

# 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

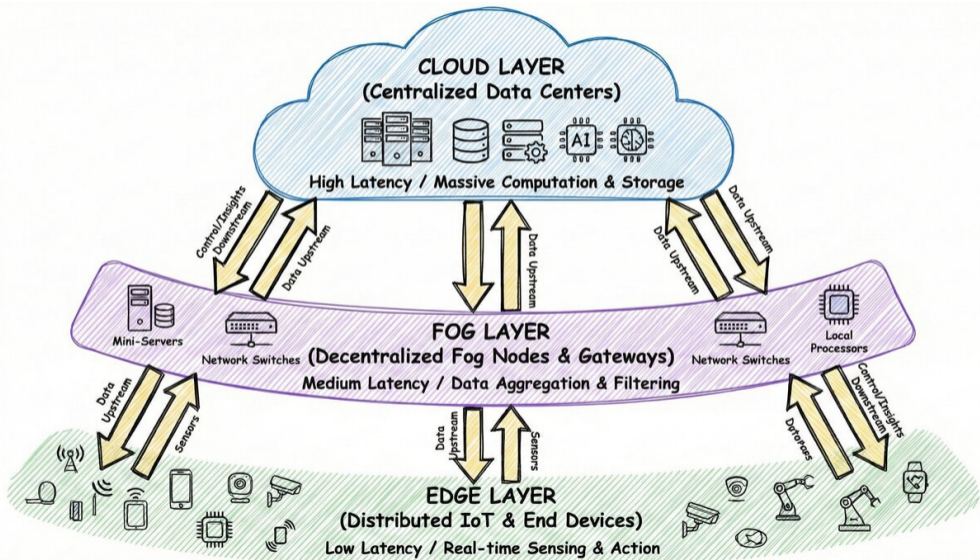


MRS  
MULTI-ROBOT  
SYSTEMS  
GROUP



**DATAMOLE**

# Computing continuum



Lecture 13: AI on the EDGE

Tomáš Bába

Computing continuum

Constraints

Compression Techniques

Accelerators

PEFT

AI in drone research

Control

Computer vision

Mapping and Localization

Task execution

References

# Edge layer

Lecture  
13: AI on  
the EDGE

Tomáš  
Báča

Computing  
continuum

Constraints

Compression  
Techniques

Accelerators

PEFT

AI in drone  
research

Control

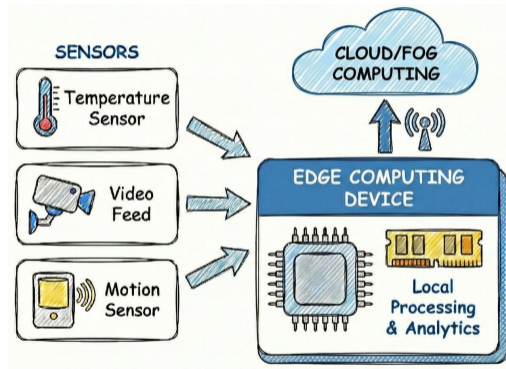
Computer  
vision

Mapping and  
Localization

Task  
execution

References

- 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  
Báča

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

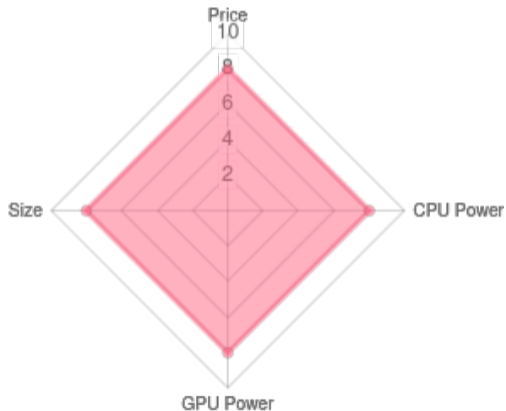
## “GPU power”

- for running the models

## CPU power

- for other data-handling, controls, etc.

