)

- More complex task than the ones given before
  - higher number of factors: access and the egress times, waiting times (initial and on interchanges), the number of interchanges
  - Interchanges contribute to rating a route (the one with the lowest value is chosen)

$$v = tt + e^{(n_i \times S_1) - 1} \times s_2$$

with:

$v$: the route rating

$tt$: travel time

$n_i$: number of interchanges

$S_1$: scaling value 1 (=1)

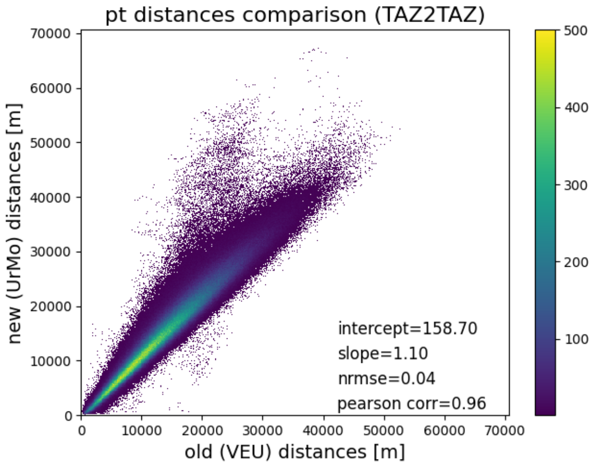
$S_2$: scaling value 2 (=4)

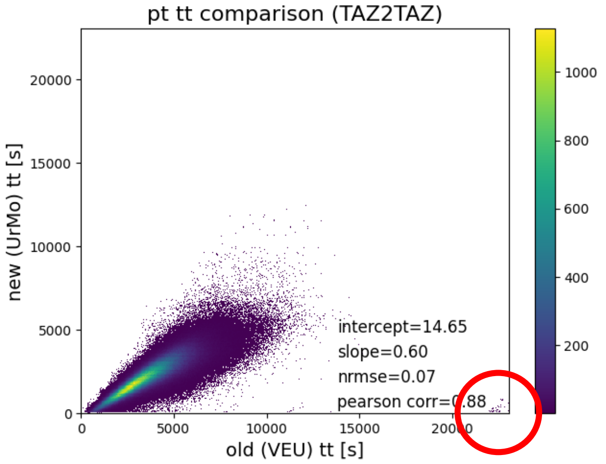
# Results Comparison
## Public Transport (2)

- The initial approach did not deliver distances for PT, only travel times
  - Instead, the walking distances were used

  →the differences in the old and new distances, can be explained by detours some PT lines contain

- Regarding travel times
  - the old computation delivered very high values of almost 6 hours in some few cases, denoted by the red circle
  - They are not realistic for the city of Berlin. Overall, travel times are lower when using the new approach.



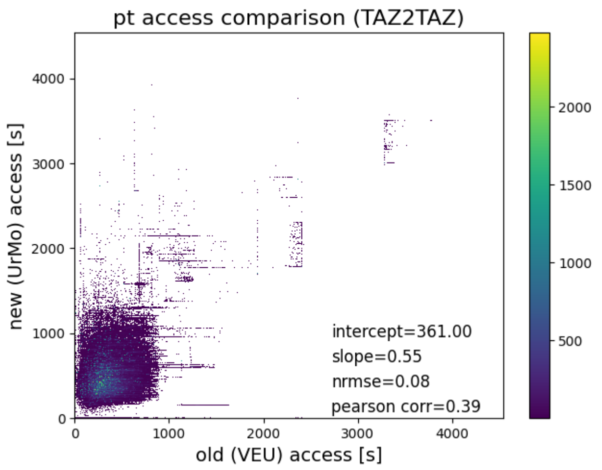



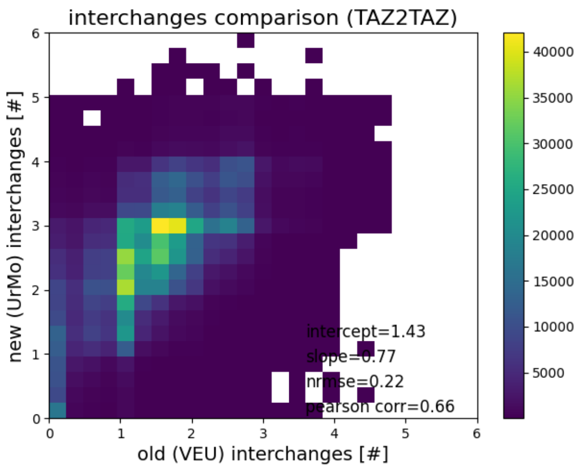
# Results Comparison
## Public Transport (3)

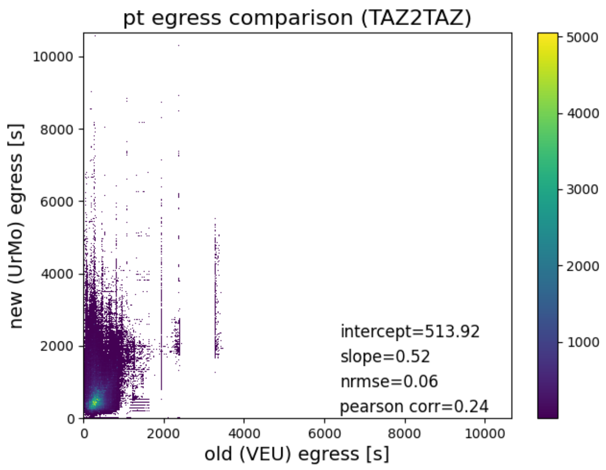
- Additional PT values show higher differences
  - High access times (>3000 s) – though in both, new and old, methods
  - Partially higher numbers of intersections in the new attempt, though slope <1
  - Too high egress times in the new attempt

→Further research (evaluation of real-world data and models) necessary

→Additional limits for access and egress



# Next Steps

- Improvement of computing public transport measures
- Including slopes and traffic lights when computing walking and using a bicycle
  - Both are implicitly regarded for MIT as travel times from simulation runs are read

# Thank you!

## Any questions?

Please find UrMoAC at:
https://github.com/DLR-VF/UrMoAC

Please find TAPAS at:
https://github.com/DLR-VF/TAPAS

# Impressum

Topic: **Computation of mode-dependent travel time matrices for an agent-based demand model computed using a standalone accessibility tool**

Date: 23rd-25th of April 2024

Authors: Daniel Krajzewicz, Alain Schengen

Institute: Institute of Transport Research

Images by: DLR, ESA