IoT 2025
Vienna, Austria

Active Inference for Digital Twins: Predicting and Optimising IoT Processing Service Performance

15th International Conference on the Internet of Things (IoT 2025)

Elena Pretel[1], Boris Sedlak[2], Víctor Casamayor-Pujol[2], Elena Navarro[1], Víctor López-Jaquero[1], Pascual González[1], and Schahram Dustdar[2,3]

[1] University of Castilla-La Mancha, Albacete, Spain
[2] Universitat Pompeu Fabra, Barcelona, Spain
[3] TU Wien, Vienna, Austria

Theoretical Foundations: Digital Twins (DTs)

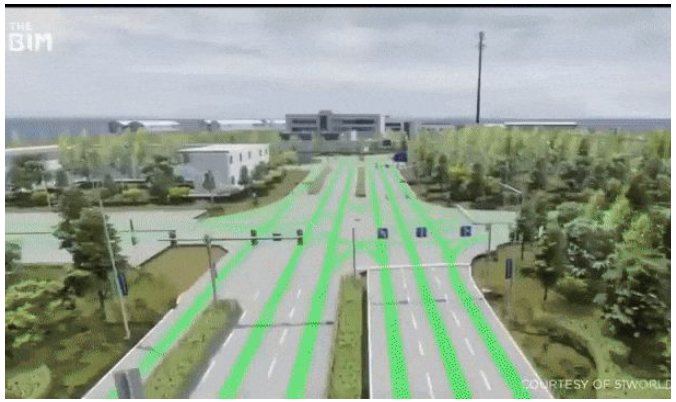
A DT is a virtual replica that allows monitoring and controlling a physical (processing) system.

https://www.unrealengine.com/en-US/spotlights/51world-creates-digital-twin-of-the-entire-city-of-shanghai

Wynne et al., Dynamic testing and analysis of the world's first metal 3D printed bridge

Theoretical Foundations: Digital Twins (DTs)

A DT is a virtual replica that allows monitoring and controlling a physical (processing) system.



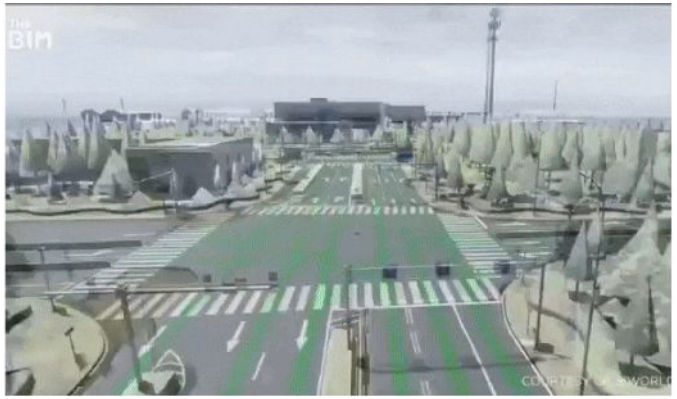
Model Shanghai in Unreal Engine

https://www.unrealengine.com/en-US/spotlights/51world-creates-digital-twin-of-the-entire-city-of-shanghai

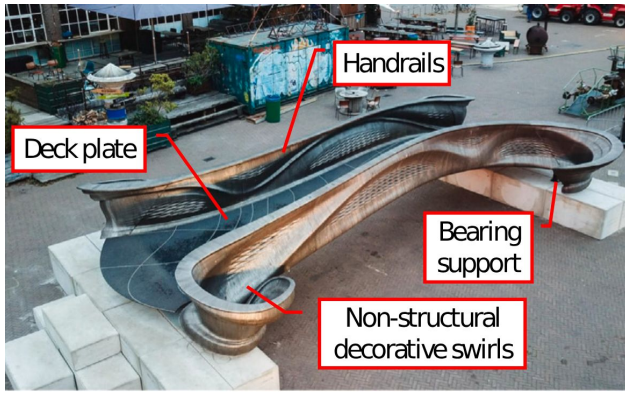
Wynne et al., Dynamic testing and analysis of the world's first metal 3D printed bridge

Theoretical Foundations: Digital Twins (DTs)

A DT is a virtual replica that allows monitoring and controlling a physical (processing) system.



Model Shanghai in Unreal Engine



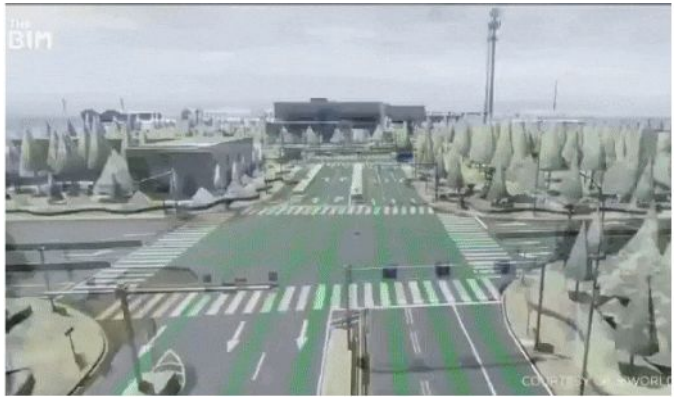
Create a bridge from a 3D printer

https://www.unrealengine.com/en-US/spotlights/51world-creates-digital-twin-of-the-entire-city-of-shanghai

