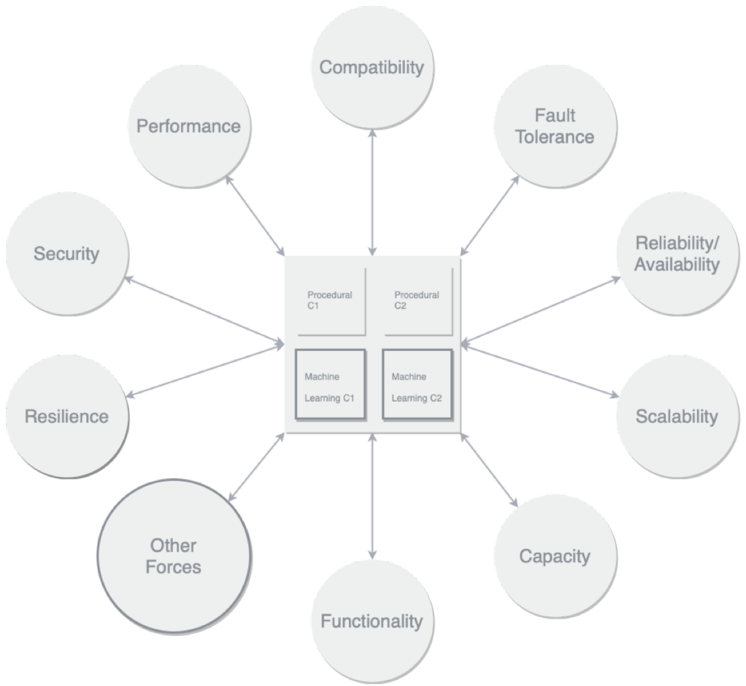
Towards using probabilistic models to design software systems with inherent uncertainty

Alex Serban, Erik Poll, Joost Visser

Architecture design

- Software architecture design is a problem that deals with a resolution of forces
- All forces can not be satisfied optimally - in most cases the final solution involves *trade-offs* between different forces
- The question we ask is which are the forces that drive architectural *design* when using machine learning (ML) components?
- The core differences between ML components and other components are: (1) it is not possible to verify that ML components will always *satisfy their intended functionality*, and (2) it is not possible to verify that ML components can *cope with stochastic event during operation*

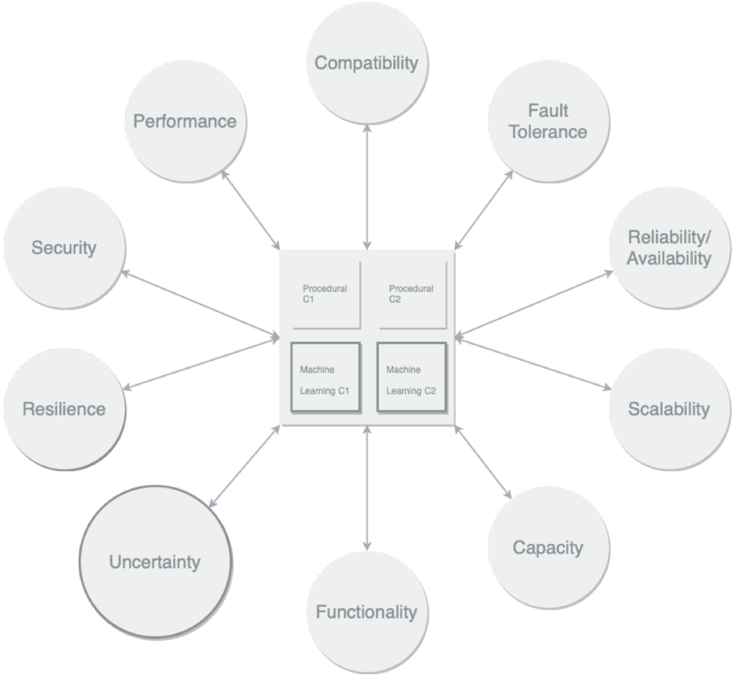


Architecture design

- The question that we ask is which are the forces that drive architectural design when using machine learning (ML) components?

