ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

LAB05: On device learning

Davide Nadalini – d.nadalini@unibo.it
Lorenzo Lamberti – lorenzo.lamberti@unibo.it
Alberto Dequino – alberto.dequino@unibo.it
Francesco Conti – f.conti@unibo.it

Objective of the Class

Intro: On-Device Learning on MicroControllers

Tasks: Implement the C code to train a simple DNN on-device:

- Visualize tensors
- Forward, loss, backward, update: familiarize with the C code
- Implement a loss function on-device
- Train a DNN the GVSoC simulator with sample data and label

Programming Language: C

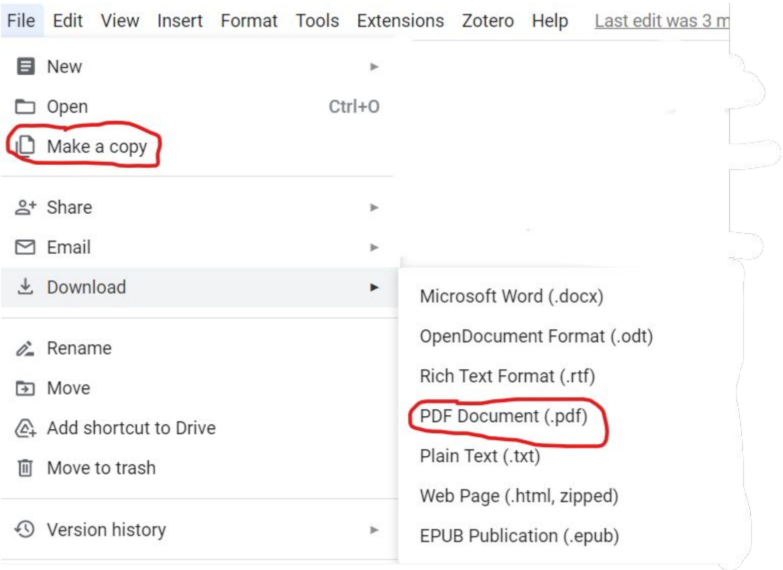
Lab duration: 3h

The class is meant to be interactive: coding together and on your own!

How to deliver the Assignment

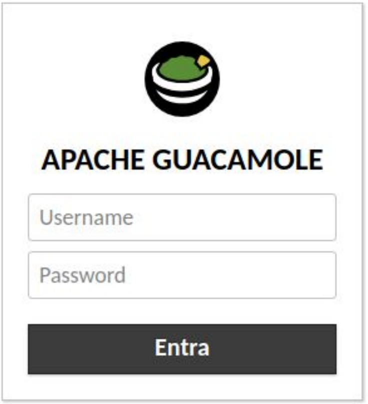
You will deliver ONLY the GDOC assignment, no code

- Copy the google doc to your drive, so that you can modify it. (File -> make a copy)
- Fill the tasks on this google doc.
- Export to pdf format.
- Rename the file to: LAB<number_of_the_lesson>_APAI_<your_name>.pdf
- Use Virtuale platform to load ONLY your .pdf file



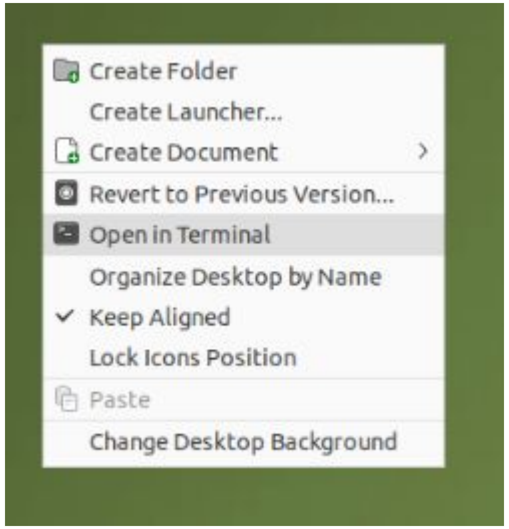
SETUP: How to access the server

1. Open this web page: https://compute.eees.dei.unibo.it:8443/guacamole/ **(works only from ALMA WIFI NETWORK!)**



2. Login. We distribute credentials by hand.

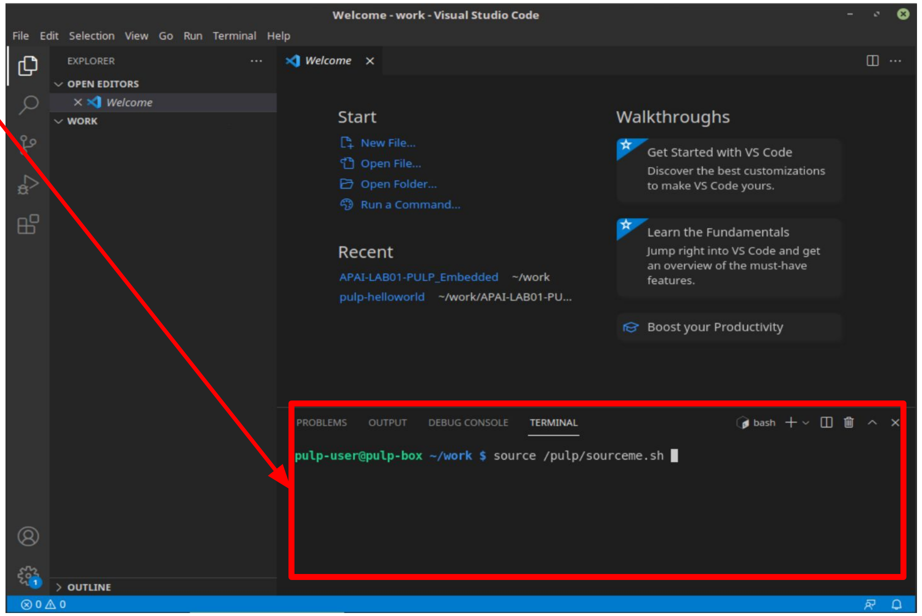
3. Open a terminal (right click – open a new terminal)



