Ricardo
Energy & Environment

Application of Real-World Driving Emissions for Air Quality Modelling and Impact Assessments

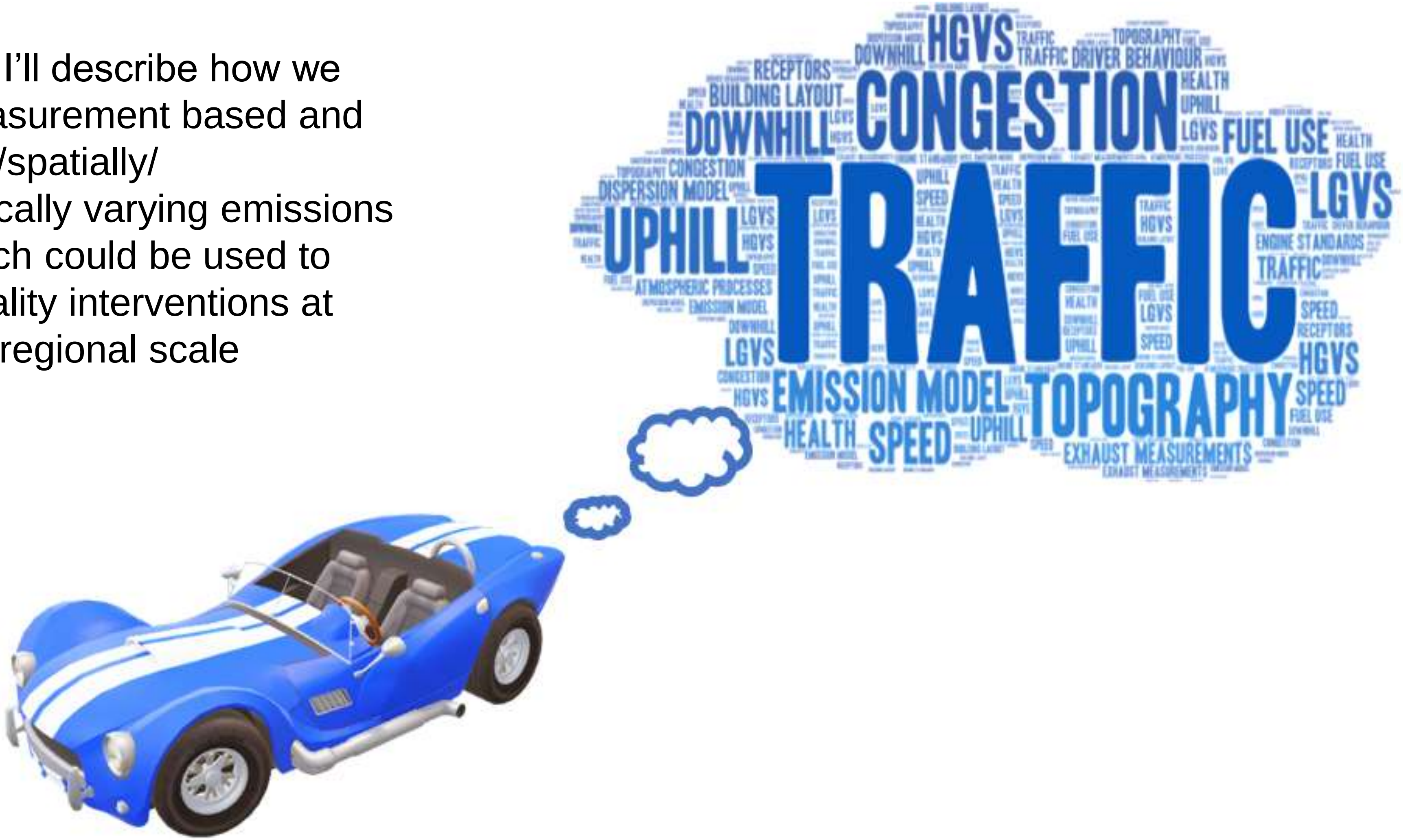
Nicola Masey, Scott Hamilton
Air Quality Modelling Group, Ricardo
12th February 2019

Introduction

- Traditional air quality models predict annual average concentrations
- Increasing availability of microsimulation traffic models – detailed traffic info lost during processing in AQ studies
- UK Government Plans to tackle air pollution focus on mitigation measures to reduce pollution, some of which are subtle when temporally aggregated to annual data
- Drive to model high spatial and high temporal resolution concentrations (but this is complex and hard to do reproducibly….)
- Emission models are never validated in the local domain. We did that here, in the process developing new methods we can reuse in future projects.
- We can now model city and regional scale cases, taking account of all contributing factors in generating traffic emissions, which can be freely varied in time and we can validate emissions with measurement surveys- this is quite novel in the UK context.

Traffic emission modelling for complex cases using programming

In this talk I'll describe how we built a measurement based and temporally/spatially/ technologically varying emissions model which could be used to test air quality interventions at micro and regional scale



Python’s big hitters for air quality science

The fundamental libraries for us (and how we use them) are:

Numpy- creating and manipulating arrays, e.g. emission grids, billions of data points can be processed, VERY fast

Pandas- built on top of Numpy- reading and processing 2D data tables, e.g. traffic activity observation processing

SciPy – huge collection of scientific modules- we use the image processing library to process emission grids

GDAL – every geospatial function you could ever want, we use it for processing spatial data in dispersion models

