

# AI on the EDGE

## BECM33MLE — Machine Learning Engineering

Dr. Tomáš Báča

Multi-Robot Systems group, Faculty of Electrical Engineering  
Czech Technical University in Prague



FACULTY  
OF ELECTRICAL  
ENGINEERING  
CTU IN PRAGUE

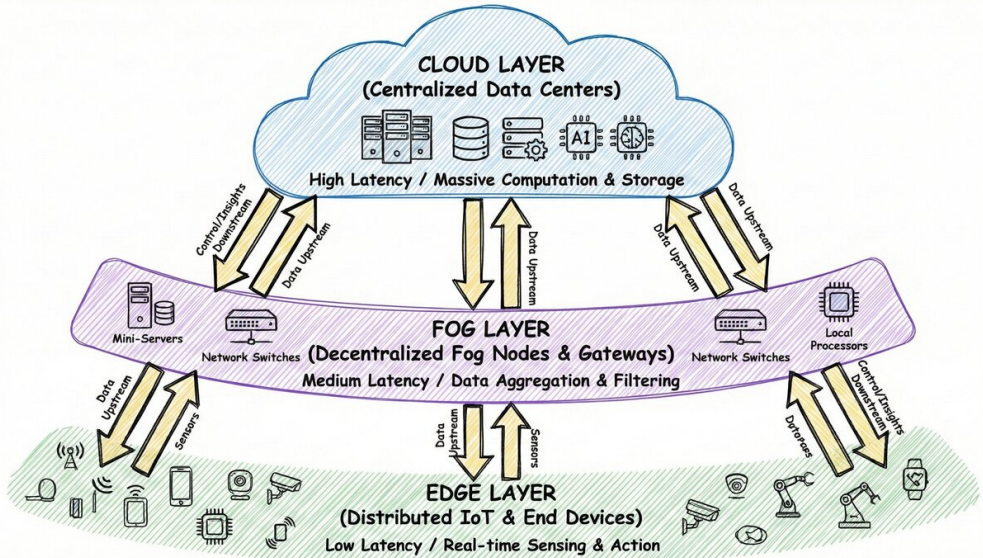


MRS  
MULTI-ROBOT  
SYSTEMS  
GROUP



DATAMOLE

# Computing continuum

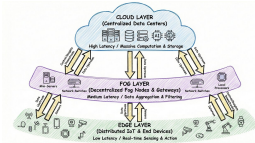


## Lecture 13: AI on the EDGE

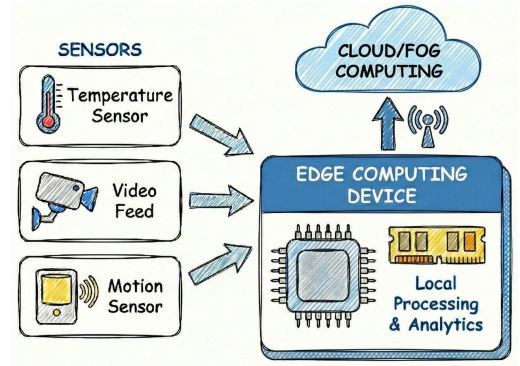
### Computing continuum

#### Computing continuum

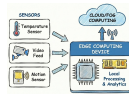
2026-01-11



- Low latency — data process at the source
- Reducing bandwidth — sending post-processed data to the Fog or Cloud
- Real-time decision making
- Better security — the data might not leave the site
- Offline-capable
- Scalable



- Low latency — data process at the source
- Reducing bandwidth — sending post-processed data to the Fog or Cloud
- Real-time decision making
- Better security — the data might not leave the site
- Offline-capable
- Scalable



# Edge constraints & Trade offs

Lecture 13: AI on the EDGE

Tomáš Bába

Computing continuum

Constraints

Compression Techniques

Accelerators

PEFT

AI in drone research

Control

Computer vision

Mapping and Localization

Task execution

References

## Connectivity

- (or) the lack off it
- bandwidth

## Price

- as always

## Size

- usually defined quite well

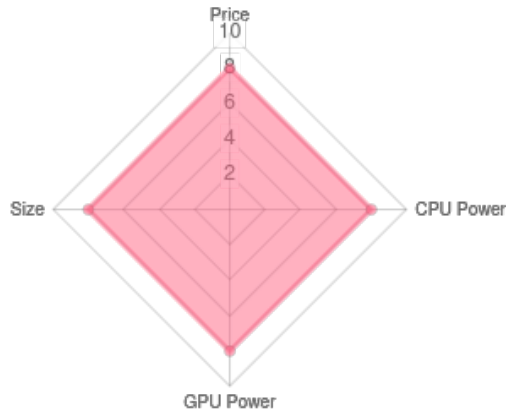
## "GPU power"

- for running the models

## CPU power

- for other data-handling, controls, etc.

Major Edge constraints



Tomáš Bába (CTU in Prague)

Lecture 13: AI on the EDGE

January 6th, 2026

4 / 33

## Lecture 13: AI on the EDGE

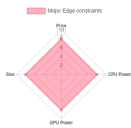
### Constraints

### Edge constraints & Trade offs

2026-01-11

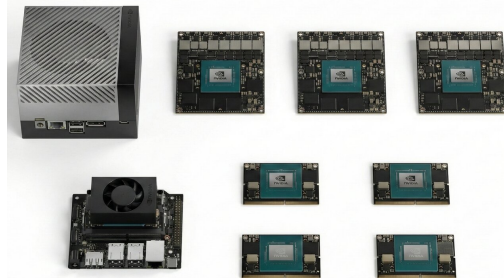
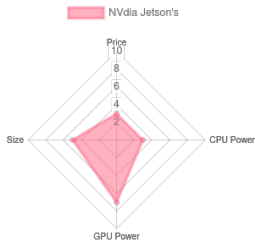
Edge constraints & Trade offs

|   |
|---|
| <b>Connectivity</b>                       |
| • (or) the lack off it                    |
| • bandwidth                               |
| <b>Price</b>                              |
| • as always                               |
| <b>Size</b>                               |
| • usually defined quite well              |
| <b>"GPU power"</b>                        |
| • for running the models                  |
| <b>CPU power</b>                          |
| • for other data-handling, controls, etc. |



## NVIDIA Jetson lineup

- Between 30 and 300 TOPs
- Between \$300 and \$2200
- Between 4 and 64 GB of memory
- small (and large) compute modules
- require custom mother boards



## Lecture 13: AI on the EDGE

### Constraints

NVIDIA Edge computing ecosystem

### NVIDIA Jetson lineup

- Between 30 and 300 TOPs
- Between \$300 and \$2200
- Between 4 and 64 GB of memory
- small (and large) compute modules
- require custom mother boards