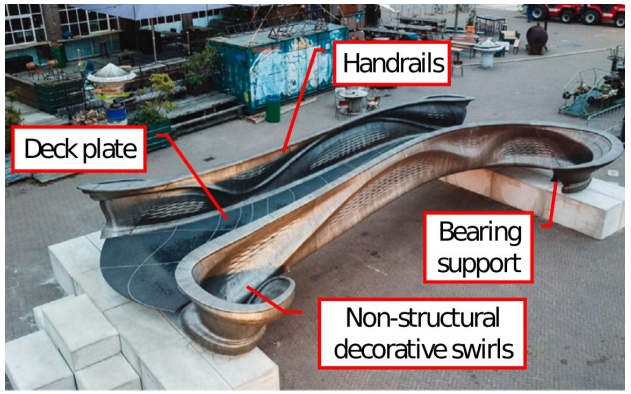
Wynne et al., Dynamic testing and analysis of the world's first metal 3D printed bridge

Theoretical Foundations: Digital Twins (DTs)

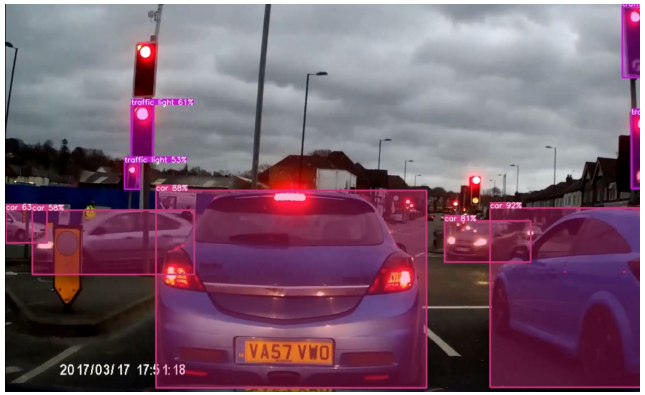
A DT is a virtual replica that allows monitoring and controlling a physical (processing) system.



Model Shanghai in Unreal Engine



Create a bridge from a 3D printer



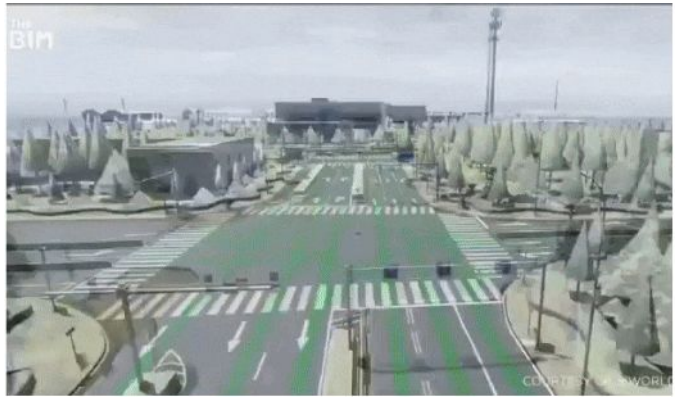
Run object detection with Yolov8

https://www.unrealengine.com/en-US/spotlights/51world-creates-digital-twin-of-the-entire-city-of-shanghai

Wynne et al., Dynamic testing and analysis of the world's first metal 3D printed bridge

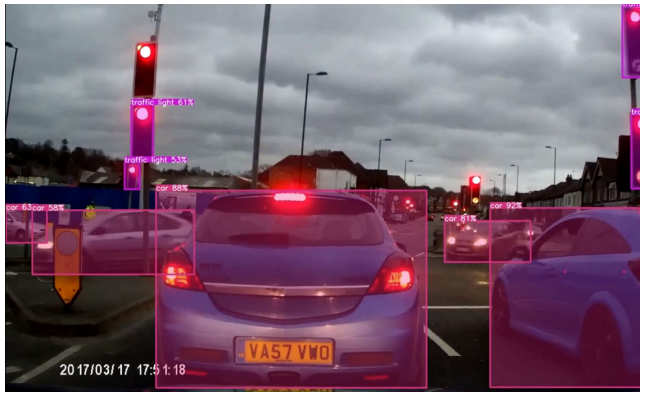
Theoretical Foundations: Digital Twins (DTs)

A DT is a virtual replica that allows monitoring and controlling a physical (processing) system.



Model Shanghai in Unreal Engine

deploy
optimise



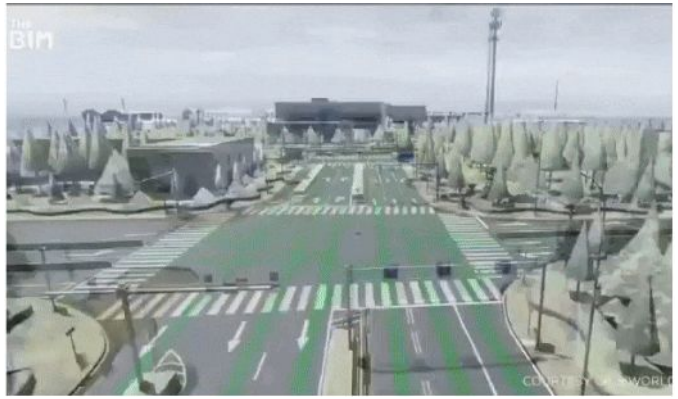
Run object detection with Yolov8

https://www.unrealengine.com/en-US/spotlights/51world-creates-digital-twin-of-the-entire-city-of-shanghai