- The core difference between ML components and other component are: (1) it is not possible to verify that ML components will always *satisfy their intended functionality*, and (2) it is not possible to verify that ML components can *cope with stochastic event during operation*

- These differences are due to *inherent uncertainty* in ML components



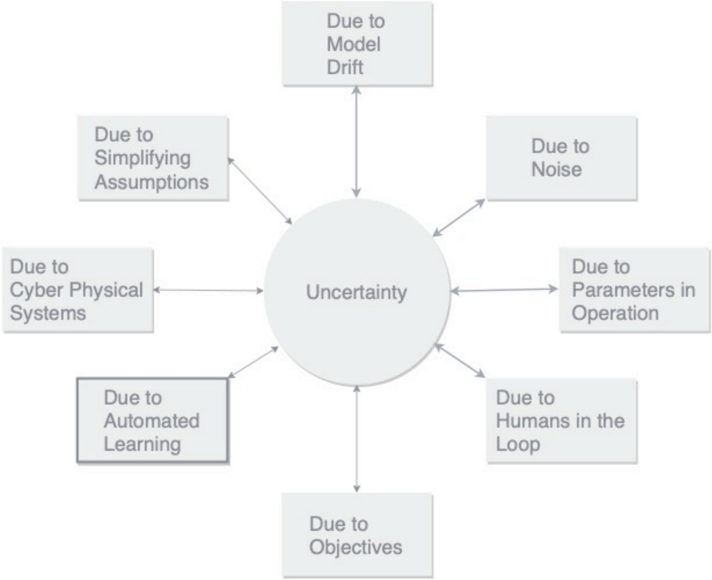
Uncertainty in software architecture

- Uncertainty is not a new concept in software architecture

- Many types of uncertainty have been tackled at *design* time, or at *run* time (self-adaptation)

- However, the uncertainty related to *automated learning* has so far been tackled only at run time (and not to a great extent)*

- In many case decisions have to be made at *design time* (e.g., autonomous vehicles – SOTIF)



*Mahdavi-Hezavehi Sara, Avgeriou Paris, Weyns, Danny - A Protocol for a Classification Framework of Uncertainty in Architecture-based Self-Adaptive Systems with Multiple Quality Requirements

Taxonomy by Perez-Palacin, Diego Mirandola, Raffaela,- Uncertainties in the modeling of self-adaptive systems: A taxonomy and an example of availability evaluation

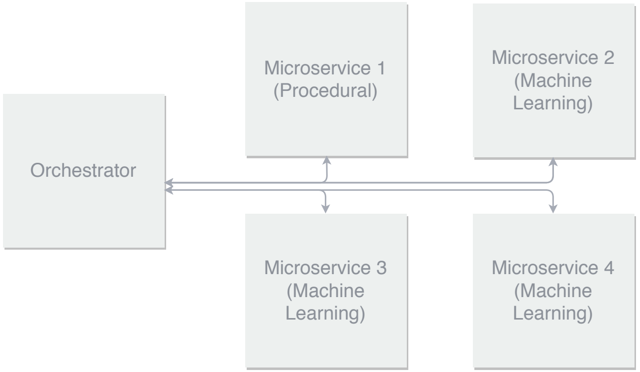
Modeling uncertainty during design

- We propose to elevate uncertainty due to automated learning at design time, and use it as an architectural decision driver (e.g., as a way to divide / orchestrate microservices)

- And use a modeling approach that can:

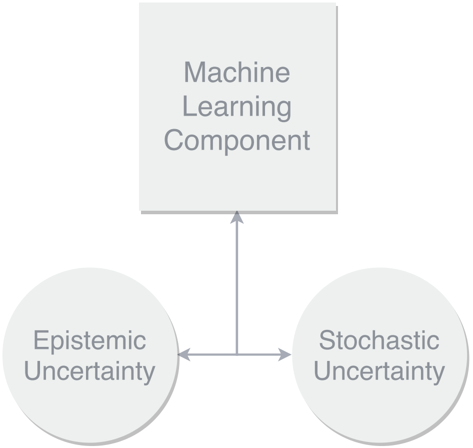
    - Take into account the fact that at design time the uncertainty estimates can be subject to change (i.e., the prior information about a component is incomplete)

    - Evaluate uncertainty locally (as it impacts one component) and globally, as it propagates through a system



Types of uncertainty

- All ML algorithms are subject to two types of uncertainty:
  - *Epistemic* uncertainty - captures our ignorance of the correct model that generates the data. This uncertainty can be removed given enough training data
  - *Stochastic* uncertainty - captures the response of a ML component to stochastic events in the operational environment (e.g., noise in the observations).
- We propose to use these 2 types of uncertainty as design elements



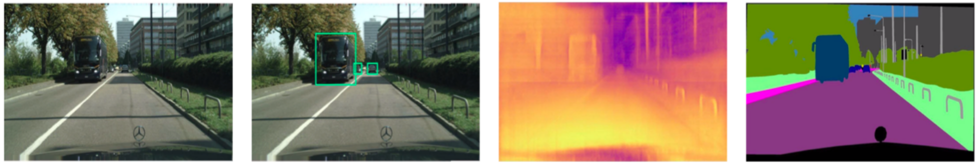
Use case

- In order to validate if uncertainty can be used at design time, we selected a perception system for autonomous vehicles

