Towards explainable AI

through digital twins, generative models and advanced simulations

2019-06-27, Science forum

Andrey Ustyuzhanin

NRU HSE

YSDA

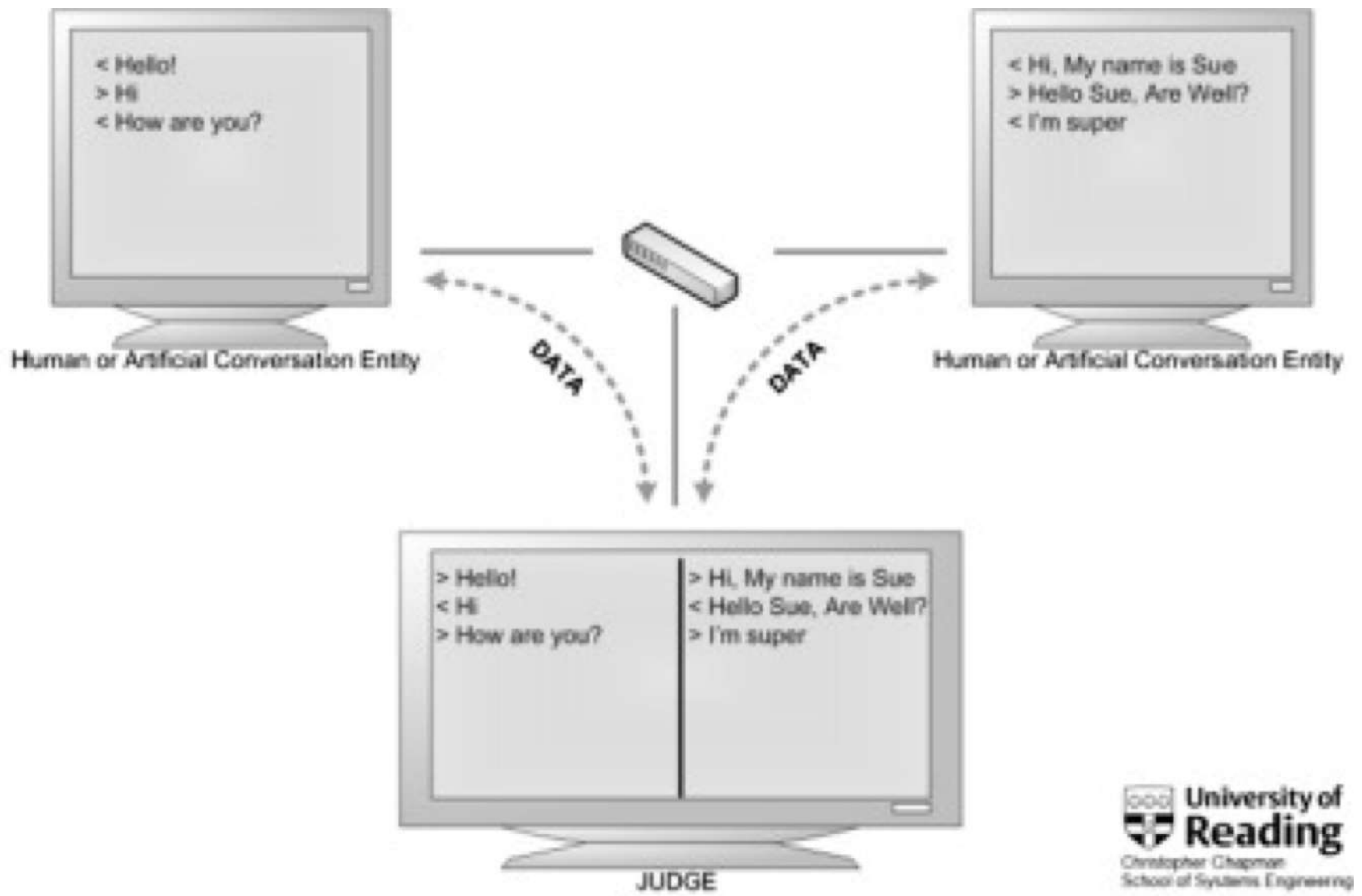
ICL

Turing Test

> “I believe that in about 50 years’ time it will be possible, to programme computers to make them play the imitation game so well that an average interrogator will not have more than 70 % chance of making the right identification after 5 min of questioning” Alan Turing, 1950 [1]

Has been passed in 2014

(a)



(b)

Explainable AI (XAI)

The predictions, and the ensuing advice for decision-making, provided by AI (or machine learning), should be accompanied by explanations [2, 3]

Open Questions:

- Can automated reasoning contribute to the development of a definition, or even a theory, of explanation?
- Could explanations be derived by computational inference?
- How can we bridge the gap between the statistical inferences of machine learning and the logical inferences of reasoning, applying the latter to extract, build, or speculate and test, explanations of the former?
- How can we bridge the gap between the apparently very different abstraction levels of explanation and explanation as in XAI?

Why bother with explainability?

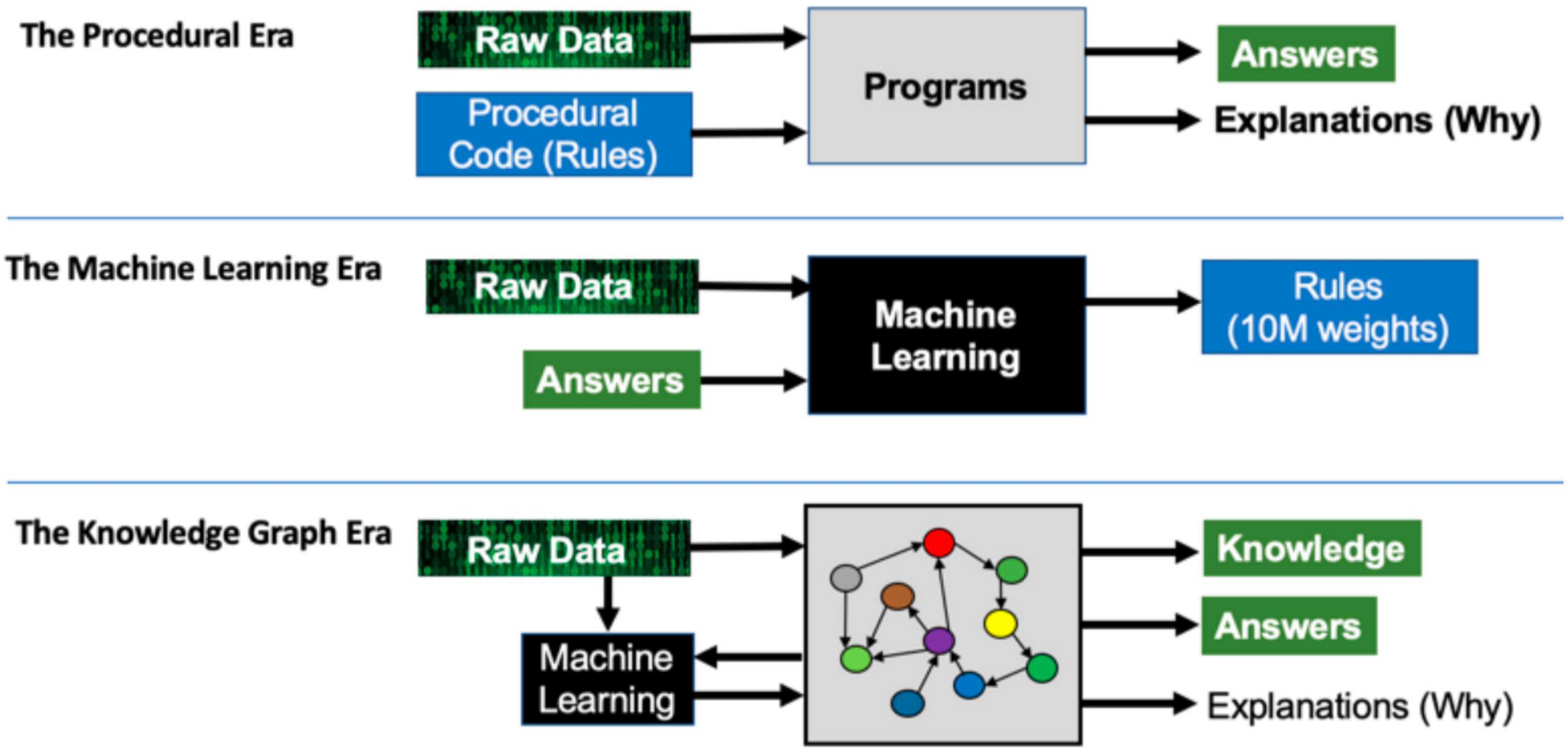
An explanation should at least provide the human user with information on what could go wrong by following the machine's prediction or advice [2]