Lecture 13: AI on the EDGE  
 Tomáš Bába  
 Computing continuum  
 Constraints  
 Compression Techniques  
 Accelerators  
 PEFT  
 AI in drone research  
 Control  
 Computer vision  
 Mapping and Localization  
 Task execution  
 References

|                          | Jetson AGX Orin series  |                      |                            | Jetson Orin NX series  |   | Jetson Orin Nano series  |  |                      |  |
|--------------------------|---|----------------------|----------------------------|--|---|--|--|----------------------|--|
|                          | Jetson AGX Orin Developer Kit                                 | Jetson AGX Orin 64GB | Jetson AGX Orin Industrial | Jetson AGX Orin 32GB   | Jetson Orin NX 16GB   | Jetson Orin NX 8GB   | Jetson Orin Nano Super Developer Kit                           | Jetson Orin Nano 8GB | Jetson Orin Nano 4GB   |
| <b>AI Performance</b>    | 275 TOPS  |                      | 248 TOPs                   | 200 TOPS   | 157 TOPS  | 117 TOPS   | 67 TOPS  | 67 TOPS              | 34 TOPS  |
| <b>GPU</b>               | 2048-core NVIDIA Ampere architecture GPU with 64 Tensor Cores |                      |                            | 1792-core NVIDIA Ampere c GPU with 56 Tensor Cores           | 1024-core NVIDIA Ampere architecture GPU with 32 Tensor Cores |  | 1024-core NVIDIA Ampere architecture GPU with 32 Tensor Cores  |                      | 512-core NVIDIA Ampere architecture GPU with 16 Tensor Cores |
| <b>GPU Max Frequency</b> | 1.3 GHz   |                      | 1.2 GHz                    | 930 MHz  | 1173MHz   | 1173MHz  | 1020MHz  | 1020MHz              | 1020MHz  |
| <b>CPU</b>               | 12-core Arm® Cortex®-A78AE v8.2 64-bit CPU<br>3MB L2 + 6MB L3 |                      |                            | 8-core Arm® Cortex®-A78AE v8.2 64-bit CPU<br>2MB L2 + 4MB L3 | 8-core Arm® Cortex®-A78AE v8.2 64-bit CPU<br>2MB L2 + 4MB L3  | 6-core Arm® Cortex®-A78AE v8.2 64-bit CPU<br>1.5MB L2 + 4MB L3 | 6-core Arm® Cortex®-A78AE v8.2 64-bit CPU<br>1.5MB L2 + 4MB L3 |                      |  |

Figure 1: Jetson lineup: from the smaller Orin Nano to the large Orin AGX

## Lecture 13: AI on the EDGE

### Constraints

#### NVIDIA Jetson Ecosystem

2026-01-11

NVIDIA Jetson Ecosystem

| Model             | Jetson AGX Orin series  |                      | Jetson Orin NX series   |                    | Jetson Orin Nano series  |                      |
|-------------------|---|----------------------|---|--------------------|--|----------------------|
|                   | Jetson AGX Orin Developer Kit                                 | Jetson AGX Orin 64GB | Jetson Orin NX 16GB   | Jetson Orin NX 8GB | Jetson Orin Nano Super Developer Kit                           | Jetson Orin Nano 8GB |
| AI Performance    | 275 TOPS  | 248 TOPs             | 157 TOPS  | 117 TOPS           | 67 TOPS  | 67 TOPS              |
| GPU               | 2048-core NVIDIA Ampere architecture GPU with 64 Tensor Cores |                      | 1024-core NVIDIA Ampere architecture GPU with 32 Tensor Cores |                    | 1024-core NVIDIA Ampere architecture GPU with 32 Tensor Cores  |                      |
| GPU Max Frequency | 1.3 GHz   | 1.2 GHz              | 1173MHz   | 1173MHz            | 1020MHz  | 1020MHz              |
| CPU               | 12-core Arm® Cortex®-A78AE v8.2 64-bit CPU<br>3MB L2 + 6MB L3 |                      | 8-core Arm® Cortex®-A78AE v8.2 64-bit CPU<br>2MB L2 + 4MB L3  |                    | 6-core Arm® Cortex®-A78AE v8.2 64-bit CPU<br>1.5MB L2 + 4MB L3 |                      |

Figure 1: Jetson lineup: from the smaller Orin Nano to the large Orin AGX

## NVIDIA Jetson lineup

- Very new player on the market
- 1000 TOPS
- Up to 128 GB of memory for models
- \$4000 USD

## Early adopters burn

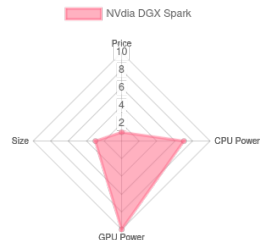
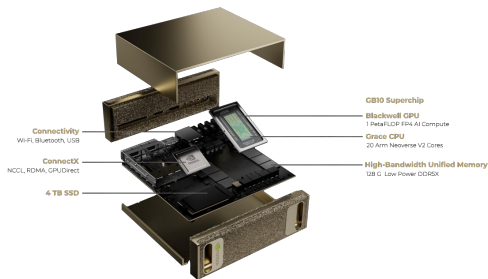


**John Carmack**  
@ID\_AA\_Carmack



DGX Spark appears to be maxing out at only 100 watts power draw, less than half of the rated 240 watts, and it only seems to be delivering about half the quoted performance (assuming 1 PF sparse FP4 = 125 TF dense BF16). It gets quite hot even at this level, and I saw a report of spontaneous rebooting on a long run, so was it de-rated before launch?

4:28 PM · Oct 27, 2025 · 135K Views



## Lecture 13: AI on the EDGE

### Constraints

#### NVIDIA DGX Spark

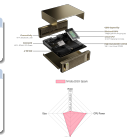
### NVIDIA DGX Spark

#### NVIDIA Jetson lineup

- Very new player on the market
- 1000 TOPS
- Up to 128 GB of memory for models
- \$4000 USD

#### Early adopters burn

DGX Spark appears to be maxing out at only 100 watts power draw, less than half of the rated 240 watts, and it only seems to be delivering about half the quoted performance (assuming 1 PF sparse FP4 = 125 TF dense BF16). It gets quite hot even at this level, and I saw a report of spontaneous rebooting on a long run, so was it de-rated before launch?



## Embedded hardware (Microcontrollers)

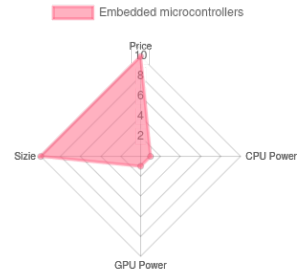
- ESP32
- STM32F7
- Arduino

## Why?

- embedded, tiny packages, can be soldered directly to a custom board
- very small, very low power (under 1 Watt)
- fast startup time, easily part of a physical product
- few GOPS at best



Figure 2: ESP32 cam



## Lecture 13: AI on the EDGE

### Constraints

Microcontrollers, Internet of Things devices

### Embedded hardware (Microcontrollers)

- ESP32
- STM32F7
- Arduino

### Why?

- embedded, tiny packages, can be soldered directly to a custom board
- very small, very low power (under 1 Watt)
- fast startup time, easily part of a physical product
- few GOPS at best



