CRIStAL
Centre de Recherche en Informatique Signal et Automatique de Lille

cnrs

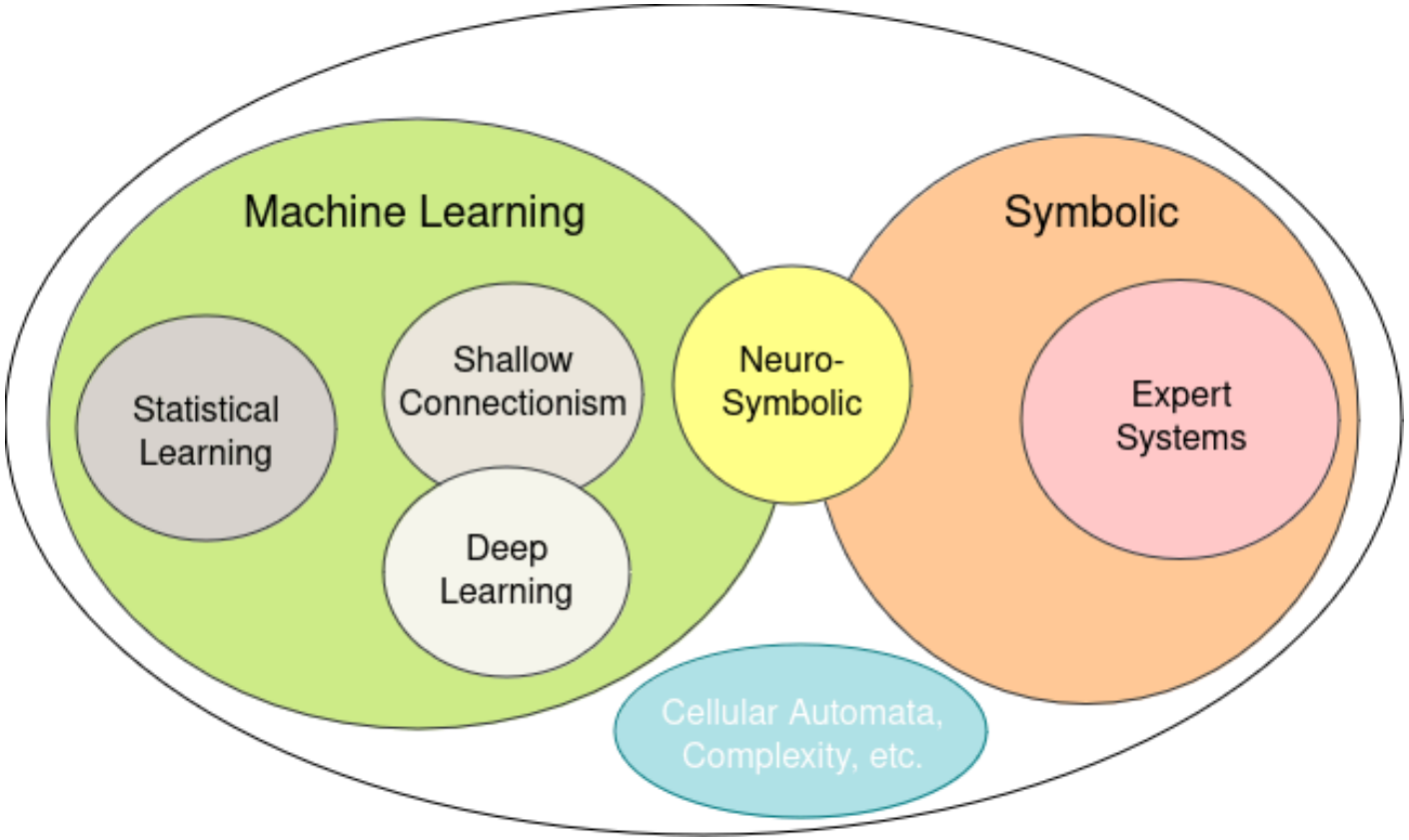
Région Hauts-de-France

Frugal AI

Romain Belmonte

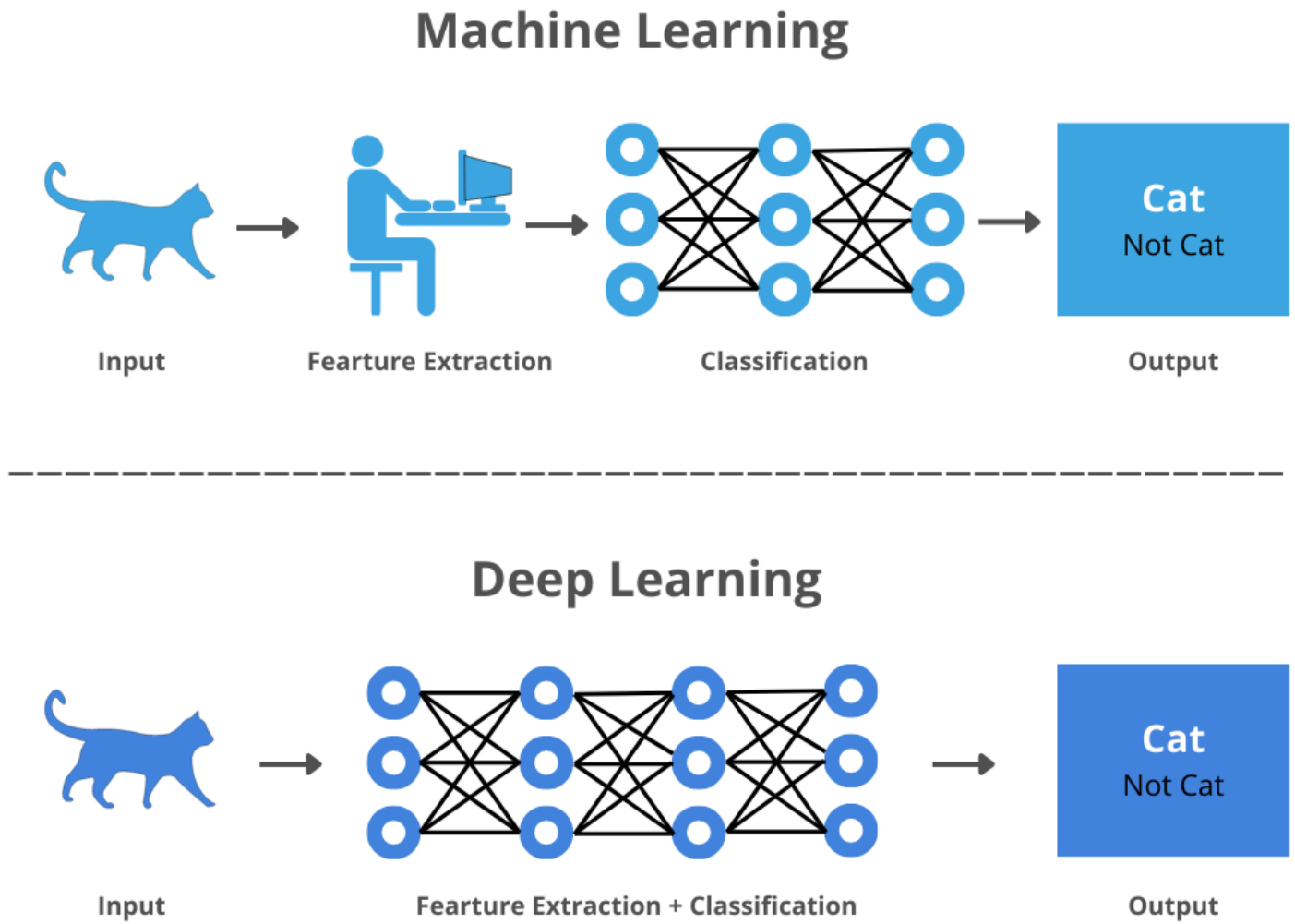
(september 24)

Artificial Intelligence



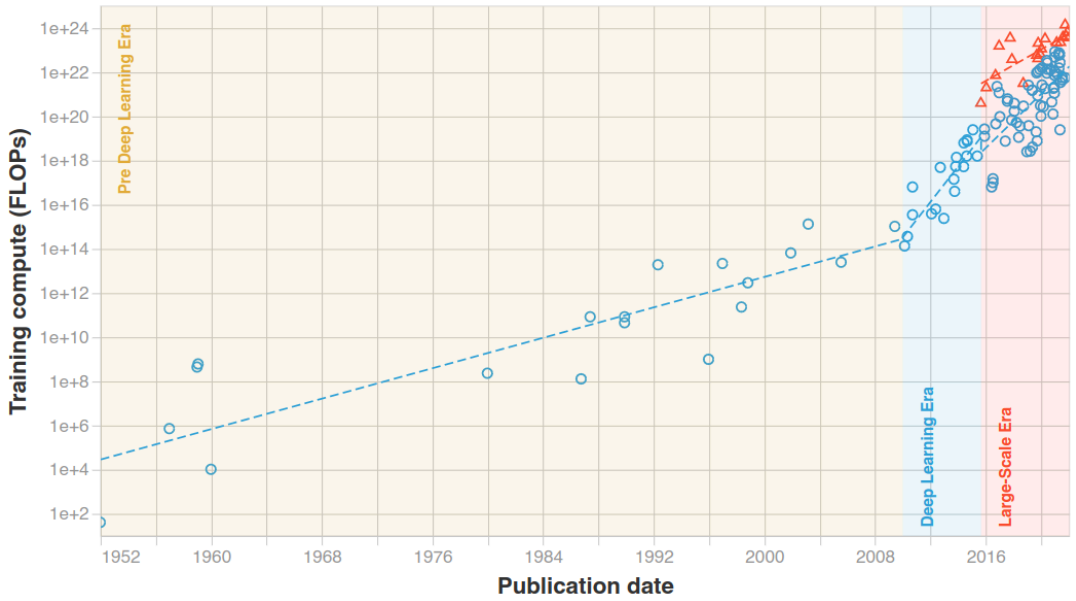
[1]

ML / DL


Machine Learning
Cat
Not Cat
Input
Fearture Extraction
Classification
Output
Deep Learning
Cat
Not Cat
Input
Fearture Extraction + Classification
Output

[2]

AI Scaling Laws



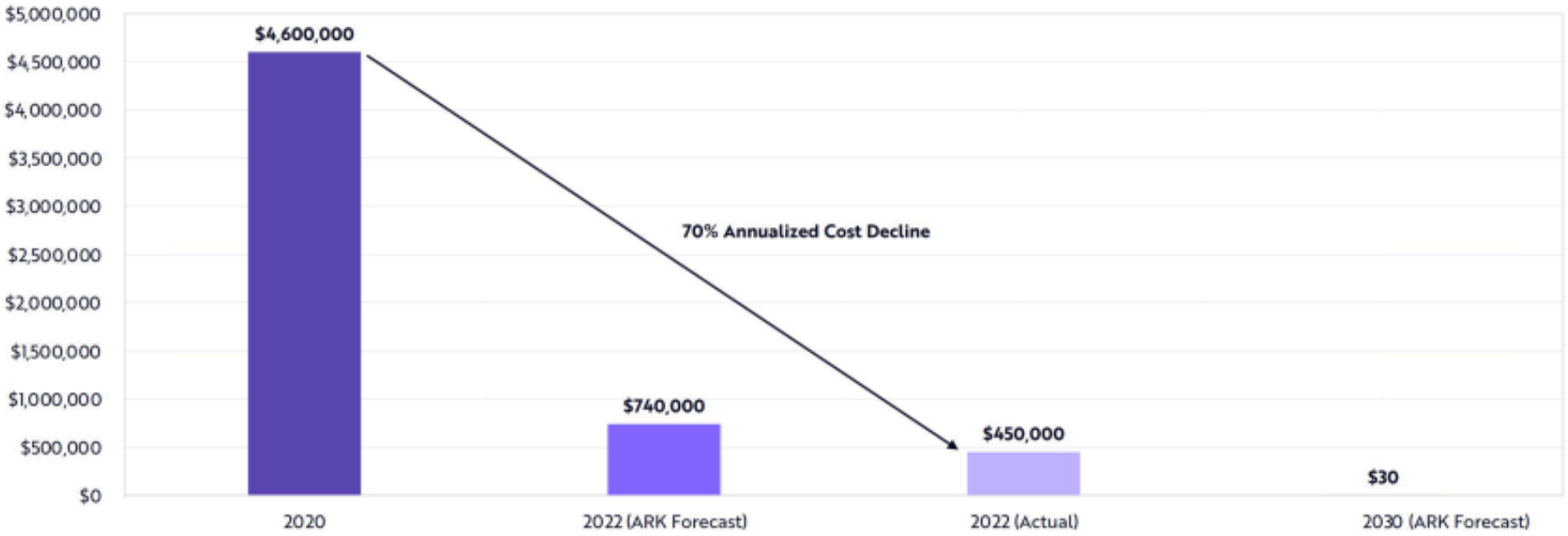
- Loss scales as a power-law with model size, dataset size, and the amount of compute used for training

[3, 4]

Footprint/Cost

Llama 3.1 - 405b

- 16 thousand H100 = a cumulative of 30.84M GPU hours of computation
- 8930 tons of CO2e (renewable energy) - an average French person = 9 tons/year

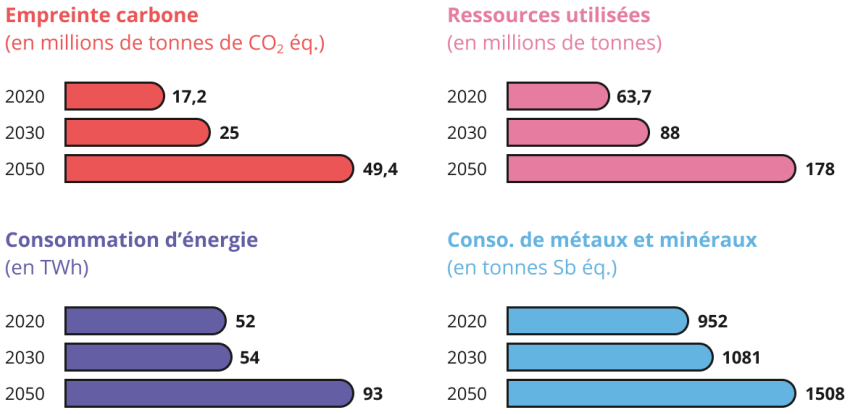


[5, 6]

IPCC guidelines

Des indicateurs issus d'une méthode normalisée d'analyse de cycle de vie qui comporte des définitions précises

Evolution de 4 indicateurs de l'impact environnemental du numérique dans le scénario tendanciel, en valeurs absolues.