Gives a hint towards more difficult version of Turing test

Path towards XAI

- Representation
  - Knowledge (objective) representations
  - Experience (subjective) representation
- Advanced environment simulation models
  - Comprehensive
  - Realistic
  - Fast
- Model of an object within the environment (digital twin)

Knowledge/experience representation



The Knowledge Graph Era: MLcontinuously reads raw data, combines this with existing knowledge and produces new knowledge, answers and explanations

http://bit.ly/2XuXAPE

Simulators

Already heavily used in

- Physics, Chemistry
- Medicine
- Space sciences

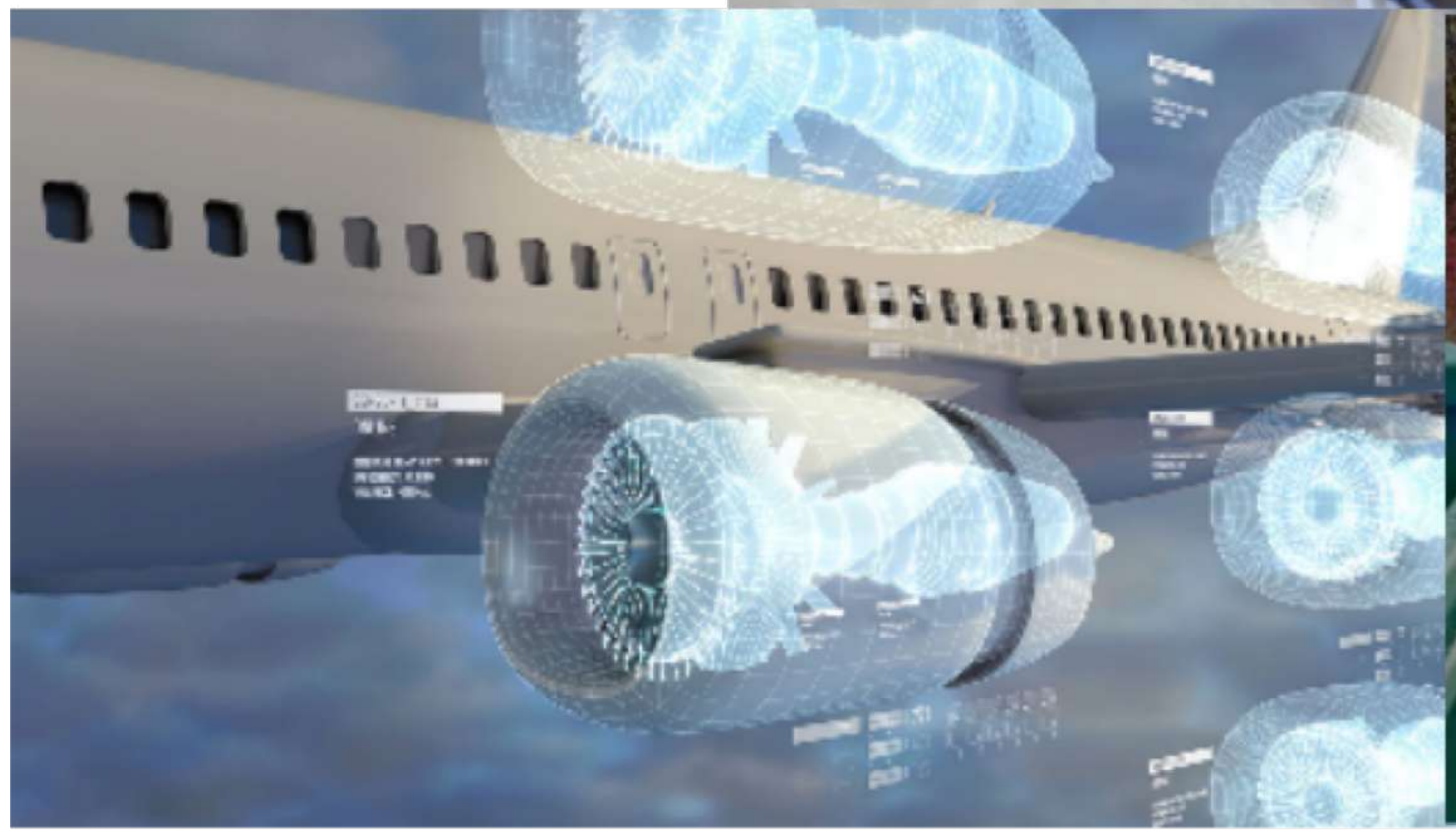
Give the luxury to learn statistical and causal dependencies