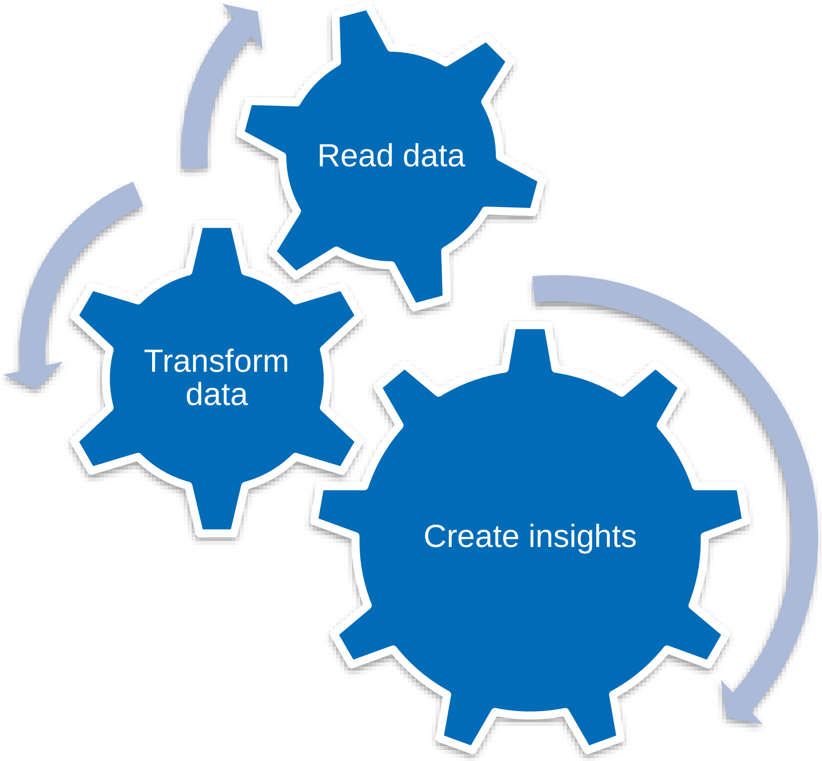
Honourable mentions go to
Anaconda- excellent python distribution/package manager

Jupyter- the ‘Notebook’ is becoming the benchmark for prototyping and sharing scientific code

Matplotlib- fundamental plotting library

Advantages of programmatic data processing (Python particularly)

- **Readability-** python is very easy to read, so QA is easy
- **Flexibility-** the language has thousands of libraries tuned to all manner of data manipulation problems
- **Reproducibility-** complex data manipulation routines can be checked and recreated by third parties
- **Reusability-** a coded solution should be a reusable solution. Intellectual capital can be built up over time and reused
- **Efficiency-** compared to other languages, python offers a rapid development cycle. The 'time to solution' is minimised



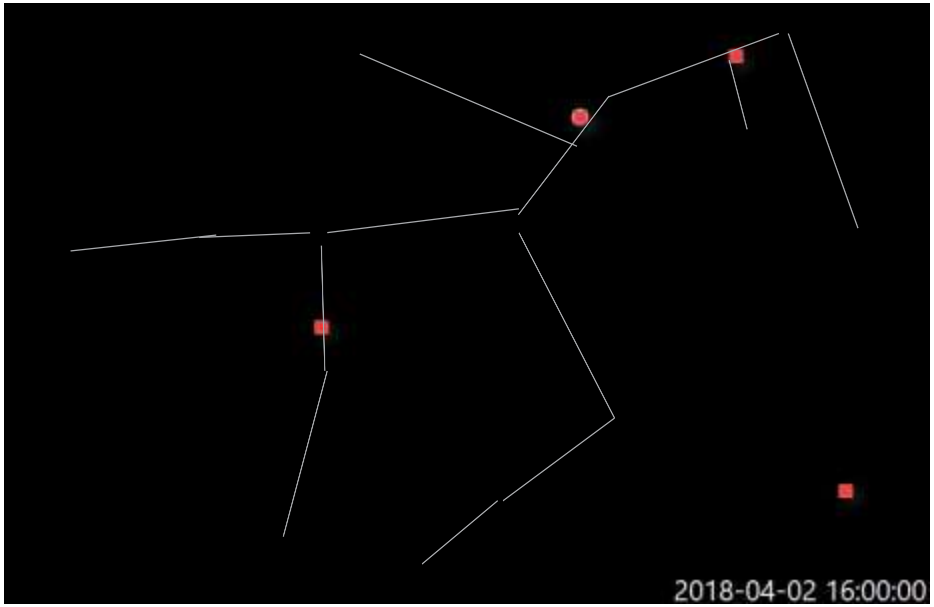
Neat things can be done with a little programming

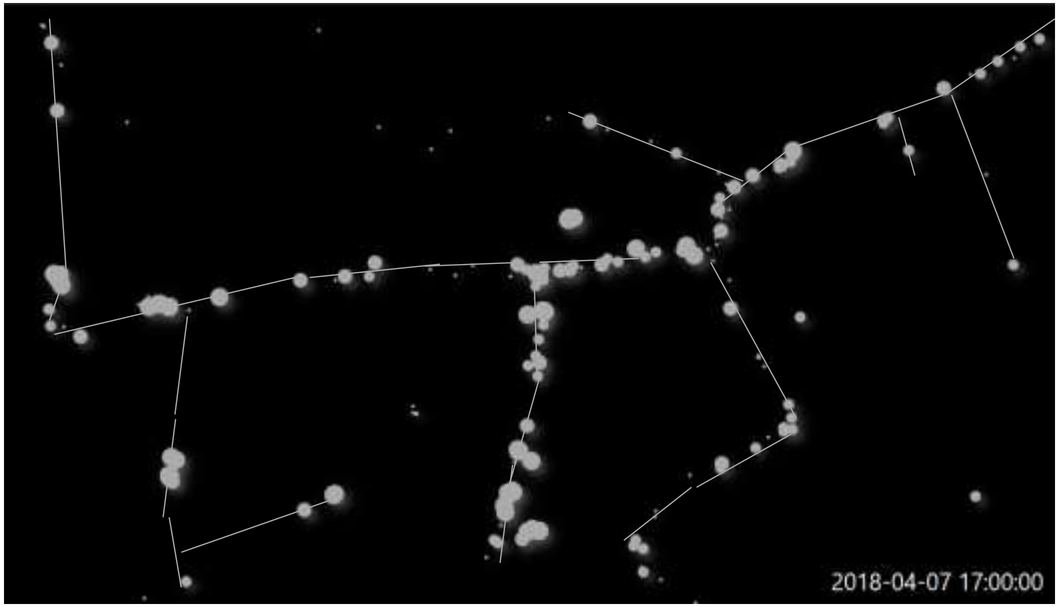
Spatial allocation of vehicle emissions from GPS data