Empreinte carbone : émissions de gaz à effets de serre exprimées en équivalent CO_2.

Ressources utilisées : indicateur MIPS qui considère cinq types de ressources, comprenant les ressources abiotiques (matériaux, énergie fossile...), la biomasse, les déplacements de terre mécaniques ou par érosion, l'eau, et l'air. Il donne une idée de l'effort effectué pour produire nos biens et services.

Consommation d'énergie finale : désigne l'énergie directement utilisée par l'utilisateur final, sous forme d'électricité ou de carburant.

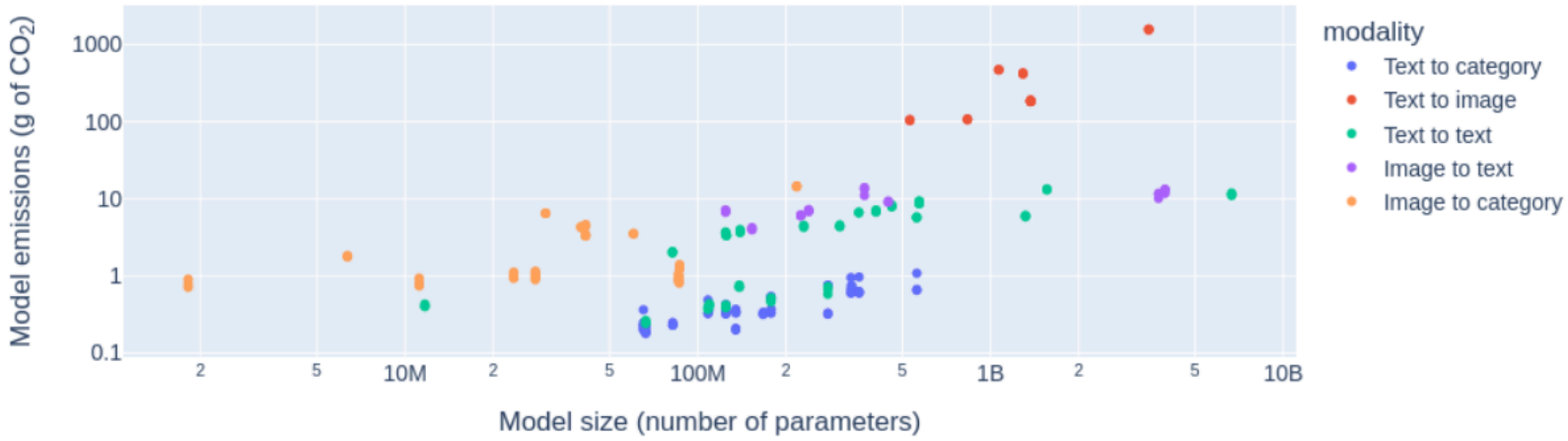
Consommation de métaux et minéraux : cet indicateur évalue la quantité de ressources minérales et métalliques extraites de la nature en équivalent antimoine (un élément chimique dont on retrouve le symbole Sb dans le tableau périodique des éléments). C'est un standard des analyses de cycle de vie qui permet de mesurer l'épuisement des ressources naturelles.

[7, 8]

Green/Red AI

Green/Red AI: Making efficiency an evaluation criterion + reporting the financial cost of developing, training, and running models

	BLOOMz-7B	BLOOMz-3B	BLOOMz-1B	BLOOMz-560M
Training energy (kWh)	51,686	25,634	17,052	10,505
Finetuning energy (kWh)	7,571	3,242	1,081	543
Inference energy (kWh)	1.0×10^{-4}	7.3×10^{-5}	6.2×10^{-5}	5.4×10^{-5}
Cost parity (# inferences)	592,570,000	395,602,740	292,467,741	204,592,592

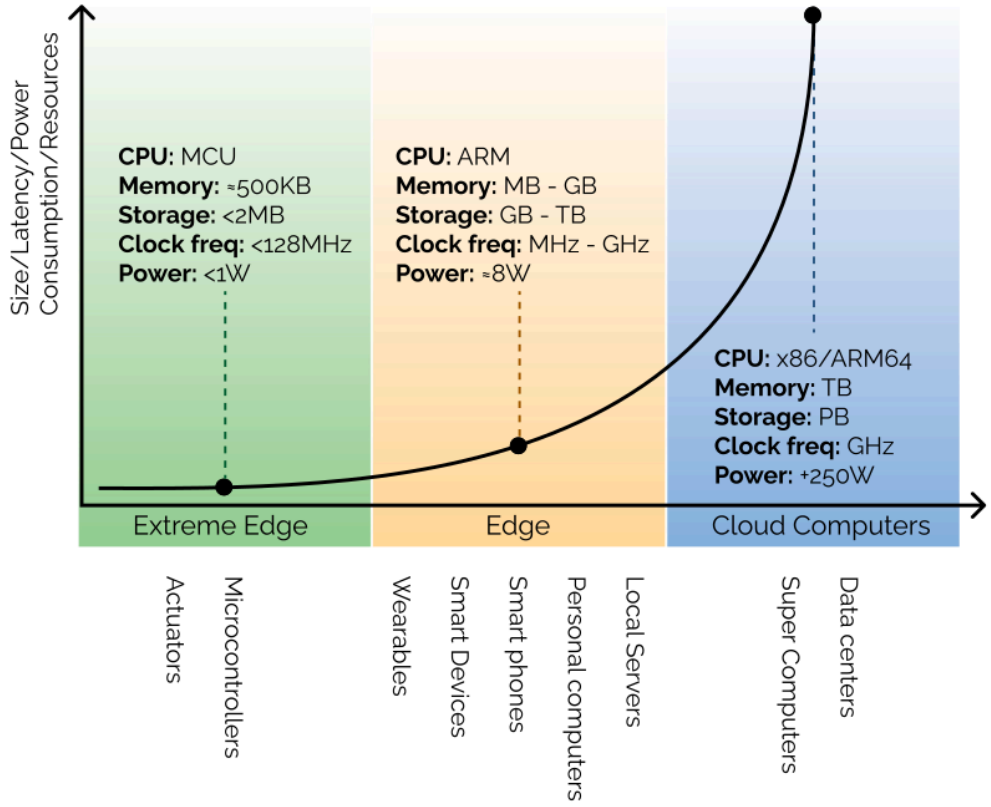


100 km with an electric car = 17 kWh; 1 h of video streaming = 70 g of CO2

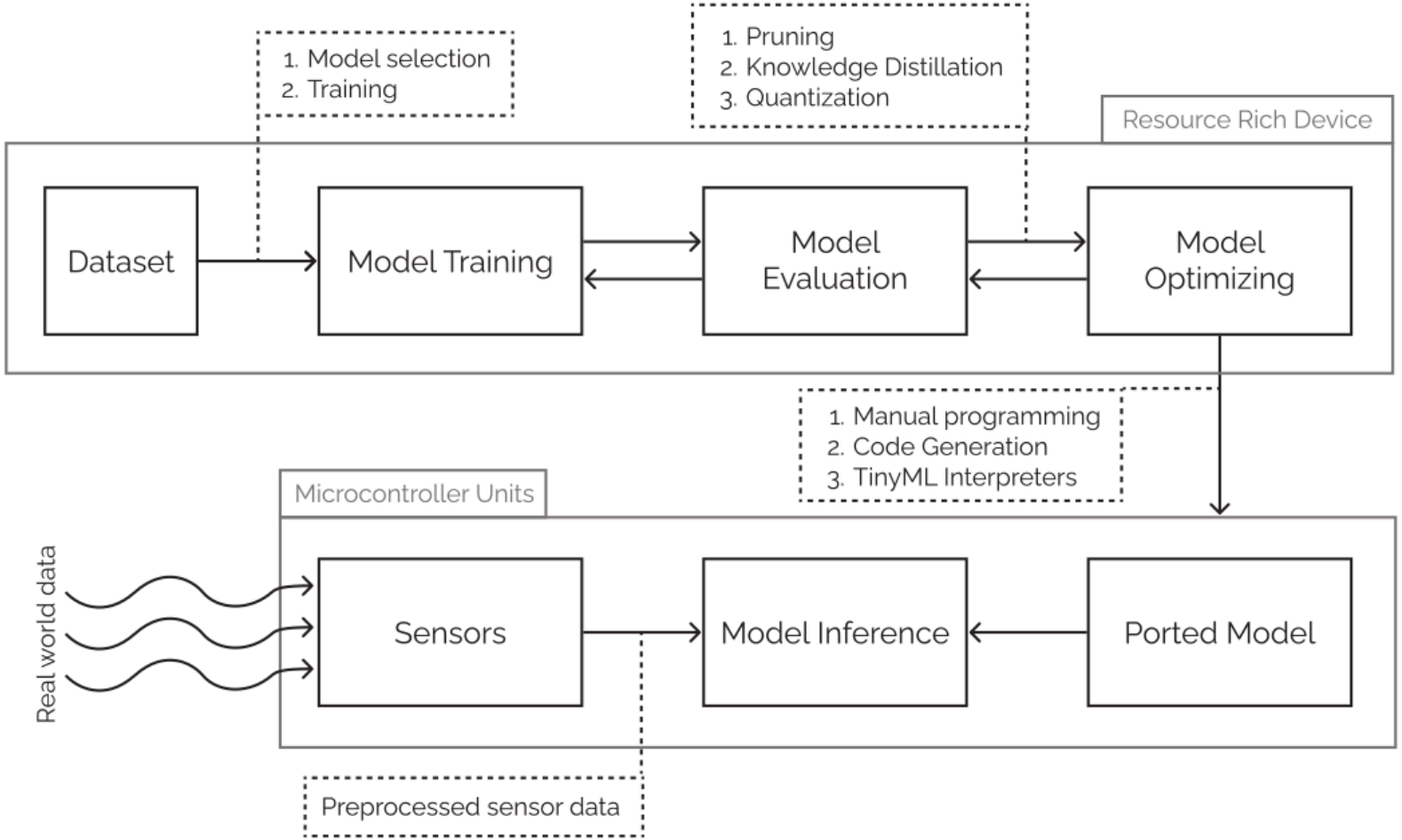
[9, 10, 11]

Challenges

- Limited Computational Power
- Memory Limitations
- Energy Efficiency
- Deployment/Maintenance Complexity
- Ethics, Privacy and Security Concerns



Pipeline


1. Model selection
2. Training
1. Pruning
2. Knowledge Distillation
3. Quantization
Resource Rich Device
Dataset
Model Training
Model Evaluation
Model Optimizing
1. Manual programming
2. Code Generation
3. TinyML Interpreters
Microcontroller Units
Real world data
Sensors
Model Inference
Ported Model
Preprocessed sensor data

[12]

Overview

1. **Data**
   - Data overload
   - Unlabeled data
   - Limited data
   - Data management
2. **Model**
   - Energy consumption
   - Memory efficiency
   - Training overhead
3. **Miscellaneous**
   - Tools
   - Resources
   - Hardware
   - Advances


DATA

Data

- **Data overload:** selection, dimensionality reduction, sampling, distillation
- **Unlabeled data:** crowdsourcing, generative models, active learning, semi-supervision, weak supervision, self-supervision
- **Limited data:** augmentation, synthetic generation, transfer learning, external data, regularization, physics-informed, few-shot learning
- **Data management:** imputation, cleaning, feature engineering, preprocessing, ensembling, cross-validation, incremental learning, memory augmented network

Data

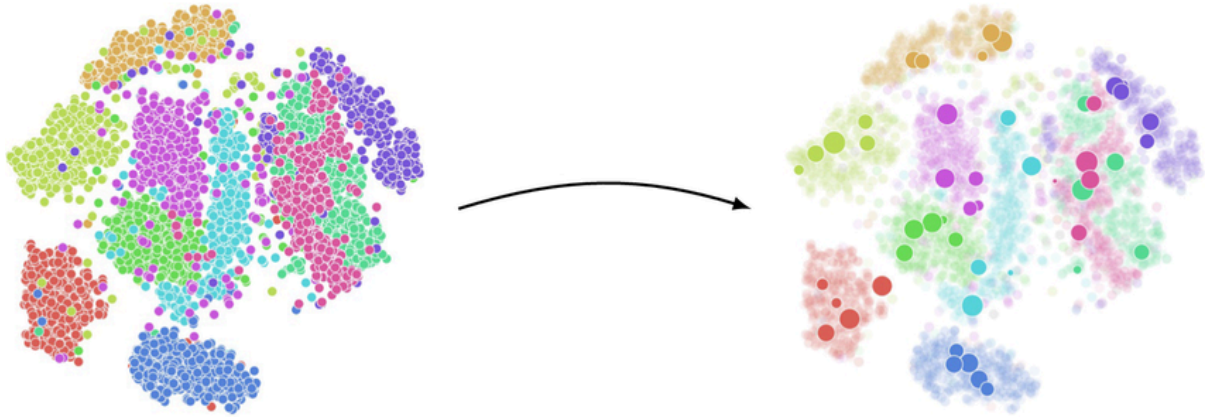
Data overload: selection, dimensionality reduction, sampling, distillation