Major Edge constraints



# NVIDIA Edge computing ecosystem

Lecture  
13: AI on  
the EDGE

Tomáš  
Báča

Computing  
continuum

Constraints

Compression  
Techniques

Accelerators

PEFT

AI in drone  
research

Control

Computer  
vision

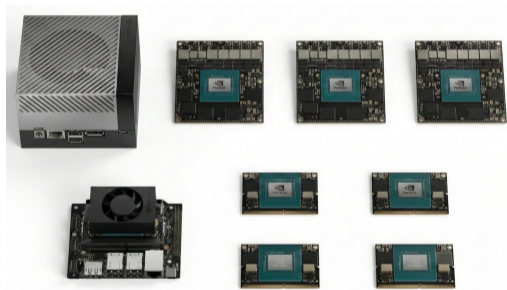
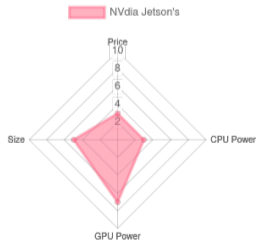
Mapping and  
Localization

Task  
execution

References

## 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



# NVidia Jetson Ecosystem

Lecture 13: AI on the EDGE

Tomáš Báča

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 3MB L2 + 6MB L3			8-core Arm® Cortex®-A78AE v8.2 64-bit CPU 2MB L2 + 4MB L3	8-core Arm® Cortex®-A78AE v8.2 64-bit CPU 2MB L2 + 4MB L3	6-core Arm® Cortex®-A78AE v8.2 64-bit CPU 1.5MB L2 + 4MB L3	6-core Arm® Cortex®-A78AE v8.2 64-bit CPU 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

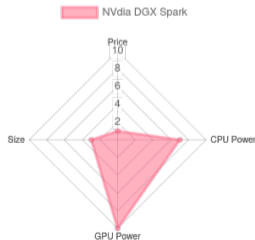
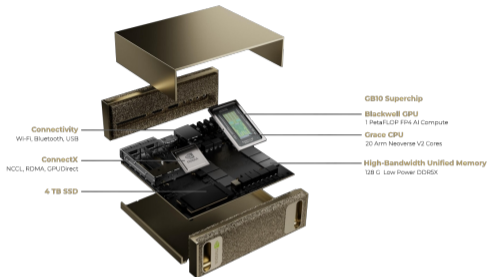