Figure 2: ESP32 cam



# Making smaller models

Lecture 13: AI on the EDGE

Tomáš Bába

Computing continuum

Constraints

Compression Techniques

Accelerators

PEFT

AI in drone research

Control

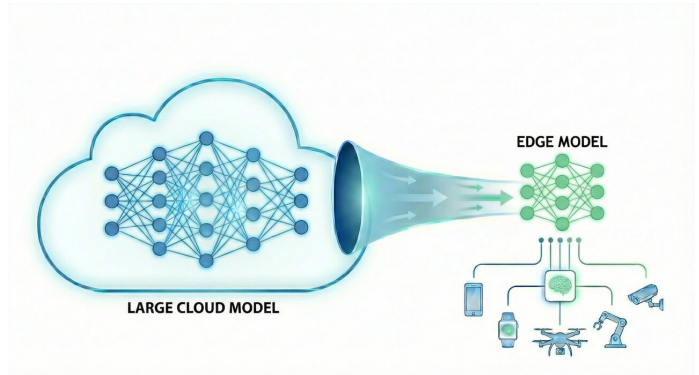
Computer vision

Mapping and Localization

Task execution

References

- reusing models is cheap and useful (if they serve the purpose)
- training larger models is easier
  - you usually don't know how big it really should be to cover the domain
  - searching for a critically-sized model is a job on its own
- somebody has spent the resources already (HuggingFace)



Tomáš Bába (CTU in Prague)

Lecture 13: AI on the EDGE

January 6th, 2026

9 / 33

Lecture 13: AI on the EDGE

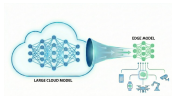
└─ Compression Techniques

└─ Making smaller models

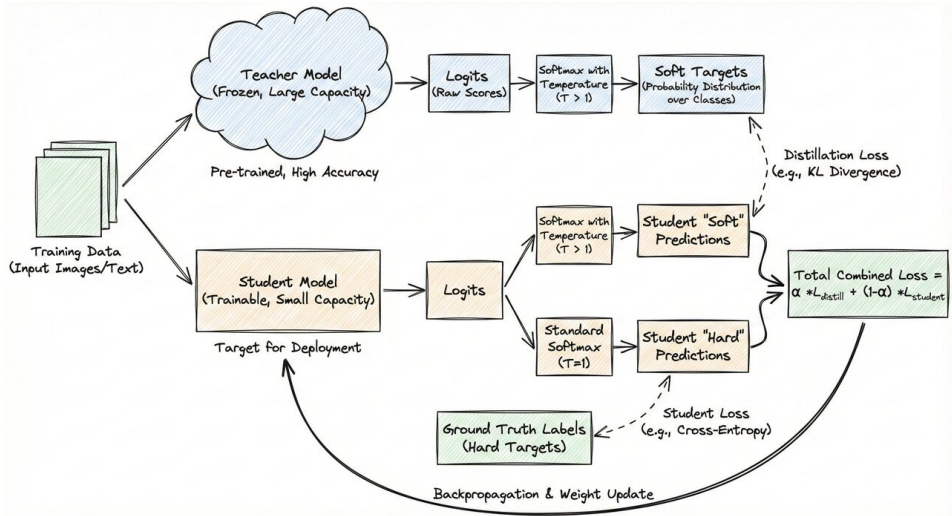
2026-01-11

Making smaller models

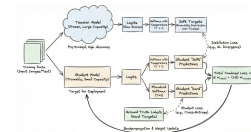
- reusing models is cheap and useful (if they serve the purpose)
- training larger models is easier
  - you usually don't know how big it really should be to cover the domain
  - searching for a critically-sized model is a job on its own
- somebody has spent the resources already (HuggingFace)



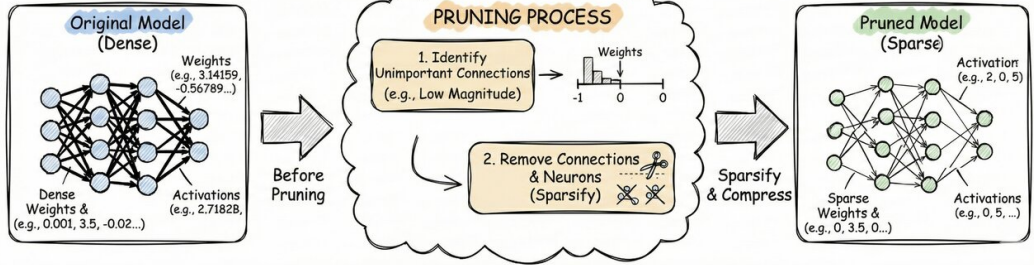
# Knowledge distillation [1]



Knowledge distillation [1]



# Pruning [2]

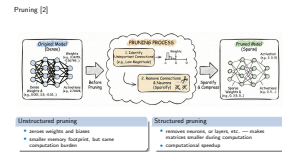


## Unstructured pruning

- zeroes weights and biases
- smaller memory footprint, but same computation burden

## Structured pruning

- removes neurons, or layers, etc. — makes matrices smaller during computation
- computational speedup



- can be static (offline, after training)
- can be dynamic (online, during inference)
- Methods for identifying paths with small contributions:
  - By weight: prune connections with small weights since they contribute less to the output.
  - By activation: prune neurons that contribute by a small amount (on average) on a validation dataset.
  - By gradient saliency: prune those with the smallest impact on the loss.
  - Synaptic Flow: sign-stripping the network, propagation of a synthetic signal (ones) through the network, then back propagation of the loss (sum of all outputs), finally: assigning weights a saliency score based on the gradient  $\times$  weight. This method is by far the safest to avoid layer collapse.

# Quantization

Lecture 13: AI on the EDGE

Tomáš Bába

Computing continuum

Constraints

Compression Techniques

Accelerators

PEFT

AI in drone research

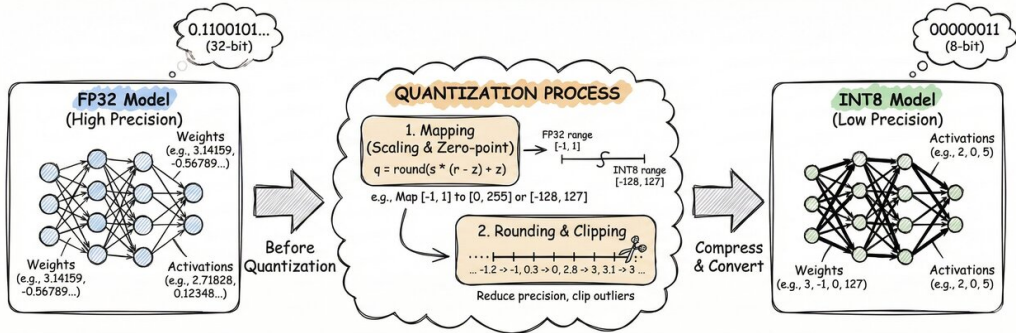
Control

Computer vision

Mapping and Localization

Task execution

References



## Post-training quantization

- taking high-fidelity model and quantizing the weights and biases
- used by HW accelerators (e.g., Hailo AI)

## Quantization-aware training

- simulation of low-precision during training

Tomáš Bába (CTU in Prague)

Lecture 13: AI on the EDGE

January 6th, 2026

12 / 33

Lecture 13: AI on the EDGE

Compression Techniques

Quantization

2026-01-11

Quantization



Post-training quantization

- taking high-fidelity model and quantizing the weights and biases
- used by HW accelerators (e.g., Hailo AI)

Quantization-aware training

- simulation of low-precision during training

## All-the-above

Distillation + Pruning + Quantization

## Embedded hardware (Microcontrollers)

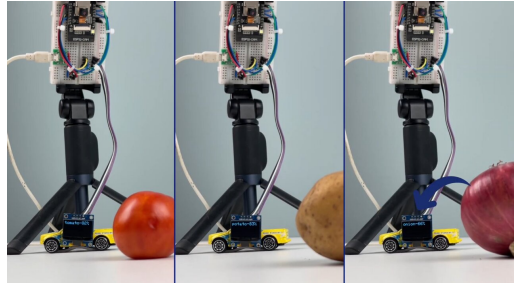
- ESP32, STM32F7, Arduino, Raspberry Pi, etc.