4. Open a text editor (For example "VSCode"):

```
$ code .
```

Now you can use the integrated terminal to run your applications!



Discover the best customizations to make VS Code yours.

Jump right into VS Code and get an overview of the must-have features.

```
pulp-user@pulp-box ~/work $ source /pulp/sourceme.sh
```

IMPORTANT: activate the pulp-sdk module file every time a new shell is open.

```
$ module load pulp-sdk
```

SETUP: How to access the server

1. Open this web page: https://compute.eees.dei.unibo.it:8443/guacamole/
   (works only from ALMA WIFI NETWORK!)

2. Login. We distribute credentials by hand.

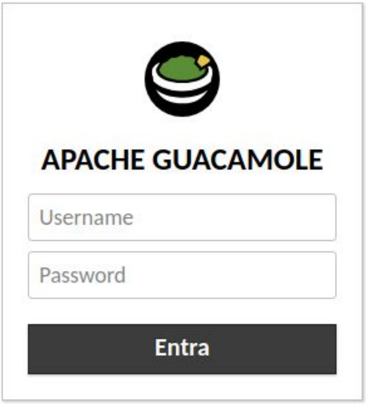
3. Open a terminal (right click – open a new terminal)

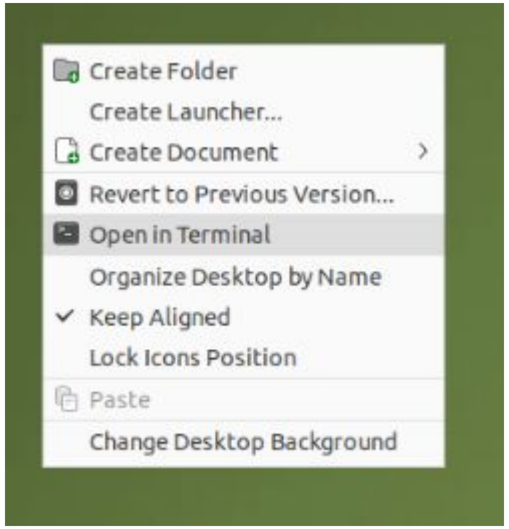
4. Clone:
   `git clone https://github.com/EEESlab/<insert_here_the_right_repo!>`

5. `module load pulp-sdk`

6. `cd <insert_here_the_right_repo!>`

7. `make clean all run`





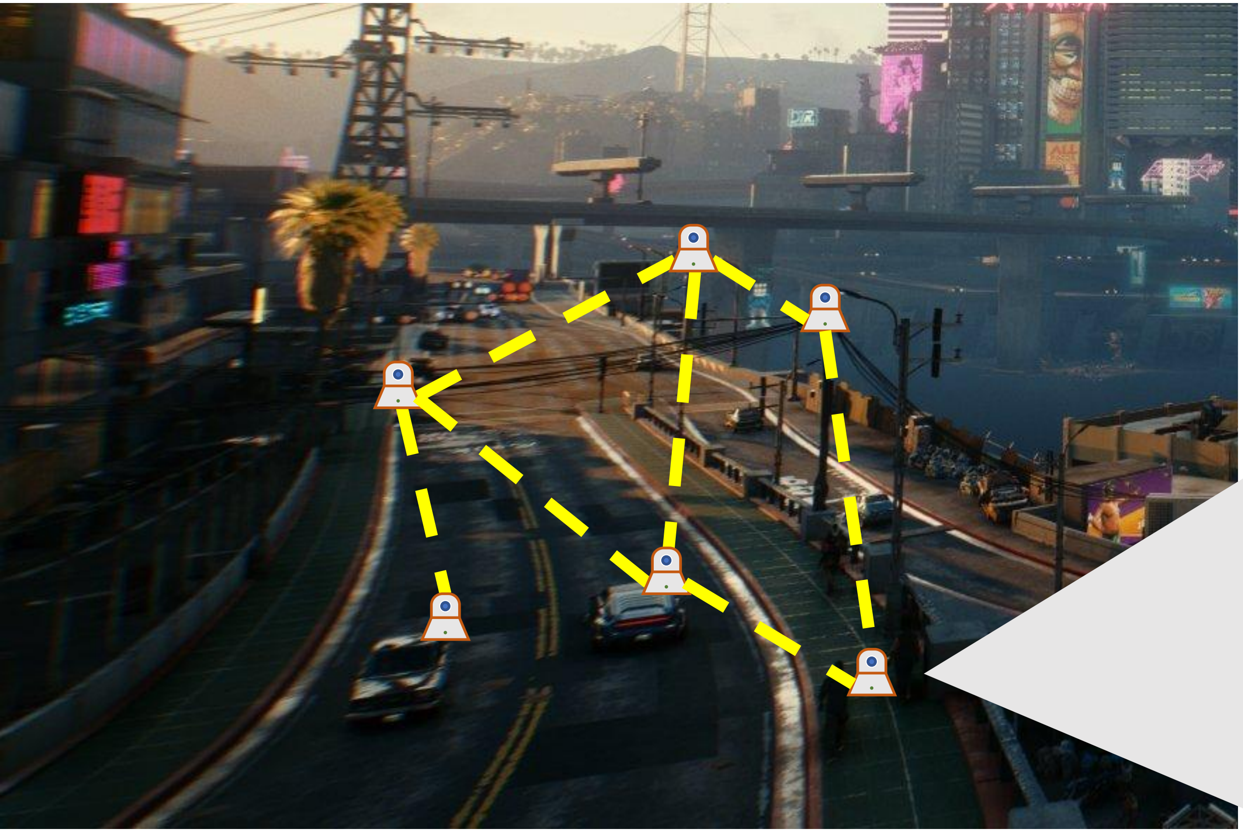
ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