John Carmack ✓  
@ID\_AA\_Carmack



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



## Embedded hardware (Microcontrollers)

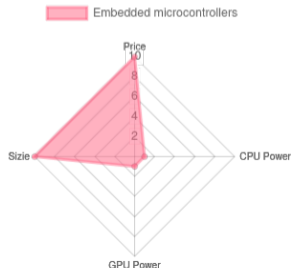
- ESP32
- STM32F7
- Arduino

## 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  
Báča

Computing  
continuum

Constraints

Compression  
Techniques

Accelerators

PEFT

AI in drone  
research

Control

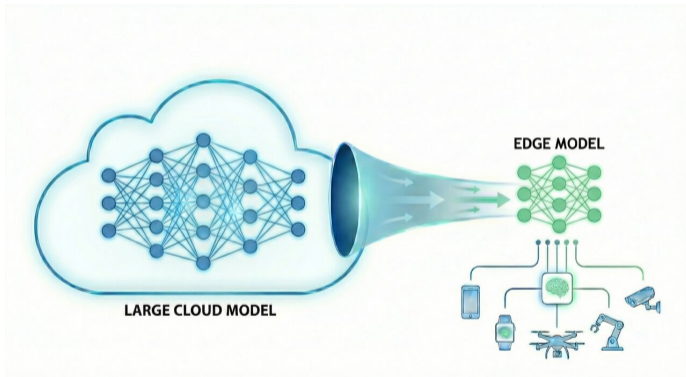
Computer  
vision

Mapping and  
Localization

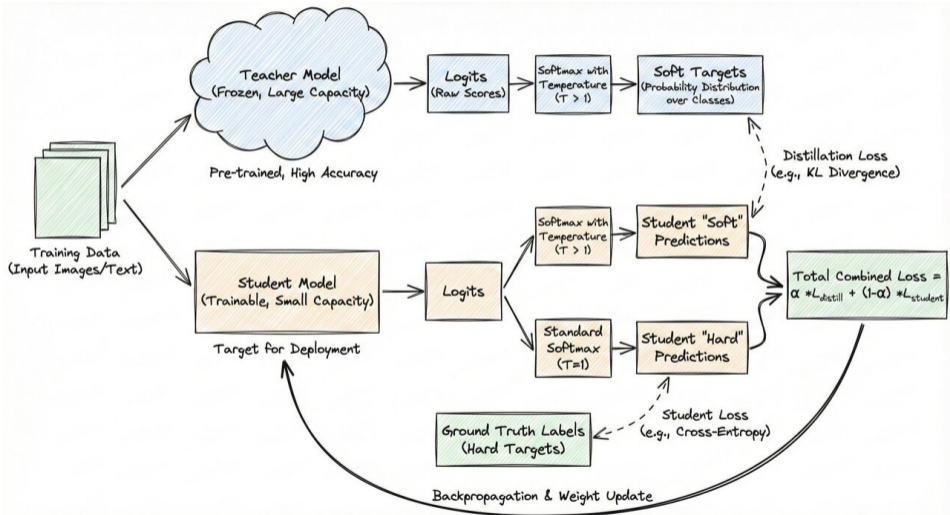
Task  
execution

References

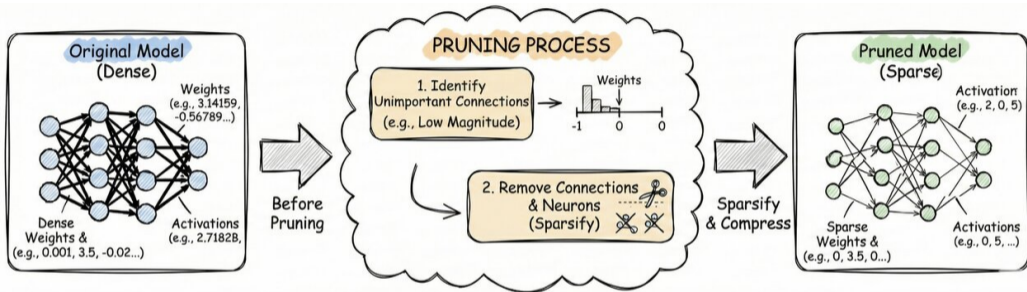
- reusing models is cheap and useful (if they serve the purpose)
- training larger models is easier
  - you usually don't know how big it really should be to cover the domain
  - searching for a critically-sized model is a job on its own
- somebody has spent the resources already (HuggingFace)



# 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

# Quantization

Lecture 13: AI on the EDGE

Tomáš Bába

Computing continuum

Constraints

Compression Techniques

Accelerators

PEFT

AI in drone research

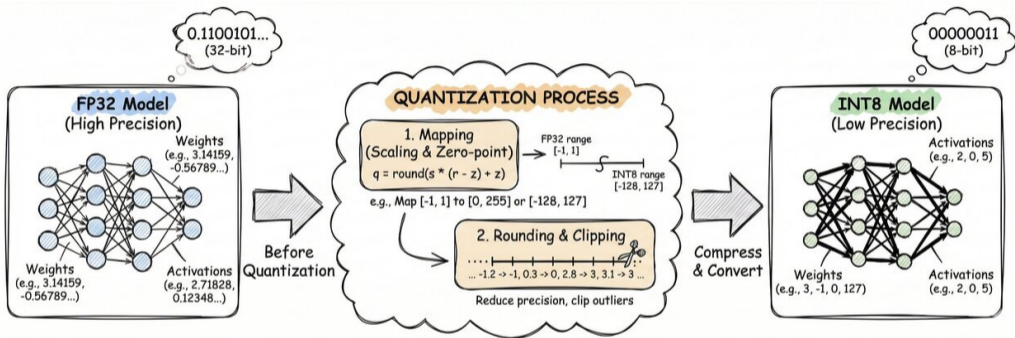
Control

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

## All-the-above

Distillation + Pruning + Quantization

## Embedded hardware (Microcontrollers)

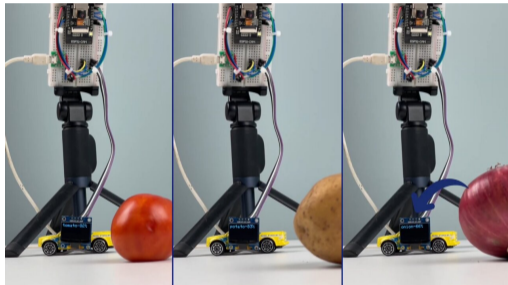
- ESP32, STM32F7, Arduino, Raspberry Pi, etc.