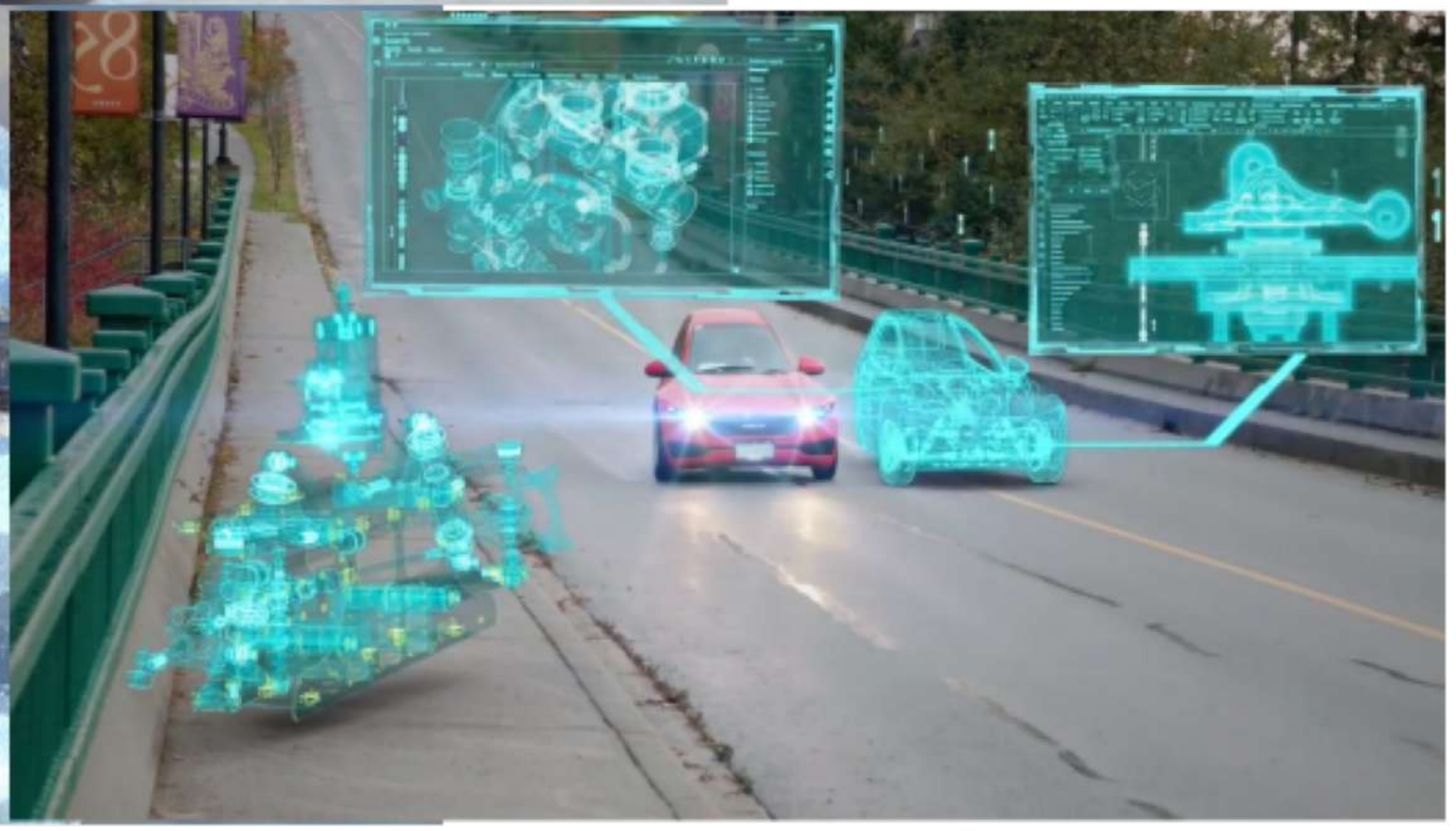
However

- Difficult to embrace uncertainties
- Have to be comprehensive
- Have to be fast

Digital Twins (DT)

DT of an object is an expressive generative model that can adequately relate current internal state, environment conditions and changes with the state distribution at the following time step.



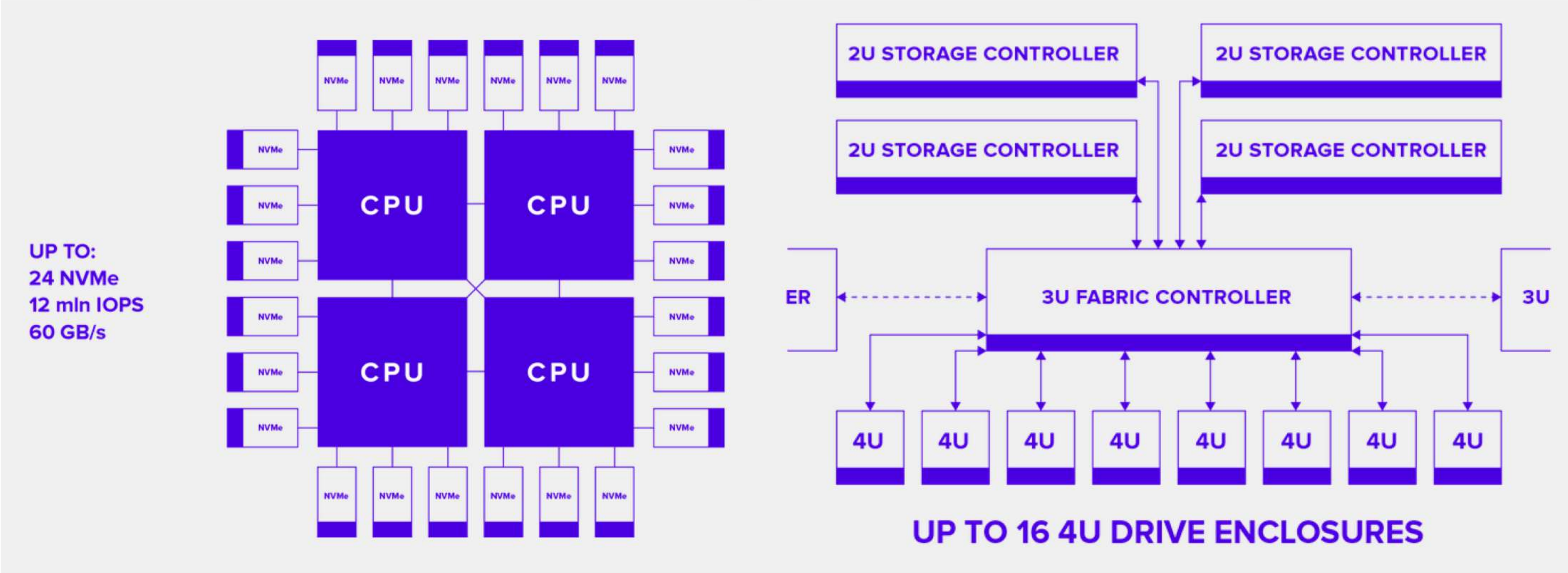


Real life example

Joint project of HSE LAMBDA with SPbPU and YADRO Ltd

Improve **fault-tolerance** of TATLIN SAN systems:

- **Complex custom system**
- **Own caching techniques**
- **Reed-Solomon encoding instead of RAID**