On-Device Learning with PULP-TrainLib

On-Device Learning (ODL)

“The process of **locally optimizing AI** models deployed on **Edge IoT Devices**”

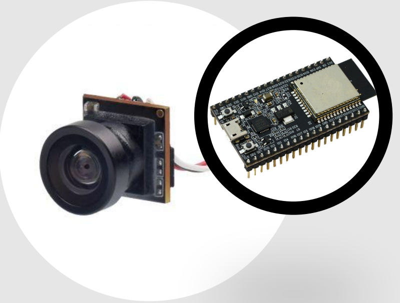
The Internet of Things (IoT)



IoT

“The process of connecting everyday physical objects to the internet”[1]

Miniaturized **low-power** and **low-cost edge** sensor nodes embedded in physical systems



[1] https://www.redhat.com/en/topics/internet-of-things/what-is-iot

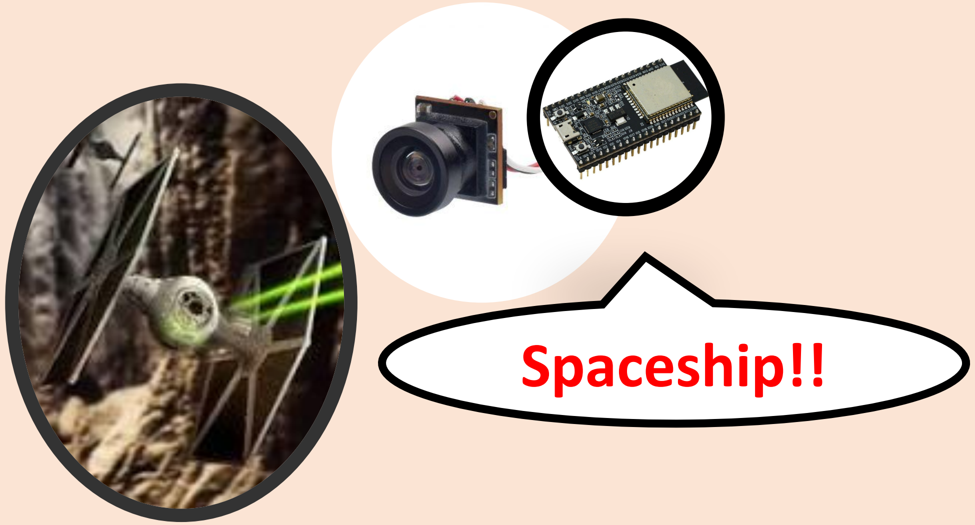
IoT Applications – Examples

KEYWORD SPOTTING

Identify & React to user’s keywords

IMAGE CLASSIFICATION

Detect the class of an image’s content



IoT Applications – Examples (cont’d)

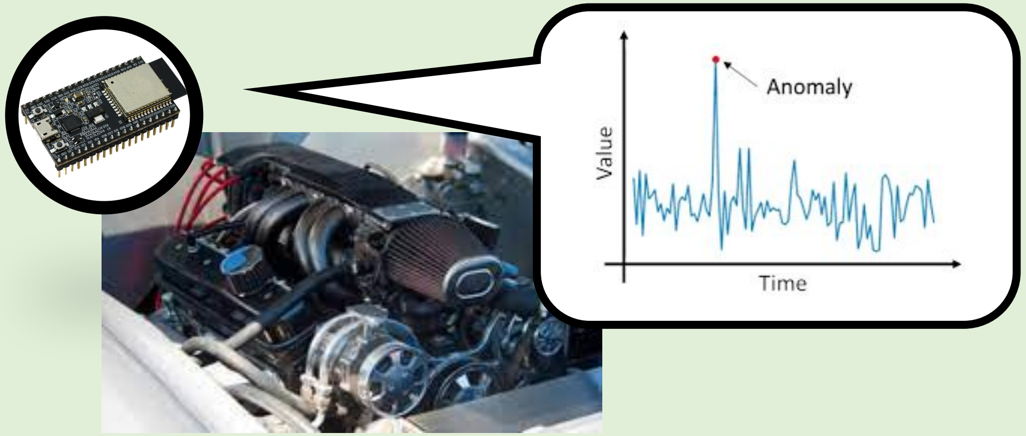
PERSON DETECTED!!