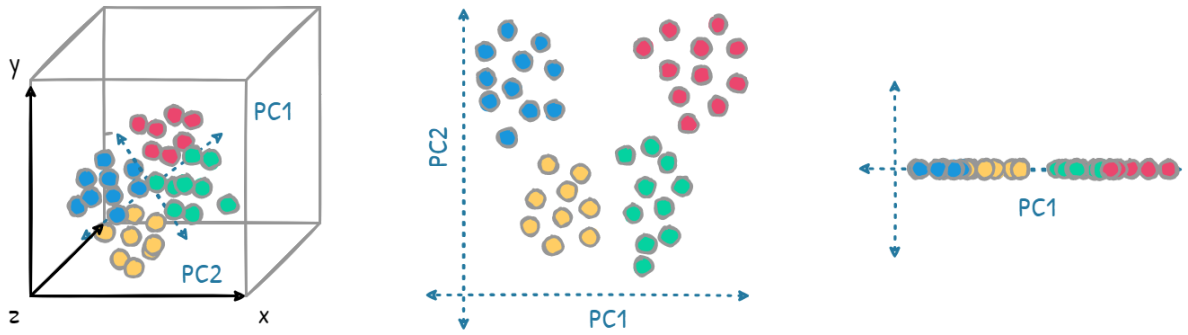
Data Overload

Garbage In, Garbage Out

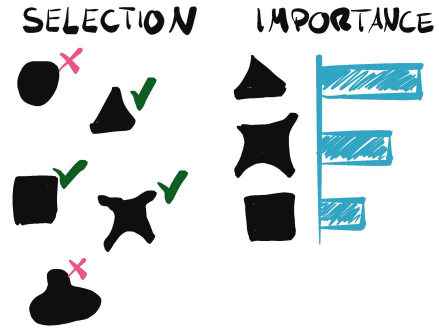
Data Overload

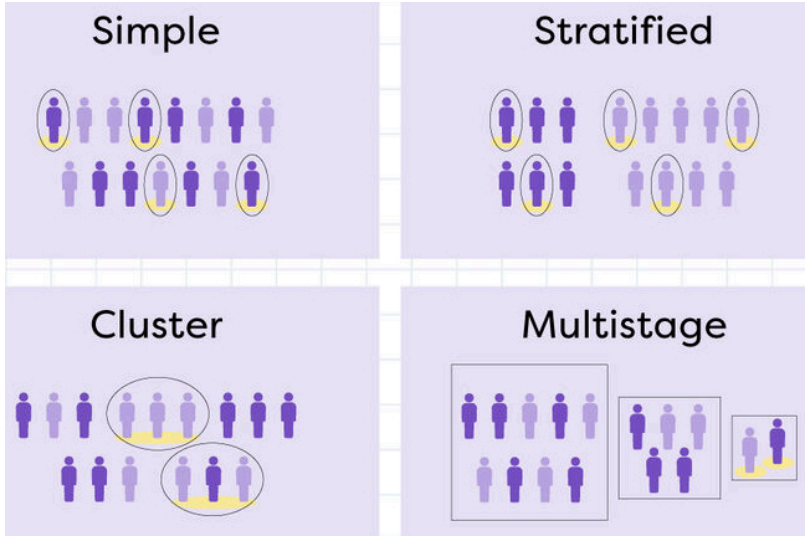
Selection, dimensionality reduction, sampling





[13]

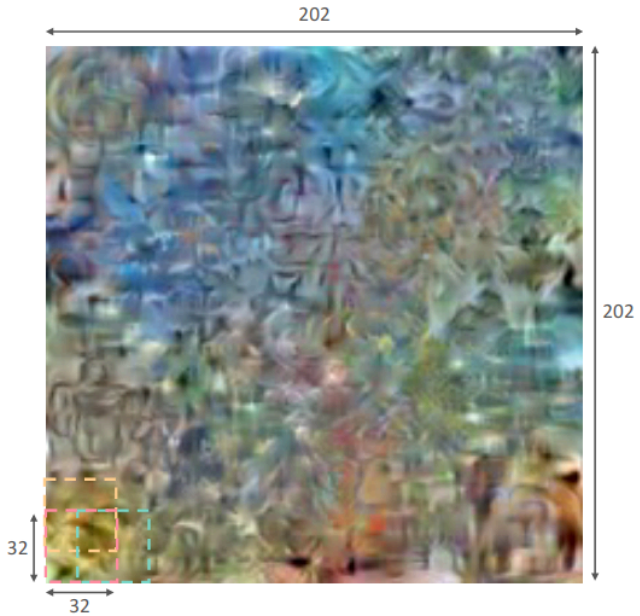




Data Overload

Distillation

Classic Dataset Distillation
1 Image-Per-Class



Poster Dataset Distillation (PoDD)
0.4 Image-Per-Class

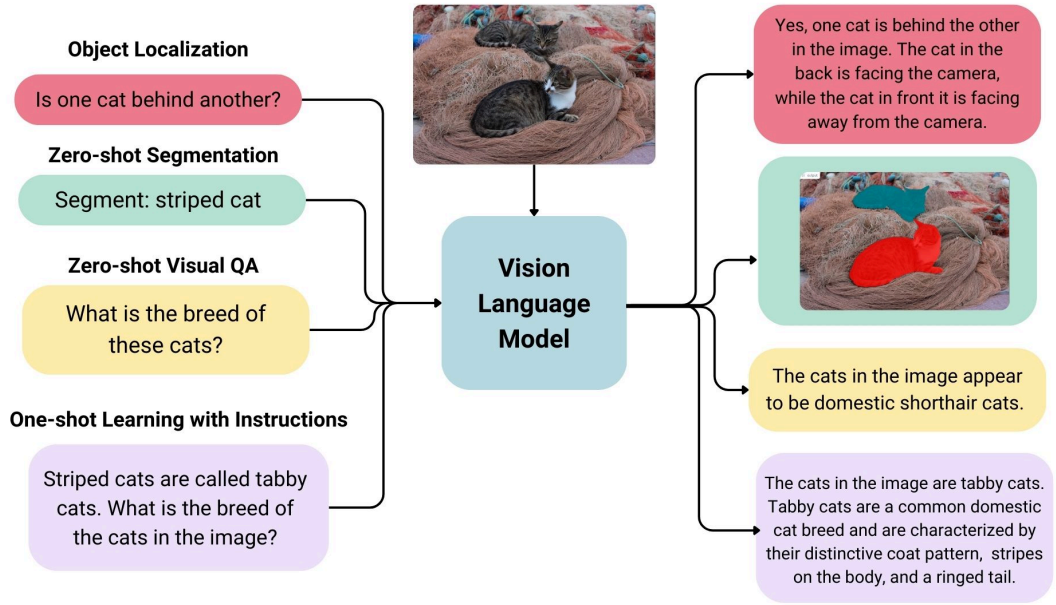
[14]

Data

Unlabeled data: crowdsourcing, generative models, active learning, semi-supervision, weak supervision, self-supervision

Unlabeled Data

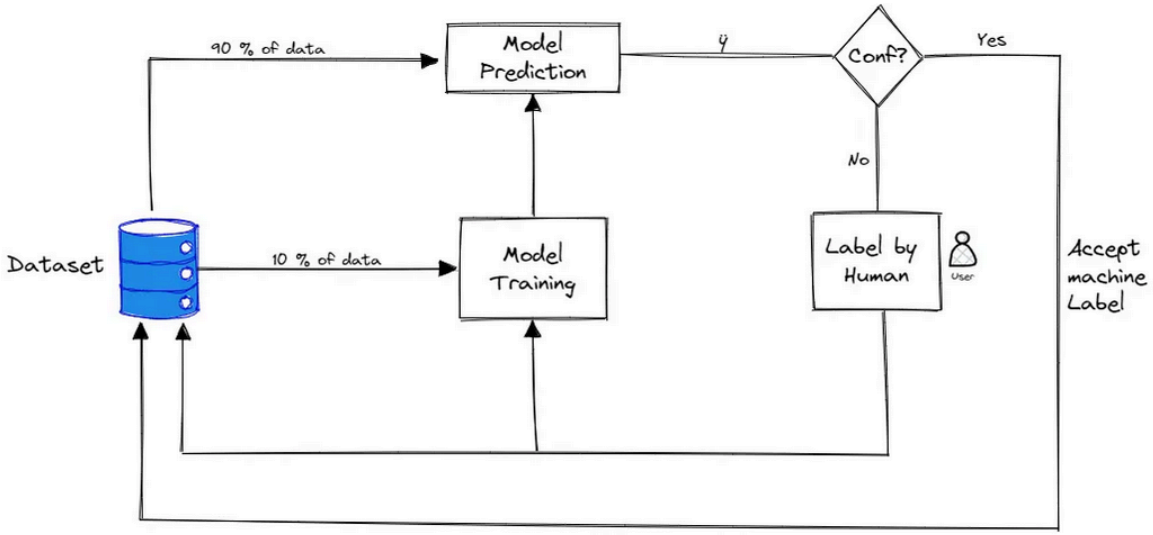
Crowdsourcing, generative models

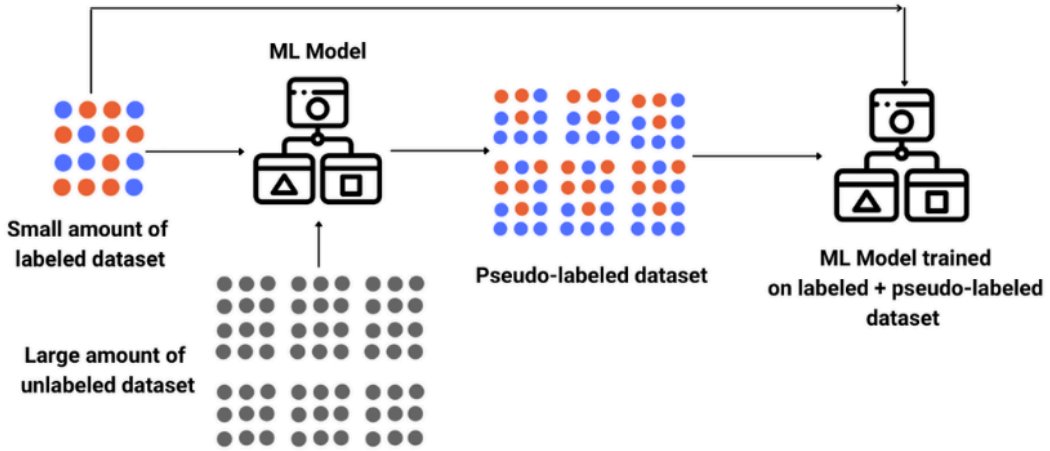


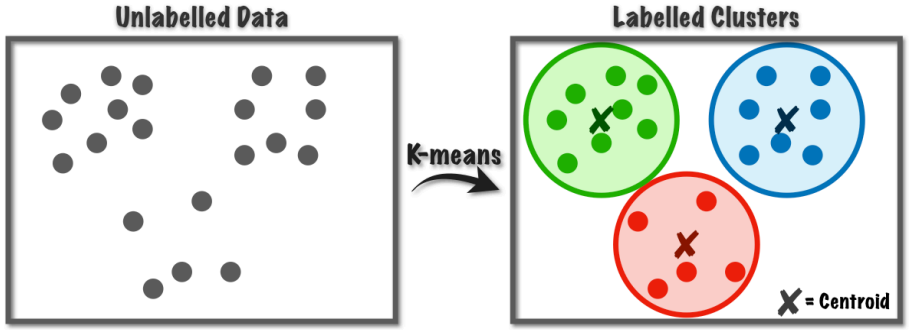
[Pro bono projects with PeopleForAI]

Unlabeled Data

Active learning, semi-supervision, weak supervision



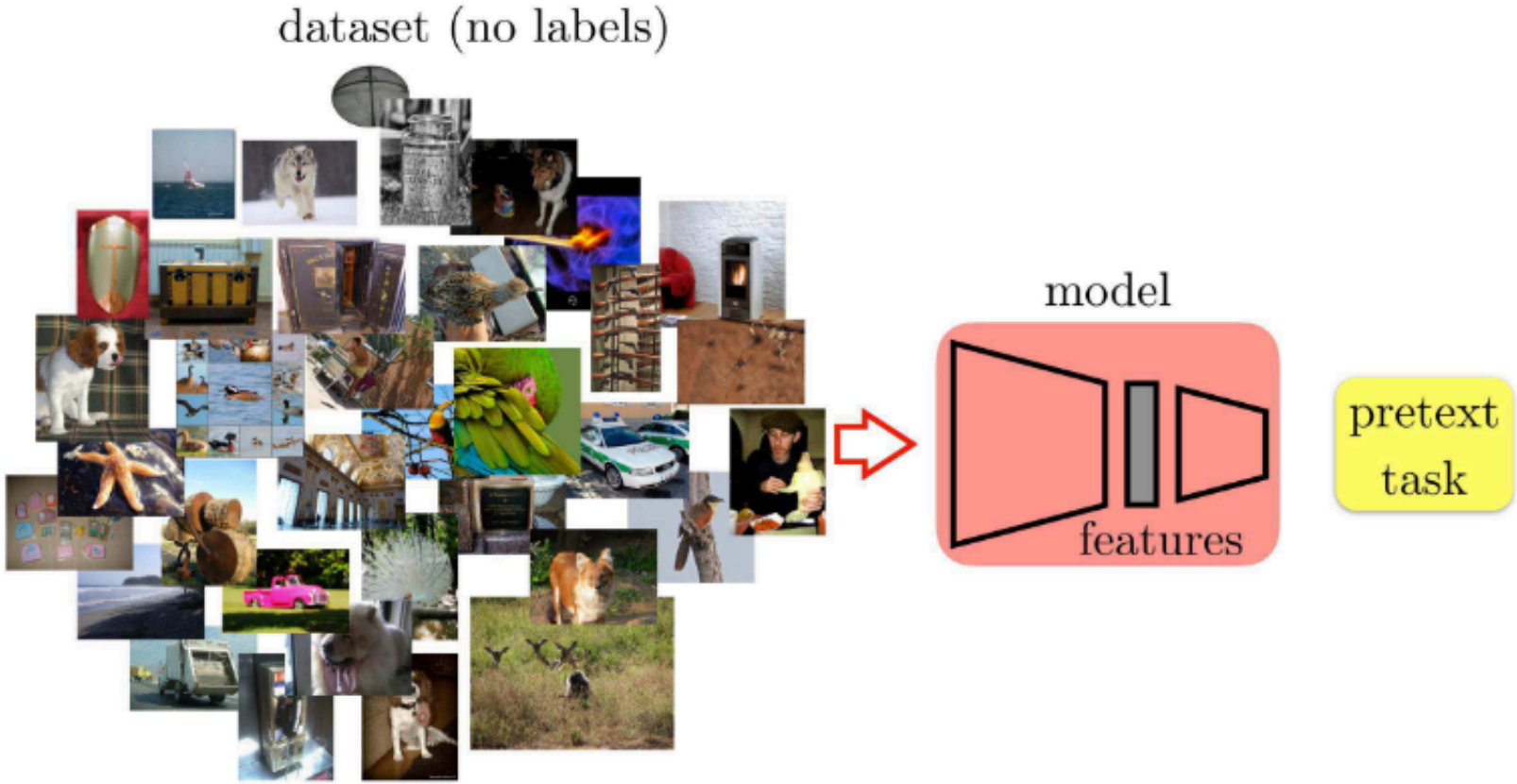




[15]