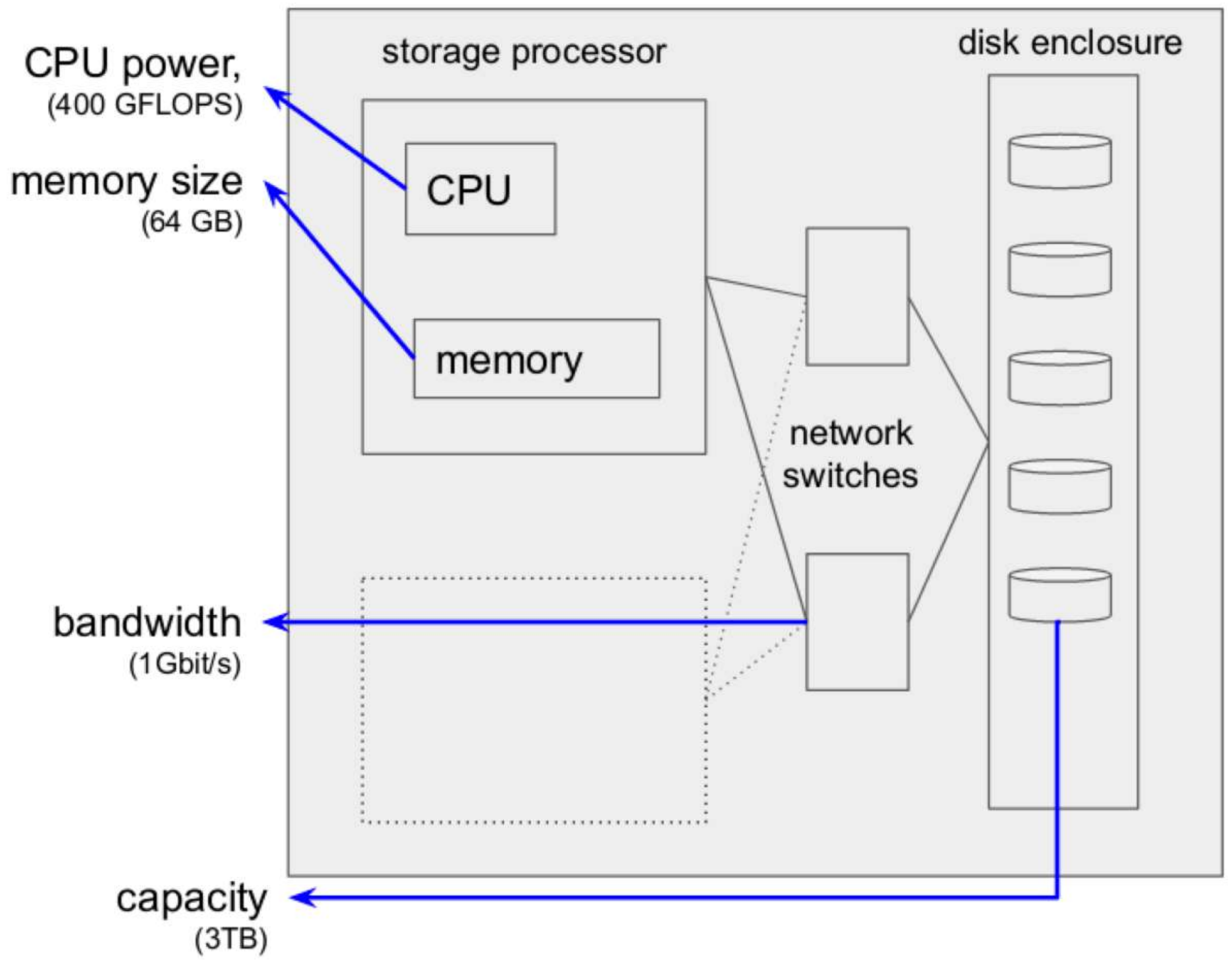
Fault-tolerance for TATLIN

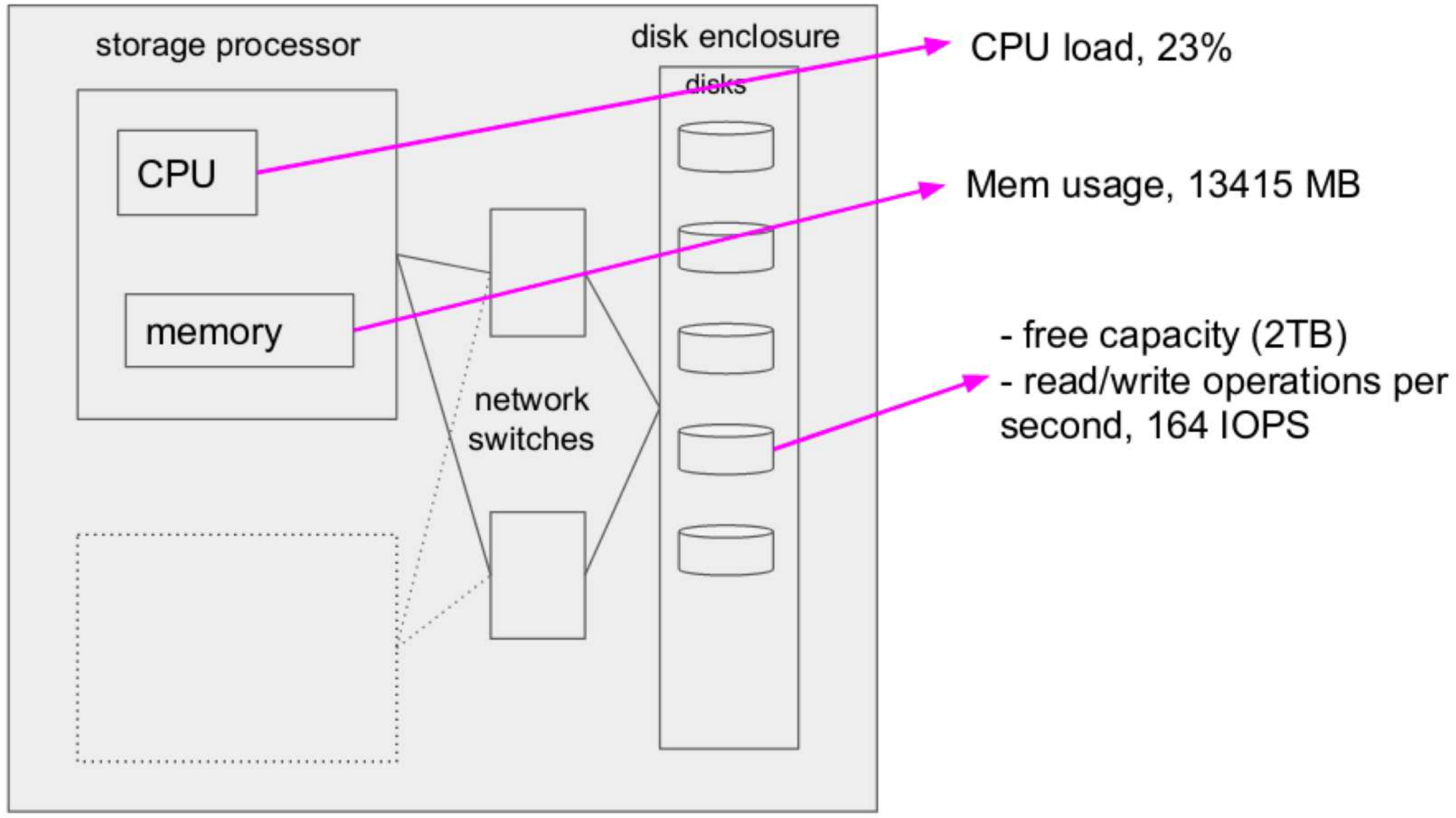
- Lack of real-life data
  - Lots of missing data
  - Takes time to collect
  - Anomalies are rare
- No ML approach can deal with these settings
  Not clear which ML approach will eventually become handy

- Let’s create TATLIN simulator!