- Diagnose the source emission signal from local government vehicles transiting small towns during business hours
- Key things we need:

1) Processing of GPS measurements
2) Computation of emissions for each time step and vehicle
3) Spatial allocation of emissions to help inventory compilation
4) More modular approach with scripts 'talking' to each other

- Added significant value to an existing dataset (GPS) - Able to spatially allocate emissions from the local authority fleet to help decision making, e.g. *"is it worth buying new LGVs?"*

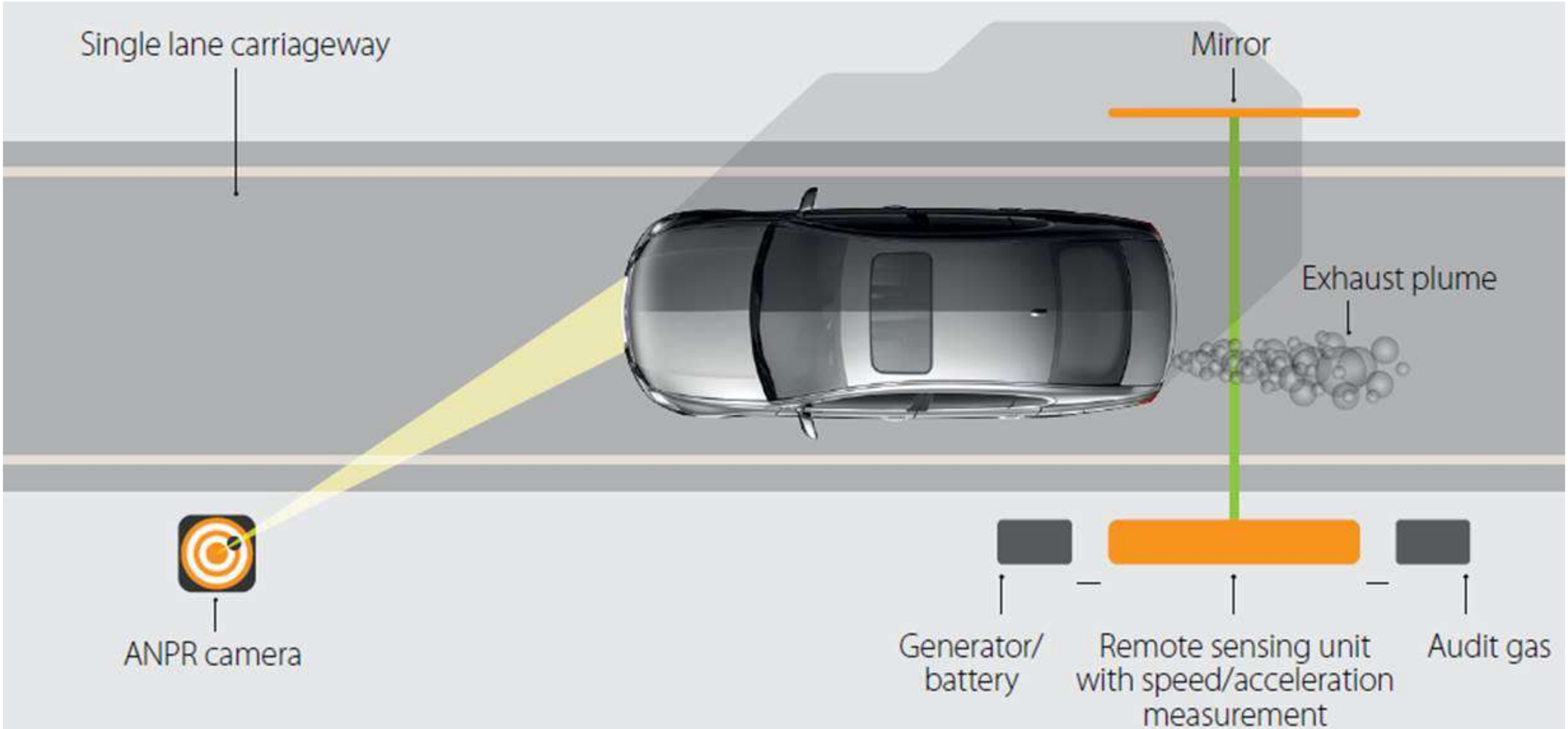




Application of real-world driving emissions


Things have moved on since our first vehicle emissions testing over 50 years ago
D.S.I.R.
DIESEL SMOKE EXPERIMENTS
WARREN SPRING LABORATORY

OPUS set up


Single lane carriageway
Mirror
Exhaust plume
ANPR camera
Generator/ battery
Remote sensing unit with speed/acceleration measurement
Audit gas