## 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)

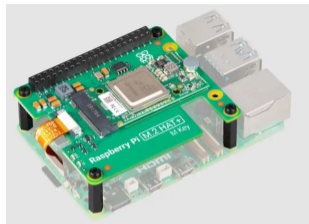


Figure 3: Hailo AI module



Figure 4: Google Coral module

## 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/>
- $\approx$  \$4000



# RoboFly + HailoAI

Lecture 13: AI on the EDGE

Tomáš Bába

Computing continuum

Constraints

Compression Techniques

Accelerators

PEFT

AI in drone research

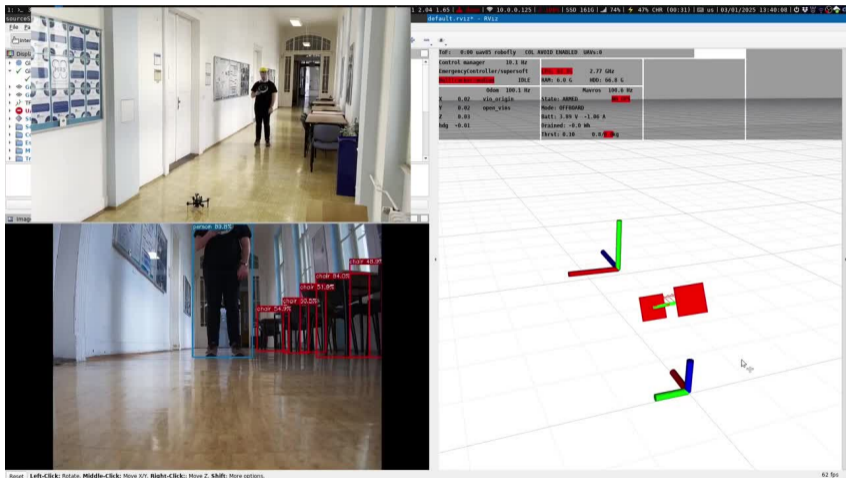
Control

Computer vision

Mapping and Localization

Task execution

References



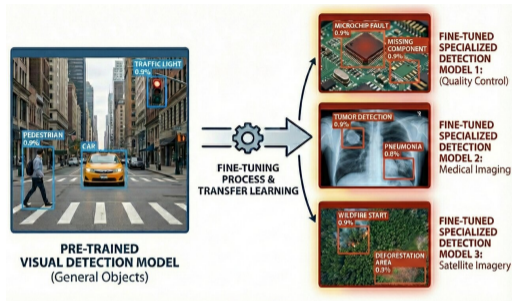
Video: [https://youtu.be/BLrIMEv\\_jFM](https://youtu.be/BLrIMEv_jFM)

## 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

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

Lecture  
13: AI on  
the EDGE

Tomáš  
Báča

Computing  
continuum

Constraints

Compression  
Techniques

Accelerators

PEFT

AI in drone  
research

Control

Computer  
vision

Mapping and  
Localization

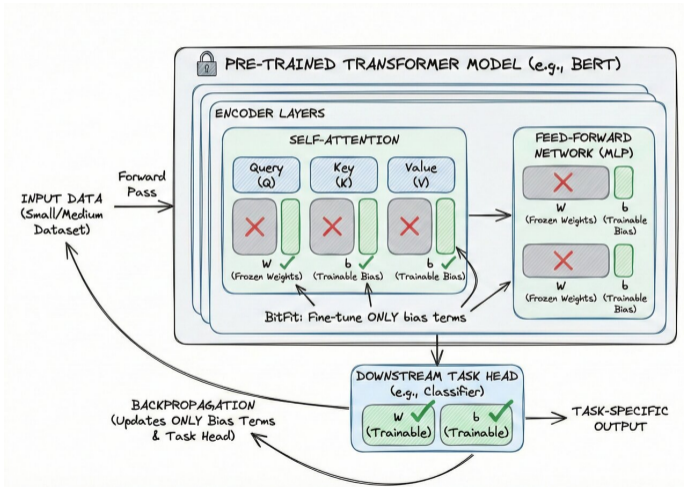
Task  
execution

References

- 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 ( $\leq 1\%$ )