Simulated environment

DEBS – Discrete event-based SAN simulator. Connects effective SAN parameters (e.g. effective CPU power) with observables (e.g. disk capacity)



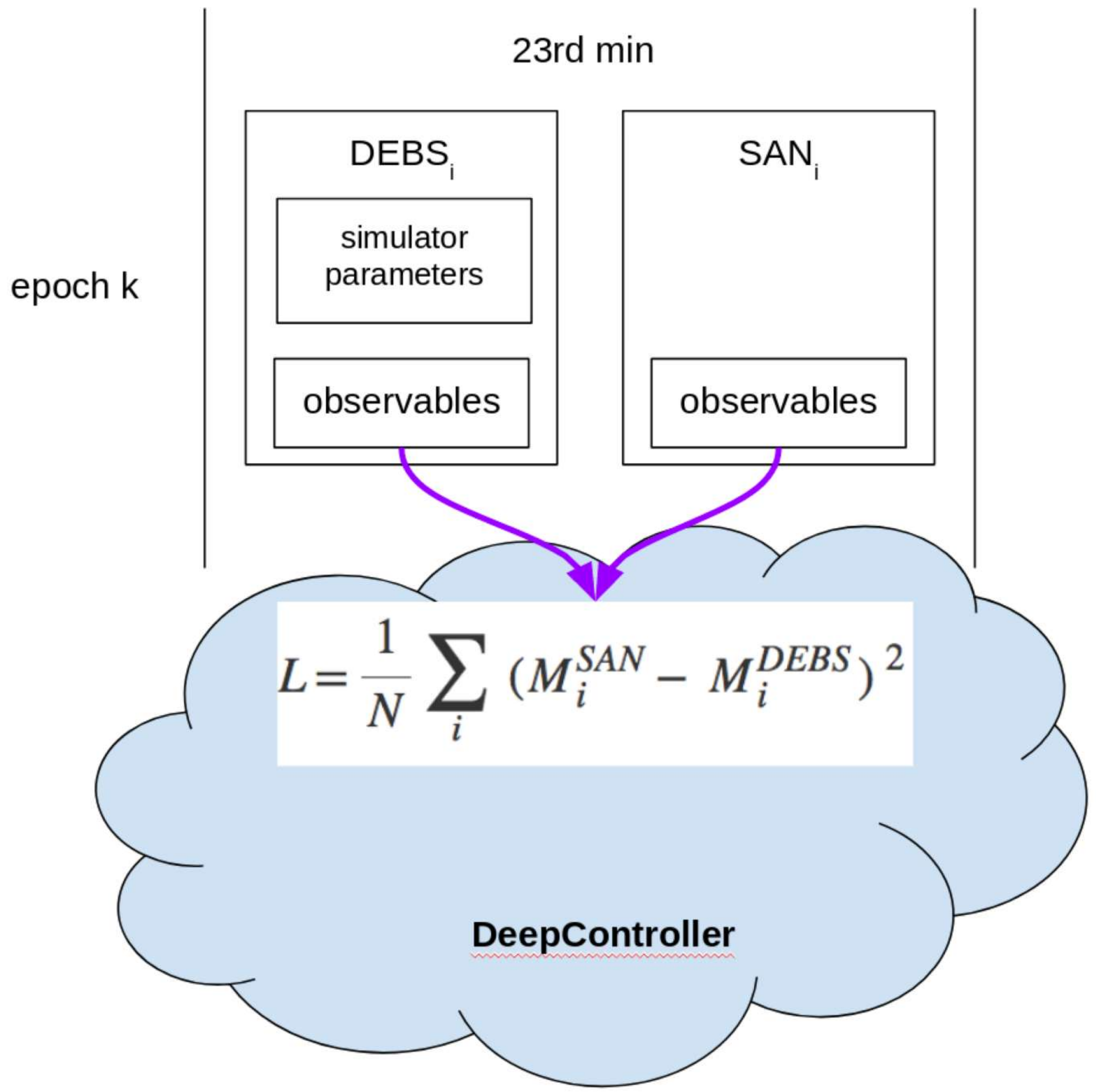


DeepController – NeuralNet-based model that is trained to tune DEBS

DeepController training

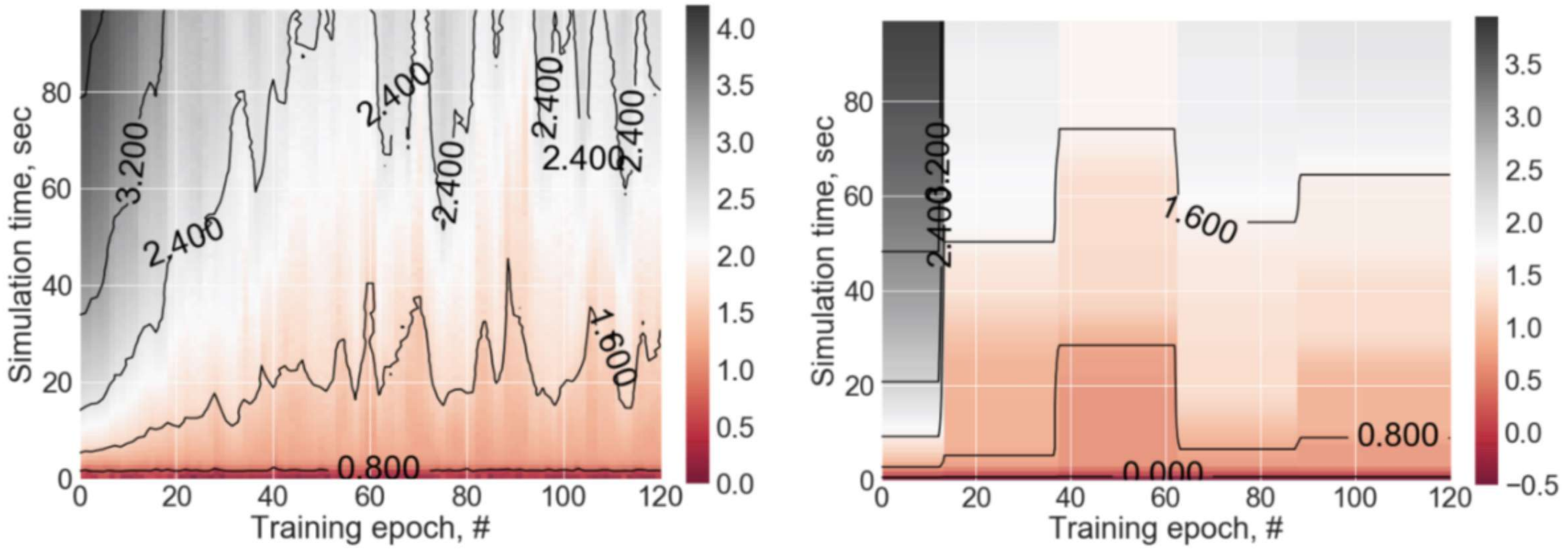
Reinforcement Learning (Deep Deterministic Policy Gradient): minimize L that is difference between observables distributions from reality and DEBS under similar conditions

i.e. neural net weights: $W^* = \text{argmin}\ L(W)$, DDPG allows to estimate dL/dW stochastically since analytical derivatives are not available



Results

The difference (loss function) as time dependency:



(a) (b)

The more data we have, the better is the result
The longer rungs the simulation the larger is the uncertainty