Wynne et al., Dynamic testing and analysis of the world's first metal 3D printed bridge

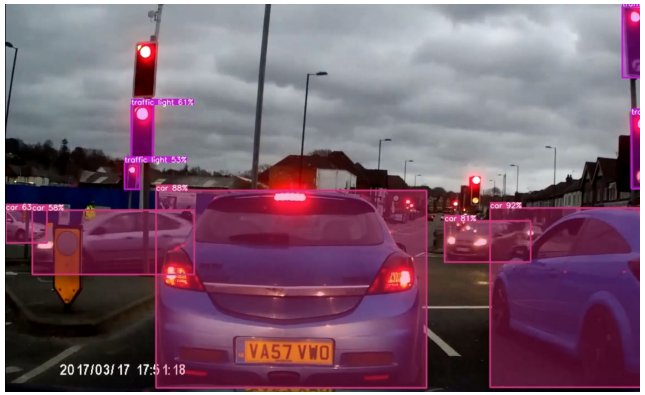
Theoretical Foundations: Digital Twins (DTs)

A DT is a virtual replica that allows monitoring and controlling a physical (processing) system.



Model Shanghai in Unreal Engine

deploy
optimise



Run object detection with Yolov8

- **Predictability** – ensure accuracy of predictions despite **changing environment**

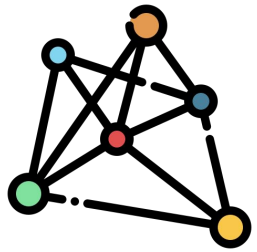
https://www.unrealengine.com/en-US/spotlights/51world-creates-digital-twin-of-the-entire-city-of-shanghai

Wynne et al., Dynamic testing and analysis of the world's first metal 3D printed bridge

So what actually is **Active Inference**?

What can it **contribute** to Digital Twins?

Theoretical Foundations: Active Inference (AIF)



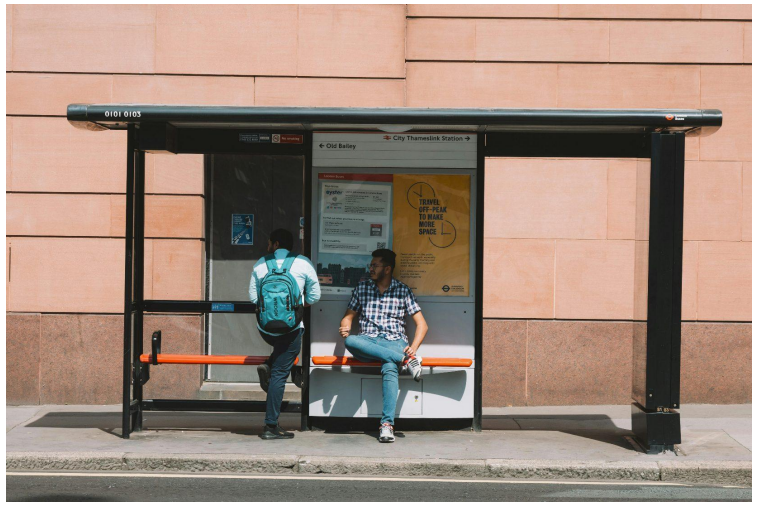
Origin: From neuroscience—systems (e.g., humans) act to minimize surprise or uncertainty. Agents create internal representations of their (processing) environment that allow them interpreting and acting efficiently.

Core idea: AIF agents constantly predict sensory inputs, compare to reality, and update their internal model. Actions not merely reactive, but reduce uncertainty.

AIF cycle:

- **Perception** → infers hidden states
- **Action** → changes the environment to meet expectations.
- **Learning** → updates beliefs from experience

Theoretical Foundations: Active Inference (AIF)



Internalize bus schedule

https://www.pexels.com/photo/men-waiting-on-bus-stop-in-london-18111366/

https://www.japan-experience.com/plan-your-trip/to-know/understanding-japan/stepping-stones

Theoretical Foundations: Active Inference (AIF)