## TinyML [3]

- Edge Impulse [4], [5]
- TensorFlow Lite [6]
- X-Cube-AI (for STMs) [7]

## TED Talk about TinyML

<https://www.youtube.com/watch?v=rfFg1gLLaAo>



Video: <https://youtu.be/bZIKVaD3dRk>



## Hailo AI

- around 13 TOPS or more
- $\approx 1.5$  Watts
- HailoRT ecosystem
- Models are post processed: the inference is split into batches that are sent to the module to be computed
- Performance of the quantized models:  
<https://hailo.ai/products/hailo-software/model-explorer-vision/>

## Google Coral

- Old (essentially dead)
- $\approx 2.0$  Watts
- 4 TOPS (int8) (MobileNet v2)

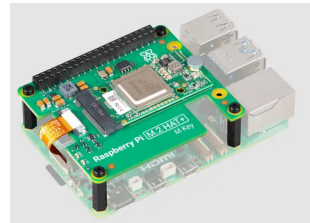


Figure 3: Hailo AI module

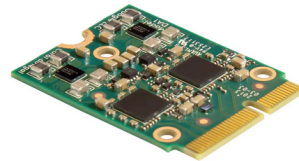


Figure 4: Google Coral module



Figure 3: Hailo AI module



Figure 4: Google Coral module

## RoboFly

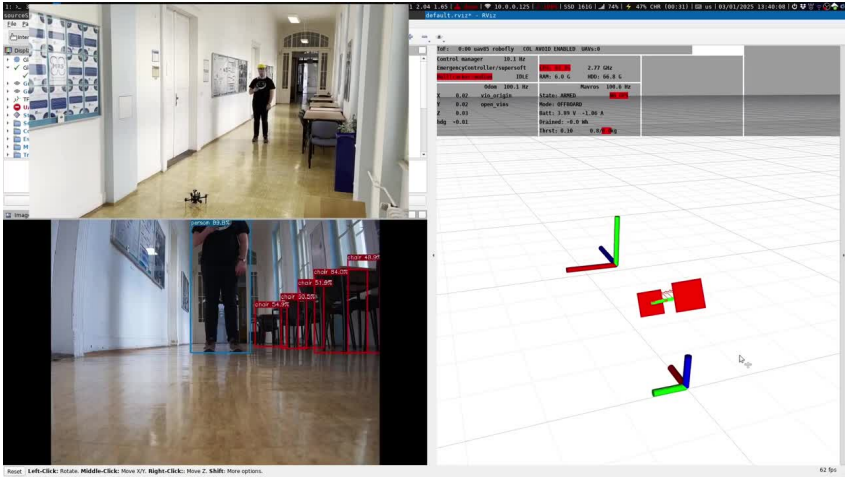
- Our own miniature drone for research and AI research
- Raspberry PI + Hailo AI 8L
- GNSS-denied flight using VIO
- ROS2 for control, estimation, planning, etc.
- <https://fly4future.com/robofly/>
- ≈ \$4000



### RoboFly

- Our own miniature drone for research and AI research
- Raspberry PI + Hailo AI 8L
- GNSS-denied flight using VIO
- ROS2 for control, estimation, planning, etc.
- <https://fly4future.com/robofly/>
- ≈ \$4000



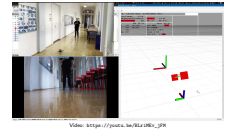


Video: [https://youtu.be/BLRiMEv\\_jFM](https://youtu.be/BLRiMEv_jFM)

## Lecture 13: AI on the EDGE

### Accelerators

#### RoboFly + HailoAI

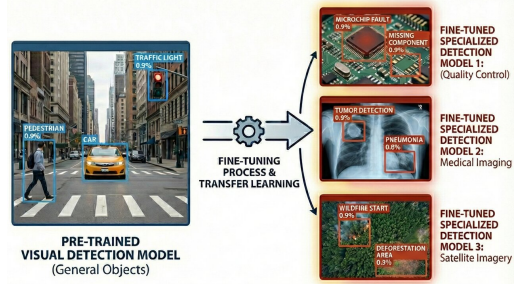


## Why finetuning?

- optimizing model for a downstream task:
- Example for an LLM:
  - tooling
  - summarization
  - sentiment analysis
  - domain Adaptation
  - style tuning

## Full-parameter fine-tuning

- intractable: requires the same resources as the original training
- memory requirements are prohibitive
- back propagating through the entire network is slow
- making a copy of the whole model for a downstream task is costly



## Parameter-Efficient Fine Tuning

- models are usually overparametrized
- selectively tuning just 1% of the parameters can significantly alter the behavior
- can be done with consumer hardware even for large models

Fine tuning

**Why finetuning?**

- optimizing model for a downstream task.
- Example for an LLM:
  - tooling
  - summarization
  - sentiment analysis
  - domain Adaptation
  - style tuning

**Full-parameter fine-tuning**

- intractable: requires the same resources as the original training
- memory requirements are prohibitive
- back propagating through the entire network is slow
- making a copy of the whole model for a downstream task is costly

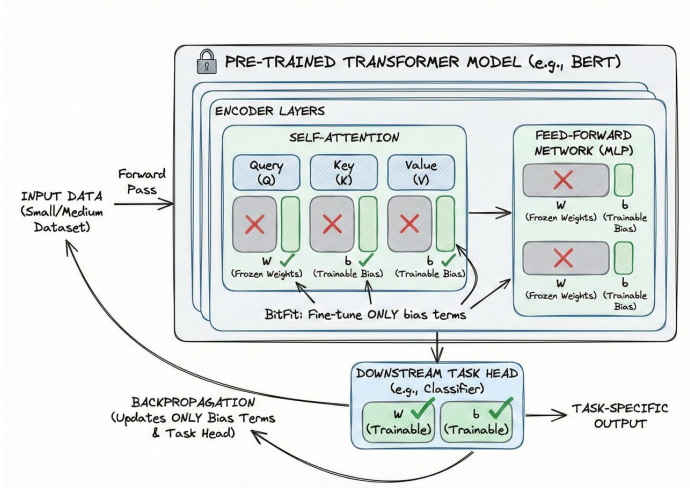
**Parameter-Efficient Fine Tuning**

- models are usually overparametrized
- selectively tuning just 1% of the parameters can significantly alter the behavior
- can be done with consumer hardware even for large models

- fine-tuning by only training the biases and,
- affine shift of the activations
- faster training (storing gradients only for biases)
- changes only tiny portion of the parameters ( $\leq 1\%$ )

### Adding new head

- relying on the original architecture
- adopting it to new output (new classes, labels)



## Lecture 13: AI on the EDGE

### PEFT

#### Bias-only Fine-Tuning (BitFit) [8]

2026-01-11

BitFit-only Fine-Tuning (BitFit) [8]

- fine-tuning by only training the biases and,
- affine shift of the activations
- faster training (storing gradients only for biases)
- changes only tiny portion of the parameters ( $\leq 1\%$ )

