A Tale of Two Challenges

How I Learned to Stop Worrying and Released My Benchmark

SILVIU S. CRACIUNAS
SENIOR PRINCIPAL SCIENTIST, TRUSTMOTION
ASSOCIATE PROFESSOR, TECHNICAL UNIVERSITY OF DENMARK

15th International Real-Time Systems
Open Problems Seminar (RTSOPS 2026)

trustmotion
part of NXP

TrustMotion and NXP

25+ Years Experience Provides the Foundation for Automotive Safety

1998

Our Heritage

Our technological roots trace back to 1998, when TTTech Computertechnik was founded in Austria for fail-safe operational systems in the aerospace and transportation industry.

2014

Driver Assistance

Start of development of the first industry-wide, fully integrated driver assistance system (zFAS) in collaboration with Audi.

2017

Safety Middleware

We launch MotionWise, a scalable, series-proven safety middleware platform for automated driving.

2018

Spin-off: TTTech Auto

TTTech Auto is established with participation from Audi, Infineon, Samsung, and TTTech Computertechnik.

2022

Unicorn Status

The company secures a USD 285 million investment from Aptiv and Audi with a valuation above 1 billion USD.

2023

4SDV

Continued High Growth

Safety Consulting Services launched and entering a long-term collaboration with Mobileye The company also introduces the term 4SDV, emphasizing System, Safety, and Security for the Software-Defined Vehicle.

2024

Elevated Strategic Positioning & Branding

We elevate the strategic positioning and branding by focusing on the 4SDV approach and being an enabler & driver of the transformation towards safe SDVs.

2025

We are now part of NXP

NXP acquires TrustMotion to enhance its CoreRide platform with MotionWise safety-critical middleware while TrustMotion remains neutral and open to all SoC suppliers, OEMs, Tier-1s and 3rd party suppliers.

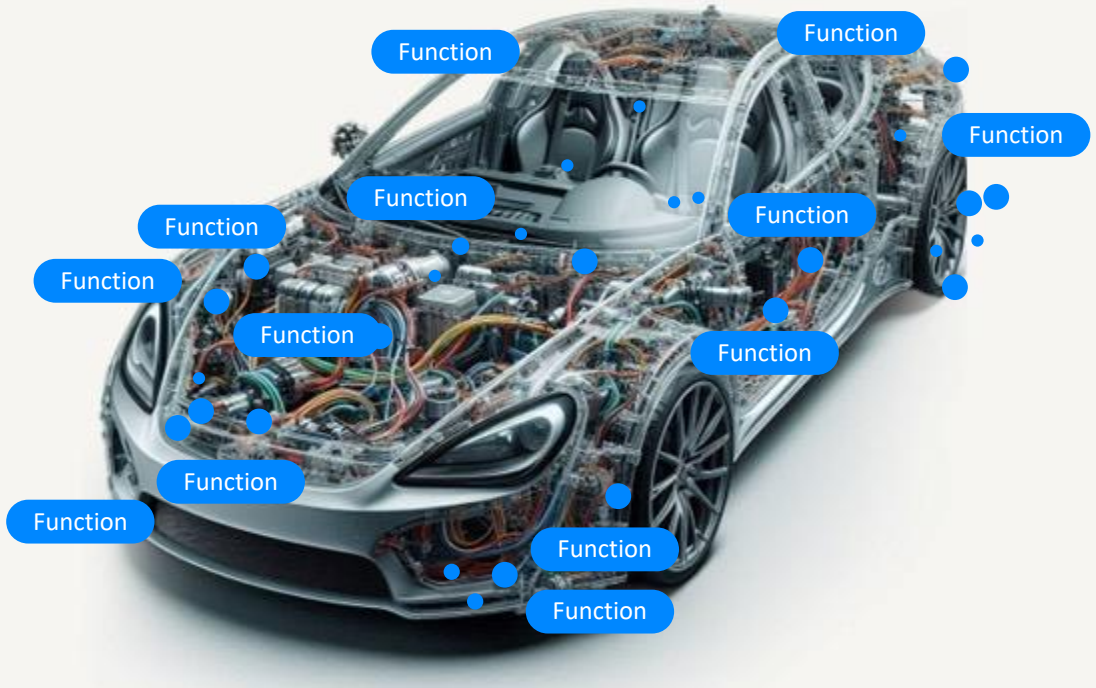
2026

New Name and Neutrality

As TrustMotion we are neutral in the market and continue to work with all SoC suppliers, OEMs, Tier 1s and 3rd party suppliers