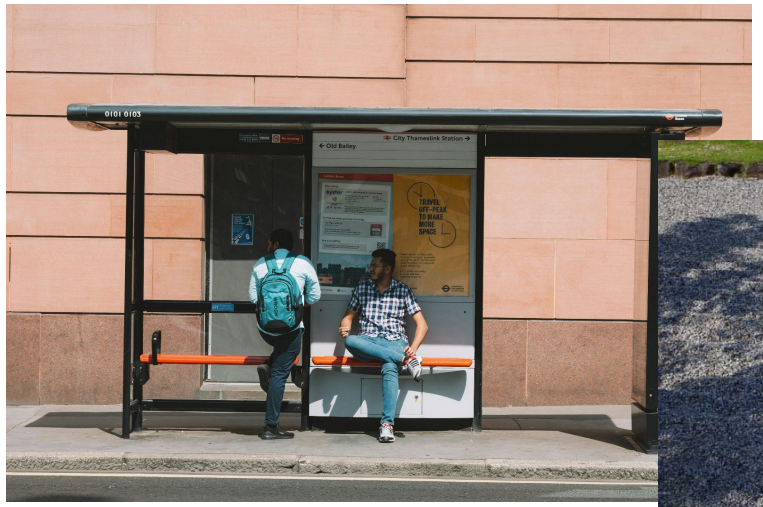
Internalize bus schedule

Building hypothesis of actions

https://www.pexels.com/photo/men-waiting-on-bus-stop-in-london-18111366/

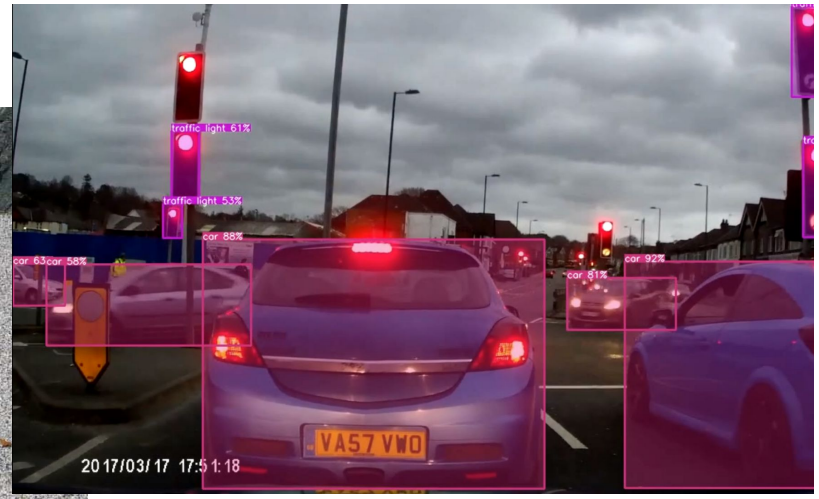
https://www.japan-experience.com/plan-your-trip/to-know/understanding-japan/stepping-stones

Theoretical Foundations: Active Inference (AIF)



Internalize bus schedule

Building hypothesis of actions



Predict IoT Performance

https://www.pexels.com/photo/men-waiting-on-bus-stop-in-london-18111366/

https://www.japan-experience.com/plan-your-trip/to-know/understanding-japan/stepping-stones

Theoretical Foundations: Active Inference (AIF)

Create Digital Twin by **using** X

X = {Physics Engine, Kafka, Active Inference, etc}

Using Active Inference → creating a Digital Twin

Theoretical Foundations: Active Inference (AIF)

~~Create Digital Twin by **using** X~~

X = {Physics Engine, Kafka, Active Inference, etc}

Using Active Inference ⟶ creating a Digital Twin

Proposal and Architecture

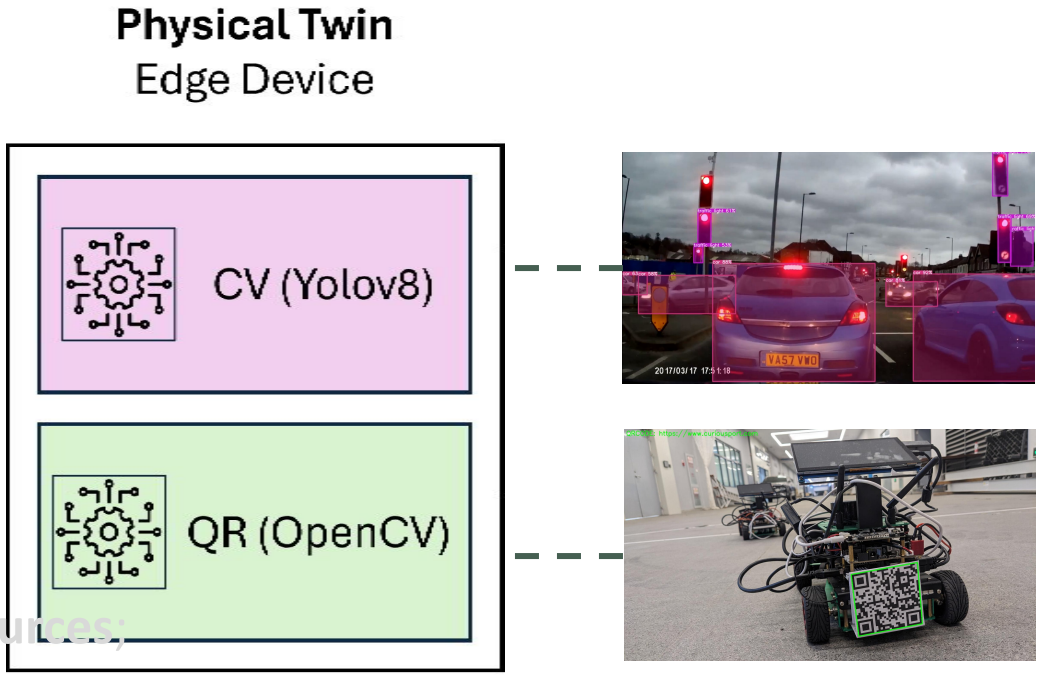
Case Study: AIF-Enabled DT in Edge Computing