Unlabeled Data

Self-supervision



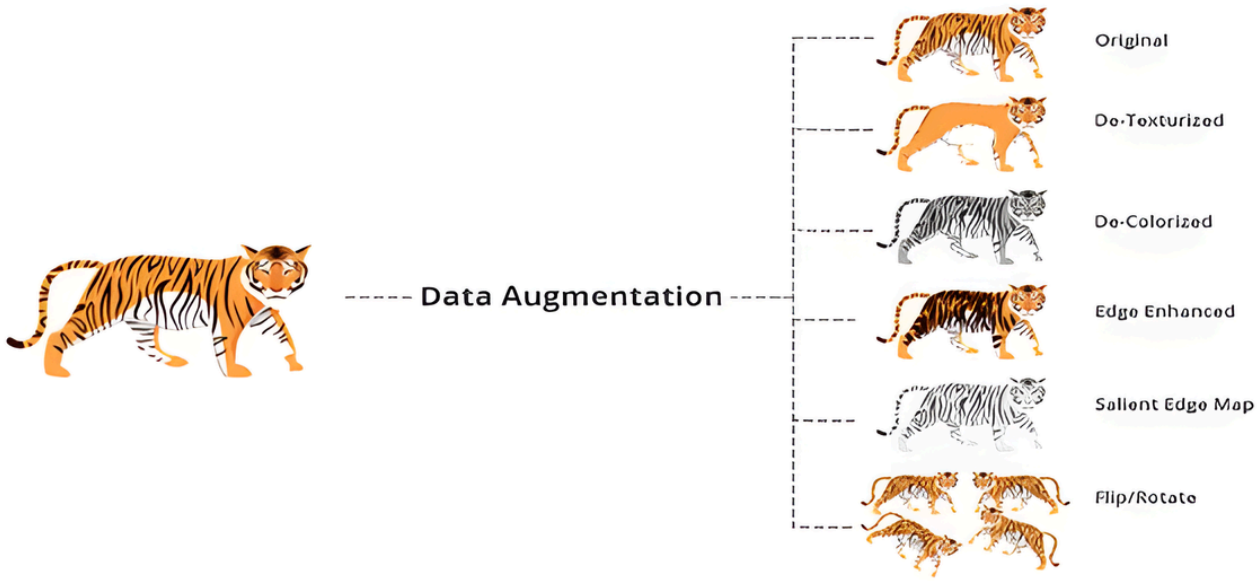
[16]

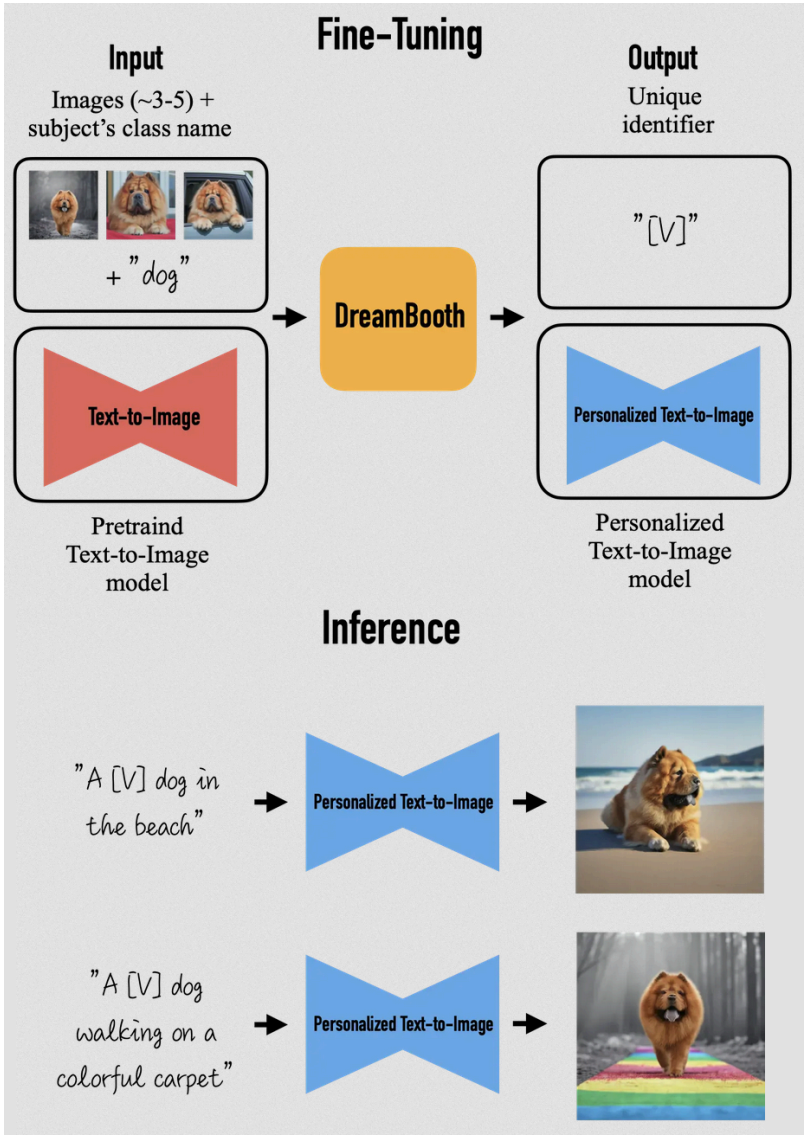
Data

Limited data: augmentation, synthetic generation, transfer learning, external data, regularization, physics-informed, few-shot learning

Limited Data

Augmentation, synthetic generation

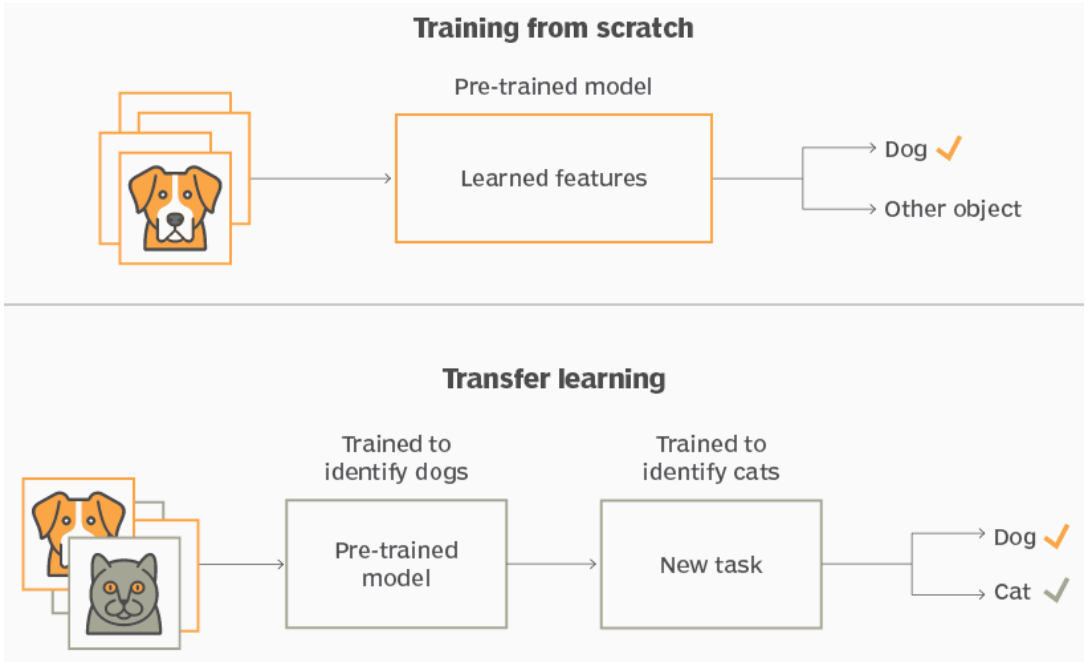


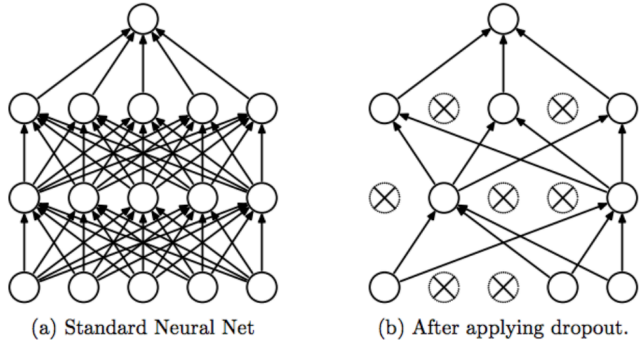


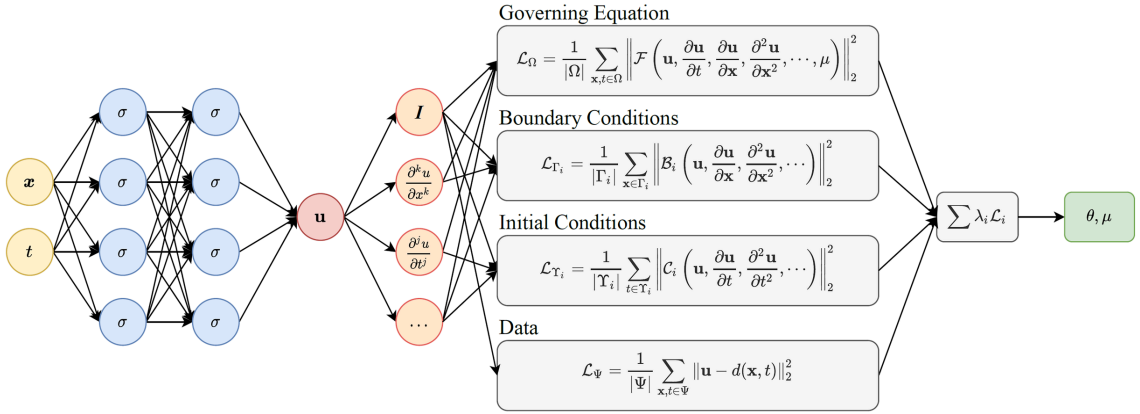
[Nvidia Omniverse]

Limited Data

Transfer learning, external data, regularization, physics-informed



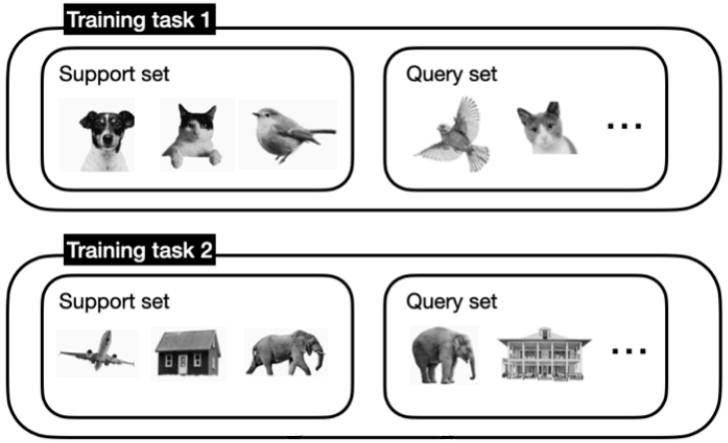


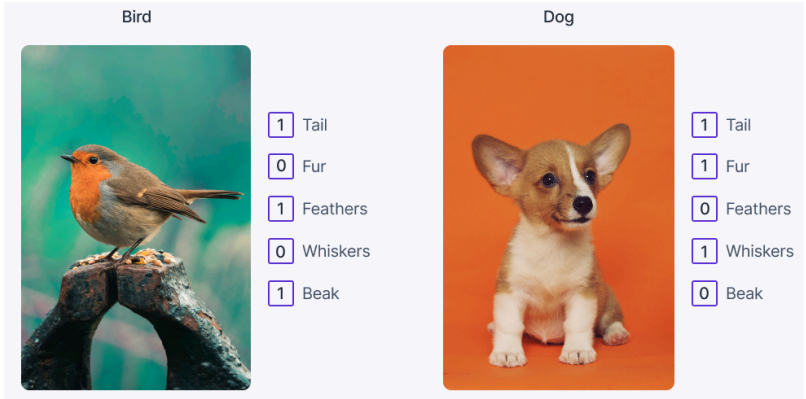


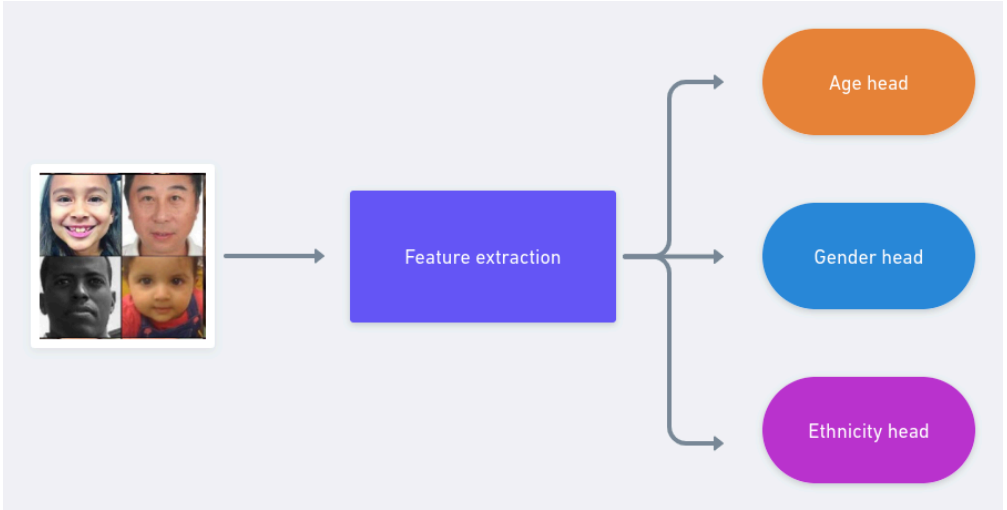
[17]