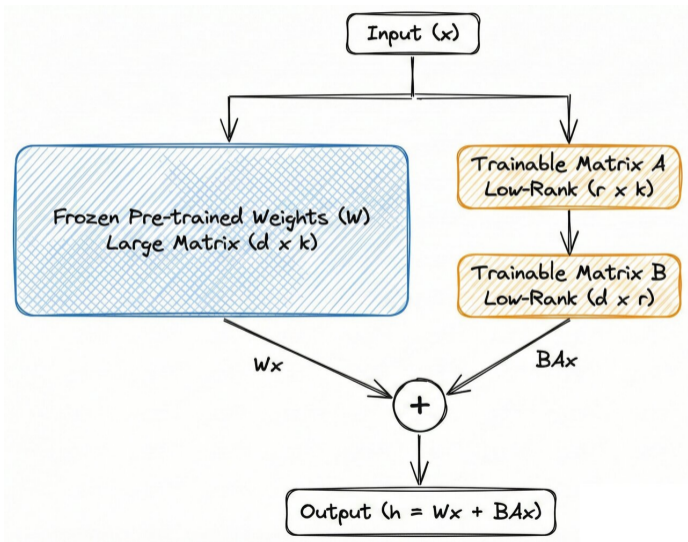
## Adding new head

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



## 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



# AI in robotics — Moravec's paradox

Lecture  
13: AI on  
the EDGE

Tomáš  
Báča

Computing  
continuum

Constraints

Compression  
Techniques

Accelerators

PEFT

AI in drone  
research

Control

Computer  
vision

Mapping and  
Localization

Task  
execution

References

- **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



# Autonomous Aerial System Navigation Pipeline

Lecture  
13: AI on  
the EDGE

Tomáš  
Báča

Computing  
continuum

Constraints

Compression  
Techniques

Accelerators

PEFT

AI in drone  
research

Control

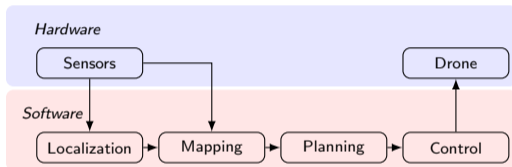
Computer  
vision

Mapping and  
Localization

Task  
execution

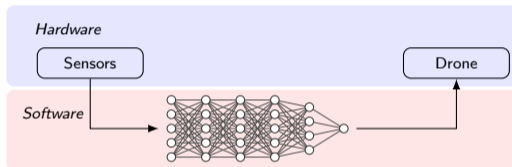
References

## Classical navigation pipeline



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

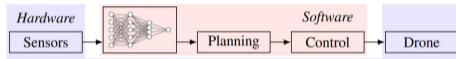
## End-to-End navigation



- 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**.

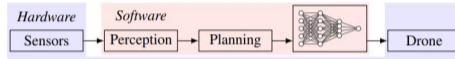
- In many areas of the pipeline used for the aerial robots.
- Localization and Mapping can be summarized as **Perception**.

## 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. Bauersfeld, A. Romero, R. Penicka, Y. Song, *et al.*, *Autonomous drone racing: A survey*, 2023. [Online]. Available: <https://arxiv.org/abs/2301.01755>