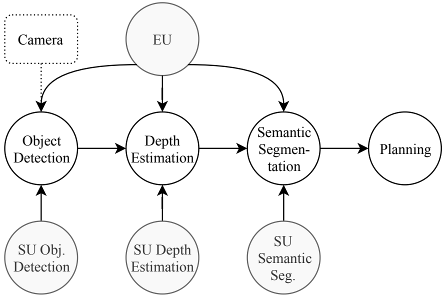
- The system performs there tasks which can only be implemented using ML

- For this system we have selected two architectures from literature

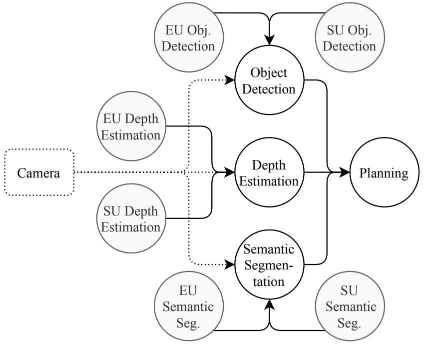
- We propose to explicitly model the two types of uncertainty in the architecture (i.e., to annotate the architectures)



(a) Camera Input (b) Object Detection (OD) (c) Depth Estimation (DE) (d) Semantic Segmentation (SS)



End-to-end architecture



Component-based architecture

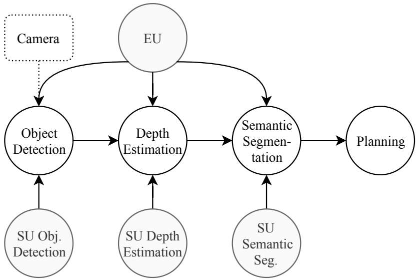
Implementation details

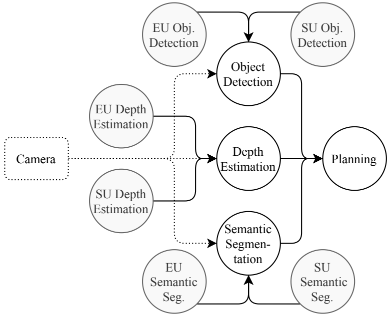
- The end-to-end architecture is implemented with a variant of MultiNet, which shares a base encoder for all tasks and has different individual decoders
    - All components have the same epistemic uncertainty (EU),
    - But different stochastic uncertainties (SU)

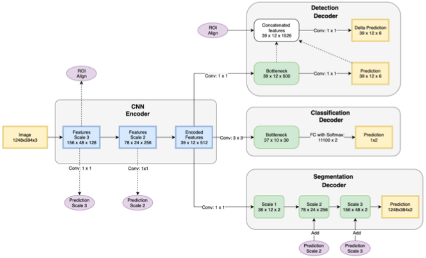
- In the component based architecture we train separate models for all tasks
    - All components have different EU and SU

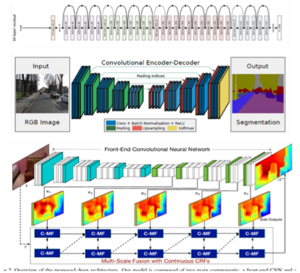
- We extract individual uncertainties and their propagation from the CityScape dataset

- For simplicity, we only represent discrete values for uncertainty (high and low)