Limited Data

Few-shot learning







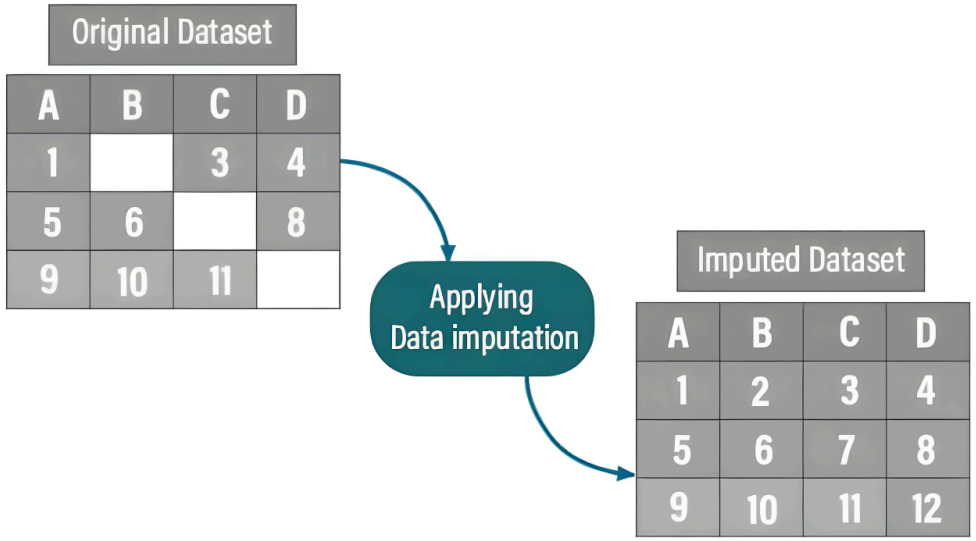
[18, 19]

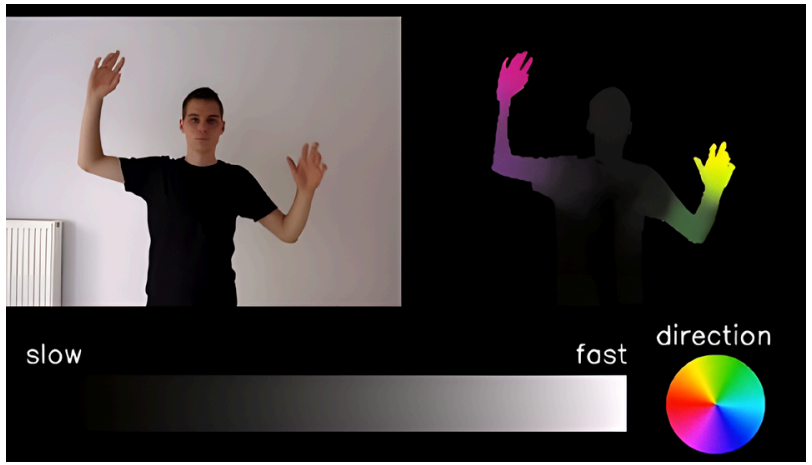
Data

Data management: imputation, cleaning, feature engineering, preprocessing, ensembling, cross-validation, incremental learning, memory augmented network

Data Management

Imputation, cleaning, feature engineering, preprocessing



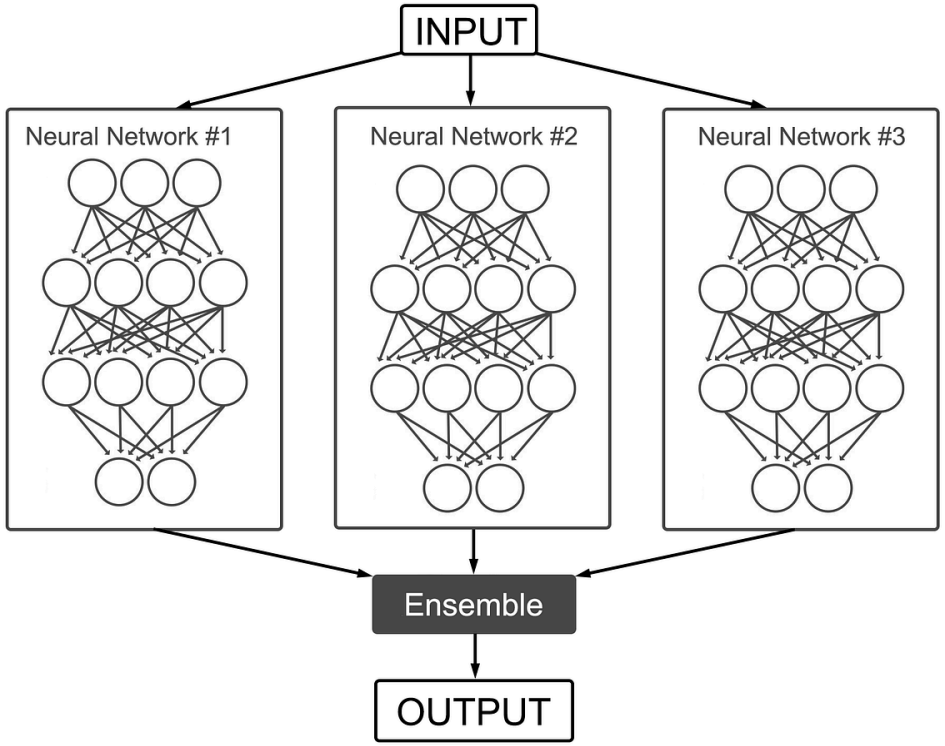


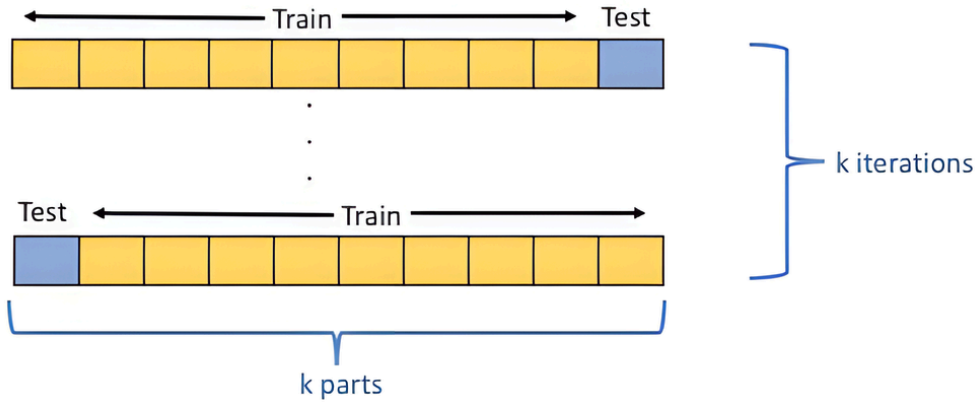
Original

Canny

Data Management

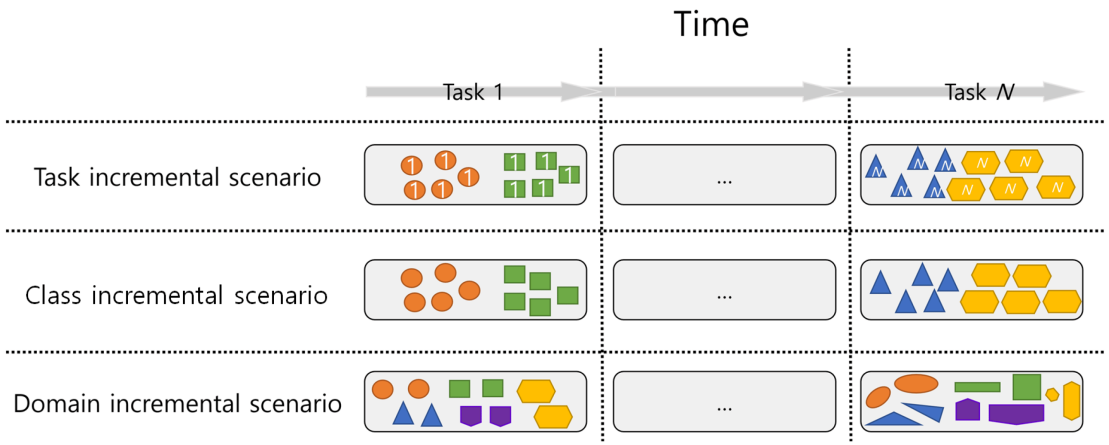
Ensembling, cross-validation

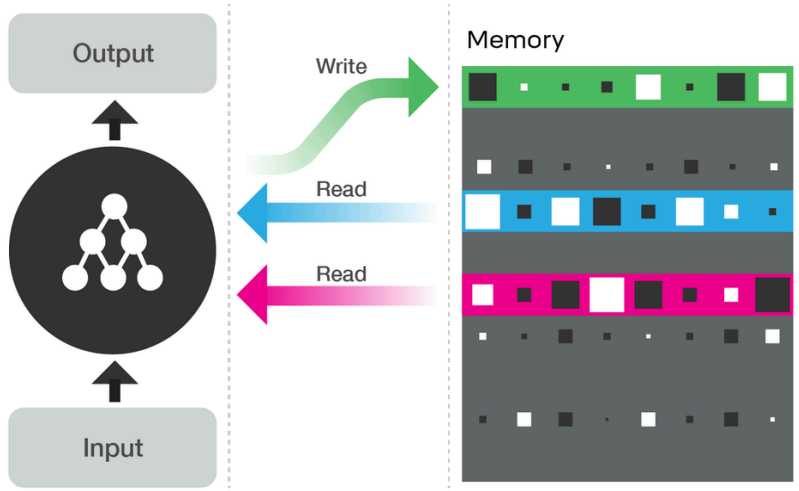




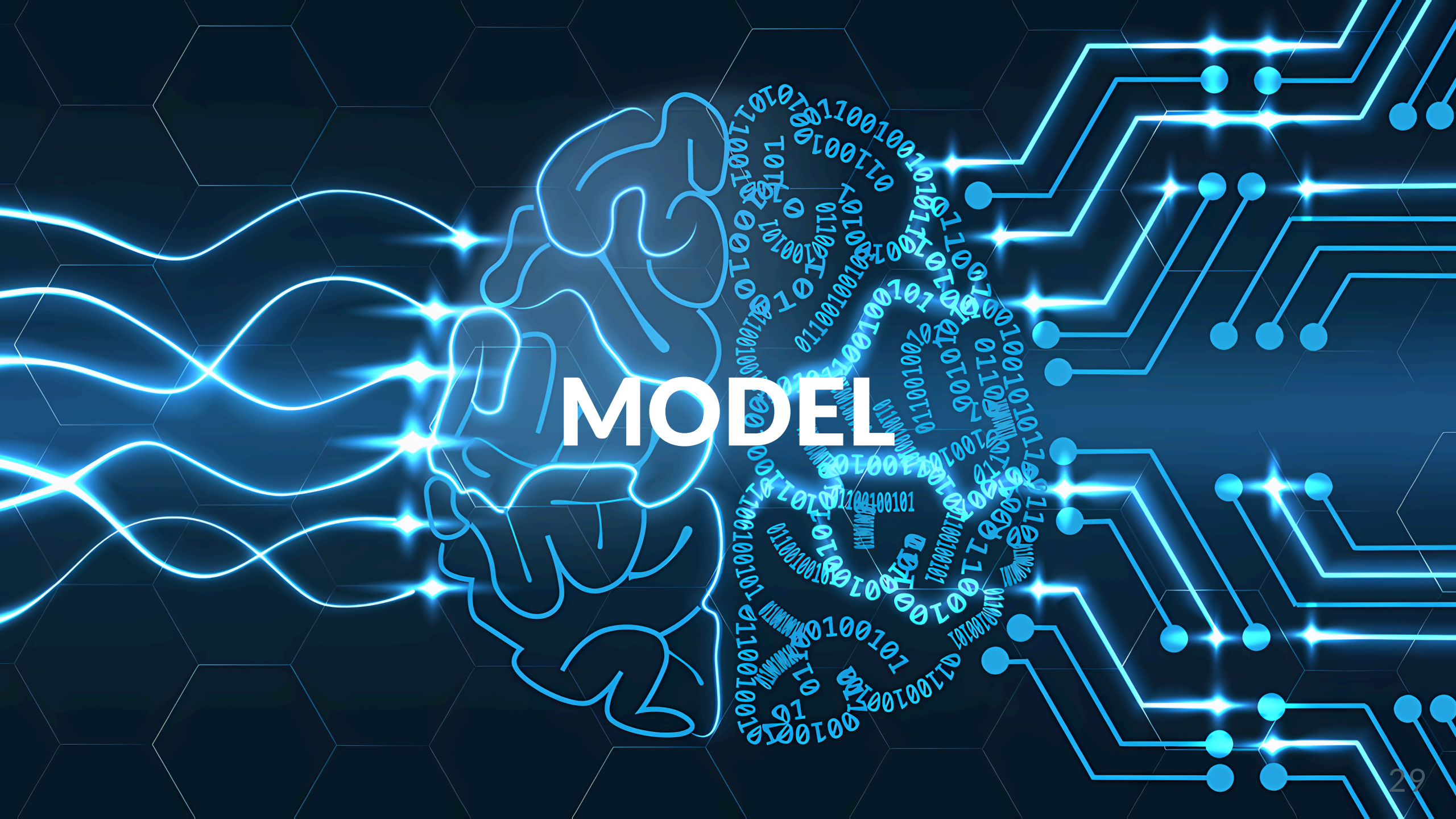
Data Management

Incremental learning, memory augmented network





[20, 21]