Physical twin: Edge device with limited CPU power

Two concurrent processing services:
- YOLOv8 video inference
- QR code reading service

Observe and configure services:
- CV: throughput, data quality, cores, model size
- QR: throughput, data quality, cores

Both services **compete for limited CPU resources**; thus, creating a dynamic optimization scenario



Proposal and Architecture

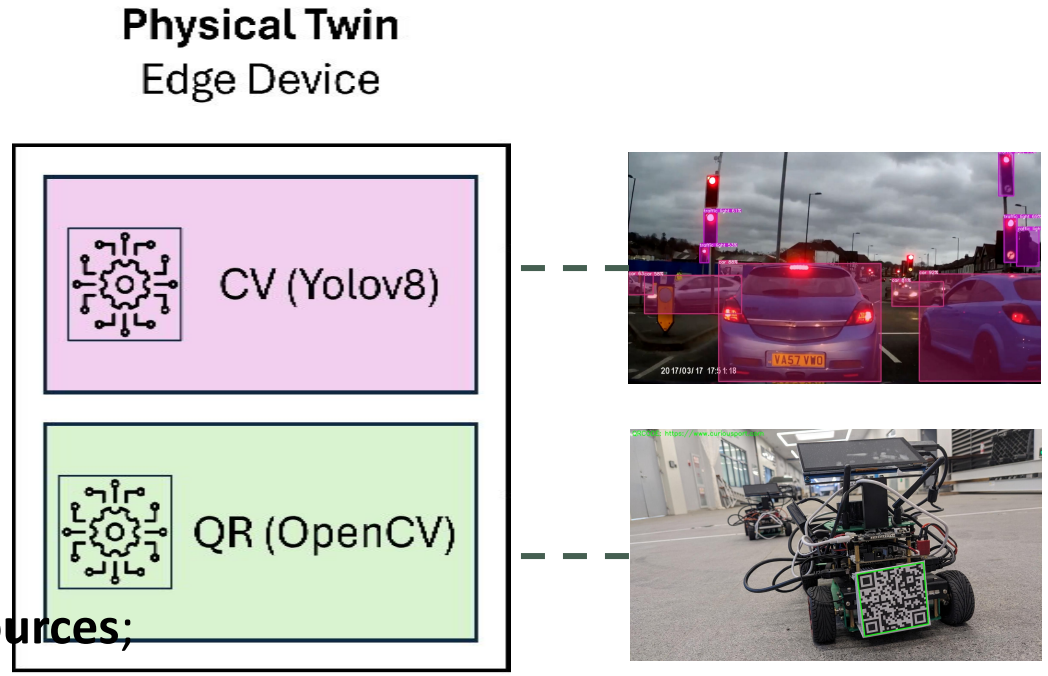
Case Study: AIF-Enabled DT in Edge Computing

Physical twin: Edge device with limited CPU power

Two concurrent processing services:
- YOLOv8 video inference
- QR code reading service

Observe and configure services:
- CV: throughput, data quality, cores, model size
- QR: throughput, data quality, cores

Both services **compete for limited CPU resources**; thus, creating a dynamic optimization scenario



Proposal and Architecture

Case Study: AIF-Enabled DT in Edge Computing

Physical twin: Edge device with limited CPU power

Two concurrent processing services:

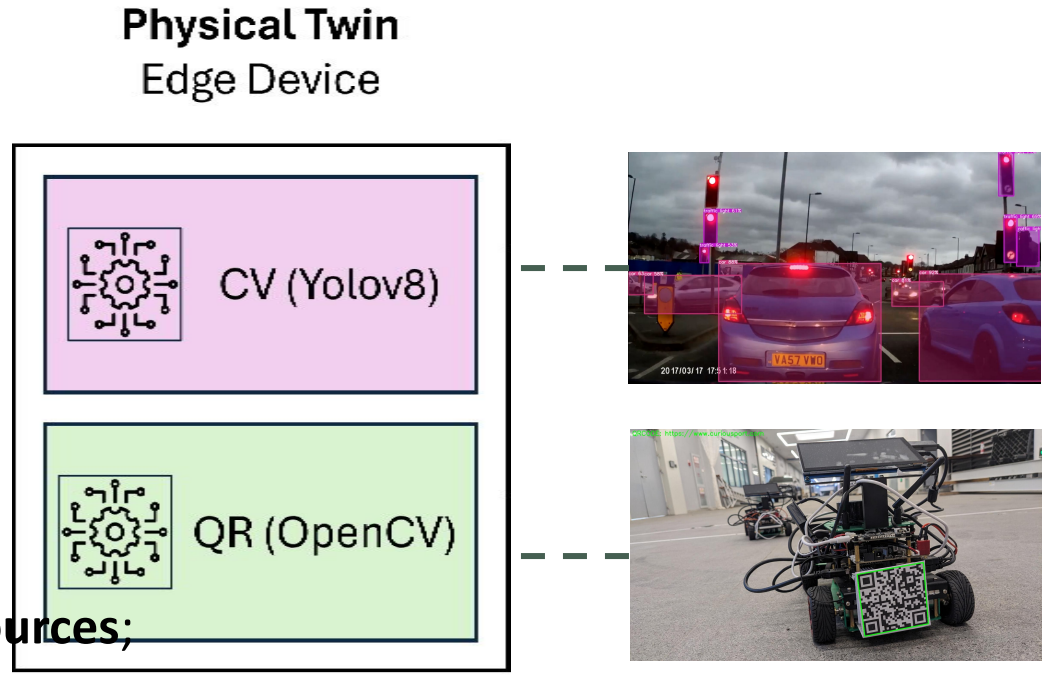
- YOLOv8 video inference
- QR code reading service