#### Adding new head

- relying on the original architecture
- adopting it to new output (new classes, labels)

# Low-Rank Adaptation (LoRA) [9]

Lecture 13: AI on the EDGE

Tomáš Bába

Computing continuum

Constraints

Compression Techniques

Accelerators

PEFT

AI in drone research

Control

Computer vision

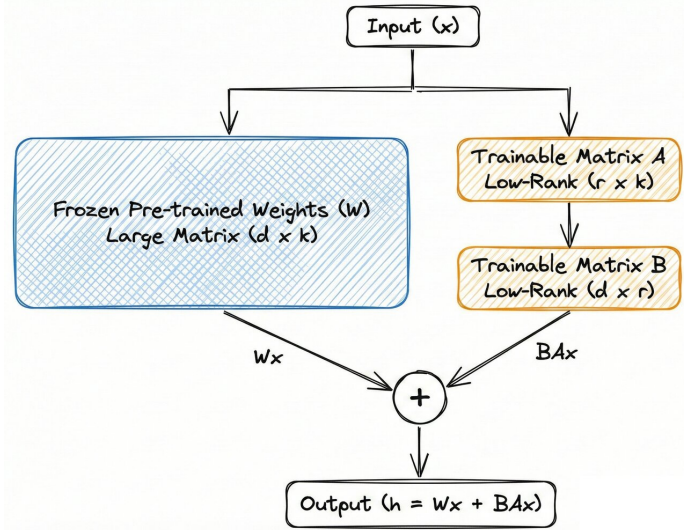
Mapping and Localization

Task execution

References

## The future?

- Devices (phones first) shipped with a single foundation model
- Vendors LoRA adapters will alter it to different use case
  - summarization, chat, computer vision, etc.
- 3rd party apps could provide their own LoRA adapters to fulfill their needs



Tomáš Bába (CTU in Prague)

Lecture 13: AI on the EDGE

January 6th, 2026

19 / 33

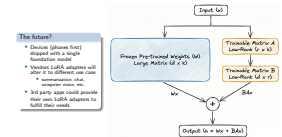
Lecture 13: AI on the EDGE

PEFT

Low-Rank Adaptation (LoRA) [9]

2026-01-11

Low-Rank Adaptation (LoRA) [9]



- **Moravec's paradox** is the observation that learning sensorimotor and perception skills is more challenging than reasoning.
- Formulated by Hans Moravec, Rodney Brooks, Marvin Minsky and others in 1980s.
- Original formulation: 'it is comparatively easy to make computers exhibit adult level performance on intelligence tests or playing checkers, and difficult or impossible to give them the skills of a one-year-old when it comes to perception and mobility' [10].

[10] H. Moravec, *Mind children: The future of robot and human intelligence*. Harvard University Press, 1988

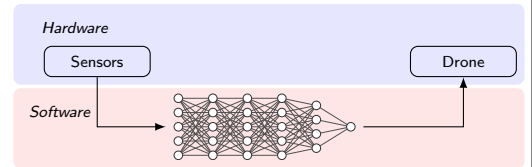


## Classical navigation pipeline

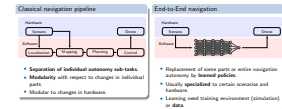


- **Separation of individual autonomy sub-tasks.**
- **Modularity** with respect to changes in individual parts.
- Modular to changes in hardware.

## End-to-End navigation

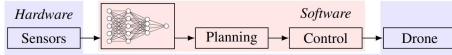


- Replacement of some parts or entire navigation autonomy by **learned policies**.
- Usually **specialized** to certain scenarios and hardware.
- Learning need training environment (simulation) or **data**.



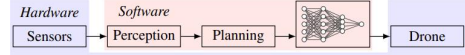
- In many areas of the pipeline used for the aerial robots.
- Localization and Mapping can be summarized as **Perception**.

## Learned Perception



- Learned perception is **very common** in even standard navigation pipeline.
- Basically application of **Computer Vision** tasks in robotics.
- Examples: detection of **landmarks for localization**, detection of **humans to follow in cinematography**, **semantic segmentation** for scene understanding.

## Learned Control



- Replaces the low-level control algorithms like PID or MPC by a **learned policy that follows given trajectory**.
- Can be faster than optimization-based solutions such as MPC.
- **Can optimize more complex objectives**, e.g., specified as reward in Reinforcement Learning.

[11] D. Hanover, A. Loquercio, L. Buaersfeld, A. Romero, R. Penicka, Y. Song, *et al.*, *Autonomous drone racing: A survey*, 2023. [Online]. Available: <https://arxiv.org/abs/2301.01755>

- In many areas of the pipeline used for the aerial robots.
- Localization and Mapping can be summarized as **Perception**.

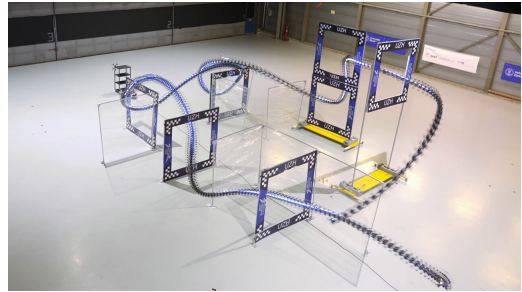
| Learned Perception  | Learned Control  |
|---|--|
| <ul style="list-style-type: none"> <li>• Learned perception is <b>very common</b> in even standard navigation pipeline.</li> <li>• Basically application of <b>Computer Vision</b> tasks in robotics.</li> <li>• Examples: detection of <b>landmarks for localization</b>, detection of <b>humans to follow in cinematography</b>, <b>semantic segmentation</b> for scene understanding.</li> </ul> | <ul style="list-style-type: none"> <li>• Replaces the low-level control algorithms like PID or MPC by a <b>learned policy that follows given trajectory</b>.</li> <li>• Can be faster than optimization-based solutions such as MPC.</li> <li>• Can optimize <b>more complex objectives</b>, e.g., specified as reward in Reinforcement Learning.</li> </ul> |

[11] D. Hanover, A. Loquercio, L. Buaersfeld, A. Romero, R. Penicka, Y. Song, *et al.*, *Autonomous drone racing: A survey*, 2023. [Online]. Available: <https://arxiv.org/abs/2301.01755>

- **Learned Planning and Control** for minimum-time flight in cluttered environment for drone racing.
- Uses **Reinforcement Learning** to learn policies in simulation that avoid collisions while minimizing the time of flight.
- Only a global guiding path is given and the policy outputs directly the body rate command and collective thrust of the UAV.