VISUAL WAKE WORDS

Detect the presence of a key object

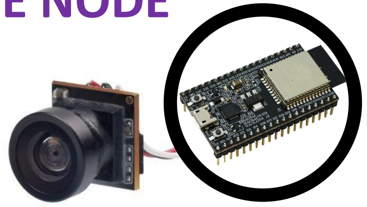
ANOMALY DETECTION

Detect when a signal indicates an anomaly



IoT Sensor Nodes: Tiny Machine Learning

LOW-POWER EDGE NODE

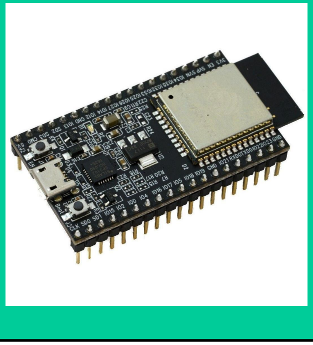


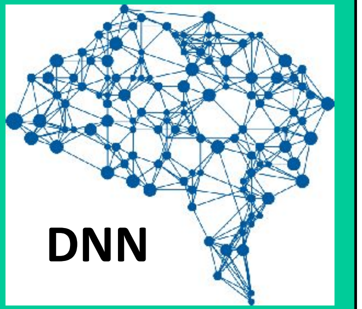
Spaceship!!

CAMERA & SENSORS

MCU
< 1 W
< 256 kB

A.I.