MODEL

Model

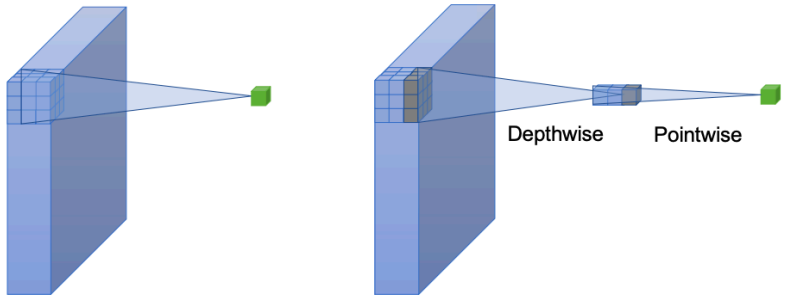
- **Energy consumption:** efficient algorithms-architectures, skip mechanism, gate mechanism, early termination, compilation, specialized hardware
- **Memory efficiency:** batching, accumulation, weight sharing, tensor decomposition, pruning, quantization, distillation, sparse representation
- **Training overhead:** early stopping, mixed precision, scheduling, distributed learning, dynamic pruning, low-rank adaptation

Model

Energy consumption: efficient algorithms-architectures, skip mechanism, gate mechanism, early termination, compilation, specialized hardware

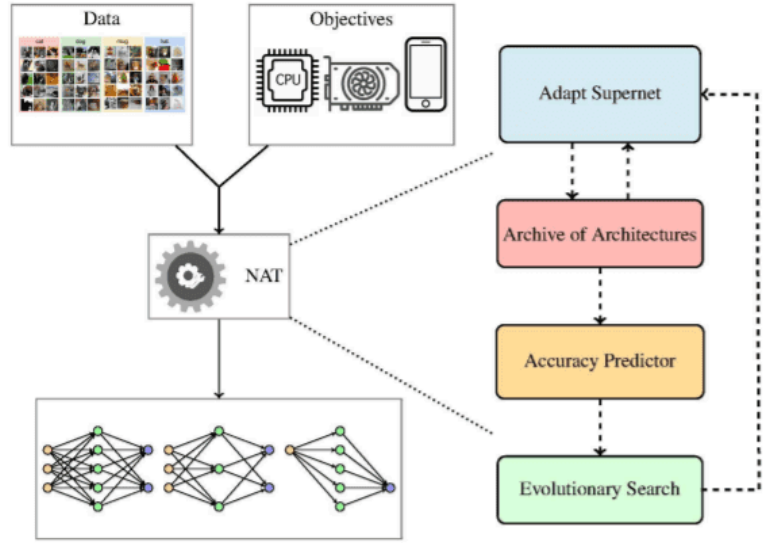
Energy Consumption

Efficient algorithms-architectures



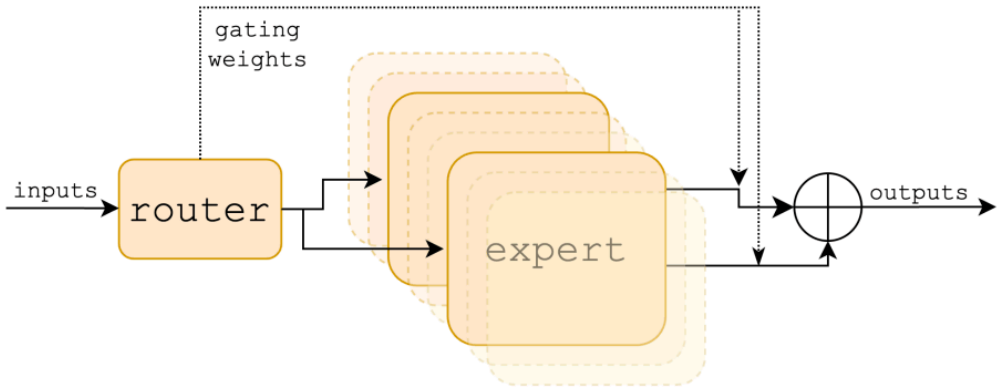
Model	Multi-Layer Perceptron (MLP)	Kolmogorov-Arnold Network (KAN)
Model (Shallow)	(a) *fixed* activation functions on *nodes*; *learnable* weights on *edges*	(b) *learnable* activation functions on *edges*; sum operation on *nodes*

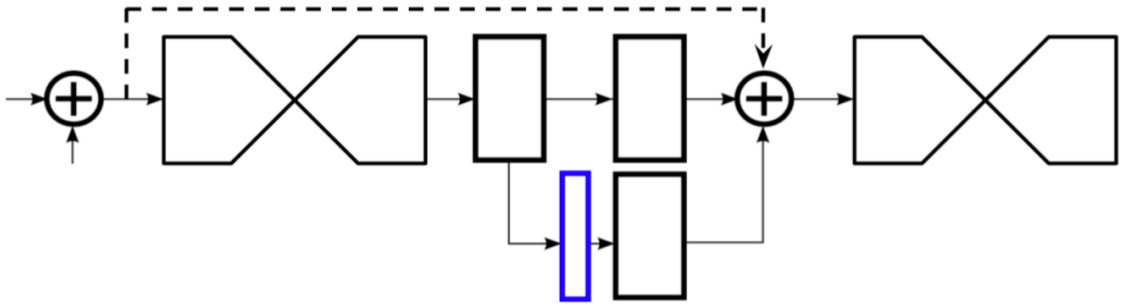
[22, 23, 24, 25]



Energy Consumption

Skip mechanism, gate mechanism, early termination

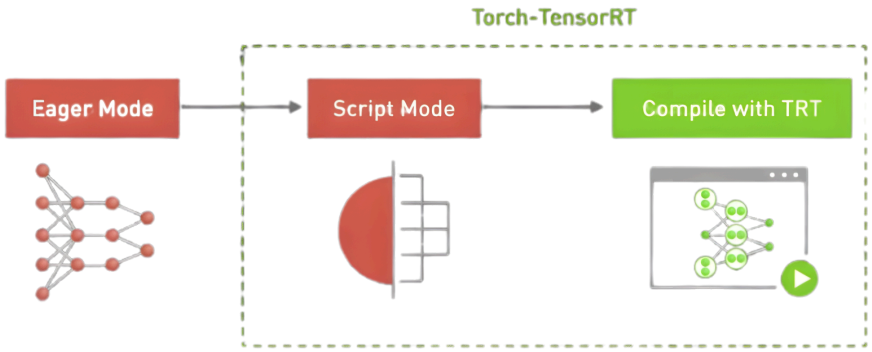


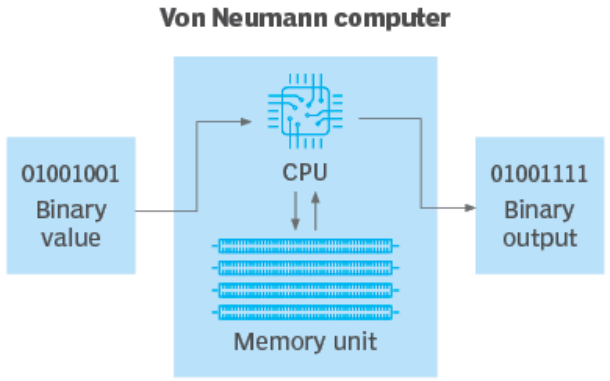


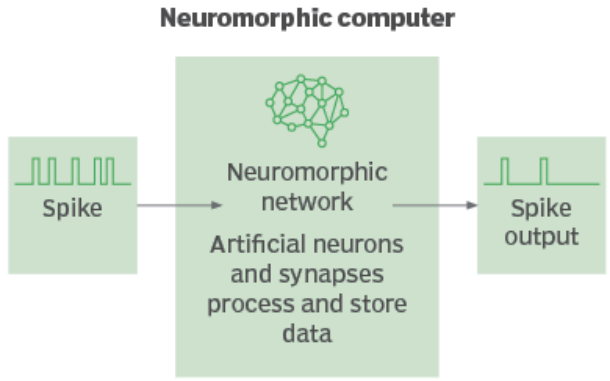
[26]

Energy Consumption

Compilation, specialized hardware







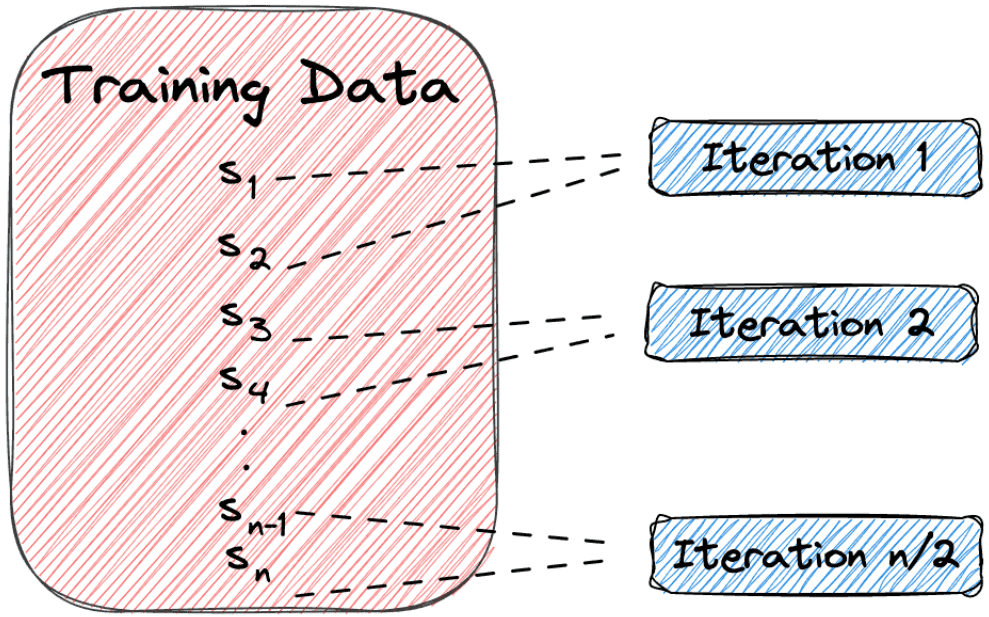
[27]

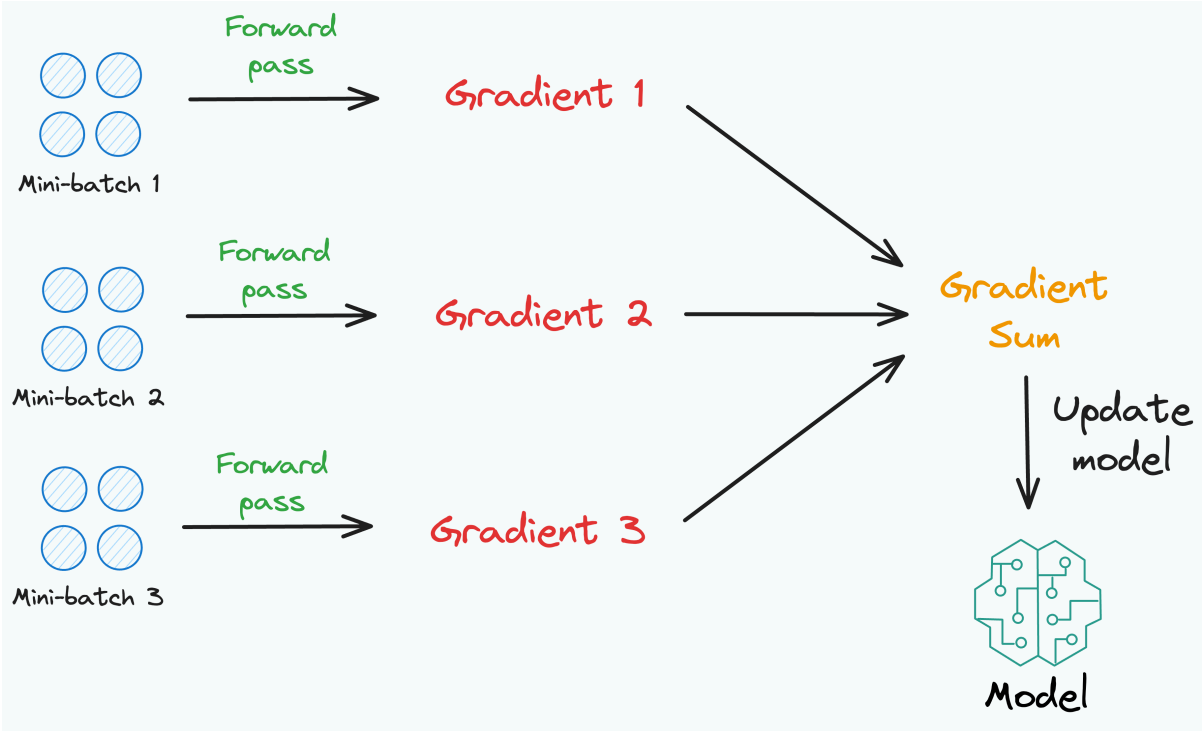
Model

Memory efficiency: batching, accumulation, weight sharing, tensor decomposition, pruning, quantization, distillation, sparse representation