## TOPFLIGHT Project (Junior Star GACR)

<https://mrs.fel.cvut.cz/projects/topflight>



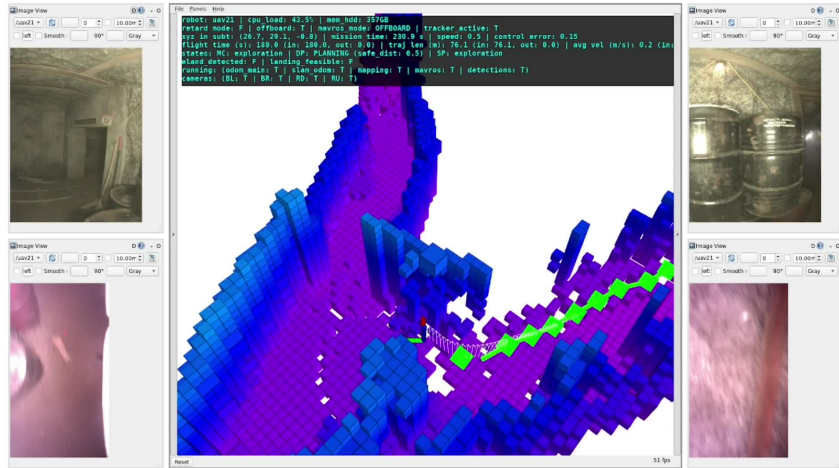
Video: <https://youtu.be/wR1niZvI3pI>

- [12] R. Penicka, Y. Song, E. Kaufmann, and D. Scaramuzza, "Learning minimum-time flight in cluttered environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7209–7216, 2022



## Object detection

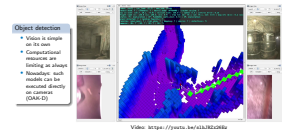
- Vision is simple on its own
- Computational resources are limiting as always
- Nowadays: such models can be executed directly on cameras (OAK-D)



Video: <https://youtu.be/s1hJRZx26Ew>

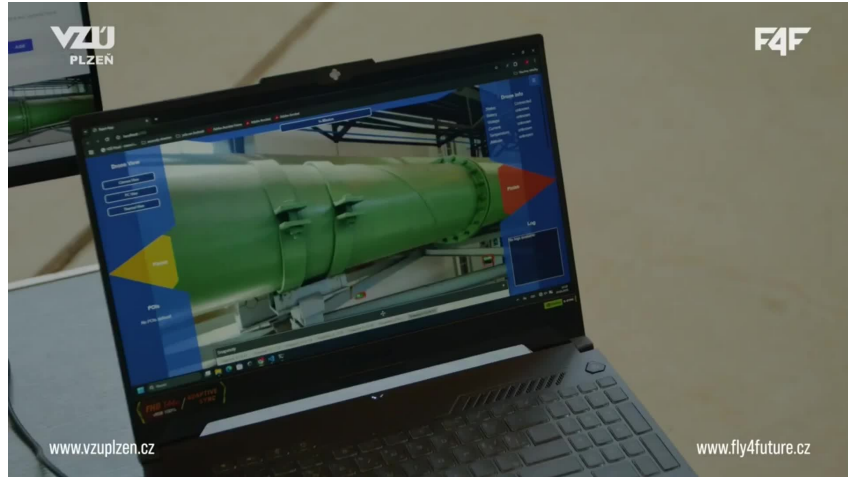
## Lecture 13: AI on the EDGE

- Control
  - Computer vision
    - Traditional pipeline + Computer vision



## Structural failure detection

- FOG approach
- Drone flies using traditional pipeline
- Images are processed on a laptop
- All data is gathered for further offline processing



## Lecture 13: AI on the EDGE

Control

Computer vision

Traditional pipeline + Computer vision

2026-01-11



## SPRIN-D Autonomous Flight Challenge

- GNSS-denied environment exploration
- GNSS-denied object semantic localization
- GNSS-denied person following
- <https://www.sprind.org/en/actions/challenges/funke-fully-autonomous-flight-2.0>

## Lockheed Martin offset programme

- GNSS-denied fixed-wing localization
- Camera-only global localization and relocalization
- Semantic scene understanding
- Differentiable models for wind prediction

## TOMSNAV Project — Junior Star GACR

- Semantic hierarchical localization and mapping
- Semantic map understanding
- Efficient map sharing
- Human-explainable actions
- Mapping, planning and mission execution without the knowledge of precise location
- <https://mrs.fel.cvut.cz/projects/gacr-tomsnav>

### Lecture 13: AI on the EDGE

Control

Computer vision

AI and ML in Aerial Robotics Research — our current funding and motivation

2026-01-11

#### SPRIN-D Autonomous Flight Challenge

- GNSS-denied environment exploration
- GNSS-denied object semantic localization
- GNSS-denied person following
- <https://www.sprind.org/en/actions/challenges/funke-fully-autonomous-flight-2.0>

#### Lockheed Martin offset programme

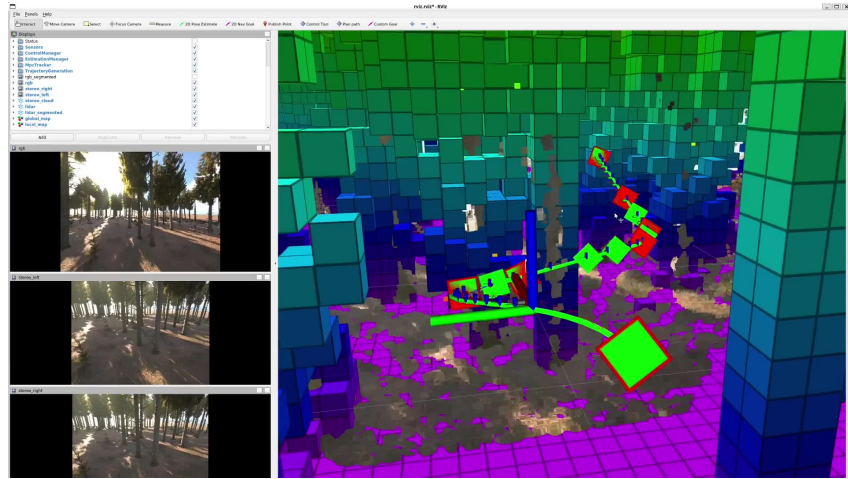
- GNSS-denied fixed-wing localization
- Camera-only global localization and relocalization
- Semantic scene understanding
- Differentiable models for wind prediction

#### TOMSNAV Project — Junior Star GACR

- Semantic hierarchical localization and mapping
- Semantic map understanding
- Efficient map sharing
- Human-explainable actions
- Mapping, planning and mission execution without the knowledge of precise location
- <https://mrs.fel.cvut.cz/projects/gacr-tomsnav>

# Mapping — traditional

- marking occupied and free space into a map
- projecting colored image into the map
- post-processing the map for planning purposes



Link: <https://youtu.be/Jtxh1hZRS1A>

## Lecture 13: AI on the EDGE

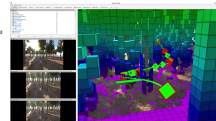
### Control

#### Mapping and Localization

##### Mapping — traditional

2026-01-11

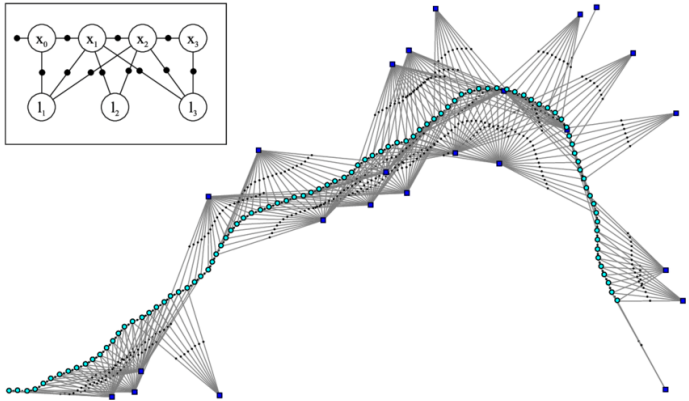
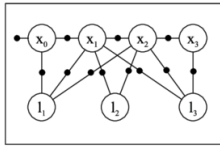
- marking occupied and free space into a map
- projecting colored image into the map
- post-processing the map for planning purposes