LOW-RESOLUTION CAMERAS AND SENSORS

ULTRA-LOW POWER

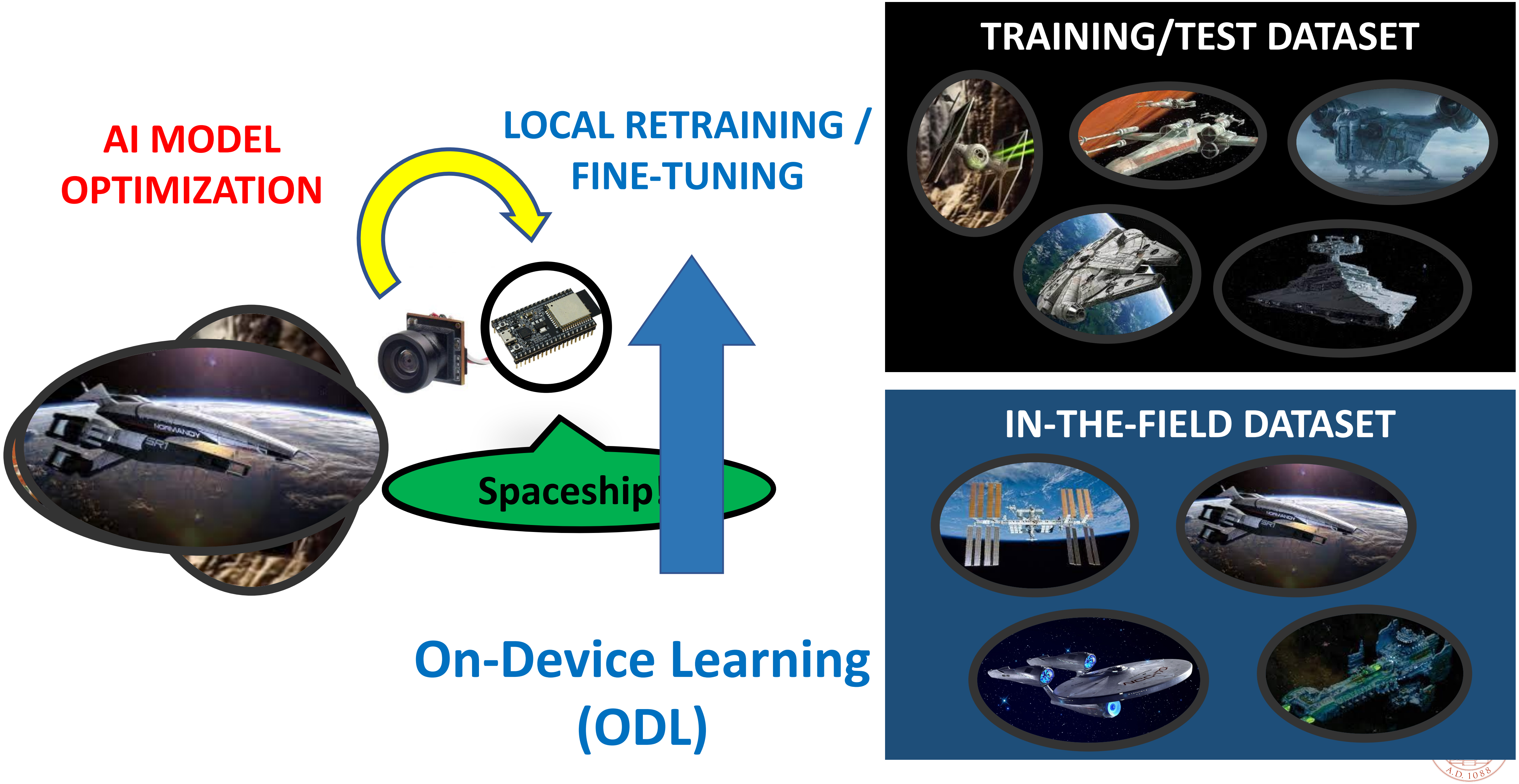
SMALL MEMORIES

LIMITED COMPUTATIONAL RESOURCES

TINY MACHINE LEARNING

Reduce the AI requirements to fit tiny Devices

The inference problem



AI MODEL OPTIMIZATION

LOCAL RETRAINING / FINE-TUNING

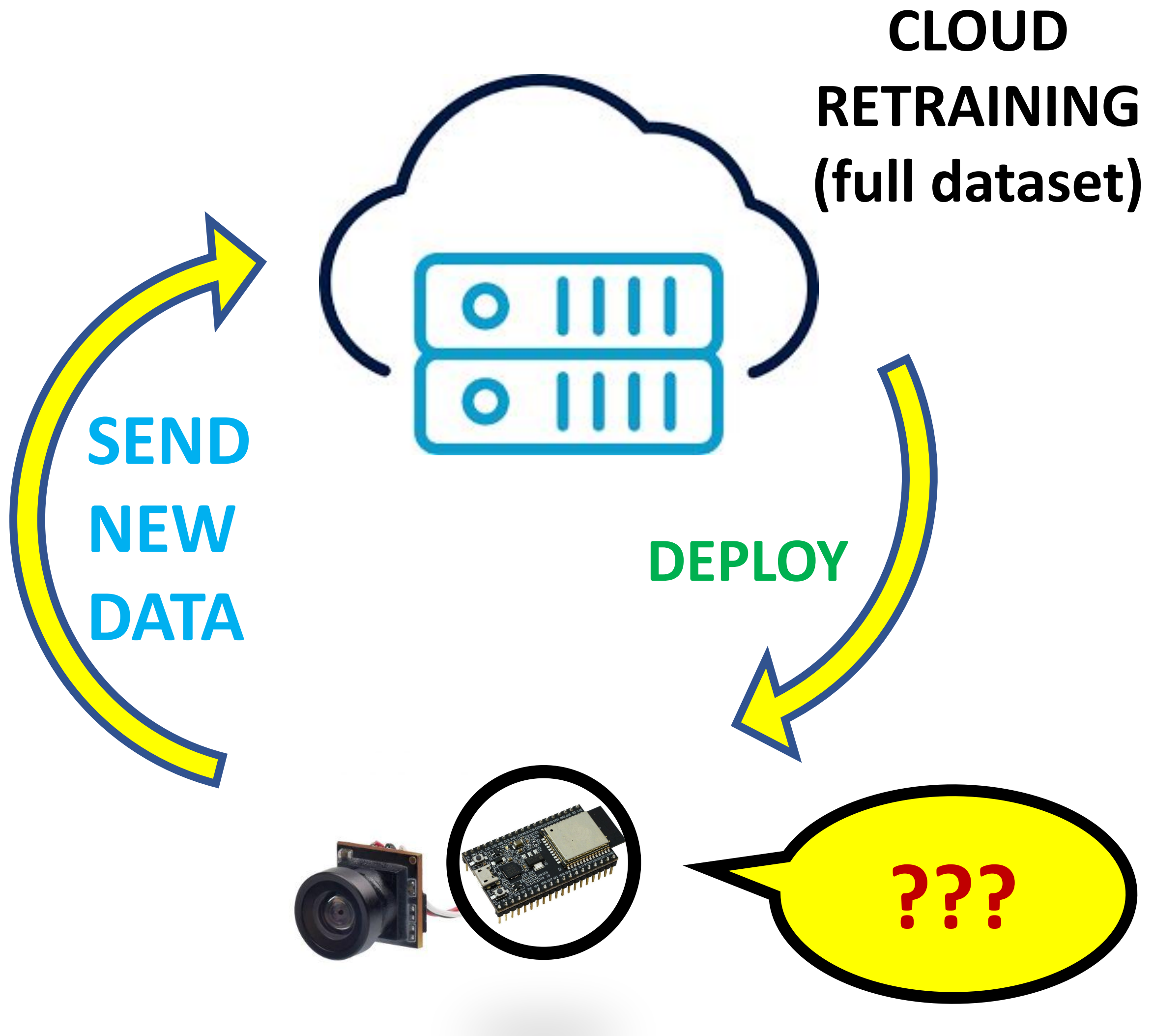
TRAINING/TEST DATASET

IN-THE-FIELD DATASET

On-Device Learning (ODL)