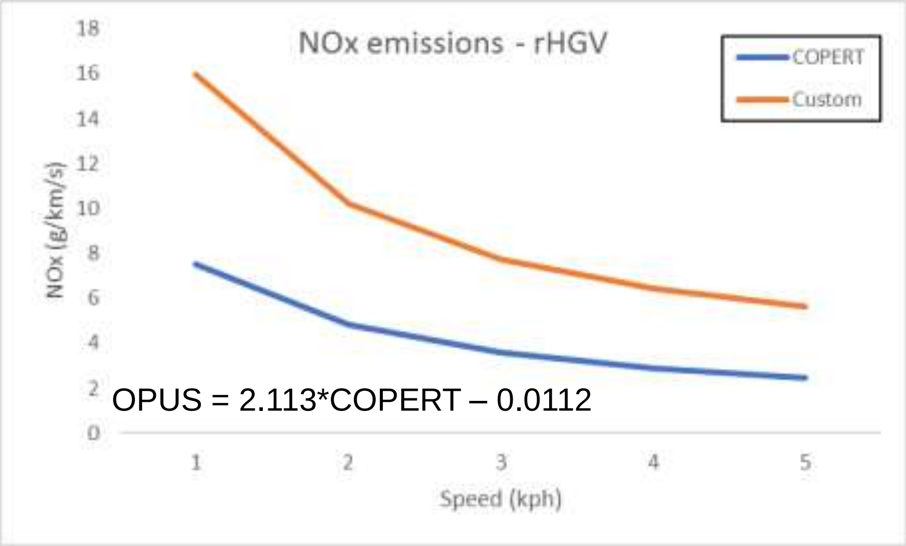
Consolidating remote sensing data with air quality models

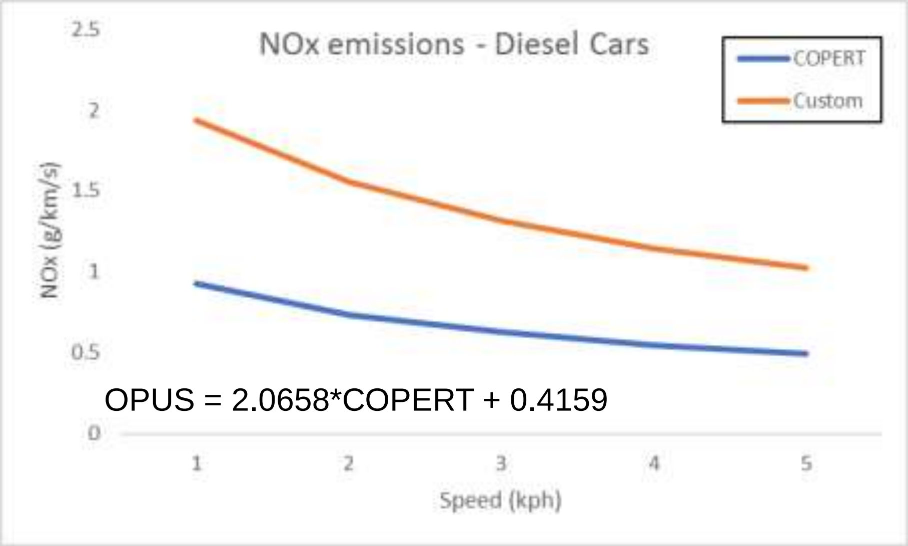
Remote Sensing

1. **Interpret** — Statistical analysis of data from the measurement campaign
2. **Compare** — Placing the measurements in the context of current local evidence- compare with local emissions models
3. **Localise** — Tune local scale emission models to reflect the new evidence. We can change coefficients in for example COPERT
4. **Diagnose** — Create/update air quality models with the localised emissions. RapidAIR is a good candidate model to make use of RS data given its speed and efficiency