Observe and configure services:

- CV: throughput, data quality, cores, model size
- QR: throughput, data quality, cores

Predict Actions

Both services **compete for limited CPU resources**; thus, creating a dynamic optimization scenario



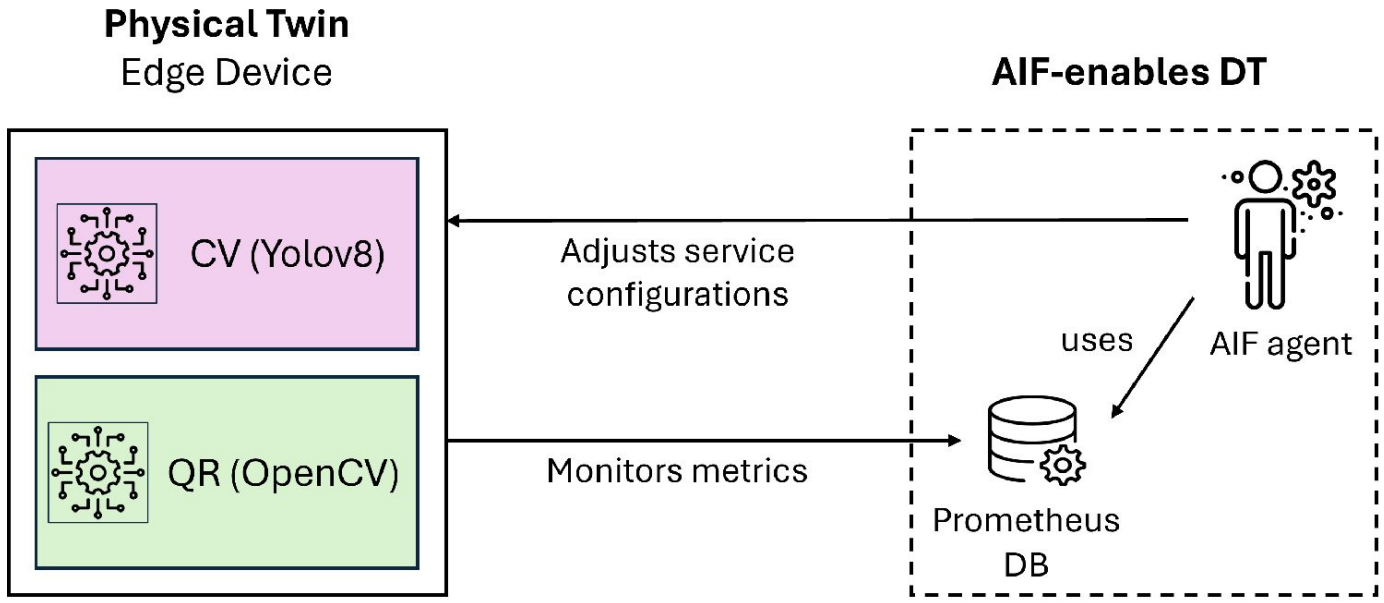
Proposal and Architecture

Architecture of the AIF-Enabled DT

Prometheus Database: collects and stores real-time service metrics (CPU usage, throughput, data quality, etc.)

AIF Agent: use service observations, update beliefs, and act to optimize model accuracy and service performance

Closed-loop control: DT observes → infers → acts → updates continuously



Proposal and Architecture

AIF agent is modeled as a POMDP

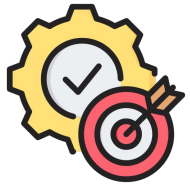
Component	Details / Variables in this Case Study
States (S)	CV throughput, CV data quality, CV cores, CV model size, QR throughput, QR data quality, QR cores. ***Constraint**:* CV_cores + QR_cores ≤ 8.
Actions (U)	- **CV actions (U_CV):** increase/decrease *quality*, *CPU cores*, *model size*, or *no-op*. - **QR actions (U_QR):** increase/decrease *quality*, *CPU cores*, or *no-op*. Combined as a **joint action** (U = [U_CV, U_QR]).
Transition Model (B)	Encodes probability of moving from one state to another after an action, give small prior about variable dependencies (e.g., cores → throughput)
Preferences (C)	Define the agents "goals" or Service Level Objectives (SLOs). - **CV:** throughput ≥ 3 FPS, high quality ≥ 288px, large model ≥ YOLOv8m. - **QR:** throughput ≥ 40 FPS, high quality ≥ 800px.