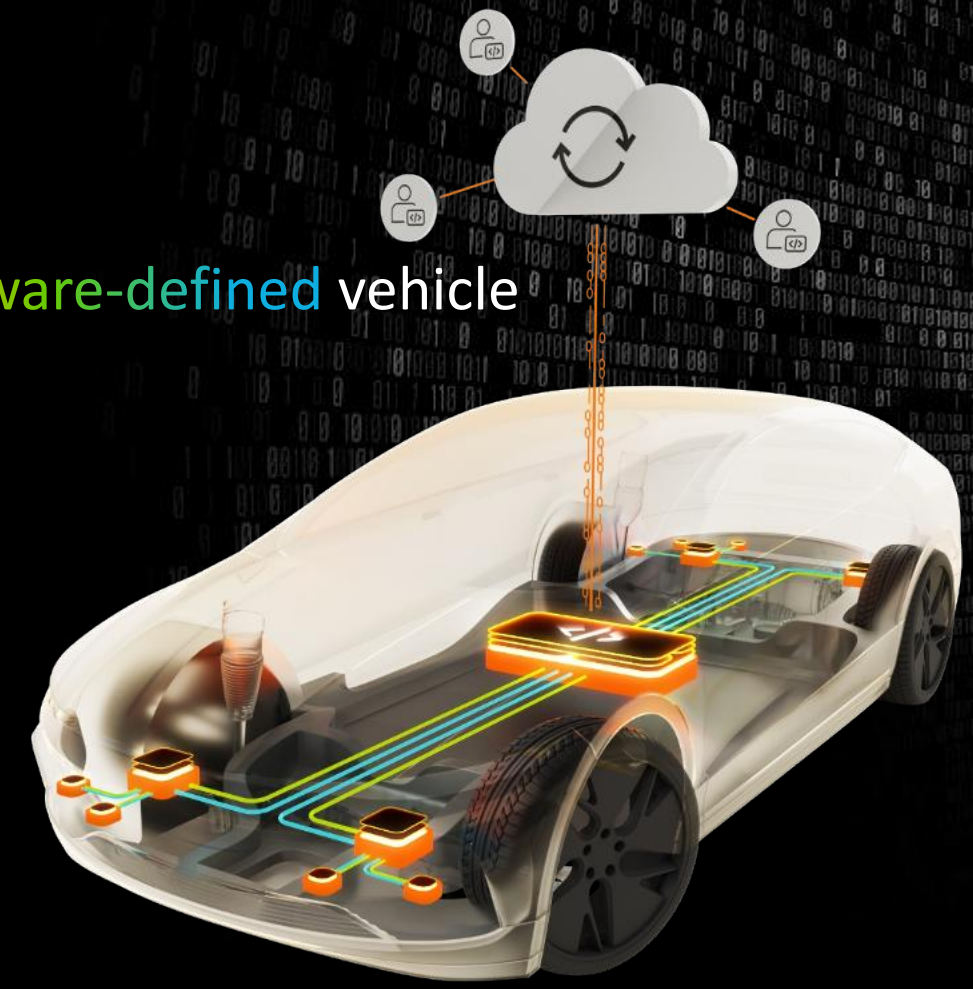
Vehicle transformation underway

From hardware-oriented vehicle



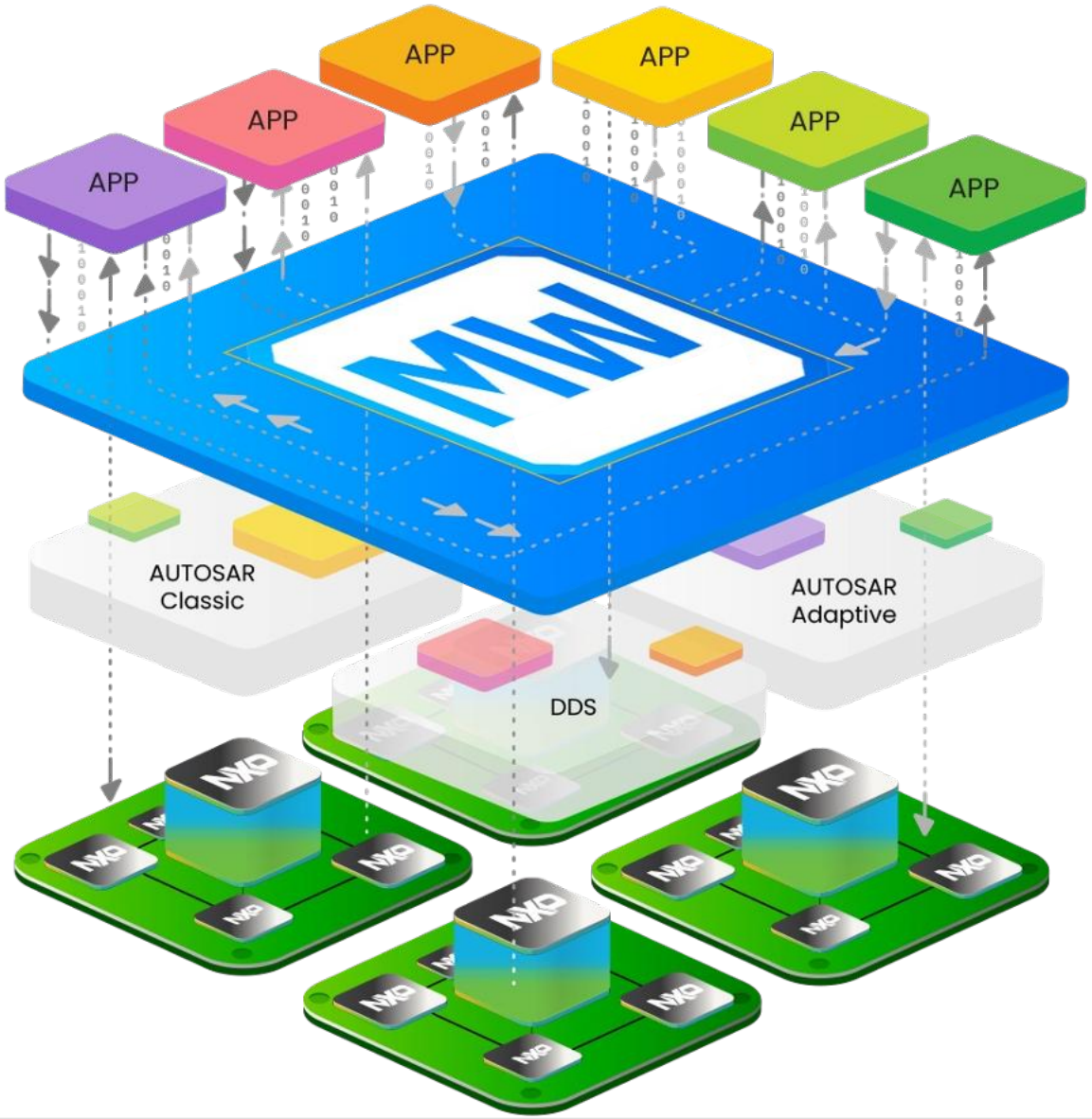
- Static
- Best performance when new
- Loses value over time

to software-defined vehicle



- Dynamic, updatable
- Performance increases over time
- Value increases over time