Memory Efficiency

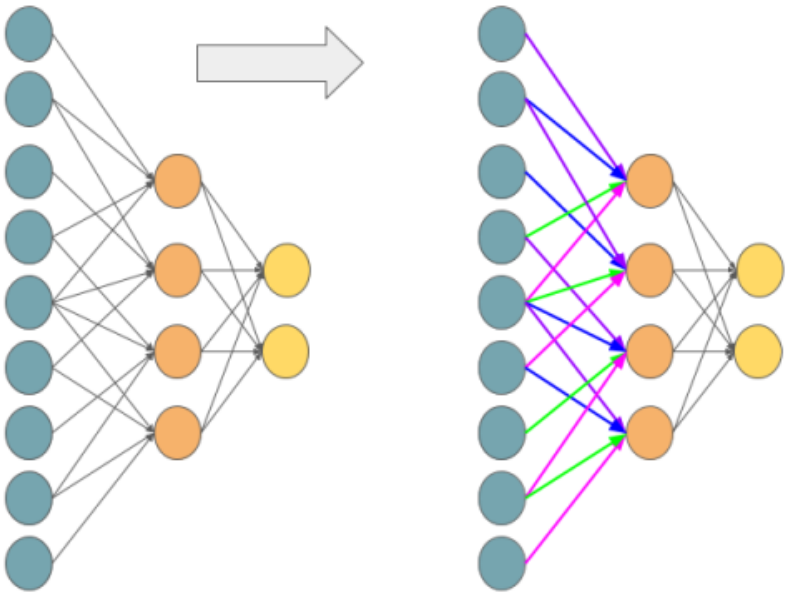
Batching, accumulation

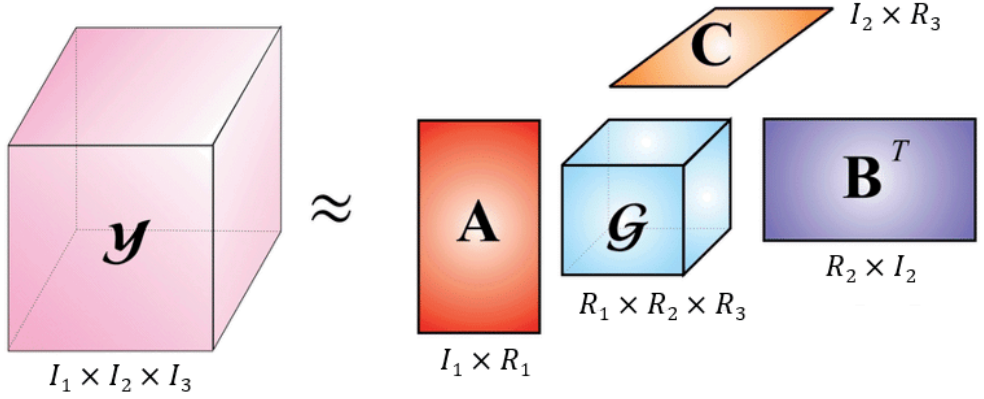




Memory Efficiency

Weight sharing, tensor decomposition

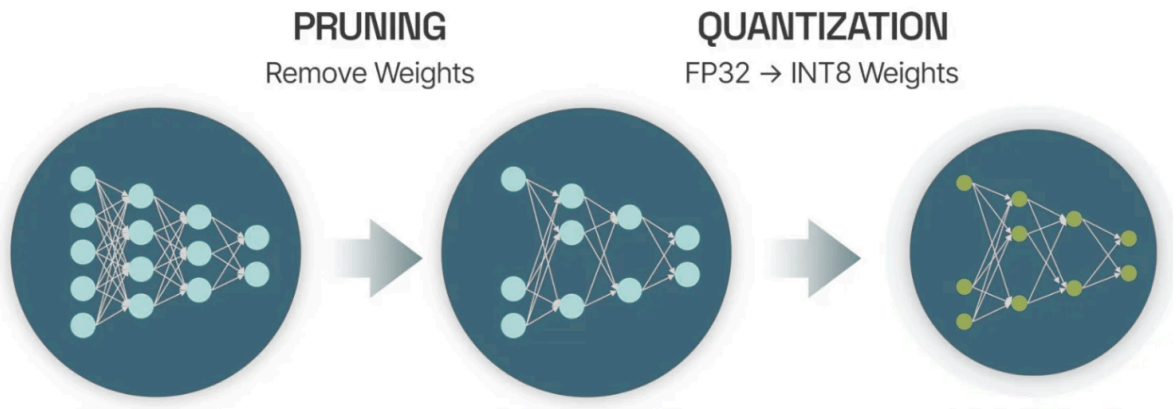


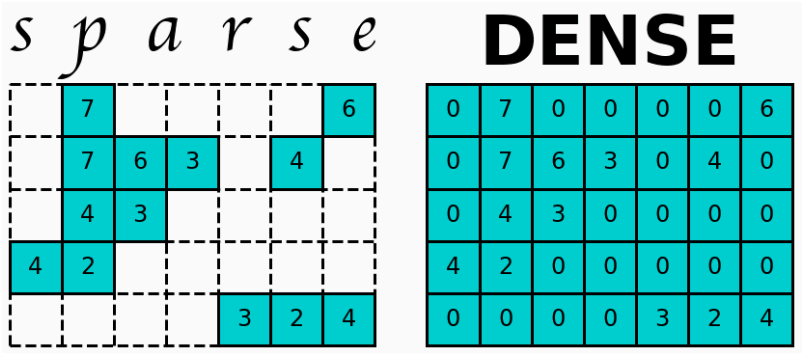


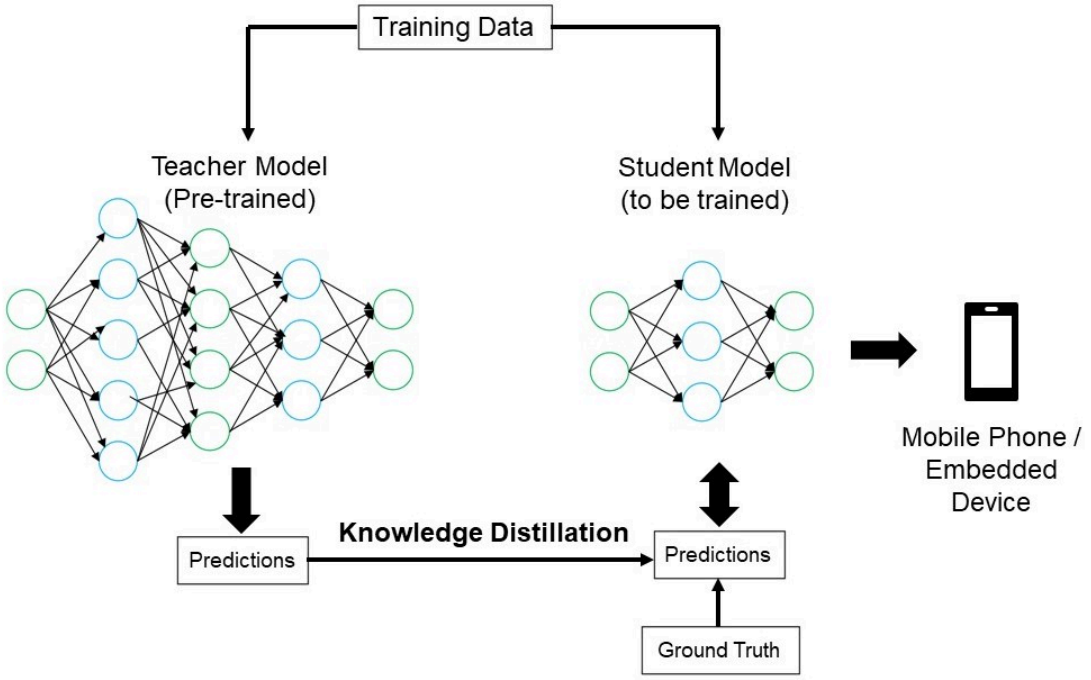
[28, 29]

Memory Efficiency

Pruning, quantization, distillation, sparse representation







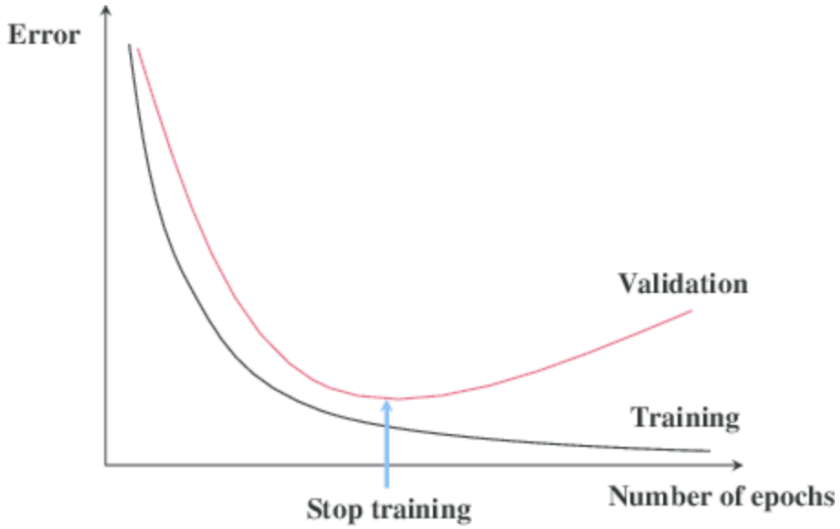
[30, 31]

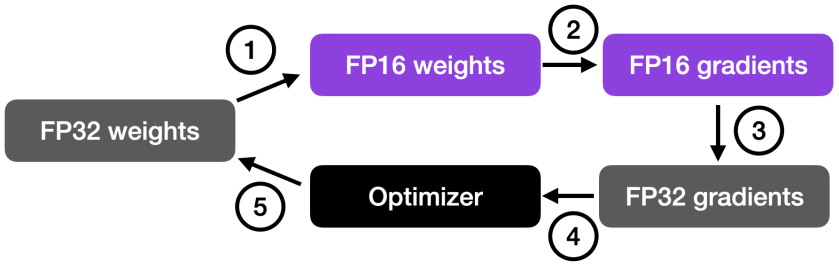
Model

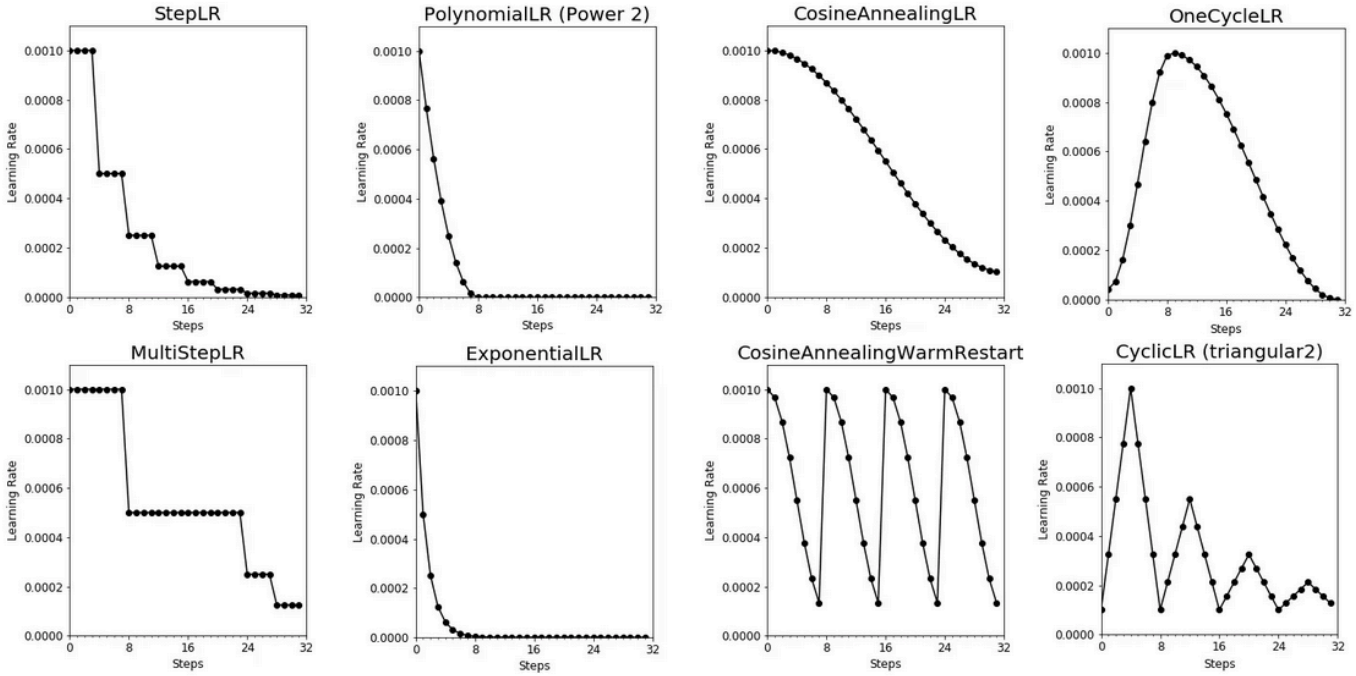
Training overhead: early stopping, mixed precision, scheduling, distributed learning, dynamic pruning, low-rank adaptation