Proposal and Architecture

AIF agent is modeled as a POMDP

Component	**Details / Variables in this Case Study**
States (S)	CV throughput, CV data quality, CV cores, CV model size, QR throughput, QR data quality, QR cores. ***Constraint**:* CV_cores + QR_cores ≤ 8.
Actions (U)	- **CV actions (U_CV):** increase/decrease *quality*, *CPU cores*, *model size*, or *no-op*. - **QR actions (U_QR):** increase/decrease *quality*, *CPU cores*, or *no-op*. Combined as a **joint action** (U = [U_CV, U_QR]).
Transition Model (B)	Encodes probability of moving from one state to another after an action, give small prior about variable dependencies (e.g., cores → throughput)
Preferences (C)	Define the agents "goals" or Service Level Objectives (SLOs). - **CV:** throughput ≥ 3 FPS, high quality ≥ 288px, large model ≥ YOLOv8m. - **QR:** throughput ≥ 40 FPS, high quality ≥ 800px.

Experiments and Results



Goal: Validate whether the **AIF-enabled DT** can

- **Predict** the physical twin's performance (RQ1)
- **Take** actions to meet **Service Level Objectives (SLOs)** (RQ2)

Experimental Setup

- **Platform**: Edge device (8 CPU cores) running two services
- **Environment**: Docker containers + Prometheus monitoring
- **AIF implementation**: pymdp library
- **Duration**: 50 AIF cycles per experiment, 30 repetitions with different seeds
- **Metrics**: RMSE (prediction accuracy), SLO fulfillment rate (%)

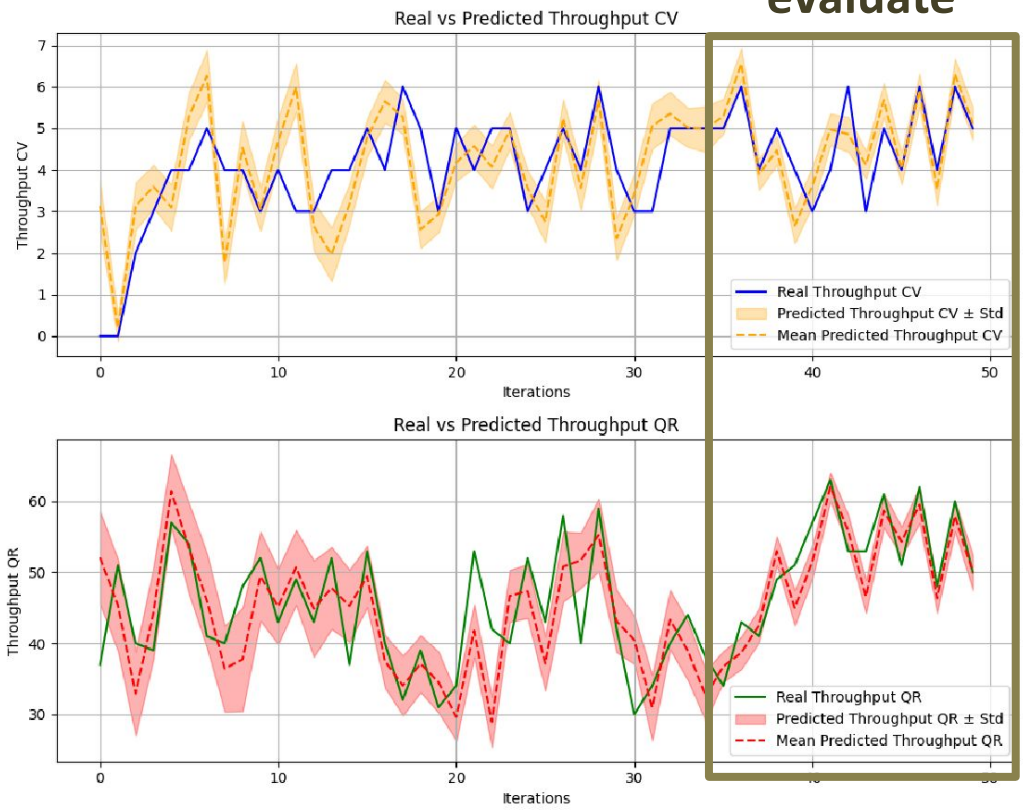
Experiments and Results

Experiment 1 (RQ1):
Predict the physical twin's performance (predicted vs. real throughput) in **last 15** iterations.

First 35 iterations learning transition model

Test	Metric	p-value	Statistical Conclusion
Mann–Whitney U	Distributions with α=0.05	CV: 0.491 QR: 0.087	No significant difference **predictions ≈ real values**