Software-Defined Vehicle



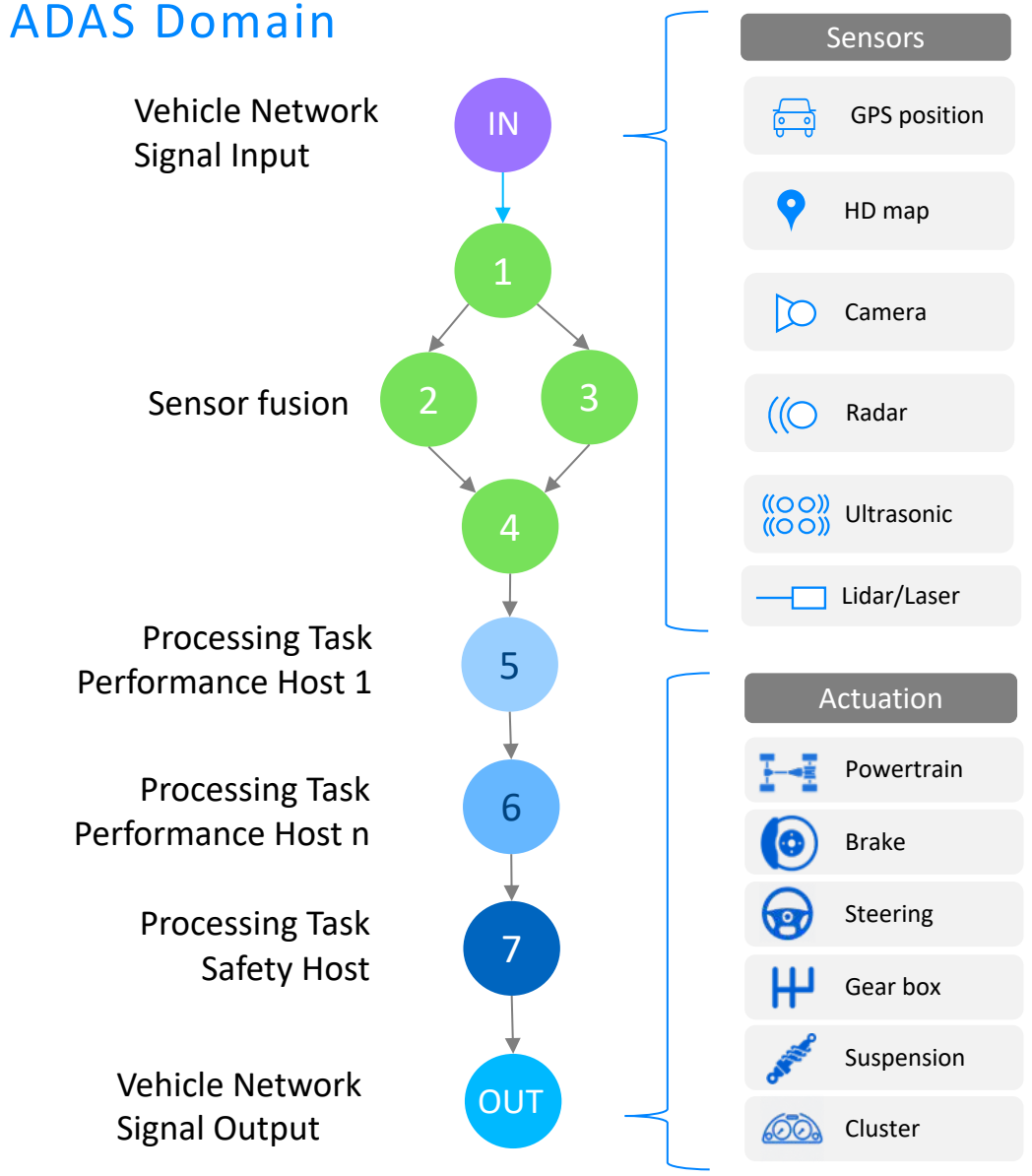
Cause-effect Chains

Cause-effect chains are commonly seen in automotive systems:

- provide additional timing and dependency requirements on the execution of tasks
- can span across multiple activation patterns
- have priorities and end-to-end latencies
- include communication latencies

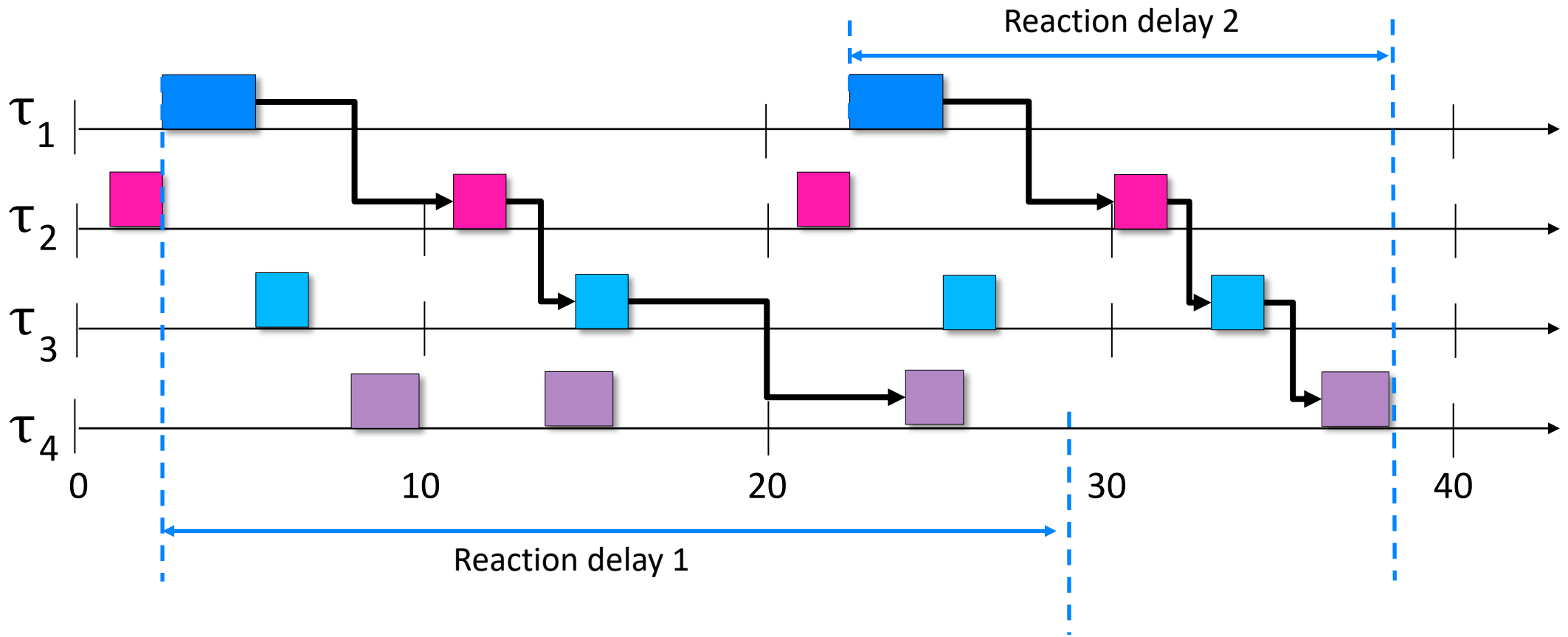
» Semantics:

- Reaction delay
- Data age
- Order



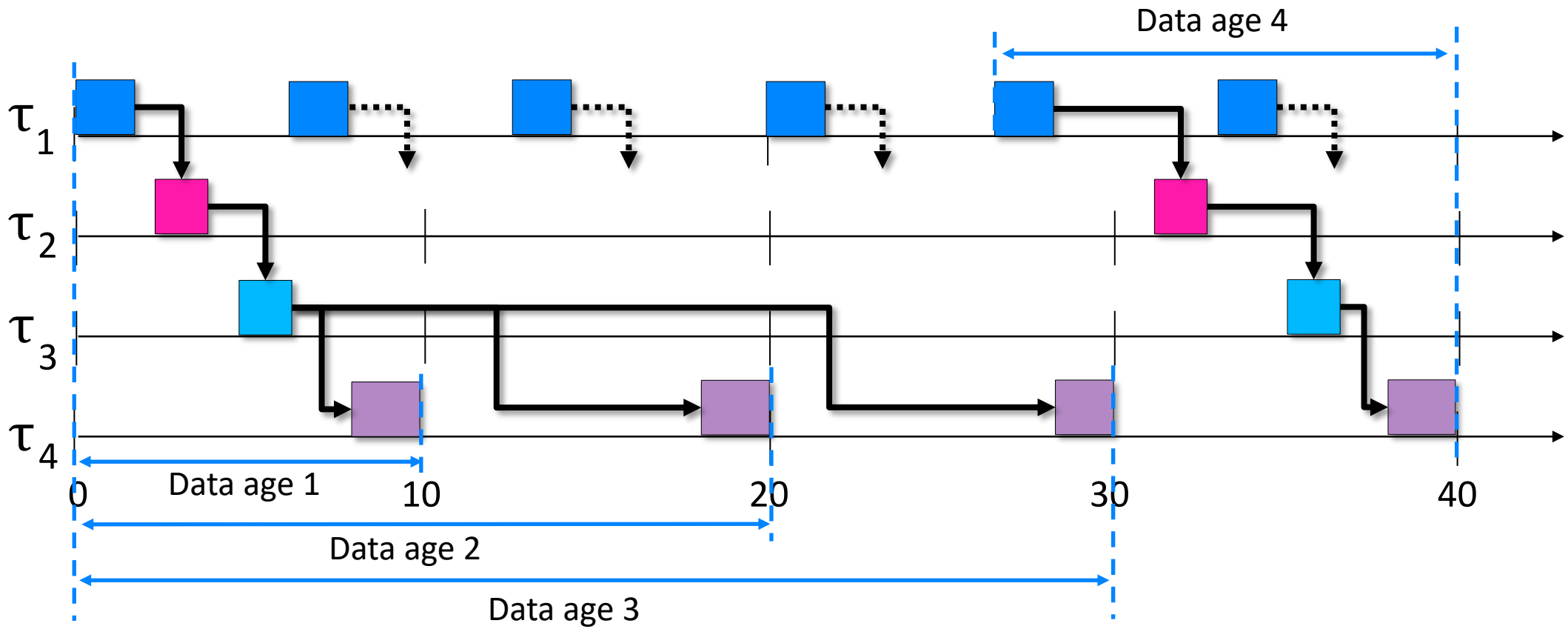
Cause-effect chains: reaction delay

- After each event, there must be a reaction within the maximum allowed time

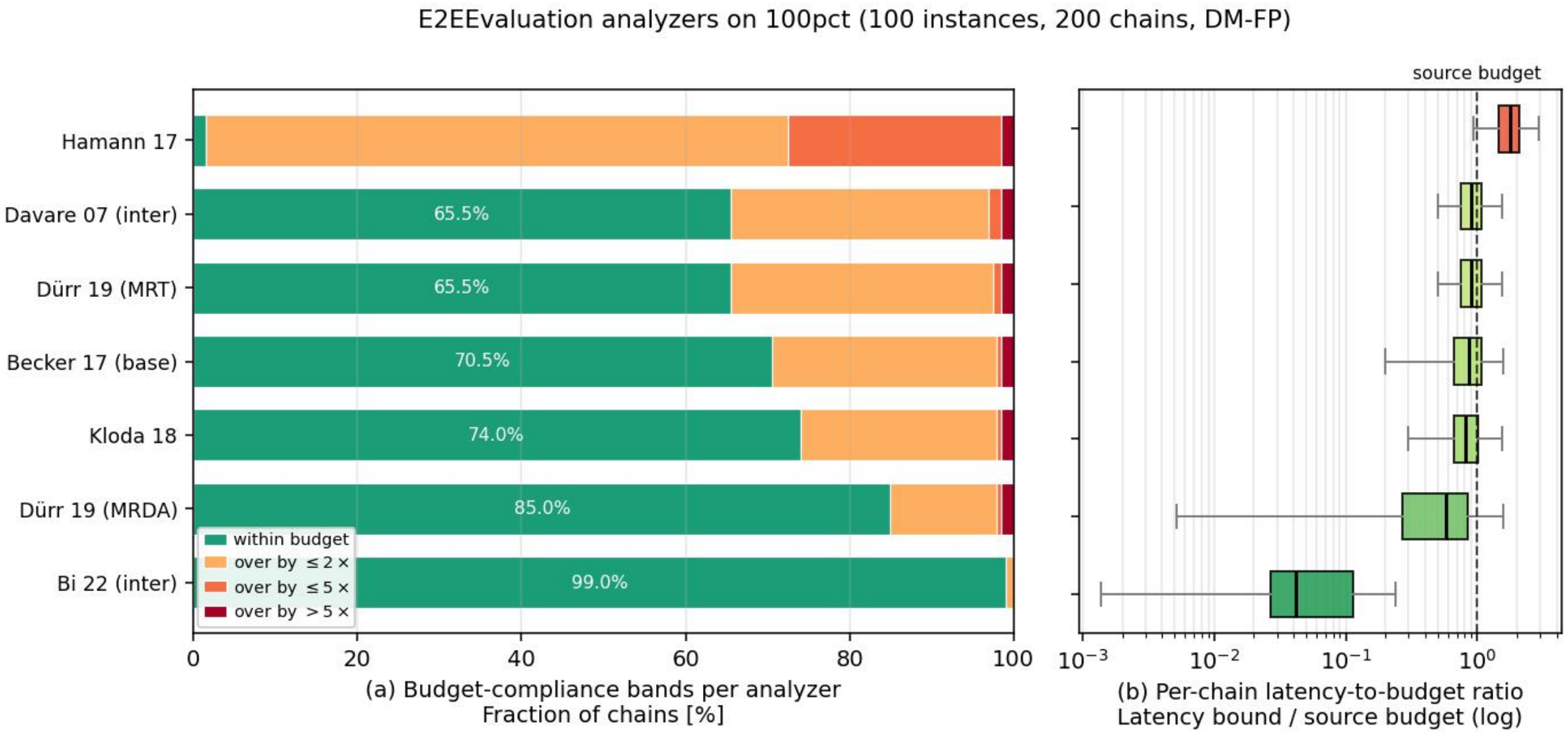


Cause-effect chains: data age

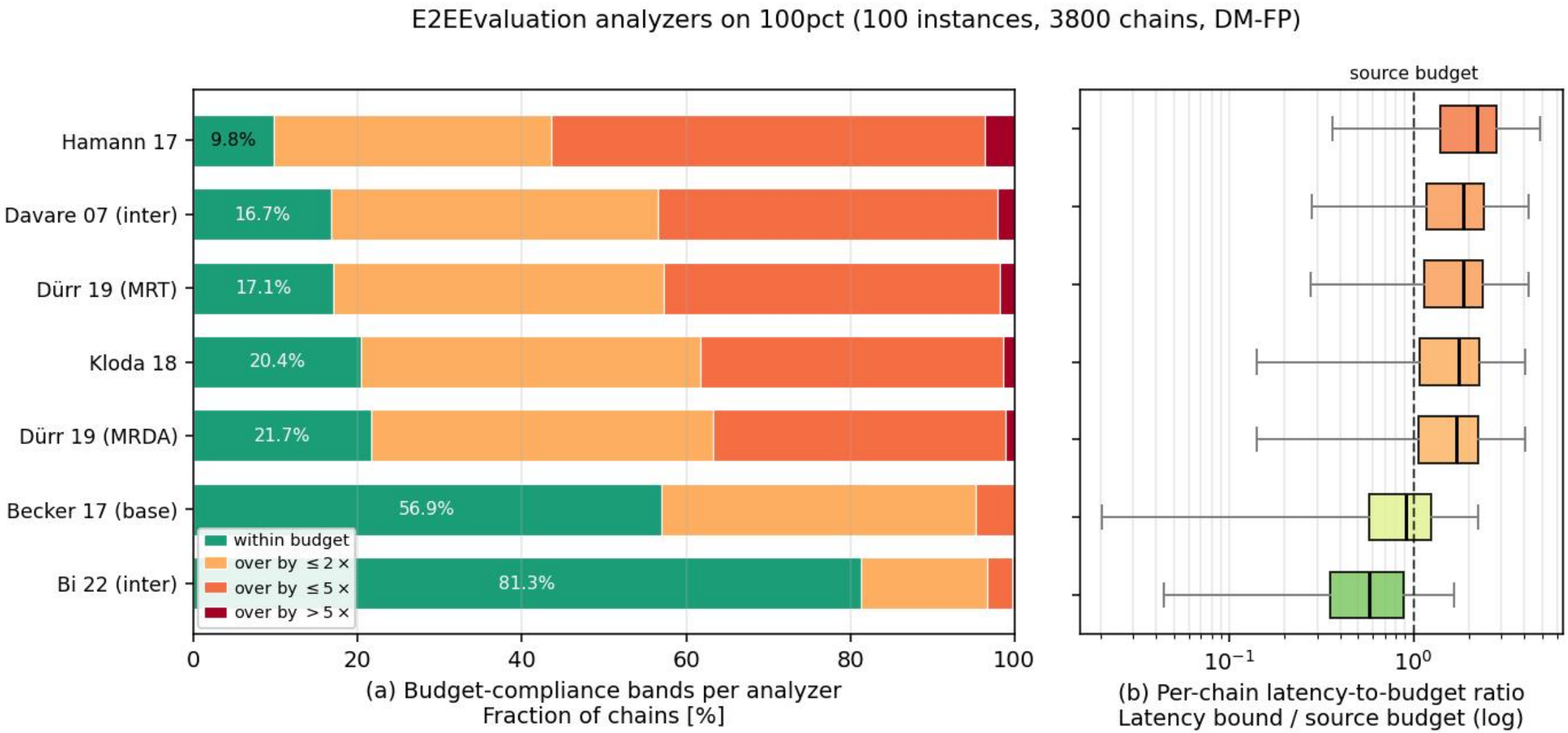
- Each action must be based on data that is not older than the given maximum



Why yet another (WATERS) challenge?



The TrustMotion ADAS challenge



The TrustMotion ADAS Challenge – what we see in the wild

Platform

3 CPUs / 9 cores

CPU	|Γ|	Σu
DM (3 cores, FP)	62	2.84
SF (2 cores, TT)	24	1.81
TC (4 cores, TT)	16	3.25
total	102	7.90

Period mix

2.5 – 80 ms

T [ms]	n	share
2.5	1	1.0 %
5	10	9.8 %
10	33	32.4 %
20	15	14.7 %
33.3	5	4.9 %
40	23	22.5 %
80	15	14.7 %

Constraints

83 % pinned

per-role	n / N	share
DM pinned	56/62	90.3 %
SF pinned	14/24	58.3 %
TC pinned	15/16	93.8 %
D < T	79	77.5 %
earliest-activation > 0	54	52.9 %

Chain lengths

39 chains

mean 10.9, max 17 - clusters at $L \in \{9, 10\}$ & $\{15, 16\}$

L	n	L	n
2	1	11	2
5–6	5	12	2
7–8	3	13–14	2
9	6	15	6
10	6	16	4
		17	2

Budget tightness

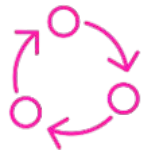
Budget tightness · $\rho = b / T\_th$

median 0.72 · mean 1.00 · p_{10} 0.36 · p_{90} 1.43

ρ bin	n	share
[0.20, 0.40)	4	10.3 %
[0.40, 0.60)	4	10.3 %
[0.60, 0.80)	12	30.8 %
[0.80, 1.00)	3	7.7 %
