Application: Co-design of future mobility systems

- **Now**: Co-design of vehicle and future mobility systems
  - Co-Design to Enable User-Friendly Tools to Assess the Impact of Future Mobility Solutions
  - Co-Design of Embodied Intelligence: A Structured Approach

- **Takeways**:
  - Using co-design, it is easy to formalize **hierarchical models** (never possible before)
  - Very **intuitive** modeling approach (no “acrobatics” needed)
  - **Rich modeling capabilities**: analytic models, catalogues, simulations
  - **Compositionality** and **modularity** allow **interdisciplinary collaboration**
  - Co-design produces **actionable information** for designers to **reason** about their problems
Co-design of vehicle and future mobility systems

- Co-Design to Enable User-Friendly Tools to Assess the Impact of Future Mobility Solutions
  - We co-design **intermodal mobility solutions** (AVs, micromobility) with the **infrastructure** (trains, roads)

- Co-Design of Embodied Intelligence: A Structured Approach
  - We co-design an AV, all the way from **hardware** (vehicle, sensors, computers, ..) to **software** (perception, control, ..) components

```
Simple approach  compositionality!  Complex approach
```

```
Moabhy system

- - - investments [USD]
- - - service level [s]
- - - externalities [kg]

Vehicle
- - - operational cost [USD/m]
- - - fix cost [USD]
- - - externalities [kg/m]
- - - autonomy performance

speed [m/s]
```

```
demand to satisfy
```
Co-design of future mobility systems

- We look at the problem from the perspective of **municipalities** and **policy makers**
  - Important decisions to make:
    - *How many AVs should we allow?*  
    - *What's the influence of AVs on public transit systems?*
    - *How performant should AVs be?*  
    - *How many trains should we buy?*
Co-design of future mobility systems

- We look at the problem from the perspective of **municipalities** and **policy makers**
  - Important decisions to make:
    - How many AVs should we allow?        What’s the influence of AVs on public transit systems?
    - How performant should they be? How many trains should we buy?

- Existing work only solves **specific problems** and does not **co-design** the whole system:
  - No joint design of **mobility solutions** and the **system** they enable
  - No **modularity** and **compositionality**: problem-specific
  - Often, not producing **actionable information** for stakeholders
Co-design of future mobility systems

- We look at the problem from the perspective of **municipalities** and **policy makers**
  - Important decisions to make:
    - *How many AVs should we allow?*
    - *What’s the influence of AVs on public transit systems?*
    - *How performant should they be?*
    - *How many trains should we buy?*

- Existing work only solves **specific problems** and does not **co-design** the whole system:
  - No **joint** design of **mobility solutions** and the **system** they enable
  - No **modularity** and **compositionality**: problem-specific
  - Often, not producing **actionable information** for stakeholders

- Several **disciplines** involved (transportation science, autonomy, economics, policy-making)

- We allow **interfaces** between them via **co-design**:
  - **Functionality**: demand to be satisfied
  - **Costs**: investments ($), externalities (CO$_2$ kg), **service level** (average waiting time, s)

- Co-design highlights the **structure** of the problem and provides **tools** to reason about it
Modeling the mobility system as a co-design problem

Mobility simulations (optimizing flow allocation on a network)

Subway:
- **Fun**: number of trains to buy
- **Res**: costs and externalities
- **Imp**: acquisition contracts

Micromobility:
- **Fun**: speed of the vehicle
- **Res**: costs and externalities
- **Imp**: vehicle models

AV:
- **Fun**: speed of the vehicle
- **Res**: costs, externalities, performance
- **Imp**: vehicle models and autonomy
Co-Design produces actionable information for stakeholders

Fixed a demand, we find the Pareto front of incomparable, minimal solutions as cost, time, and externalities

Convert externalities into cost and interpret the results:

Which one is the best? Depends on what is at upper level (policy-making, etc.)
Co-design of an autonomous vehicle

- Simple approach: a **catalogue** of existing **AVs**
- We want to model **AVs** more in detail, from the perspective of the **developers**
- We look at an example of the **methodology** to apply:
  - Can be applied to other autonomous systems
  - *Proof of concept* implementation (*no* real data)
Co-design of an autonomous vehicle

- Simple approach: a catalogue of existing AVs
- We want to model AVs more in detail, from the perspective of the developers
- We look at an example of the methodology to apply:
  - Can be applied to other autonomous systems
  - *Proof of concept* implementation (*no* real data)

**Modeling approach:**
- **Task** - *what do we need to do?*
- **Functional decomposition** - *how to decompose?*
- **Find components** - *decompose until you find components* (hardware and software)
- **Find common resources** - For instance, *size, weight, power, computation, latency*

**Implementation:**
- **Skeleton** - *write structure using the formal language*
- **Fill-in the holes**: *catalogues, analytic models, simulations*
Task abstraction and functional decomposition in autonomy

- Autonomy tasks can be usually characterized as a **design problem**:
Task abstraction and functional decomposition in autonomy

- Autonomy tasks can be usually characterized as a **design problem**:

  - Given the **sub-tasks**, we can interconnect them

  ![Diagram showing task abstraction and functional decomposition](image)

  - Note that composing tasks gives a task (compositionality)
Task abstraction and functional decomposition in autonomy

- Autonomy tasks can be usually characterized as a **design problem**: Given the sub-tasks, we can interconnect them.

- Note that composing tasks gives a task (**compositionality**).