On-Device Learning vs Training on Cloud

TRAINING ON CLOUD



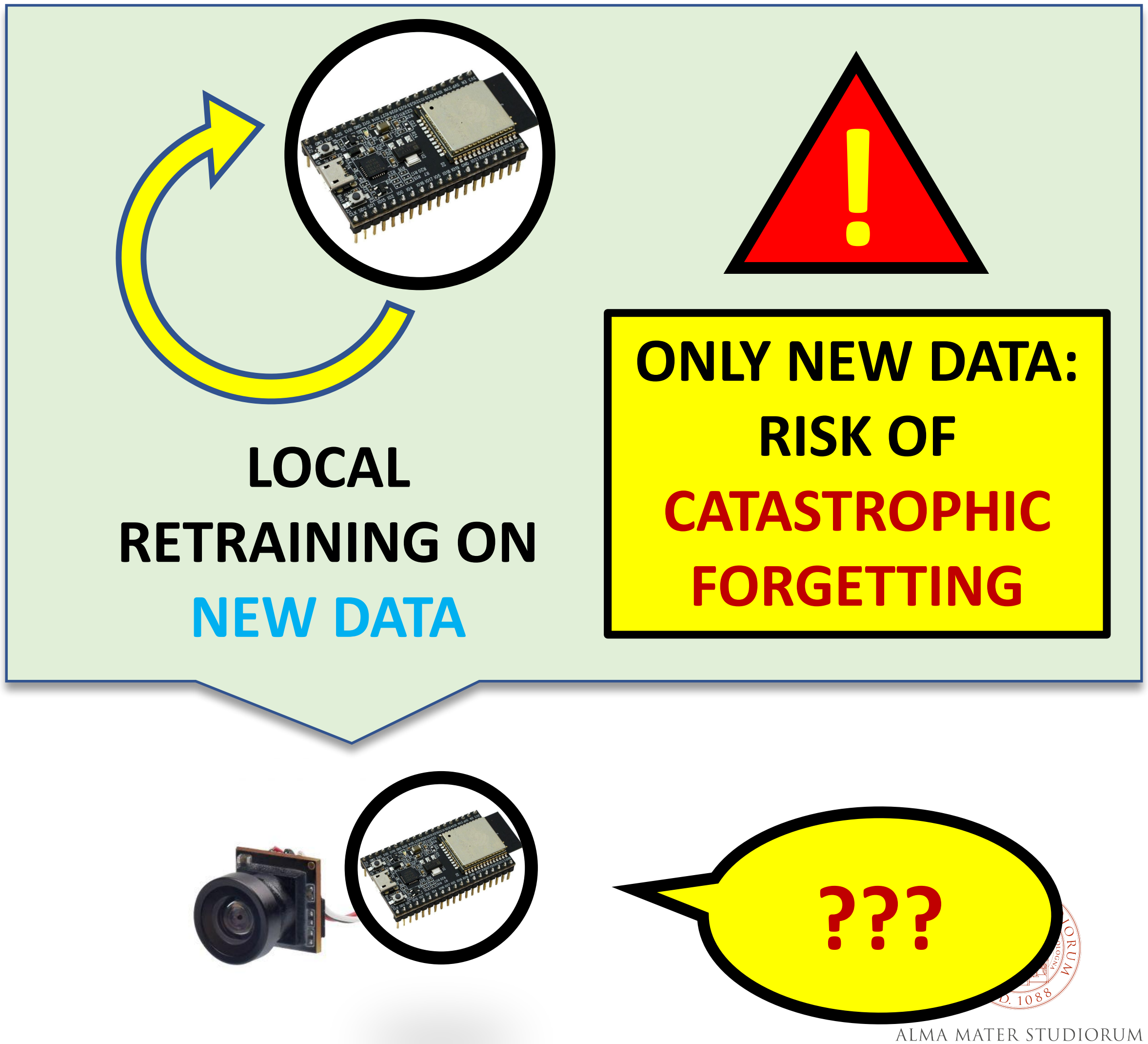
CLOUD RETRAINING (full dataset)

SEND NEW DATA

DEPLOY

???

ON DEVICE LEARNING



LOCAL RETRAINING ON NEW DATA

ONLY NEW DATA: RISK OF CATASTROPHIC FORGETTING

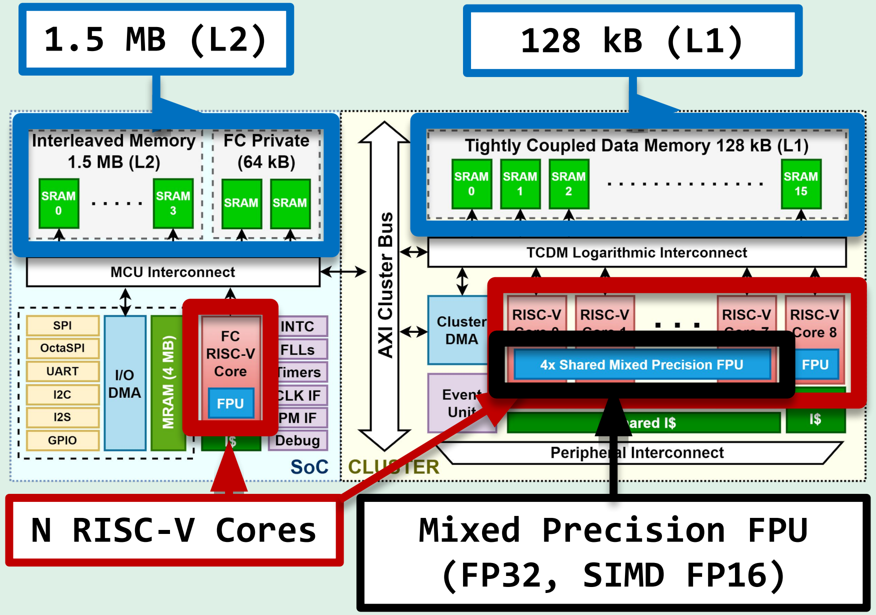
???