[1.00, 1.50)	12	30.8 %
[2.00, 3.00)	4	10.3 %

Chain shapes

Cycles



29/39 of chains contain at least one repeating task

Comm-stack and middleware tasks invoked on the receive leg are the same task invoked on the transmit leg, producing a pattern:

$$PDU_{IN} \rightarrow COMM \rightarrow MW \rightarrow \ldots \rightarrow MW \rightarrow COM \rightarrow PDU_{OUT}$$

Cyclic stateful pass-through, e.g. ultrasonic preprocessing or object fusion, where chains may contain repeated runnable references of the form:

$$\ldots \rightarrow \underbrace{T_{US} \rightarrow T_{US}}_{\text{n times}} \rightarrow \ldots \rightarrow \underbrace{T_{OBF} \rightarrow T_{OBF}}_{\text{m times}} \rightarrow \ldots$$

Variable middleware ET

MW tasks have an ET of the form:

$$ET = C + n * V$$

- C Packing independent execution time
- n Number of packed messages
- V Processing cost for one message

Tradeoff

- Less packing: shorter ET, less latency, but more MW instances and thus higher U
- More packing, less MW instances, smaller U, but more latency and larger ET

In the model we have for the moment a range:

$$WCET \in [\ WCETmin\ ,\ WCETmax\ ]$$

TrustMotion Challenge 1 – this one is for academia

Analyze chains

- **Given**: task set + chains + scheduling policy
- **Find**: tight per-chain end-to-end latency bound
- **Examples**: Bi@DAC22, Dürr@TECS19