- Let’s try with **urban driving**: lateral control (maintain lane position), longitudinal control (brake in case of obstacles).
Co-design model of an autonomous vehicle

Catalogue of vehicles to be automated

Catalogue of discomfort models

Catalogue of computers

Longitudinal control

Lateral control

Computing

Vehicle

Discomfort

Operational cost [CHF/km]

Energy externalities [g/km]

Fix cost [CHF]

Discomfort [kg \cdot m/s]

Total mass [g]

Total power [W]

Total computation [op/s]

Capacity [pax/car]

Range [m]

Speed cruise [m/s]

Dynamic performance [m/s \cdot m^2/s \cdot m/s^2]

Cost [CHF]

Power [W]

Mass [g]

Control effort

Discomfort brake

System noise

Environment

Bound

Range [m]

Catalogue of vehicles to be automated

Catalogue of discomfort models

Catalogue of computers
Co-design model of an autonomous vehicle

Longitudinal control

- latency [s]
- cost [CHF]
- mass [g]
- power [W]
- dynamic performance [m/s, m/s², m/s³]
- speed cruise [m/s]
- environment
- sensing performance
- acquisition frequency [Hz]
- implement brake control at δ [Hz]
- computation [op/s]

Lateral control

- feature extraction
- resolution [px/sterad]
- implement feature detection at δ [Hz]
- tracking error
- cost [CHF]
- mass [g]
- power [W]
- dynamic performance [m/s, m/s², m/s³]
- environment
- sensing performance
- acquisition frequency [Hz]
- implement lane control at δ [Hz]
- computation [op/s]
Co-design of a intermodal mobility system

- Lateral control can be decomposed in **sub-tasks**:

```
Lane control

Lane cameras

Implement lane control

Feature extraction

Implement feature detection

Computation of LQG solutions in closed-form
```

**Catalogue of extractors**

Implement feature extraction at $\delta$ [Hz]

Resolution [px/sterad]

Information observation

**Catalogue of algorithms**

Computation [op/s]

Tracking error control effort

Computation [op/s]

Cost [CHF]

Mass [g]

Power [W]

**Catalogue of sensors**

System noise

Environment

Observe at $\delta$ [Hz]
Co-design of a intermodal mobility system

- Longitudinal control can be decomposed in **sub-tasks**:
  
  **Brake control**
  - Implement brake control
  - Dynamic performance
  - Discomfort brake
  - Danger

  **Longitudinal sensing**
  - Sensing performance
  - Acquisition frequency

  **Longitudinal control**
  - Environment
  - Speed cruise
  - Latency

  **Catalogue of sensors**
  - Latency [s]
  - Cost [CHF]
  - Mass [g]
  - Power [W]
  - Dynamic performance [m/s · m/s² · m/s²]

  **Catalogue of algorithms**

  **Simulations of a POMDP**
We construct a poset of sensor functionalities

- Sensing performance:

![Sensor Performance Curves](image)

![FP Poset](image)

![FN Poset](image)

![Accuracy Poset](image)
Co-design of an intermodal mobility system

- The theory comes with a formal language and a solver (MCDP)

- Very intuitive to use:

```plaintext
mcdp {
    provides computation [op/s]
    requires cost [CHF]
    requires mass [g]
    requires power [W]
}

choose(
    SedanS: (load Car_SedanS),
    SedanM: (load Car_SedanM),
    SedanL: (load Car_SedanL),
    SuvS: (load Car_SuvS),
    SuvM: (load Car_SuvM),
    Minivan: (load Car_Minivan),
    Shuttle: (load Car_Shuttle),
    Hybrid: (load Car_Hybrid),
    Bev: (load Car_Bev)
)
```

Choose query type:
- Fixed the functionality, minimize the resources.
- Fixed the resources, maximize the functionality.
- Given an implementation, evaluate functionality/resources. [UI not implemented]
- Given min functionality and max resources, determine if there is a feasible implementation. [UI not implemented]
- Given min functionality and max resources, find a feasible implementation. [UI not implemented]
- "Solve for X": find the minimal component that makes the co-design problem feasible. [UI not implemented]
Co-design model of an autonomous vehicle

Catalogue of vehicles to be automated

Catalogue of discomfort models

Catalogue of computers
Solution of DPs

range [m]  
capacity [pax/car]  
speed cruise [m/s]  
environment  

operational cost [CHF/km]  
energy externalities [kg/km]  
fix cost [CHF]  
danger [kg · m/s]  
discomfort  

total computation [op/s]  
total mass [kg]  
total power [W]  
system noise  
latency [s]  

operational cost [CHF/km]  
energy externalities [kg/km]  
fix cost [CHF]  
danger [kg · m/s]  
discomfort  

AV
Solution of DPs

Details of autonomy, both hardware and software

Fix functionalities, look at minimal cost and discomfort
Solution of DPs

Fix **functionalities**, look at minimal **cost** and **discomfort**

Details of autonomy, both **hardware** and **software**

Monotonicity: Higher achievable speeds will not require less resources
Conclusions: Takeaways

‣ Using co-design, it is easy to formalize hierarchical models (never possible before)
We formalized mobility systems all the way from sensors on the vehicles to interactions of fleets of AVs with the public infrastructure of a city

‣ Very intuitive modeling approach (no acrobatics like common in optimization theory)
The interpreter allows one to easily model problems of interest

‣ Rich modeling capabilities:
    Simulation: Flow optimization for mobility network, POMDP for brake control
    Catalogues: Sensors, vehicles, computers, algorithms, ...
    Analytical: LQG closed-form solutions, discomfort models, ...

‣ Compositionality and modularity allow interdisciplinarity
We did all of it, but technically this could have been possible with different teams

‣ Co-design comes with a formal language and an optimizer
After easily modeling the problem, you can directly solve queries of your choice

‣ Co-design produces actionable information for designers to reason about their problems
We have shown actionable information for municipalities, as well as for AV developers