Link: <https://youtu.be/Jtxh1hZRS1A>

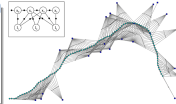
## Factor-graph SLAM

- chicken-and-egg problem of building a map while localizing
- relying on sparse and low-level image or geometrical features
- dense representation of the robot's motion
- optimization-based: *shaping the factor graph* to fit the observations
- poor scalability, does not use semantics



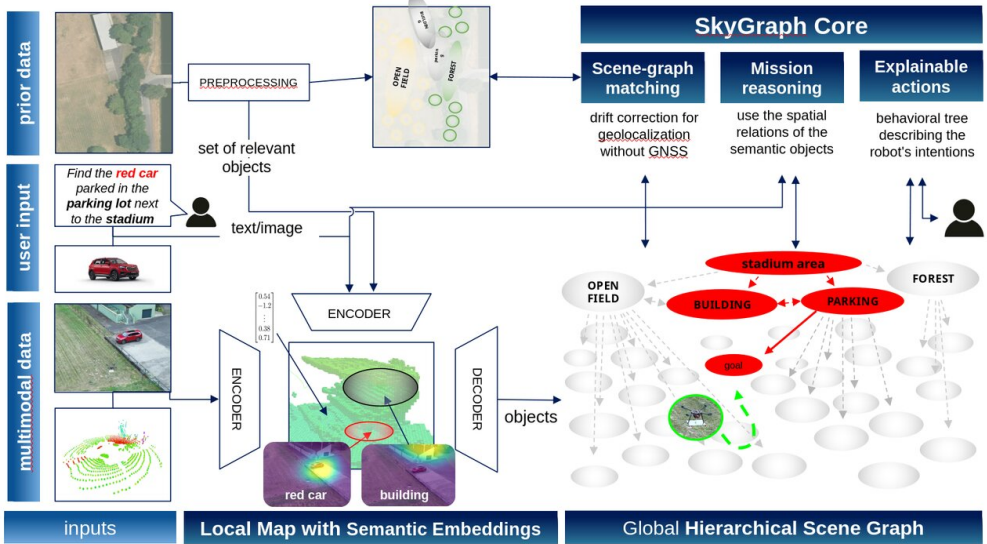
### Factor-graph SLAM

- chicken-and-egg problem of building a map while localizing
- relying on sparse and low-level image or geometrical features
- dense representation of the robot's motion
- optimization-based: *shaping the factor graph* to fit the observations
- poor scalability, does not use semantics



# Semantic Topometric Localization and Mapping

Lecture 13: AI on the EDGE  
 Tomáš Bába  
 Computing continuum  
 Constraints  
 Compression Techniques  
 Accelerators  
 PEFT  
 AI in drone research  
 Control  
 Computer vision  
 Mapping and Localization  
 Task execution  
 References



## Lecture 13: AI on the EDGE

Control

Mapping and Localization

Semantic Topometric Localization and Mapping

2026-01-11

Semantic Topometric Localization and Mapping



# Semantic mapping (current research)

Lecture 13: AI on the EDGE

Tomáš Bába

Computing continuum

Constraints

Compression Techniques

Accelerators

PEFT

AI in drone research

Control

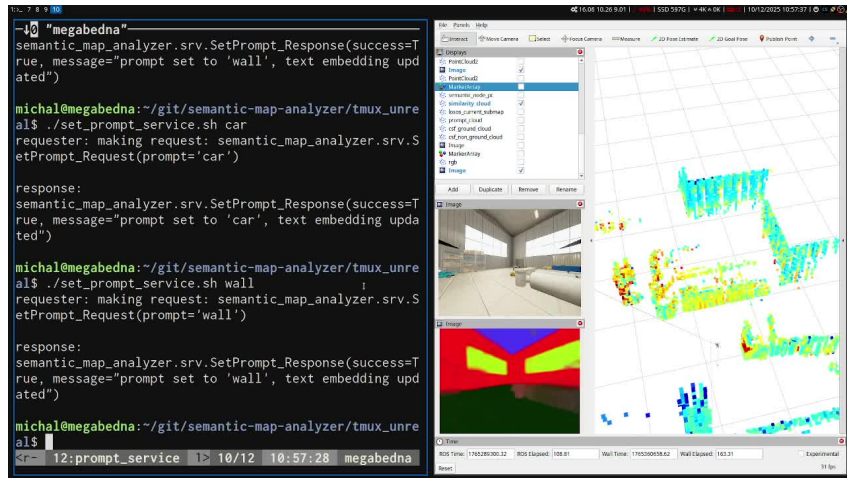
Computer vision

Mapping and Localization

Task execution

References

- Storing embeddings directly in the map's voxels
- Temporal and spatial smoothing and filtering of embeddings
- Clustering of embeddings
- Similarity search over the map



Tomáš Bába (CTU in Prague)

Lecture 13: AI on the EDGE

January 6th, 2026

30 / 33

Lecture 13: AI on the EDGE

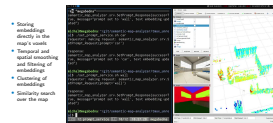
Control

Mapping and Localization

Semantic mapping (current research)

2026-01-11

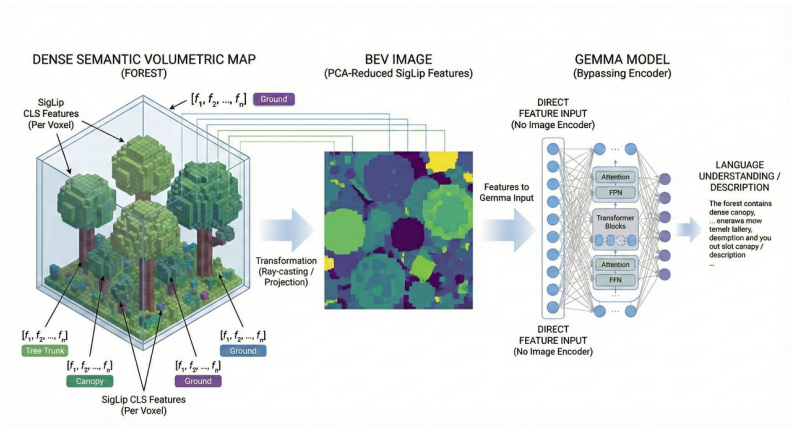
Semantic mapping (current research)



# Semantic mapping (potential research)

## Spatial reasoning

- Map populated with embeddings (SigLip)
- Birds-eye view of the scene expressed in embeddings
- The BEV image fed behind and encoder of an LLM (Gemma)
- Circumventing the language interface when possible