   'Time to interpretation' and indeed action is reduced

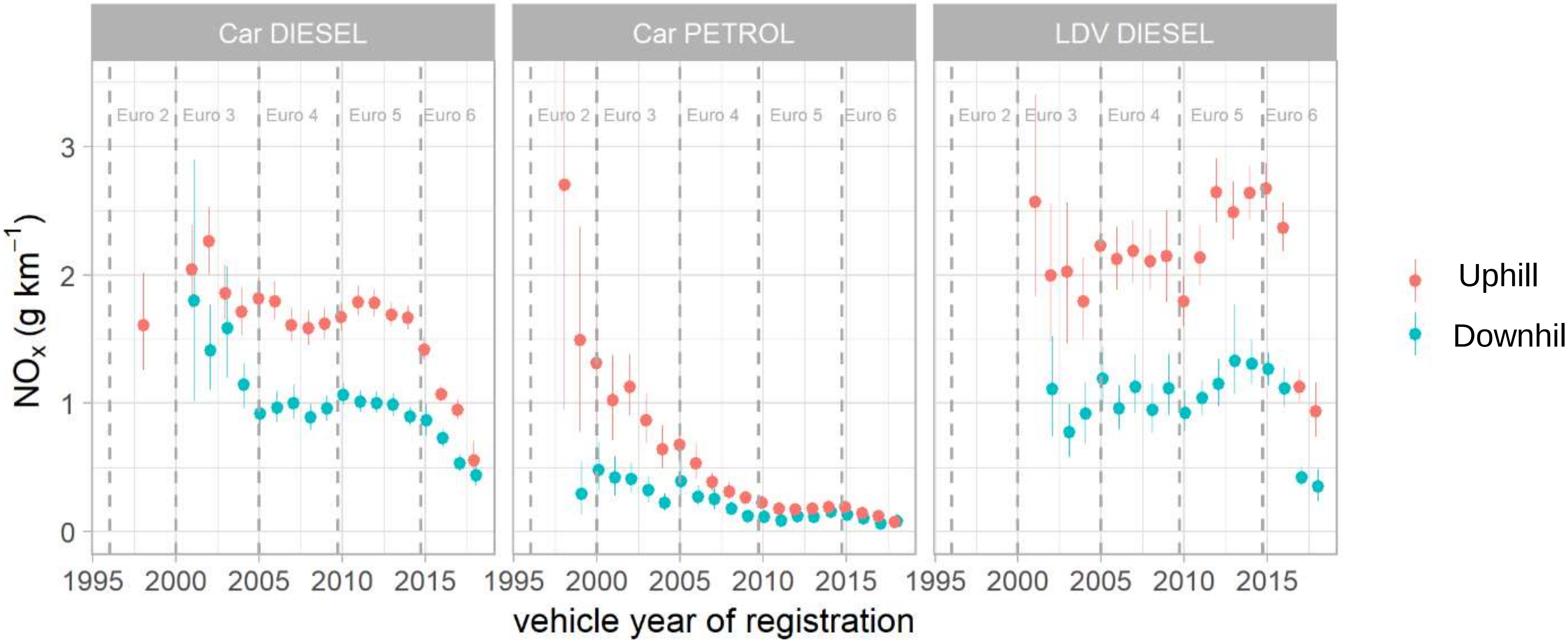
OPUS vs COPERT emissions





- These are only a few examples, our measurements suggest that every type of vehicle in COPERT was underpredicted in this domain.
- This could be due to a mix of local factors such as topography but also inherent differences in emissions from individual engine and fuel types.