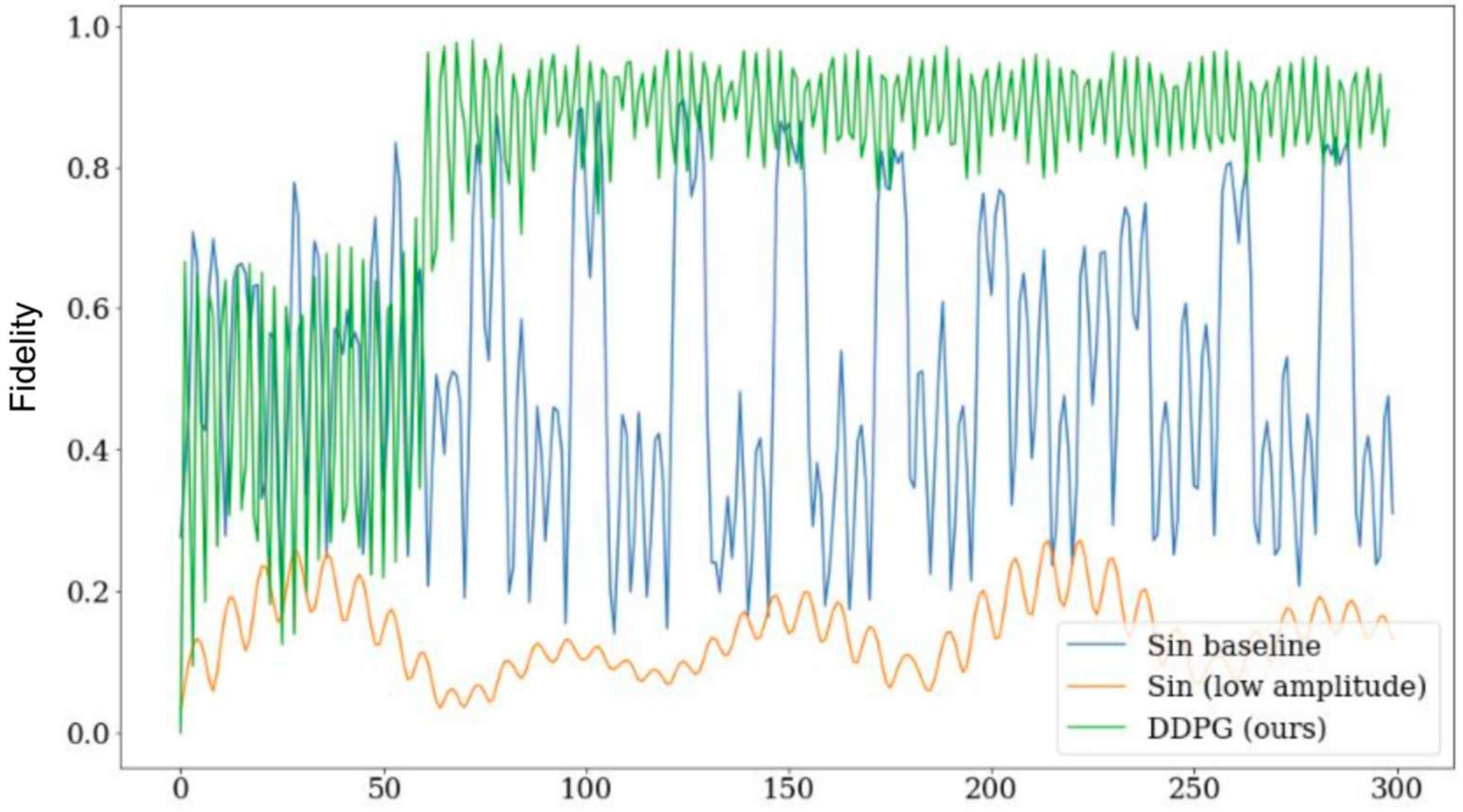
Wishful step further

Make simulator differentiable (or replace with a differentiable ‘good-enough’ surrogate model)

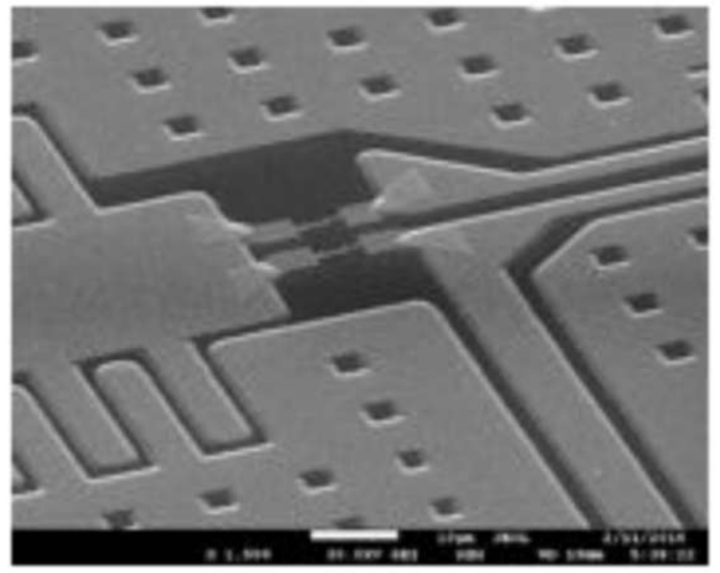
Then training will become much more stable, faster and less-dependent on initialization