# Reinforcement learning for minimum-time collision-free flight

Lecture  
13: AI on  
the EDGE

Tomáš  
Báča

Computing  
continuum

Constraints

Compression  
Techniques

Accelerators

PEFT

AI in drone  
research

Control

Computer  
vision

Mapping and  
Localization

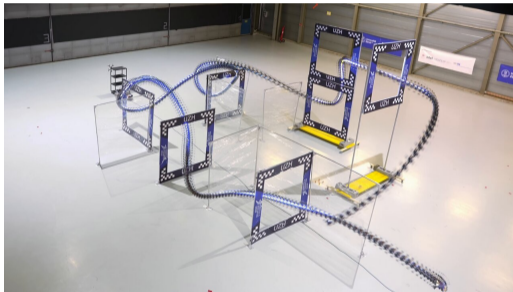
Task  
execution

References

- **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>

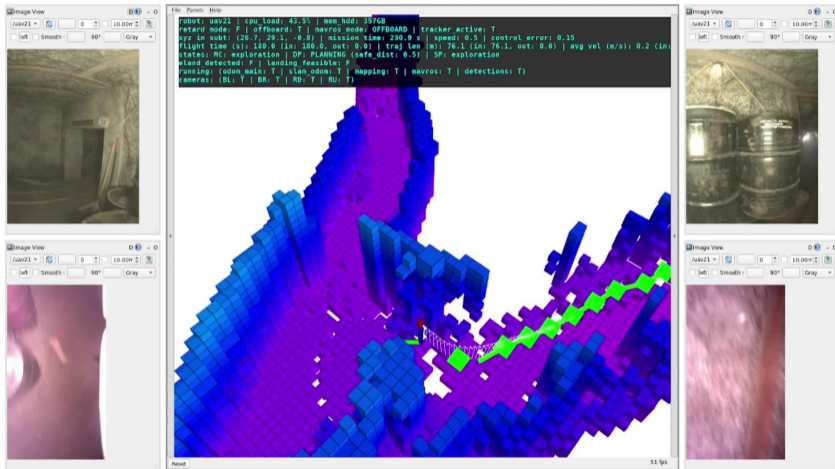


Video: <https://youtu.be/wR1niZvI3pI>

- [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

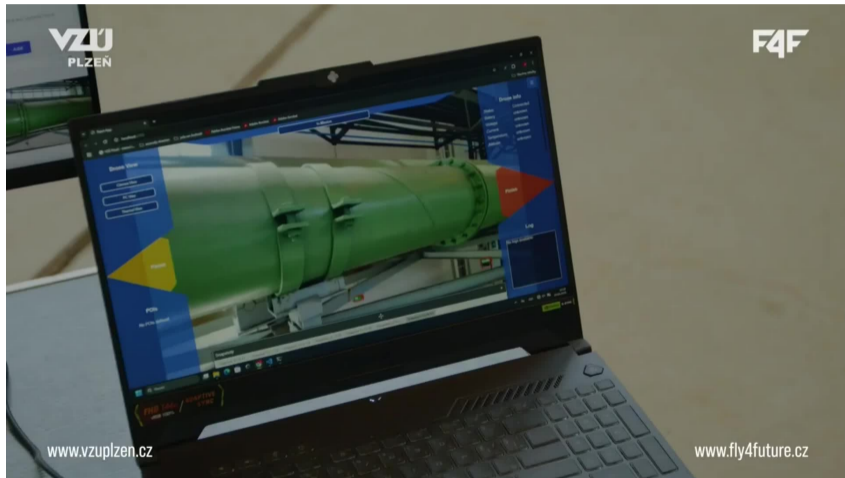
- Vision is simple on its own
- Computational resources are limiting as always
- Nowadays: such models can be executed directly on cameras (OAK-D)