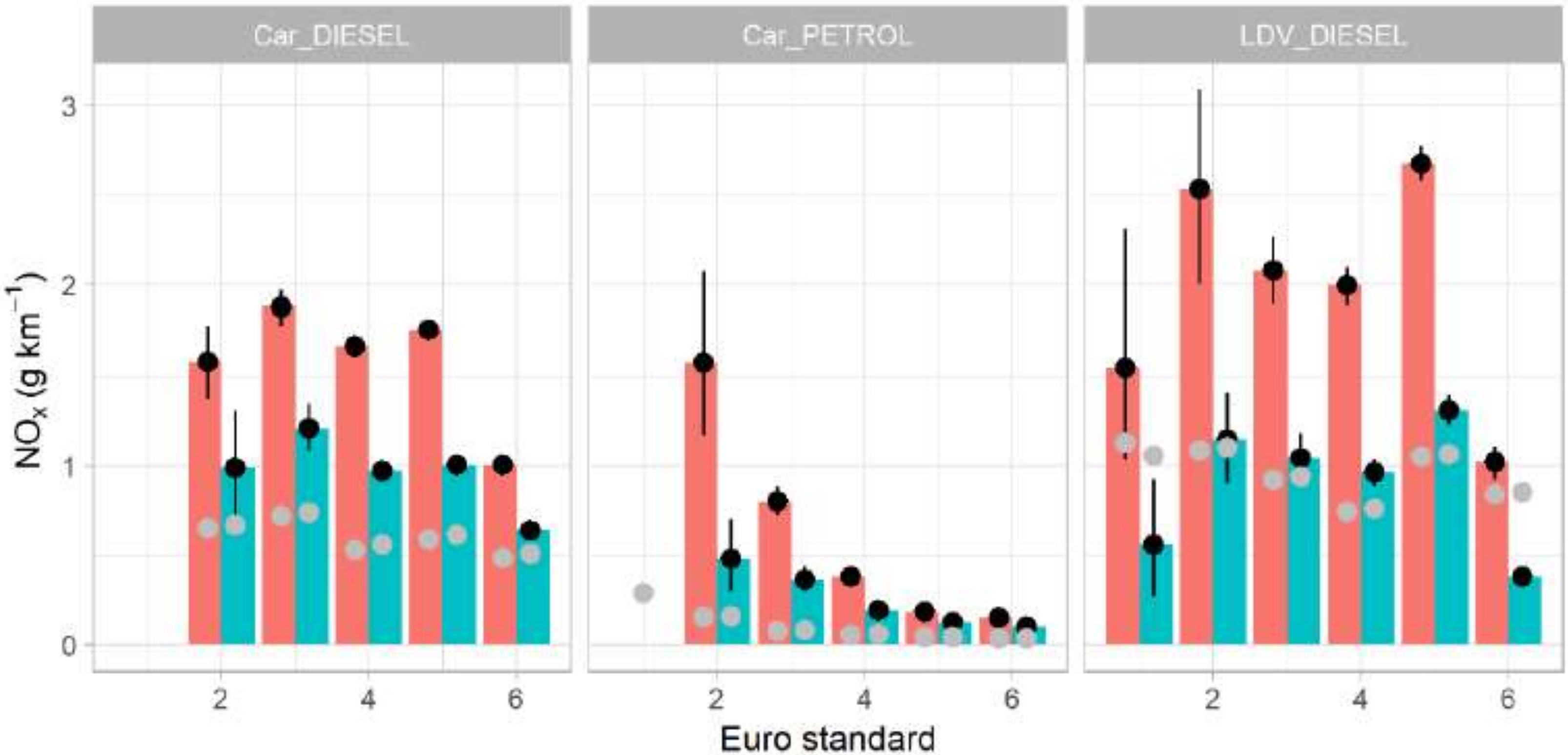
Emissions uphill vs downhill



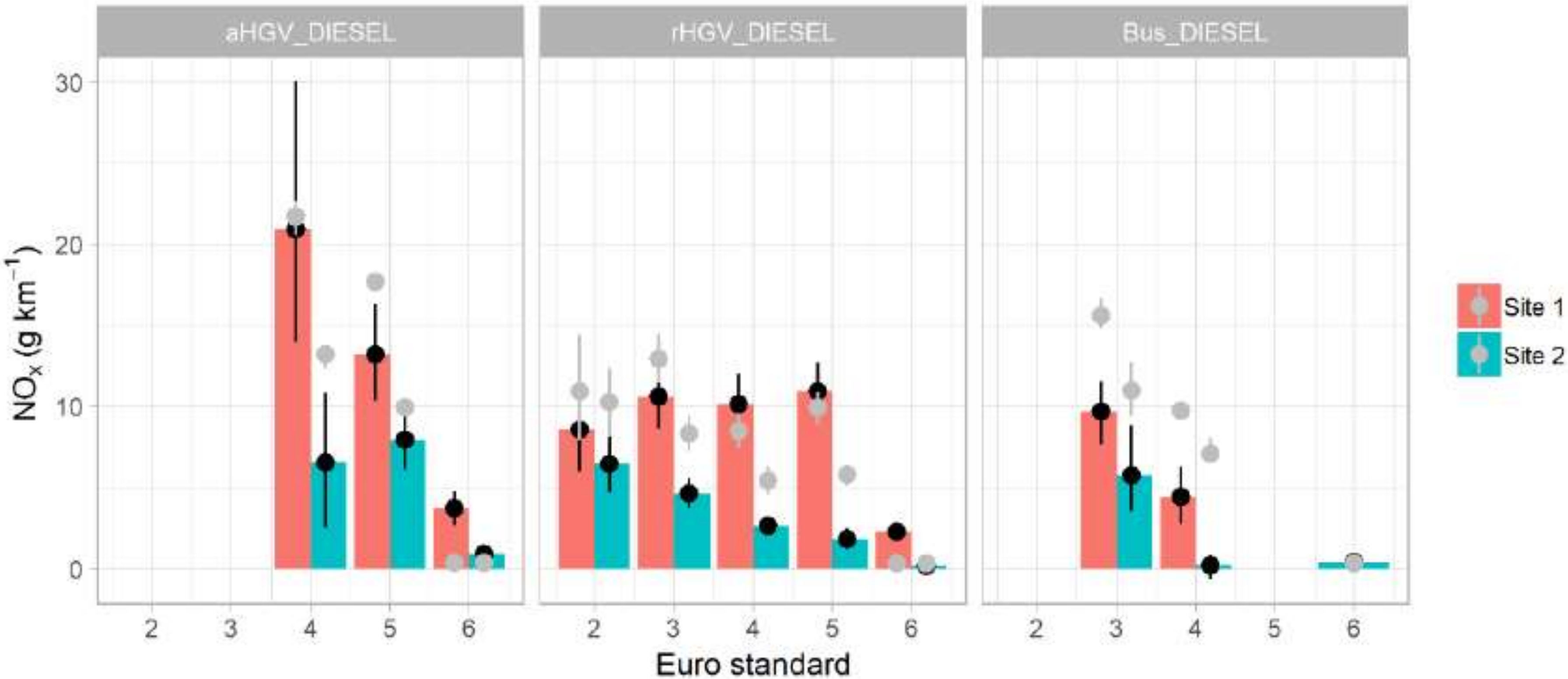
This factor becomes additionally important when receptors on one side of the road are closer to an uphill travelling fleet- the concentration response can be asymmetrical.

OPUS emission rates

- Emission rates for each vehicle type measured locally using OPUS were used in preference to the national COPERT V emissions factors.

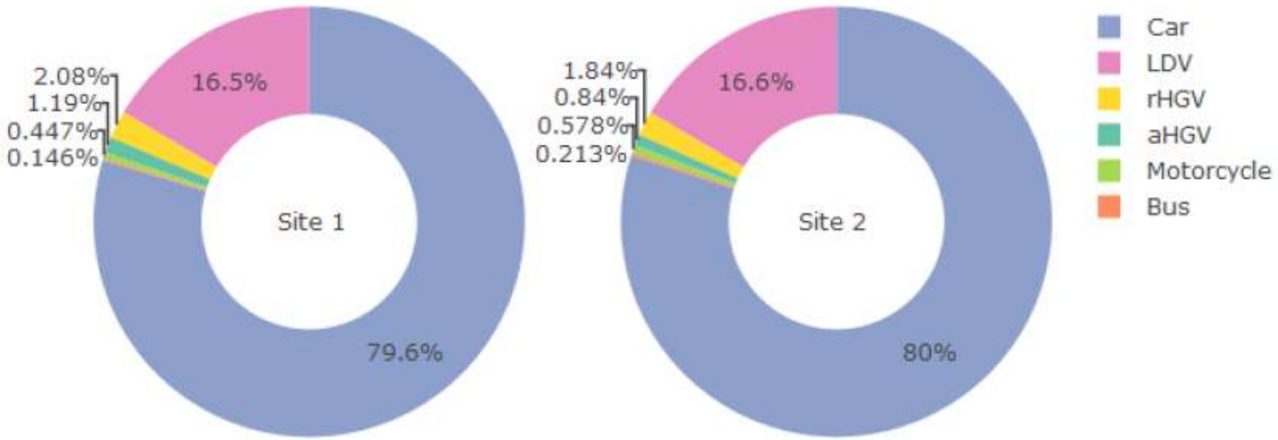


OPUS emission rates



OPUS data input to emissions model

- Data from the OPUS measurements were used as inputs to the emission model:
  - Fleet composition
  - Engine class
  - Speed
- We assume that COPERT does a decent job of characterising the shape of emission curves, but has an inherent coarseness when representing very local circumstances. For example, in COPERT an 18 tonne Euro V bus has the same emissions in Scotland as in every other EU country…. Realistic?