More examples. Quantum Control

Problem: learn to control qbit to switch from one state to another

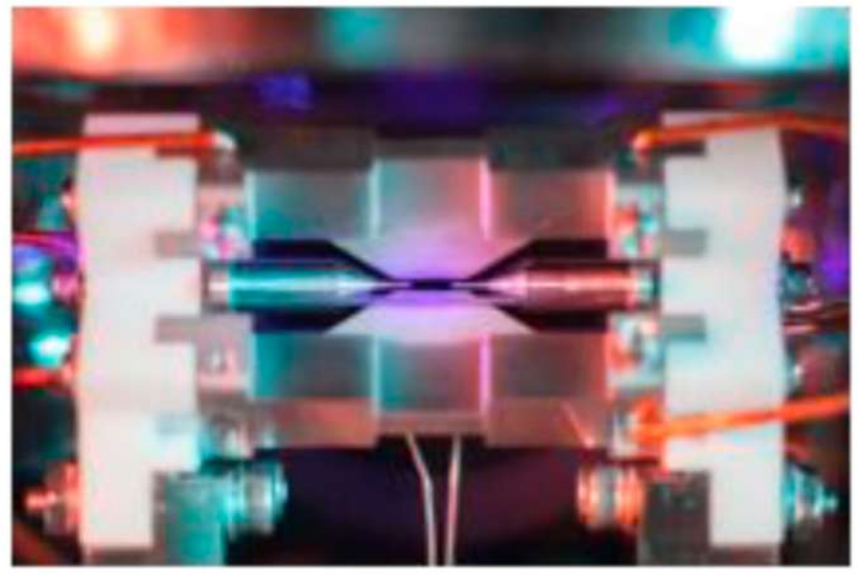


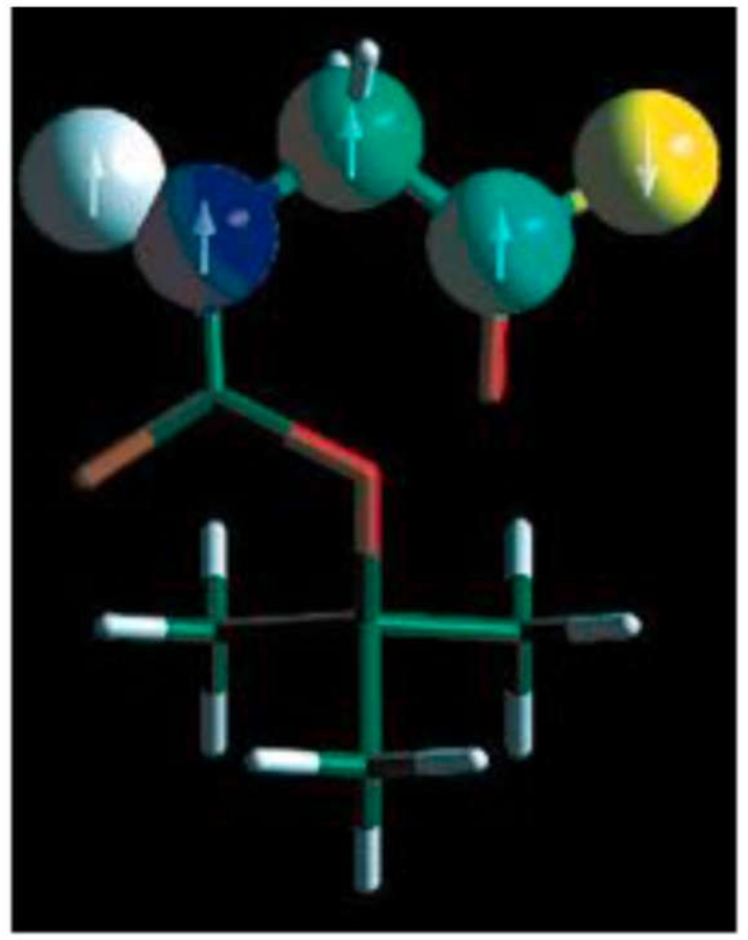
With regular simulator



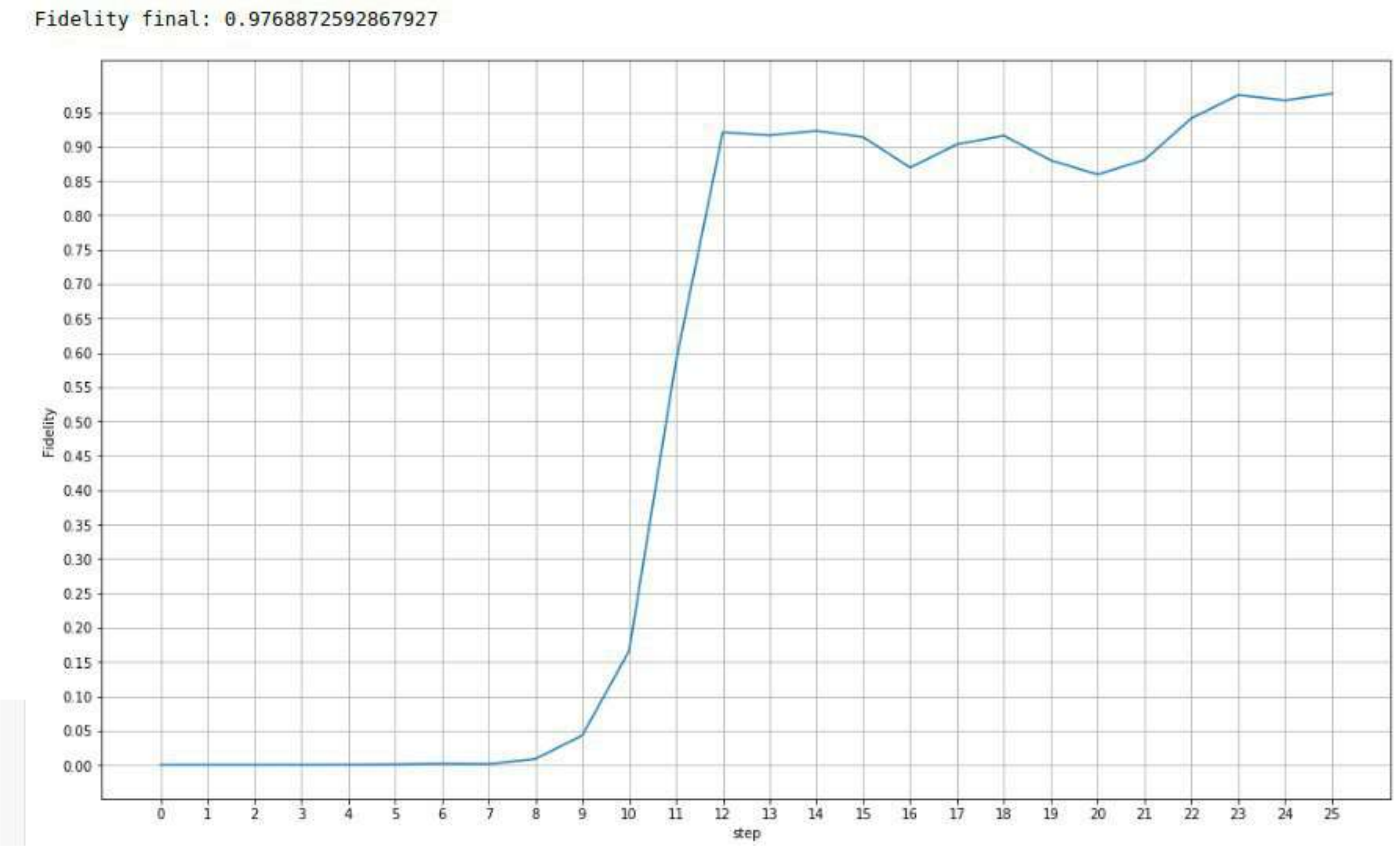
Superconducting quantum circuit

Atomic traps





NMR

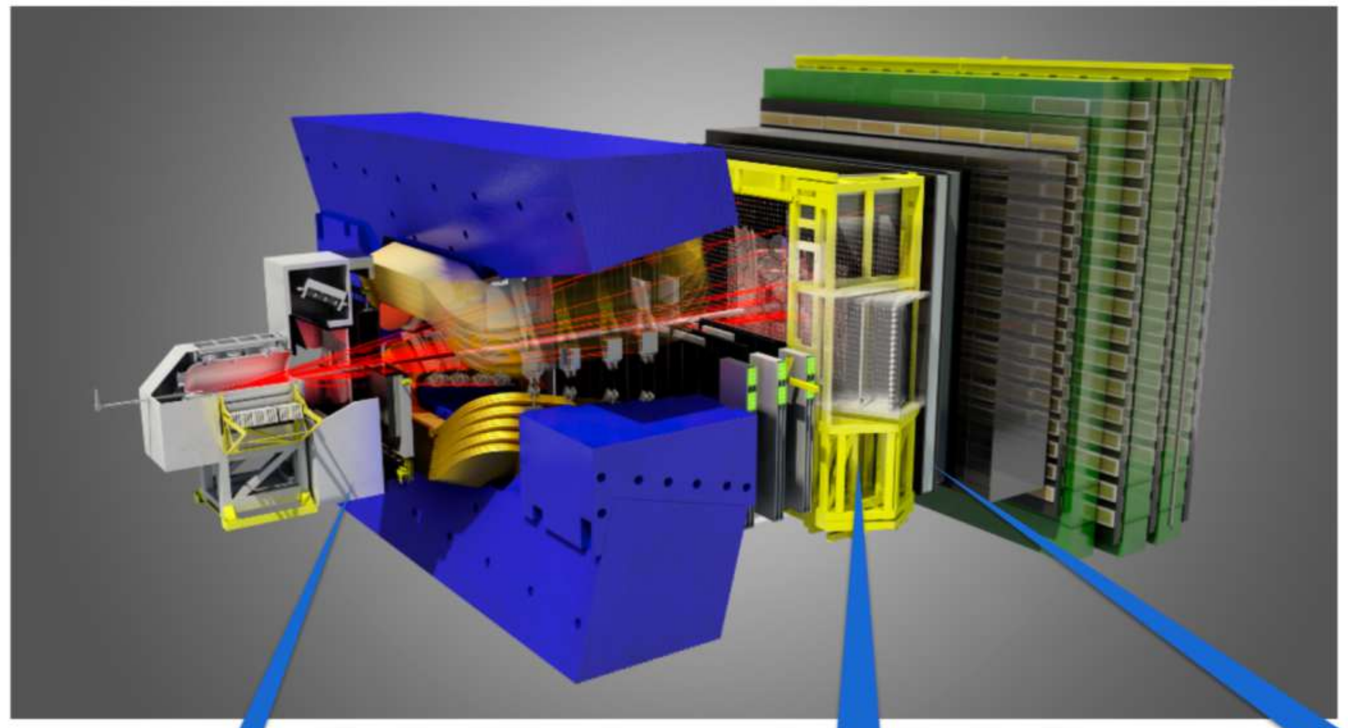


With differentiable simulator

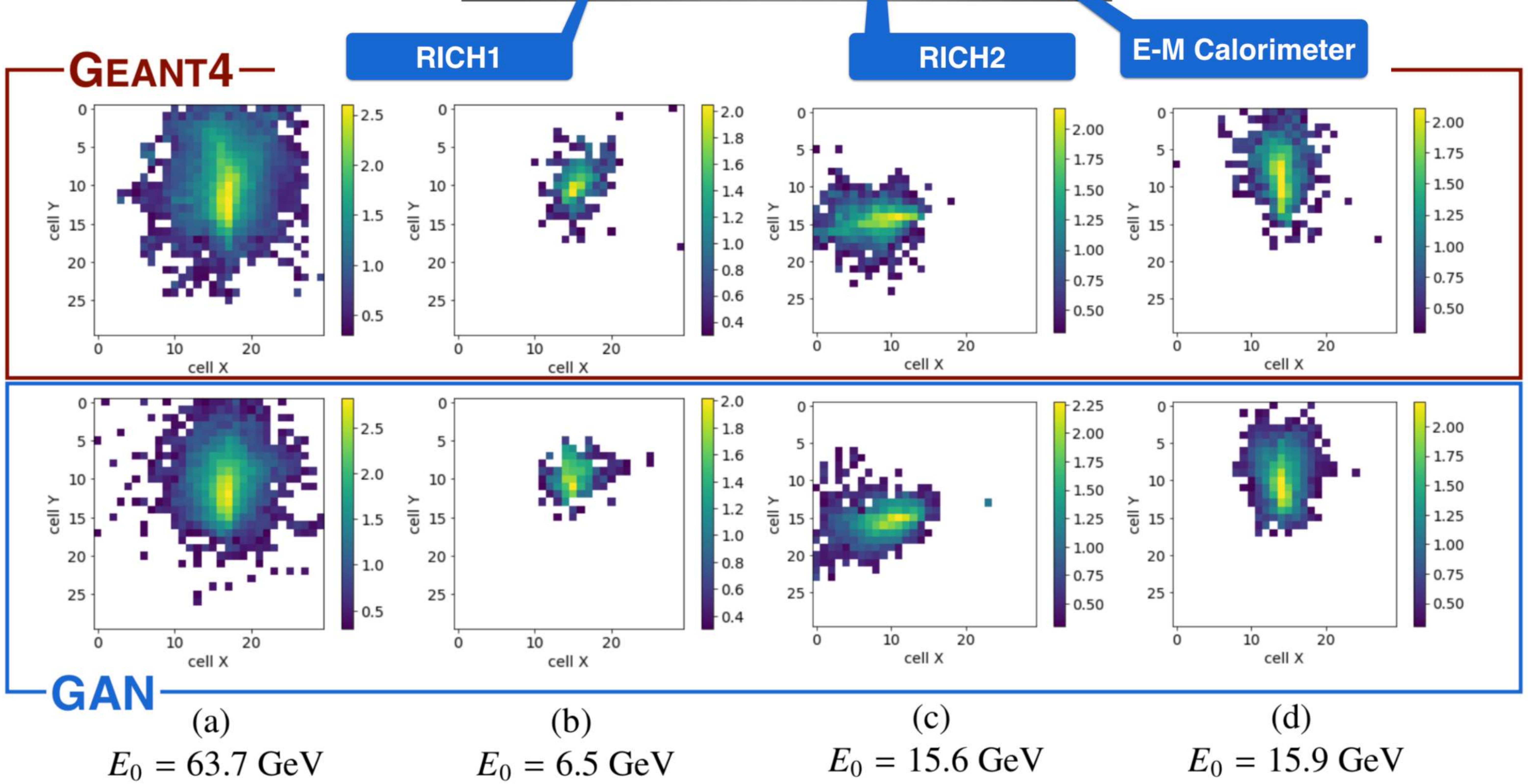
More examples. Fast LHC detector simulation

Problem: speed up simulation of particle-detector interaction

Result: simulation is 1000 times faster [4]



CERN LHCb detector



Discussion

Our lab focuses on simulation / digital twins techniques.

Open questions:

> How to embed quantitative metrics / constraints into simulators?
> How to train AI models on mixture of real and simulated data?
> How to mix detailed and surrogate simulators?
> How to make generative more transparent / explainable?
> How to extend simulation technique to embrace complexity of real-world?

Conclusion

- Explainability is the next big thing in AI development
- Knowledge/experience representation models
- Fast comprehensive simulation models (psychology, sociology, physics, economy, philosophy)
- Immersive subject representation (digital twins)
- Our lab is working on it,
    we *hire* (post docs, researchers, developers)

http://cs.hse.ru/lambda/en
anaderiRu@twitter
austyuzhanin@hse.ru

References

1. Alan Turing, "Computing machinery and intelligence", Machinery, Computing (1950): 433

2. Alison Pease, Andrew Aberdein, and Ursula Martin. The role of explanation in mathematical research, 2017. Talk, Workshop on Computer-Aided Mathematical Proof (CAMP)

3. Bonacina, Maria Paola. "Automated Reasoning for Explainable Artificial Intelligence." *ARCADE@ CADE*. 2017

4. FastCaloGAN