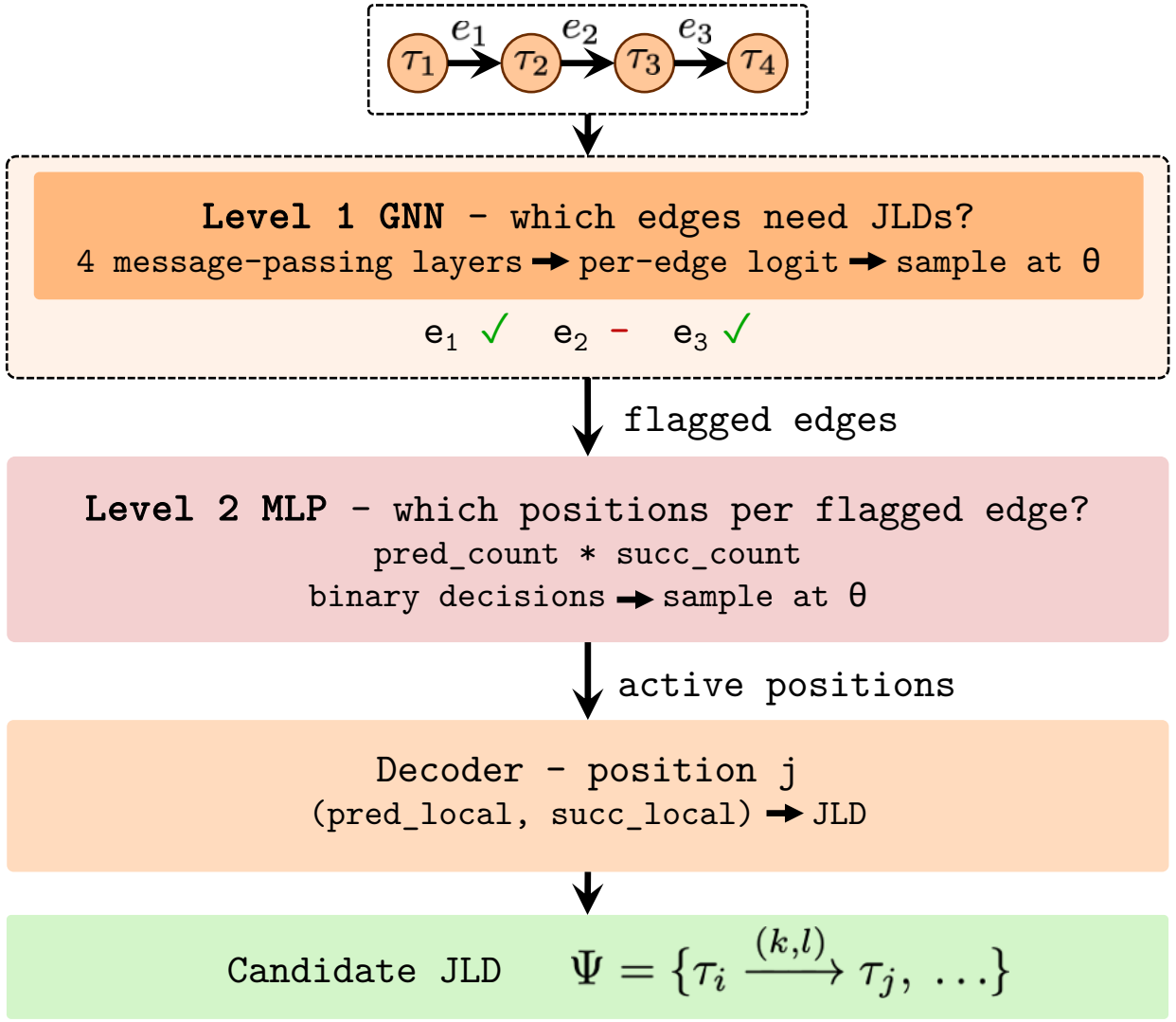
Synthesis with correct chains

- **Given:** task set + chains
- **Find:** priority / mapping / offsets / TT-slot assignment that meets every chain budget
- **Examples:** McLean@ETFA20

Job-level dependency (JLD)

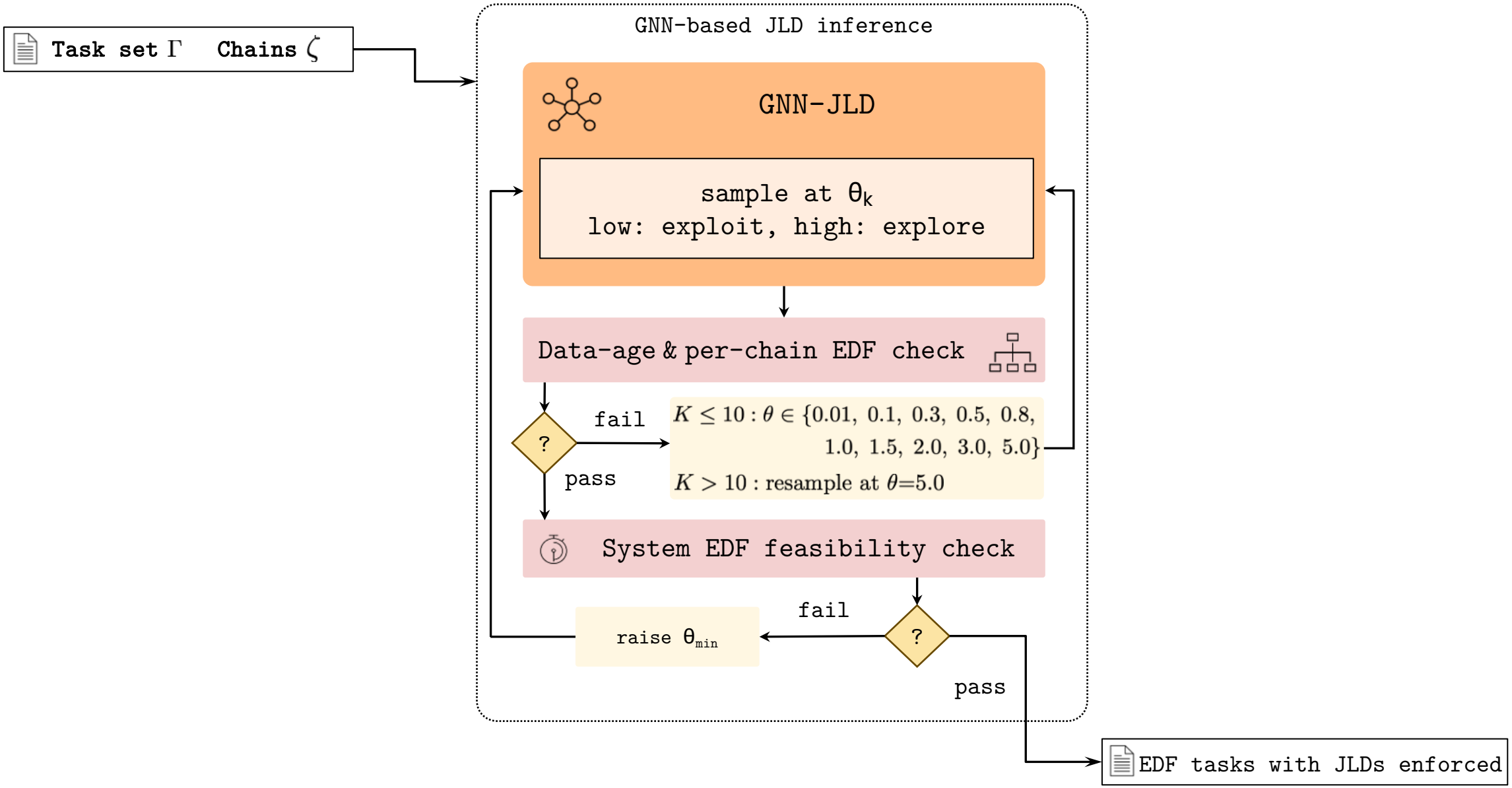
- A schedule-agnostic constraint that pins down which job of the producer feeds which job of the consumer independent of the underlying scheduler. (c.f. Becker@RTCSA16, Becker@ASP-DAC24)
- A chain's data-age budget becomes a property of the chosen JLD pattern, not of the schedule.
- **Question 1**: Are the JLDs actually enforceable under a chosen scheduling policy (FP, EDF) at run-time?
  - Need a per-chain check that checks “this JLD assignment will hold under EDF (or DM, or TT).”
- **Question 2:** With JLDs enforced across all chains simultaneously, is the full task set still schedulable?
  - Independent per-chain JLDs combine into a system-level demand-bound problem.
  - Need a system-wide test that checks "all JLDs together do not over-constrain the schedule."