evaluate

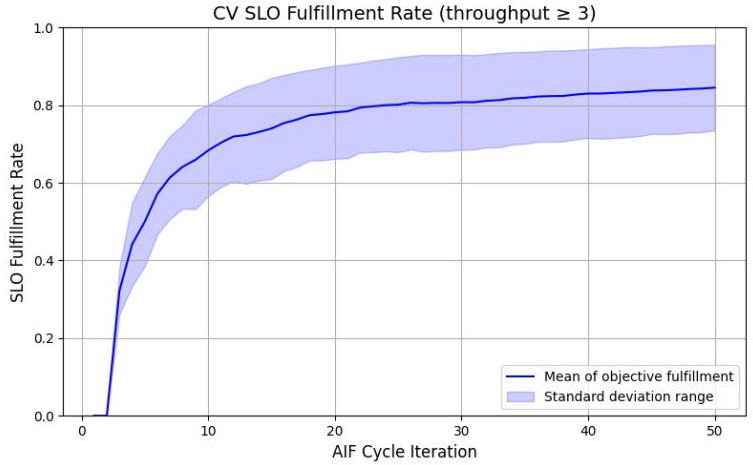


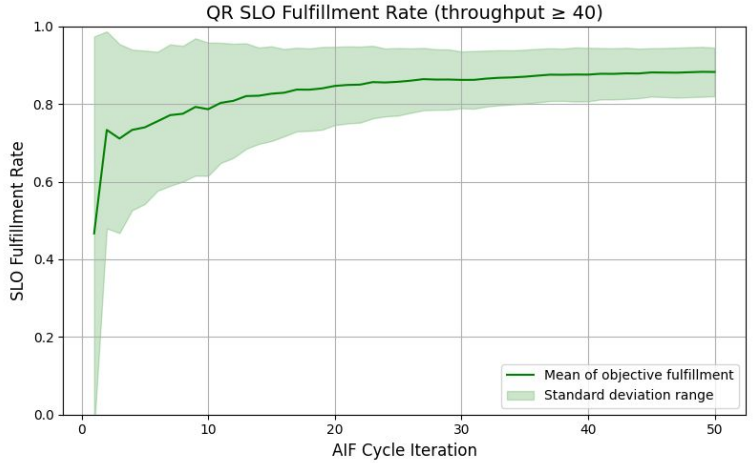
Experiments and Results

Experiment 2 (RQ2):
Use DT to optimize physical twin's performance (i.e., SLOs).

AIF agent starts with minimal prior understanding
(i.e., hint on variable dependency, but no **distributions**)

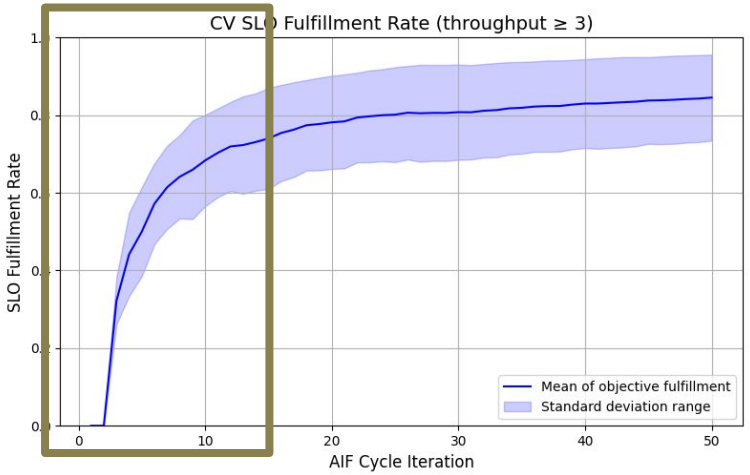
Test	Metric	p-value	Statistical Conclusion
Wilcoxon Signed-Rank	Median vs. 0.8 threshold	CV: 0.001 QR: 0.010	Median > 0.8 **Objectives met significantly**





Experiments and Results

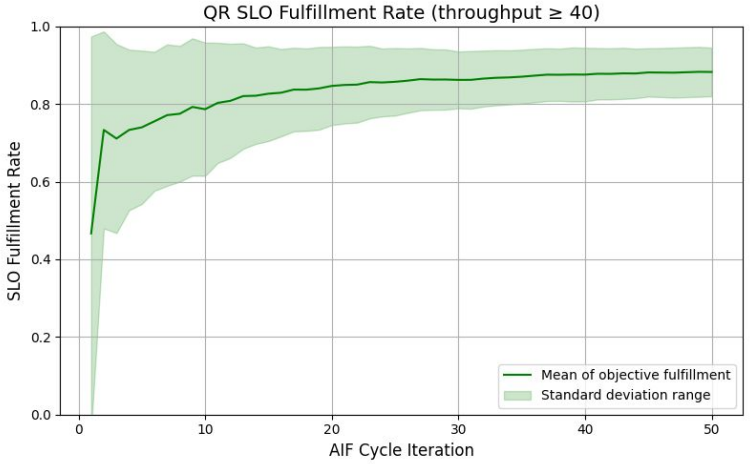
steep!



Experiment 2 (RQ2):
Use DT to optimize physical twin's performance (i.e., SLOs).

AIF agent starts with minimal prior understanding
(i.e., hint on variable dependency, but no **distributions**)

Test	Metric	p-value	Statistical Conclusion
Wilcoxon Signed-Rank	Median vs. 0.8 threshold	CV: 0.001 QR: 0.010	Median > 0.8 **Objectives met significantly**



Conclusion and Future Work

Complexity of IoT systems requires real-time monitoring and control.

Introduced a **novel approach** to enhance **predictability** of DT using **Active Inference (AIF)**.

Proposed an **AIF-enabled DT architecture** capable of **perception, prediction, learning and control** in dynamic IoT environments.

Validated that the **AIF-enabled DT**:

- Accurately **predicts** physical twin performance (RQ1).
- Effectively **selects actions** to meet Service Level Objectives (RQ2).
- Achieves >80% **SLO fulfillment** across experiments (RQ2).

Future Work: **Compare AIF-enabled DTs** with contemporary ML solutions (e.g., RL).