Modelling traffic from microsimulation models

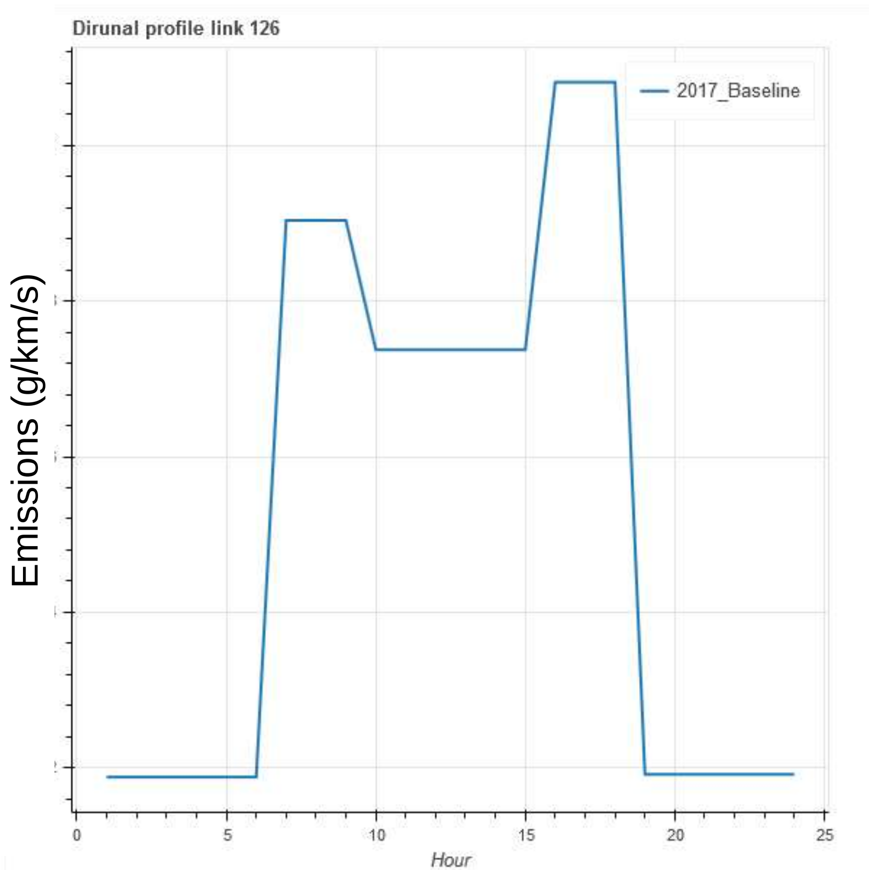
Microsimulation traffic models

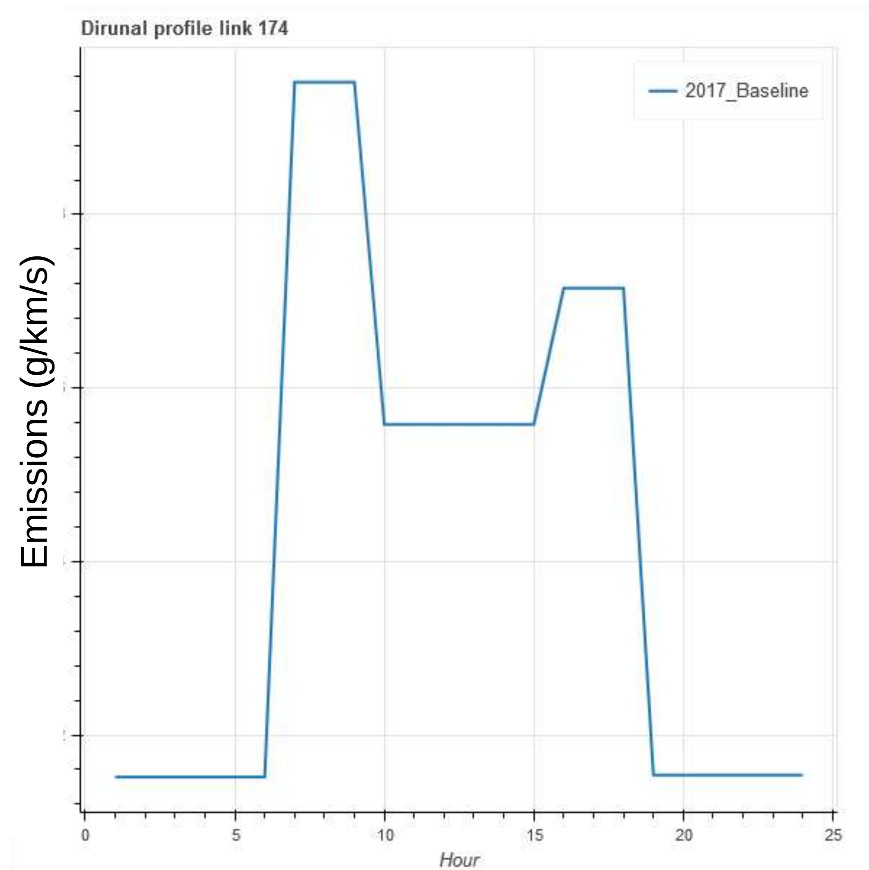
- Microsimulation traffic models provide detailed (hourly) information about speed, flows and vehicle compositions on road links
  - higher spatial and temporal resolution
- Info about mitigation scenarios e.g. peak-time vehicle bans can be included in microsimulation traffic model
- AQ modelling typically models annual concentrations – aggregation of hourly to annual data dilutes temporal information from traffic model
- Emissions for each hour modelled for each link – i.e. each link had its own flow and speed (and in some cases, technology mix)- this was a significant undertaking involving a lot of python coding to control our emissions model
- No loss of information between traffic model and modelled emissions due to aggregation of data.