Is ML a solution?

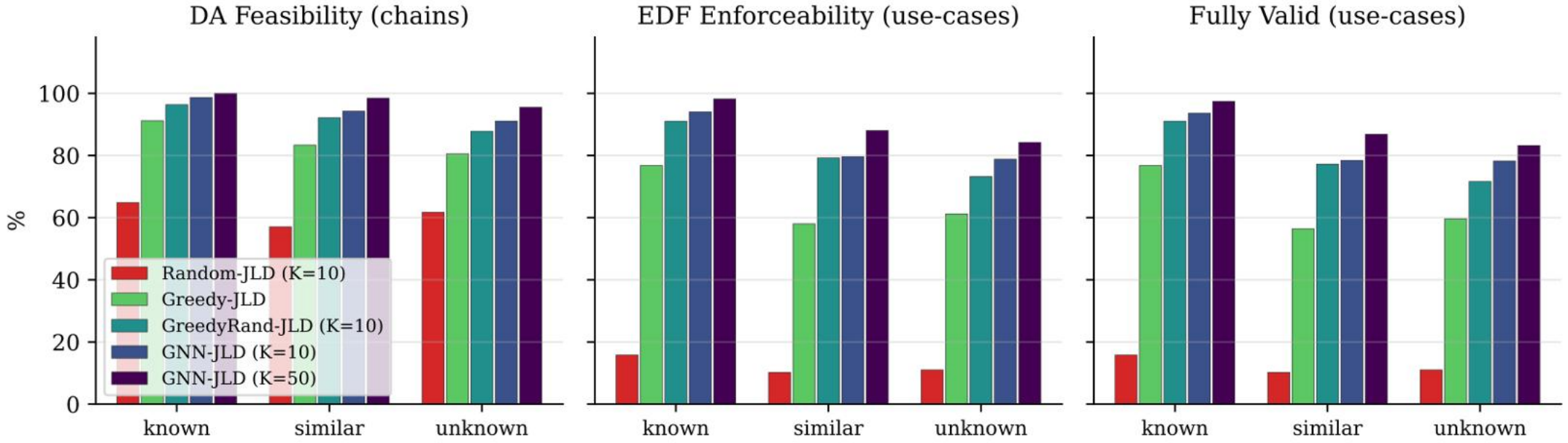


S. S. Craciunas, C. Hakert, J-J Chen, Z. Hanzálek, P. Pop - Schedulable Job-Level Dependencies for Cause-Effect Chains via Graph Neural Networks. Preprint available at: http://arxiv.org/abs/2607.02624

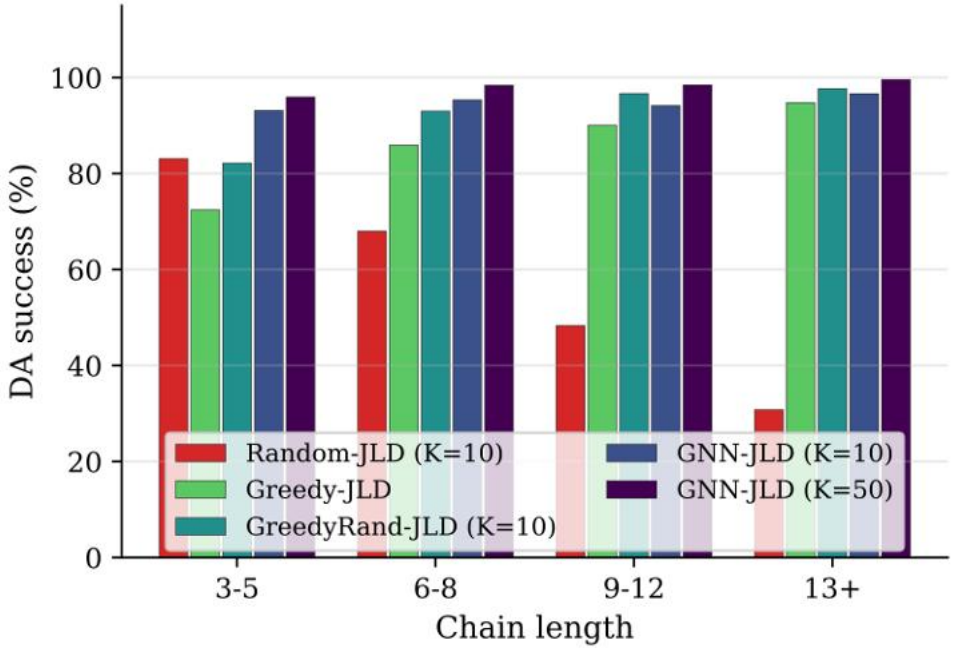
Is ML a solution?