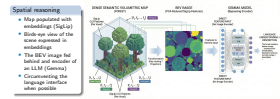


## Lecture 13: AI on the EDGE

### Control

### Mapping and Localization

### Semantic mapping (potential research)



# Task execution — creation of behavioral trees

Lecture 13: AI on the EDGE

Tomáš Bába

Computing continuum

Constraints

Compression Techniques

Accelerators

PEFT

AI in drone research

Control

Computer vision

Mapping and Localization

Task execution

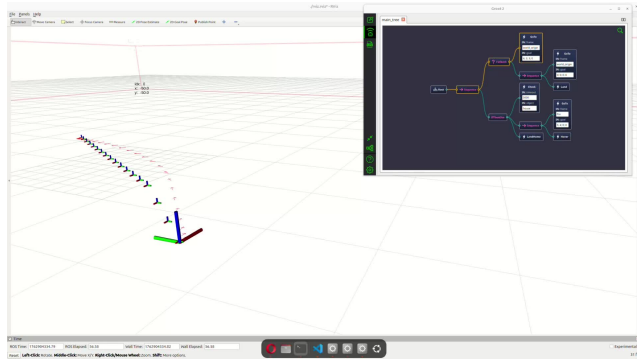
References

## Prompt

*Fly to the position (4, 3, 3, 0) in the world frame. If the action fails, fly to the position (0, 0, 0, 0) in the world frame and land. If there is a house, fly to the position (5, 4, 0, 0) in the FCU frame and hover; otherwise, land at home.*

## Properties

- understands if-statements
- Fallback behaviour in case that any component of the system failed
- Verified in simulation as well as on real hardware



Tomáš Bába (CTU in Prague)

Lecture 13: AI on the EDGE

January 6th, 2026

32 / 33

## Lecture 13: AI on the EDGE

Control

Task execution

Task execution — creation of behavioral trees

2026-01-11

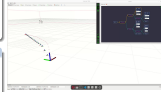
Task execution — creation of behavioral trees

### Prompt

*Fly to the position (4, 3, 3, 0) in the world frame. If the action fails, fly to the position (0, 0, 0, 0) in the world frame and land. If there is a house, fly to the position (5, 4, 0, 0) in the FCU frame and hover; otherwise, land at home.*

### Properties

- understands if-statements
- Fallback behaviour in case that any component of the system failed
- Verified in simulation as well as on real hardware



# References I

Lecture 13: AI on the EDGE

Tomáš Bába

Computing continuum

Constraints

Compression Techniques

Accelerators

PEFT

AI in drone research

Control

Computer vision

Mapping and Localization

Task execution

References

- [1] M. H. Aslam, C. Martinez, M. Pedersoli, A. Koeirich, A. Etemad, and E. Granger, *Learning from stochastic teacher representations using student-guided knowledge distillation*, 2025. arXiv: 2504.14307 [cs.LG]. [Online]. Available: <https://arxiv.org/abs/2504.14307>.
- [2] *A comprehensive guide to neural network model pruning*, <https://datafairy.io/blog/a-comprehensive-guide-to-neural-network-model-pruning>, Accessed: 2026-01-04.
- [3] J. Lin, L. Zhu, W.-M. Chen, W.-C. Wang, and S. Han, "Tiny machine learning: Progress and futures [feature]," *IEEE Circuits and Systems Magazine*, vol. 23, no. 3, 8–34, 2023, ISSN: 1556-0830. DOI: 10.1109/mcas.2023.3302182. [Online]. Available: <http://dx.doi.org/10.1109/mcas.2023.3302182>.
- [4] S. Hymel, C. Banbury, D. Situnayake, et al., *Edge impulse: An mlops platform for tiny machine learning*, 2023. arXiv: 2212.03332 [cs.DC]. [Online]. Available: <https://arxiv.org/abs/2212.03332>.
- [5] *Edge impulse*, <https://edgeimpulse.com/>, Accessed: 2026-01-04.
- [6] *Tensorflow lite*, <https://ai.google.dev/edge/litert>, Accessed: 2026-01-04.
- [7] *X-cube-ai*, <https://www.st.com/en/embedded-software/x-cube-ai.html>, Accessed: 2026-01-04.
- [8] E. B. Zaken, Y. Goldberg, and S. Ravfogel, "Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2022, pp. 1–9.
- [9] E. J. Hu, Y. Shen, P. Wallis, et al., *Lora: Low-rank adaptation of large language models*, 2021. arXiv: 2106.09685 [cs.CL]. [Online]. Available: <https://arxiv.org/abs/2106.09685>.
- [10] H. Moravec, *Mind children: The future of robot and human intelligence*. Harvard University Press, 1988.
- [11] D. Hanover, A. Loquercio, L. Bowersfeld, et al., *Autonomous drone racing: A survey*, 2023. [Online]. Available: <https://arxiv.org/abs/2301.01755>.
- [12] R. Penicka, Y. Song, E. Kaufmann, and D. Scaramuzza, "Learning minimum-time flight in cluttered environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7209–7216, 2022.

Tomáš Bába (CTU in Prague)

Lecture 13: AI on the EDGE

January 6th, 2026

33 / 33

Lecture 13: AI on the EDGE

References

References

2026-01-11

References I

- [1] M. H. Aslam, C. Martinez, M. Pedersoli, A. Koeirich, A. Etemad, and E. Granger, *Learning from stochastic teacher representations using student-guided knowledge distillation*, 2025. arXiv: 2504.14307 [cs.LG]. [Online]. Available: <https://arxiv.org/abs/2504.14307>.
- [2] *A comprehensive guide to neural network model pruning*, <https://datafairy.io/blog/a-comprehensive-guide-to-neural-network-model-pruning>, Accessed: 2026-01-04.
- [3] J. Lin, L. Zhu, W.-M. Chen, W.-C. Wang, and S. Han, "Tiny machine learning: Progress and futures [feature]," *IEEE Circuits and Systems Magazine*, vol. 23, no. 3, 8–34, 2023, ISSN: 1556-0830. DOI: 10.1109/mcas.2023.3302182. [Online]. Available: <http://dx.doi.org/10.1109/mcas.2023.3302182>.
- [4] S. Hymel, C. Banbury, D. Situnayake, et al., *Edge impulse: An mlops platform for tiny machine learning*, 2023. arXiv: 2212.03332 [cs.DC]. [Online]. Available: <https://arxiv.org/abs/2212.03332>.
- [5] *Edge impulse*, <https://edgeimpulse.com/>, Accessed: 2026-01-04.
- [6] *Tensorflow lite*, <https://ai.google.dev/edge/litert>, Accessed: 2026-01-04.
- [7] *X-cube-ai*, <https://www.st.com/en/embedded-software/x-cube-ai.html>, Accessed: 2026-01-04.
- [8] E. B. Zaken, Y. Goldberg, and S. Ravfogel, "Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2022, pp. 1–9.
- [9] E. J. Hu, Y. Shen, P. Wallis, et al., *Lora: Low-rank adaptation of large language models*, 2021. arXiv: 2106.09685 [cs.CL]. [Online]. Available: <https://arxiv.org/abs/2106.09685>.
- [10] H. Moravec, *Mind children: The future of robot and human intelligence*. Harvard University Press, 1988.
- [11] D. Hanover, A. Loquercio, L. Bowersfeld, et al., *Autonomous drone racing: A survey*, 2023. [Online]. Available: <https://arxiv.org/abs/2301.01755>.
- [12] R. Penicka, Y. Song, E. Kaufmann, and D. Scaramuzza, "Learning minimum-time flight in cluttered environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7209–7216, 2022.