Hourly emissions processing

- Emission rates from local OPUS measurements were used – unique emissions for uphill and not-uphill applied to appropriate links
- RapidEms, Ricardo's in-house emissions module, can easily integrate the localised emission rates into the emissions model
- Controlled using programming in Python
  - Reproducible- important as we ran our emission model several thousand times- impossible to manage with tools like the Emission Factor Toolkit.
  - Changes can be introduced globally (e.g. default values, scaling factors)
  - Less user intensive and faster
  - QA is much easier- only a few places things can go wrong (typically deep in a massive script but at least not in a 100,000 row Excel spreadsheet)
- Unique dispersion model ready hourly emissions for >1.1 million link situations calculated in ~3 hours. Including terrain, real world emissions, temporal changes, intervention measures etc etc. Traditional methods simply would not cut it.

Diurnal emissions from traffic models





Each link can have it’s own flow, speed, and fleet
i.e. unique emissions profiles for each link

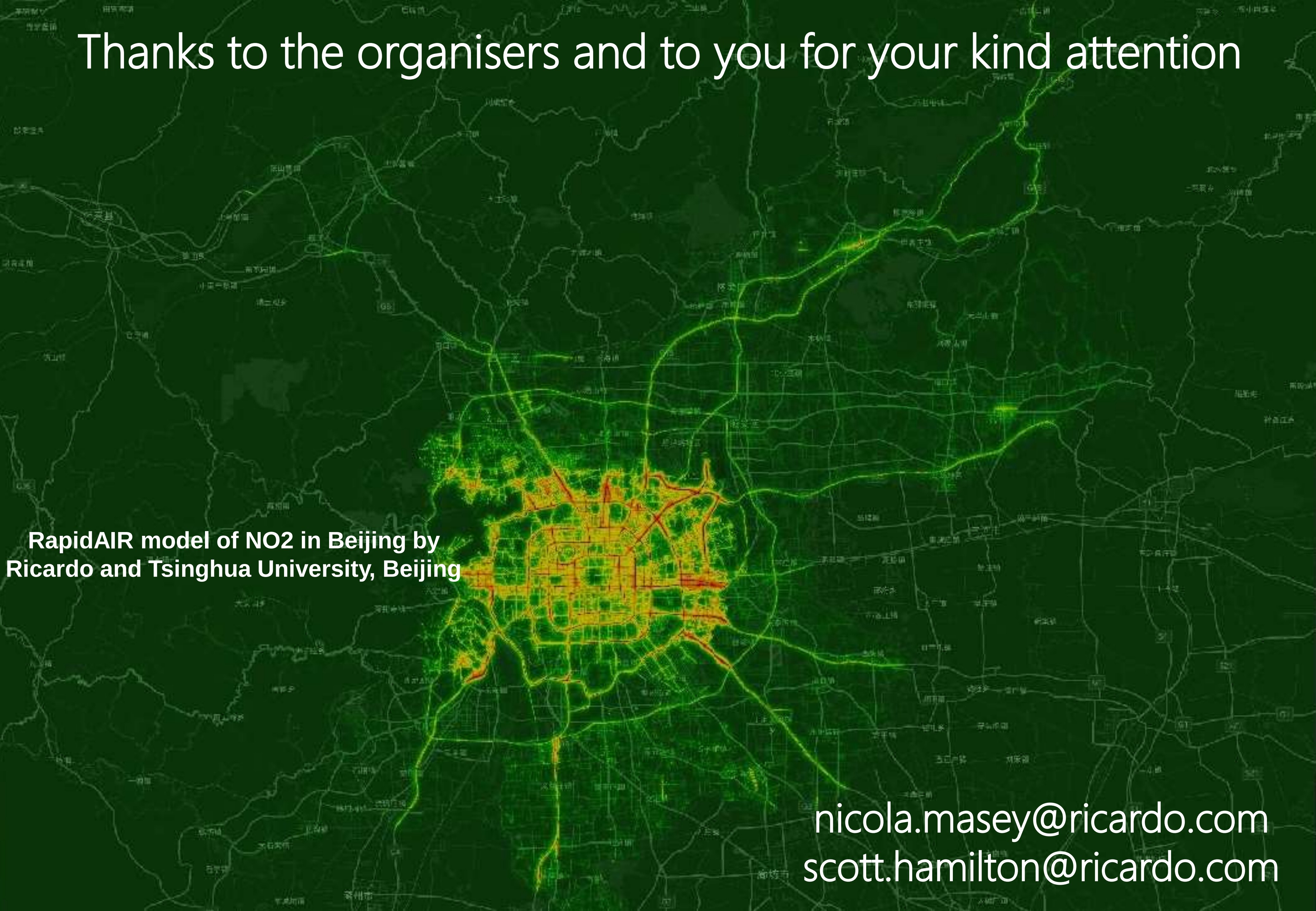
Conclusions

- Real-world driving emissions used in preference to COPERT emissions
  - Real-world emissions typically double those in COPERT
  - Large differences between uphill and non-uphill emissions
- Detail in microsimulation traffic models often lost during temporal aggregation
- Method developed to retain information from the traffic model in emissions calculations
  - Each link assigned unique flow, speed and fleet
  - Emissions calculated hourly
- Allows more thorough investigation of scenarios e.g. peak-time bans

Future Work

- Detailed modelling of hourly emissions in dispersion models (GRAL and RapidAir)
  - Hourly estimates of concentrations

Thanks to the organisers and to you for your kind attention



RapidAIR model of NO2 in Beijing by Ricardo and Tsinghua University, Beijing

nicola.masey@ricardo.com

scott.hamilton@ricardo.com

Nicola Masey and Scott Hamilton

Nicola.Masey@ricardo.com
Scott.Hamilton@ricardo.com