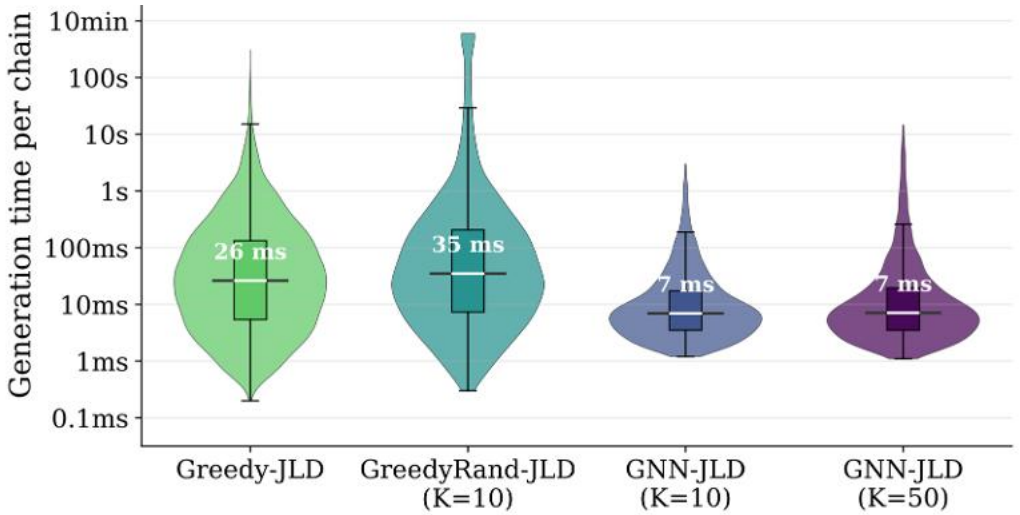


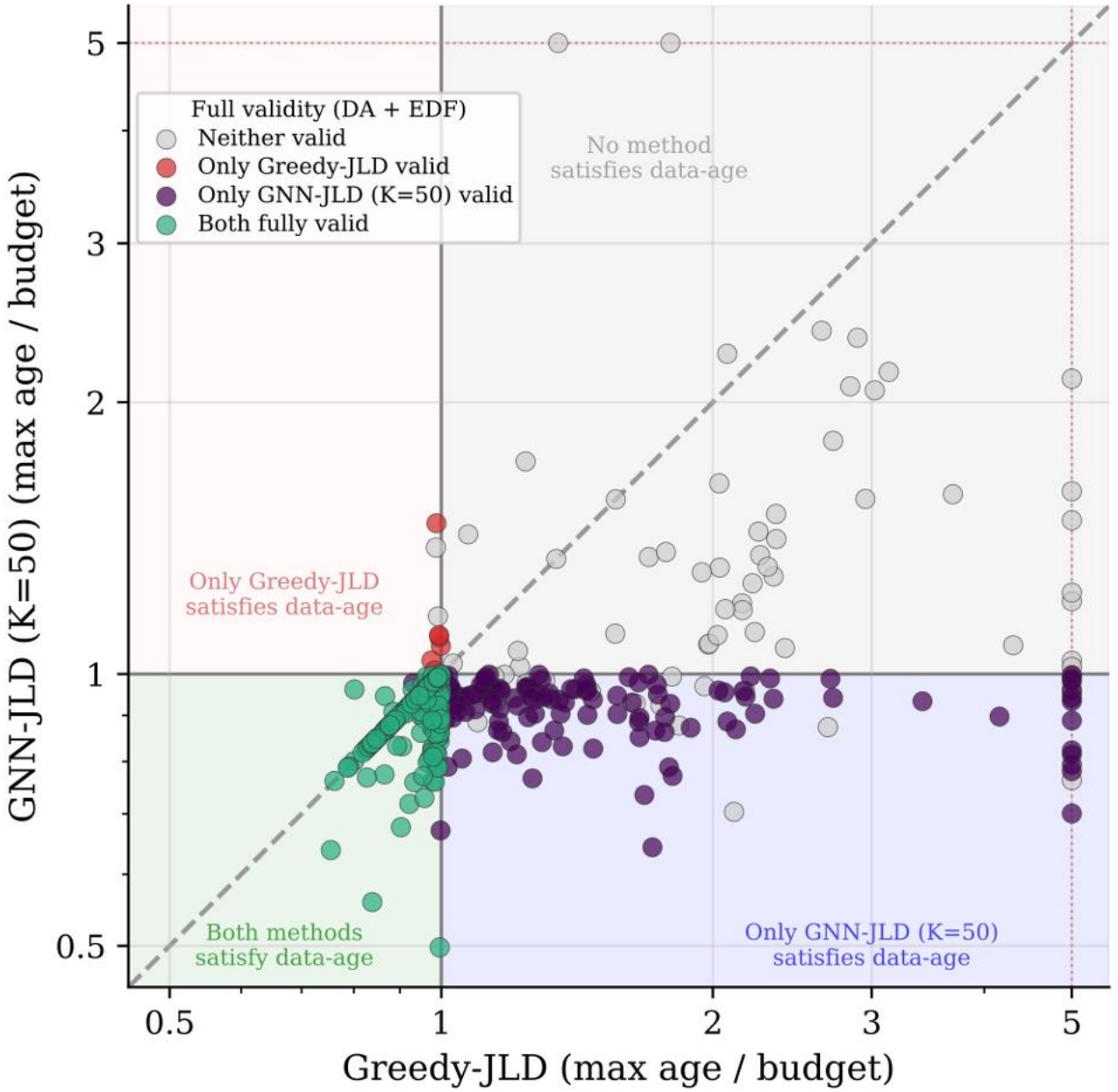
Results for EDF systems



Comparison to SOTA







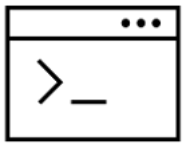
TrustMotion Challenge 2 – this one is for industry

- Real industrial workloads stay locked behind IP, NDA, and competitive concerns.
- Academic benchmarks drift away from industrial reality (Kramer@WATERS15).
- **Solution:** Publish what a workload looks like - without publishing the workload itself.

Fingerprint

- Anonymised against IP and per-task / per-chain identification.
- Not anonymised against “what kind of workload this is”.
- Enough abstraction to share publicly and enough structure to be useful for benchmarking

Generator



- Takes a fingerprint and a scale knob and draws fresh use-case instances from its per-axis distributions with same statistical shape & scaled to any size
- Bundle: 2 fingerprints (TrustMotion + WATERS-Bosch reference) & pre-generated synth data (TrustMotion, 5 × 100 = 500 instances)

https://github.com/dtu-ese/automotive-cause-effect-chain-benchmark

Challenge 2 – this one is for industry

“Come, then!” cried Defarge, in a resounding voice. “[...] friends, we are ready! The Bastille!”

— Charles Dickens, A Tale of Two Cities

- **Community effort**: let’s agree on the fingerprint for cause-effect chains
  - **v2** of the schema and generator: more activation classes, a communication backbone, LET semantics, etc.
- Release anonymised fingerprint + matching synthetic data
- **Invitation**: Maintain a collaborative bibliography of released fingerprints.



Thank you!


silviu.craciunas@nxp.com


https://scraciunas.github.io/


www.linkedin.com/in/scraciunas


```
@misc{CraciunasRTSOPS26,
  author = {Craciunas, Silviu S. and Becker, Matthias and Pop, Paul},
  title = {Automotive Cause-Effect Chains in the Wild: A Real-World Industrial Benchmark and Generator},
  month = jun,
  year = 2026,
  publisher = {Zenodo},
  version = {v1},
  doi = {10.5281/zenodo.20637979},
  url = {https://doi.org/10.5281/zenodo.20637979},
  note = {Presented at the 15th International Real-Time Systems Open Problems Seminar (RTSOPS 2026)}
}
```