Video: <https://youtu.be/slhJRZx26Ew>

## Structural failure detection

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



## 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

Lecture  
13: AI on the  
EDGE

Tomáš  
Báča

Computing  
continuum

Constraints

Compression  
Techniques

Accelerators

PEFT

AI in drone  
research

Control

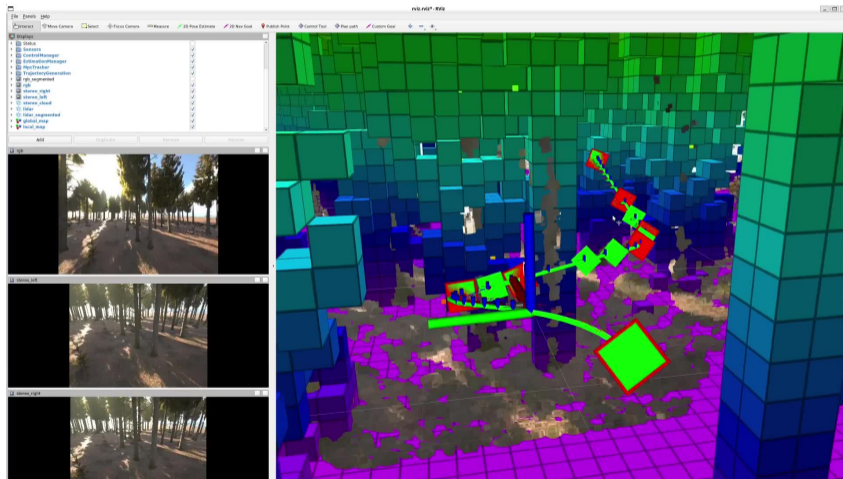
Computer  
vision

Mapping and  
Localization

Task  
execution

References

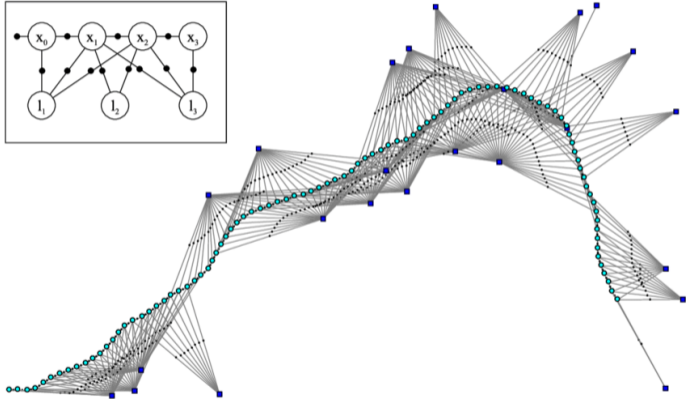
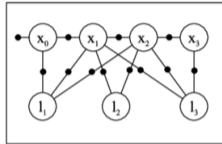
- marking occupied and free space into a map
- projecting colored image into the map
- post-processing the map for planning purposes