ADVANTAGES OF On Device Learning

- ✓ Network Scalability
- ✓ User Data Privacy
- ✓ Latency of DNN Update

On-Device Learning on PULP

Our target: the PULP Platform[1]



PULP (Parallel Ultra-Low-Power)

Scalable, energy-efficient edge computing based on **RISC-V cores**.

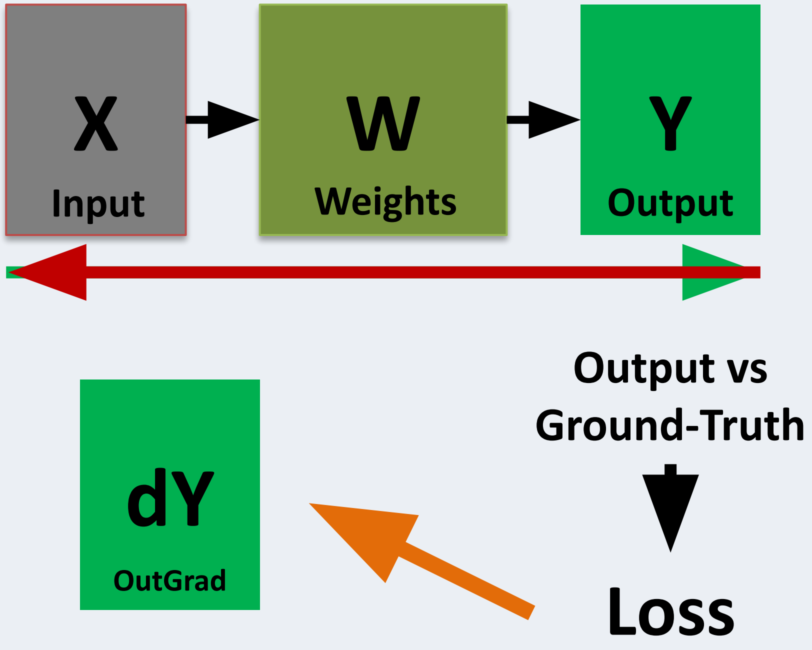
Our task: The Backpropagation Algorithm

Step 1: Forward

Step 2: Loss

Step 3: Out grad

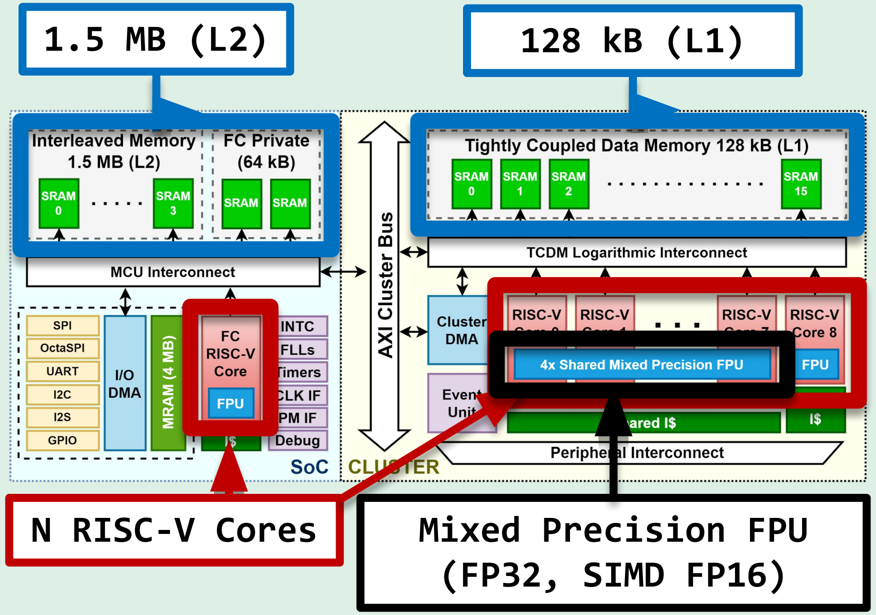
Step 4: Backward



[1] PULP-Platform: https://pulp-platform.org/

On-Device Learning on PULP

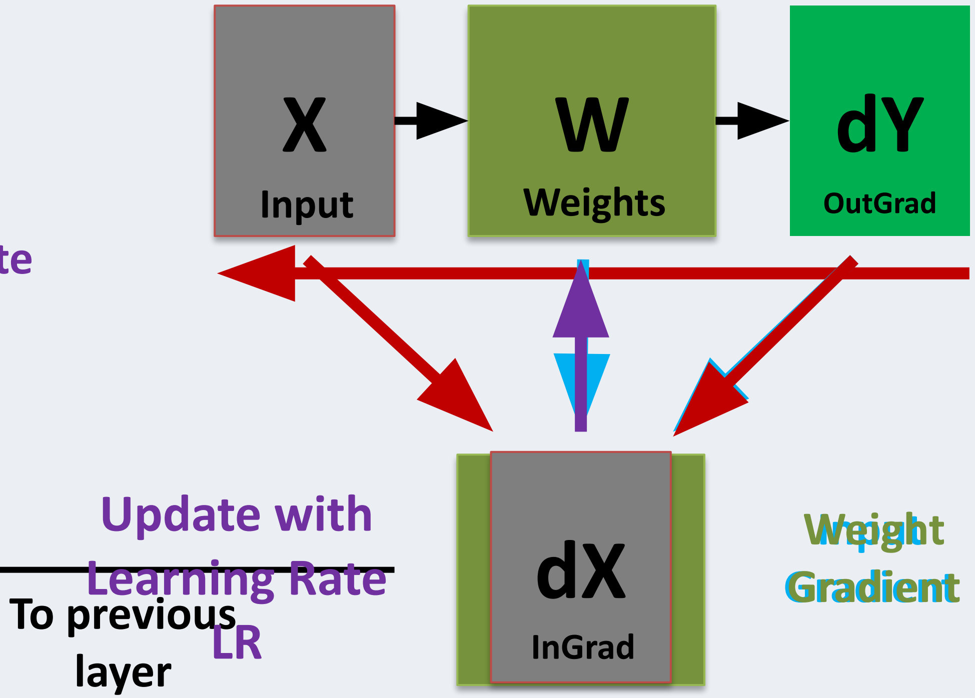
Our target: the PULP Platform[1]



PULP (Parallel Ultra-Low-Power)
Scalable, energy-efficient edge computing based on **RISC-V cores**.

Our task: The Backpropagation Algorithm

Step 1: Forward
Step 2: Loss
Step 3: Out grad
Step 4: Backward
Step 5: Weight Update



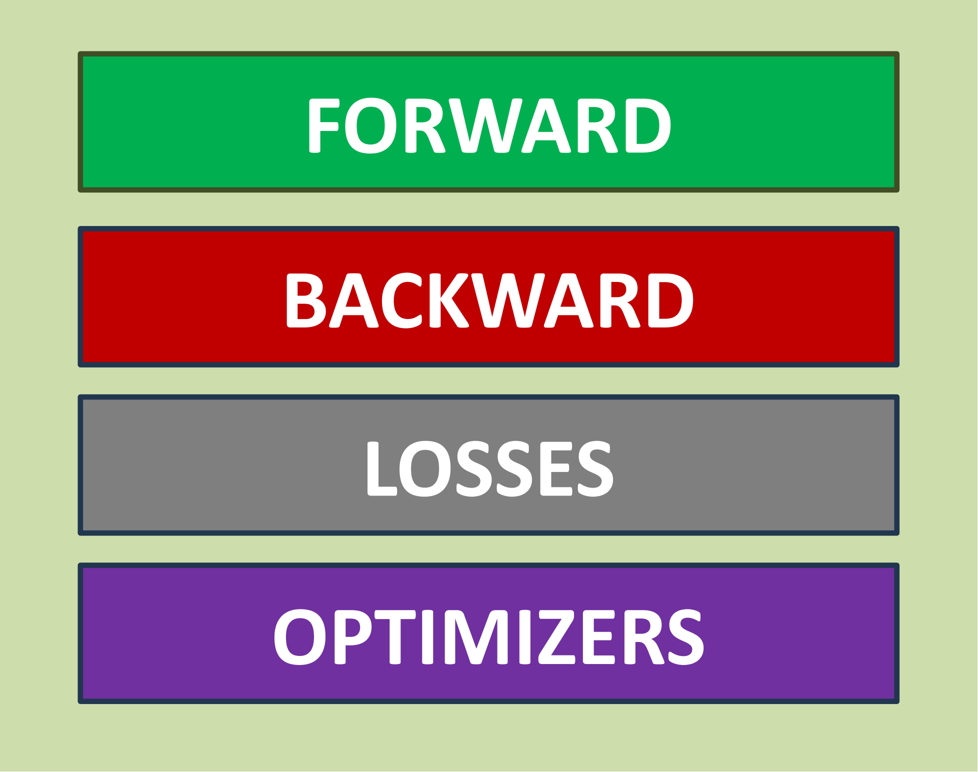
OUR GOAL
Enhancing DNNs On-Device

[1] PULP-Platform: https://pulp-platform.org/

ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

PULP-TrainLib

The first On-Device Learning library for RISC-V MultiCore MCUs (PULP)



FORWARD

BACKWARD

LOSSES

OPTIMIZERS

E.G: Conv2D Training

```
// Arguments
struct Conv2D_args C2D_args;
struct loss_args loss_args;
struct optim_args optim_args;

// Forward layer
pulp_conv2d_fp32_fw_cl(&C2D_args);

// Loss function
pulp_CrossEntropyLoss(&loss_args);

// Backward
pulp_conv2d_fp32_bw_input_grads_cl(&C2D_args);
pulp_conv2d_fp32_bw_param_grads_cl(&C2D_args);

// Update
pulp_gradient_descent_fp32(&optim_args);
```

https://github.com/pulp-platform/pulp-trainlib

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Thank you for your attention

DEI – Università di Bologna

Davide Nadalini – d.nadalini@unibo.it
Lorenzo Lamberti – lorenzo.lamberti@unibo.it
Alberto Dequino – alberto.dequino@unibo.it
Francesco Conti – f.conti@unibo.it