$P(\cdot)$	EU	$SU\ OD$	$SU\ DE$	$SU\ SS$
H	0.18	0.16	0.11	0.19

$P(DE$	EU	SU	$OD)$
0.0	L	L	L
0.13	L	L	H
0.76	L	H	L
0.85	L	H	H
0.43	H	L	L
0.78	H	L	H
0.9	H	H	L
1	H	H	H

$P(SS$	EU	SU	$DE)$
0.0	L	L	L
0.28	L	L	H
0.64	L	H	L
0.72	L	H	H
0.66	H	L	L
0.58	H	L	H
0.61	H	H	L
1	H	H	H

$P(OD$	EU	$SU)$
0.0	L	L
0.64	L	H
0.61	H	L
1	H	H

$P(Planning$	$SS)$
0.1	L
0.9	H

$P(\cdot)$	$EU\ OD$	$SU\ OD$	$EU\ DE$	$SU\ DE$	$EU\ SS$	$SU\ SS$
H	0.14	0.16	0.31	0.44	0.17	0.19

$P(OD$	EU	$SU)$
0.0	L	L
0.57	L	H
0.41	H	L
1.0	H	H

$P(DE$	EU	SGU
0.0	L	L
0.51	L	H
0.47	H	L
1	H	H

$P(SS$	EU	$SU)$
0.0	L	L
0.11	L	H
0.42	H	L
1.0	H	H

$P(Planning$	SS	DE	$OD)$
0.0	L	L	L
0.34	L	L	H
0.34	L	H	L
0.66	L	H	H
0.34	H	L	L
0.66	H	L	H
0.66	H	H	L
1	H	H	H

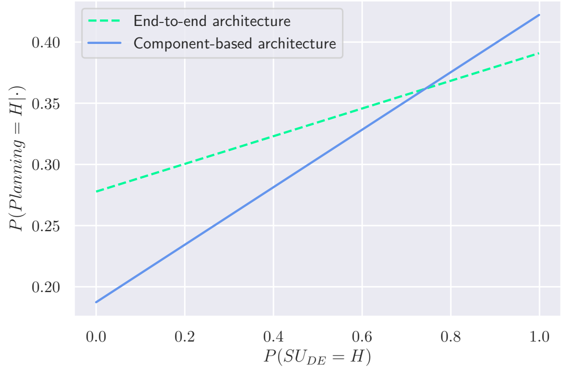
Evaluation

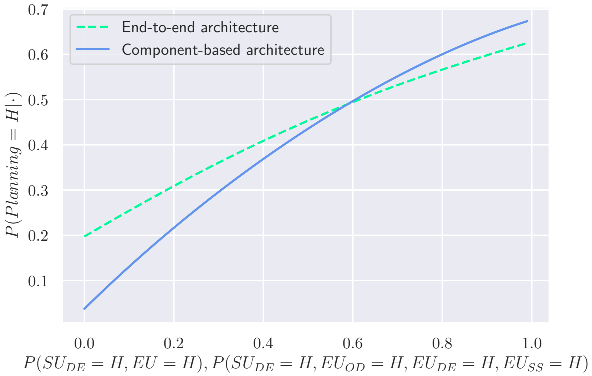
- Under the hood we use Bayesian Networks (BN) to propagate the uncertainties and evaluate trade-offs

- For example, the first plot shows the influence of high stochastic uncertainty for depth estimation in both architectures

- The second plot shows the influence of high stochastic uncertainty for depth estimation and high epistemic uncertainty for all components

- This model allows us to evaluate how any architecture copes with uncertainty, and select the one which is more resilient





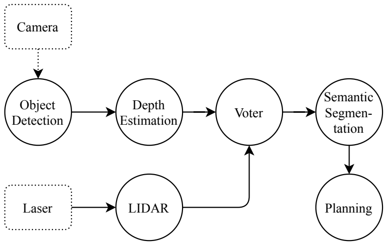
Using patterns to tackle uncertainty

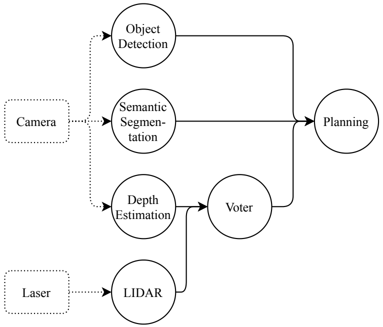
- Using the same methodology we can evaluate architectural changes

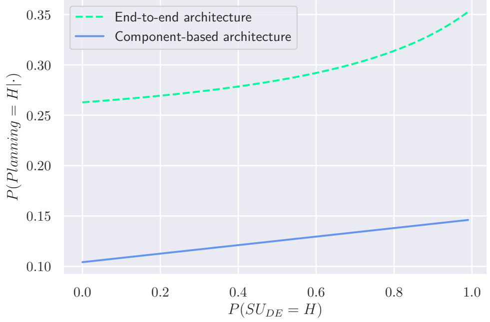
- In this case we analyze the use of an n-version programming pattern for depth estimation

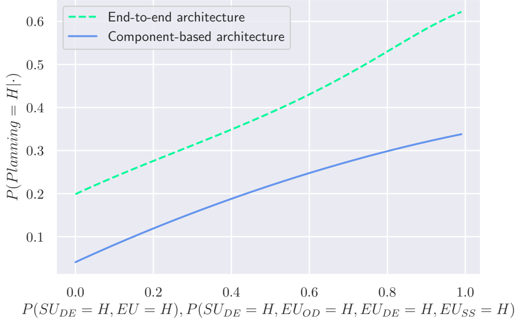
- The underlying formalism remains the same, only the uncertainty estimates have to be adjusted

- We observe that the architectures behave differently when this solution is considered









Conclusions

Learn more:

https://tinyurl.com/ml-architect

- We propose to use the inherent uncertainty of ML components in architecture design
- Elevating uncertainty as a design element can help build robust architectures with ML
- Software architectural patterns can be used to reduce uncertainty
- An interesting avenue for future research is to search for patterns that reduce uncertainty
- A limitation of the current model is the use of Bayesian Networks which allow only directed edges to be modelled (i.e., no cyclical interactions between components)