Link: <https://youtu.be/Jtxh1hZRs1A>

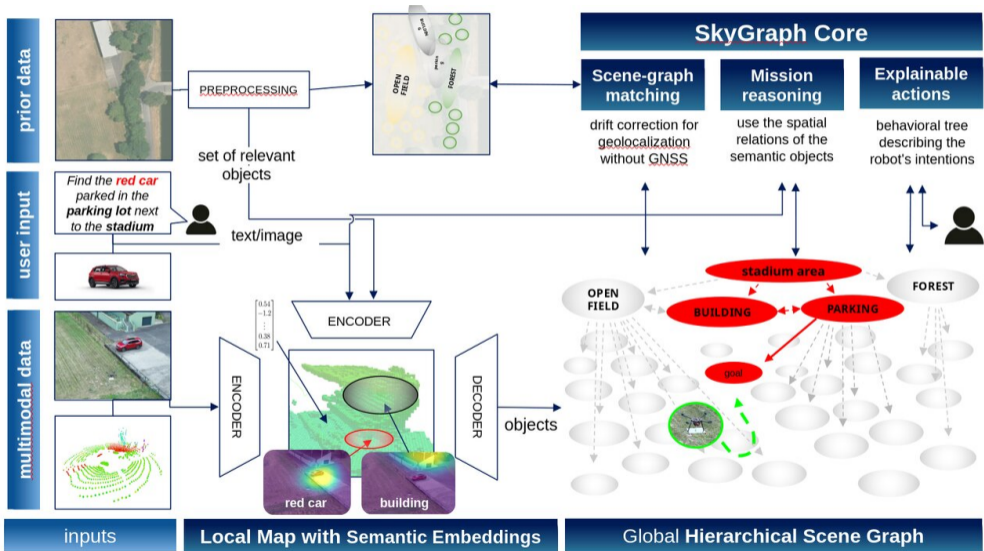
## Factor-graph SLAM

- 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



# Semantic Topometric Localization and Mapping

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



# Semantic mapping (current research)

Lecture  
13: AI on the  
EDGE

Tomáš  
Báča

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

The image shows a terminal window on the left and a 3D visualization interface on the right. The terminal window displays the following text:

```
megabedna~  
semantic_map_analyzer.srv.SetPrompt_Response(success=True, message="prompt set to 'wall', text embedding updated")  
  
michal@megabedna:~/git/semantic-map-analyzer/tmux_unre  
al$ ./set_prompt_service.sh car  
requester: making request: semantic_map_analyzer.srv.SetPrompt_Request(prompt='car')  
  
response:  
semantic_map_analyzer.srv.SetPrompt_Response(success=True, message="prompt set to 'car', text embedding updated")  
  
michal@megabedna:~/git/semantic-map-analyzer/tmux_unre  
al$ ./set_prompt_service.sh wall  
requester: making request: semantic_map_analyzer.srv.SetPrompt_Request(prompt='wall')  
  
response:  
semantic_map_analyzer.srv.SetPrompt_Response(success=True, message="prompt set to 'wall', text embedding updated")  
  
michal@megabedna:~/git/semantic-map-analyzer/tmux_unre  
al$
```

The 3D visualization interface on the right shows a 3D point cloud map of a room. The map is color-coded by semantic class. A legend on the left lists the classes: PointCloud, Image, PointCloud, MarkersMap, semantic\_node.js, similarity\_cloud, base\_ground\_plane, prompt\_label, of ground cloud, of non-ground cloud, Image, MarkersArray, and rgb. The main 3D view shows a room with a floor, walls, and ceiling. The floor is colored green, the walls are colored red and yellow, and the ceiling is colored blue. The prompt labels are visible as small colored dots on the map.

# Semantic mapping (potential research)

Lecture 13: AI on the EDGE

Tomáš Bába

Computing continuum

Constraints

Compression Techniques

Accelerators

PEFT

AI in drone research

Control

Computer vision

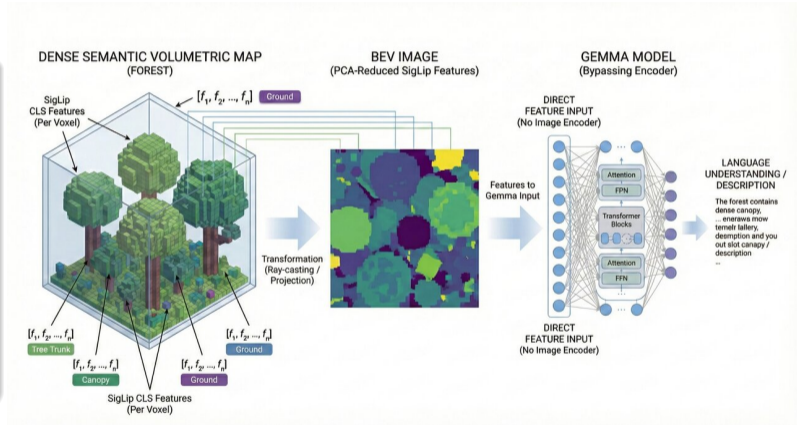
Mapping and Localization

Task execution

References

## 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



# Task execution — creation of behavioral trees

Lecture  
13: AI on  
the EDGE

Tomáš  
Báča

Computing  
continuum

Constraints

Compression  
Techniques

Accelerators

PEFT

AI in drone  
research

Control

Computer  
vision

Mapping and  
Localization

Task  
execution

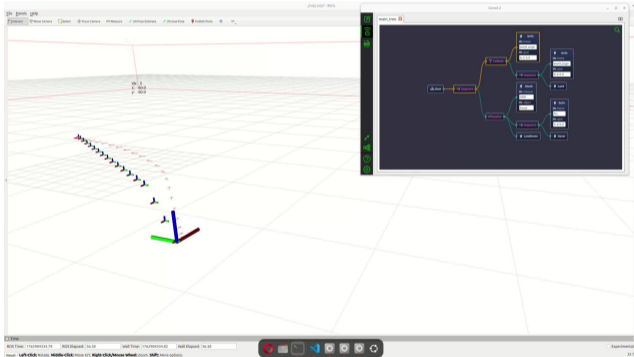
References

## 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



# References I

Lecture  
13: AI on  
the EDGE

Tomáš  
Báča

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. Koerich, 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://datature.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: 1558-0830. 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. Bauersfeld, 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.