Training Overhead

Early stopping, scheduling, mixed precision

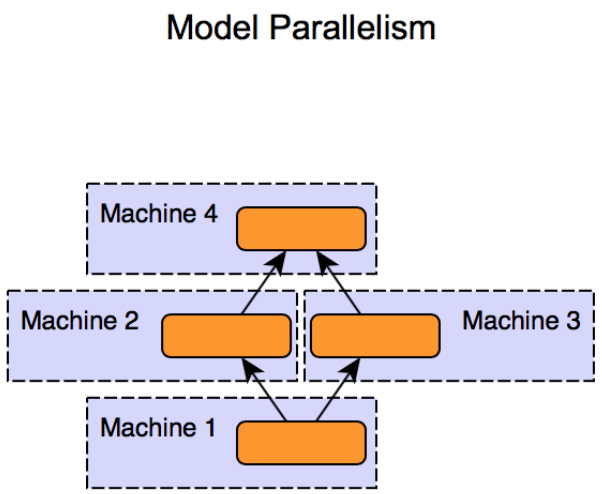


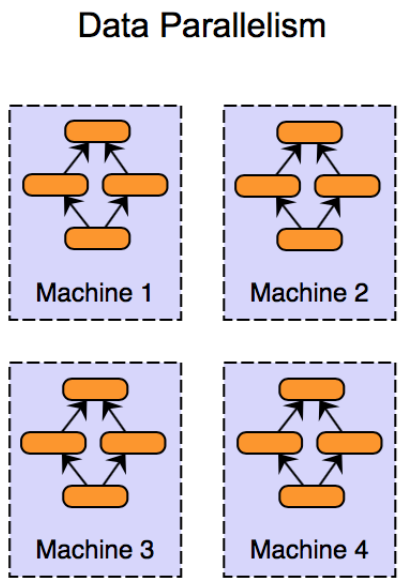


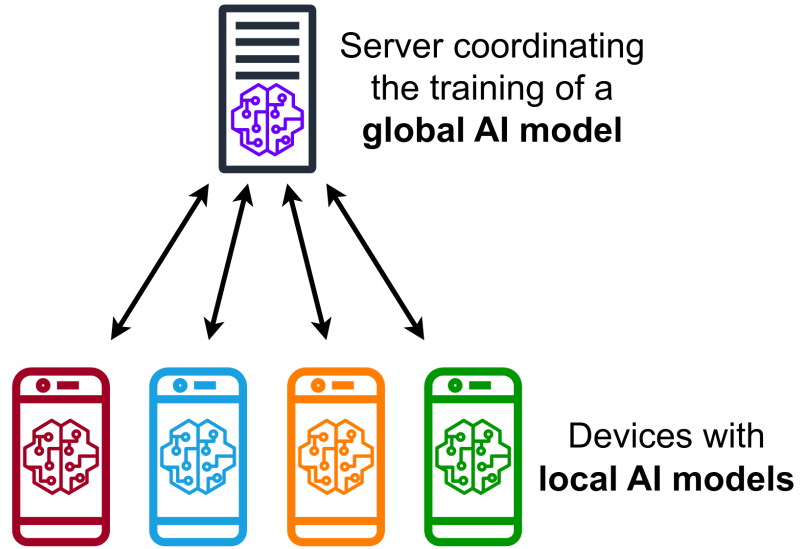


Training Overhead

Distributed learning



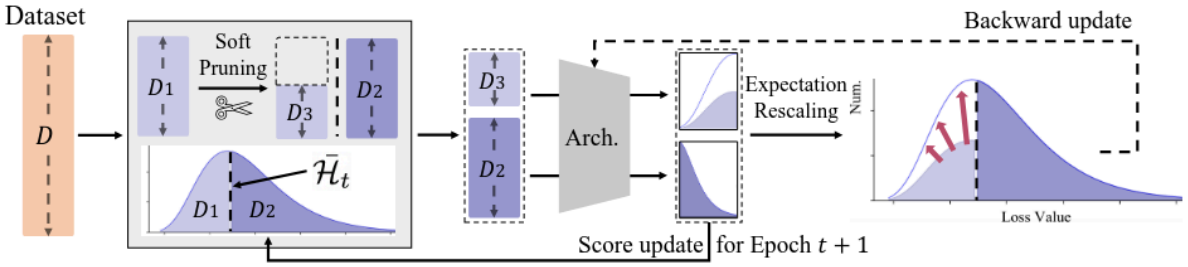


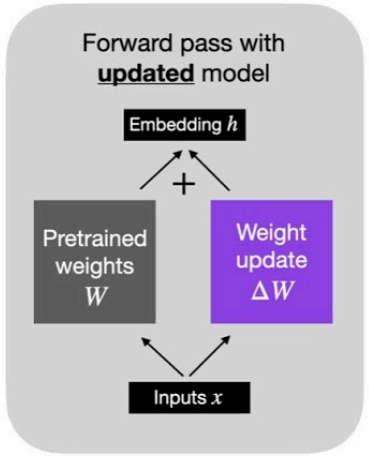


[32]

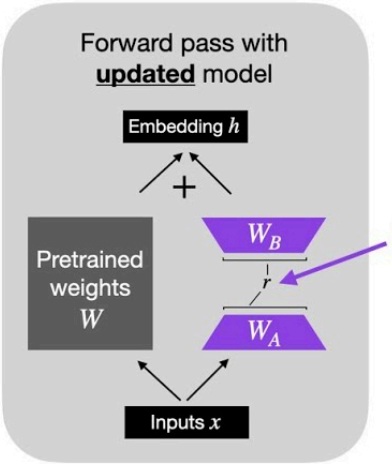
Training Overhead

Dynamic pruning, low-rank adaptation





Regular finetuning



Low-rank approximation

[33, 34]


MISCELLANEOUS

Miscellaneous

- **Tools:** data management, annotation, metadata tracker, base stack
- **Resources:** datasets, models, infrastructures
- **Hardware:** SBC, sensors, neuromorphic, deployment
- **Advances:** AI and robotics

Miscellaneous

Tools: data management, annotation, metadata tracker, base stack

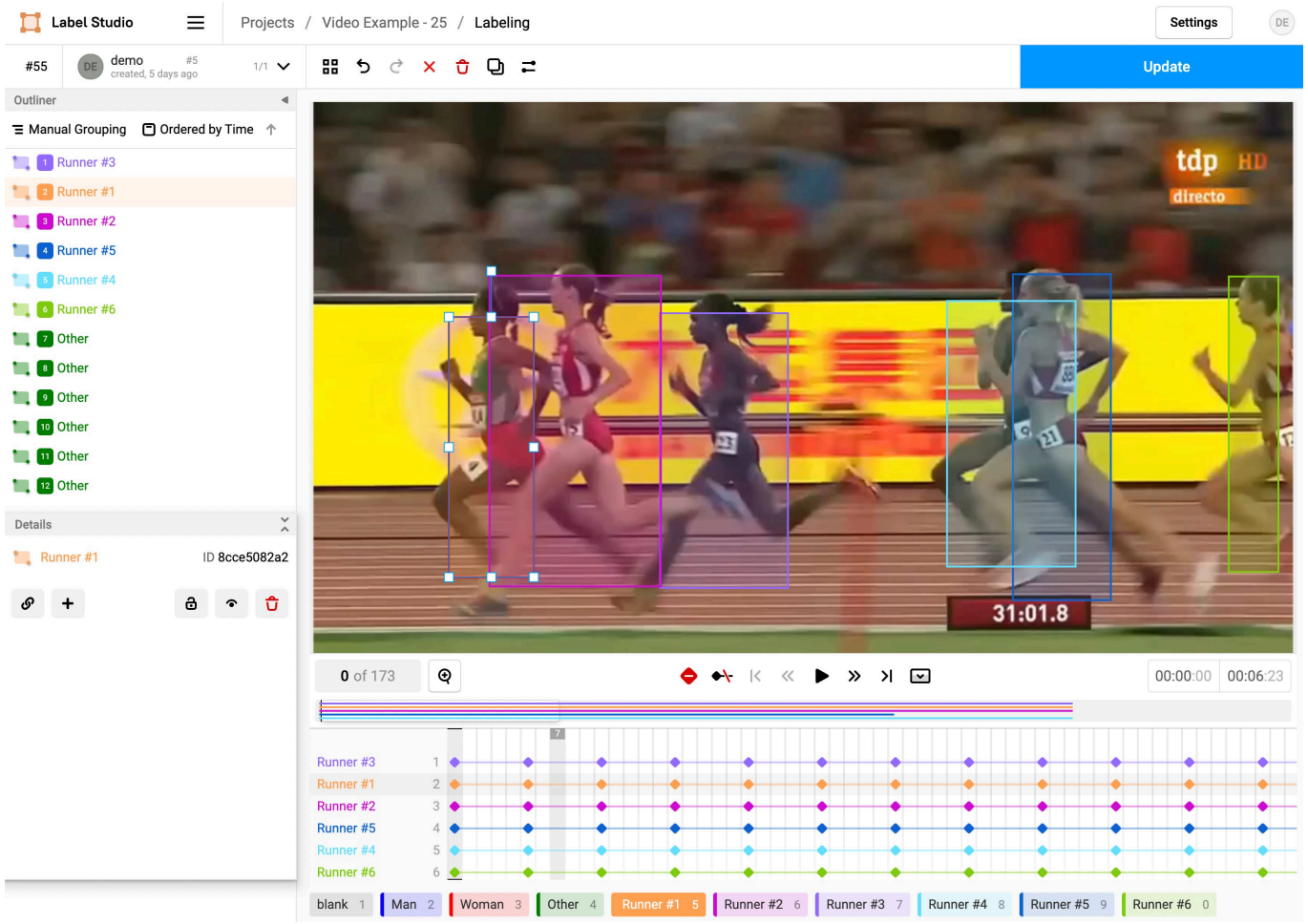
Data Management

DVC: Connect to versioned data sources and code with pipelines, track experiments, register models — all based on GitOps principles

```
dvc init
dvc remote add -d myremote /tmp/dvcstore

dvc add data/data.json
dvc push

git add data/data.json.dvc data/.gitignore
git commit -m "Add raw data"
git push

git pull
dvc pull
```

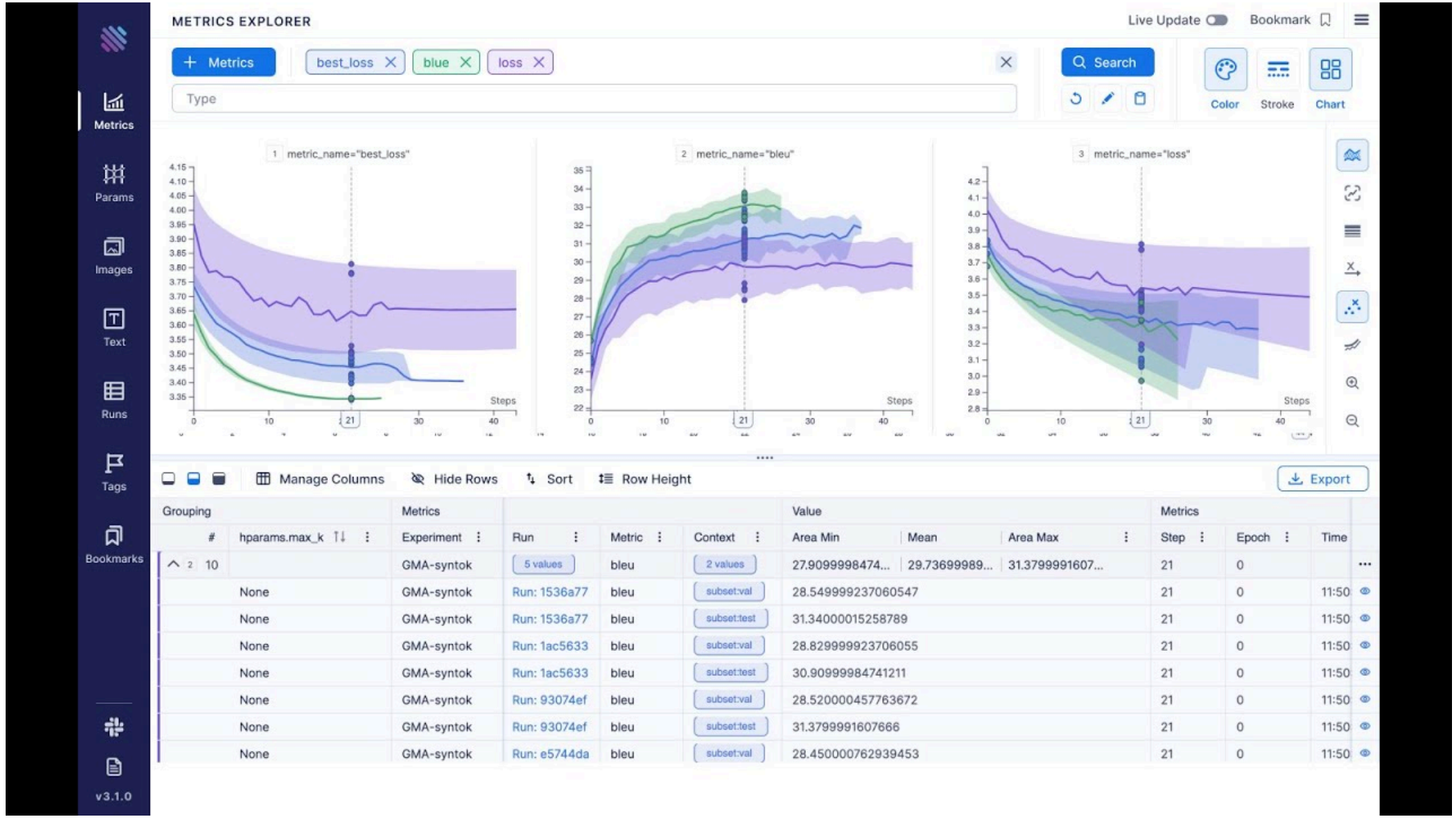
Data Annotation

Label Studio: open source data labeling tool that supports multiple projects, users, and data types



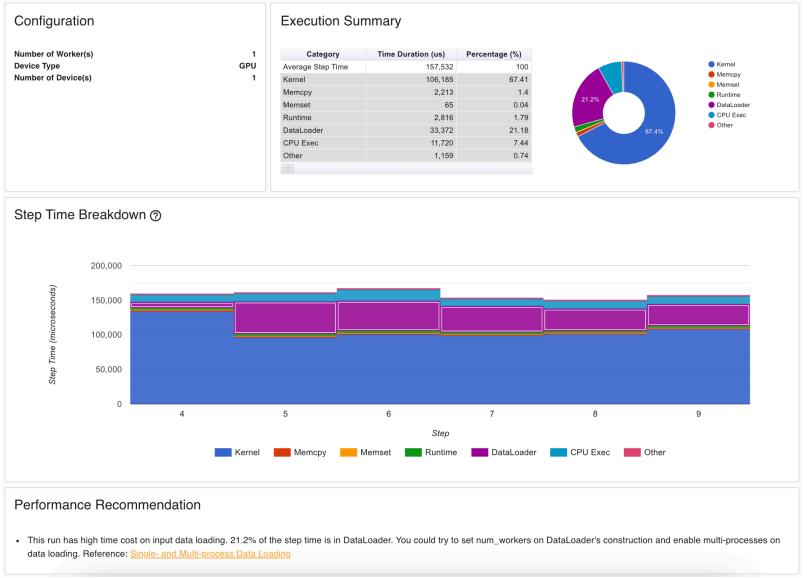
Metadata Tracker

Aimstack: Tracking and visualizing experiments



Grouping		Metrics				Value			Metrics		
#	hparams.max_k	Experiment	Run	Metric	Context	Area Min	Mean	Area Max	Step	Epoch	Time
2 10		GMA-syntok	5 values	bleu	2 values	27.9099998474...	29.73699989...	31.3799991607...	21	0	
	None	GMA-syntok	Run: 1536a77	bleu	subset:val	28.549999237060547			21	0	11:50
	None	GMA-syntok	Run: 1536a77	bleu	subset:test	31.34000015258789			21	0	11:50
	None	GMA-syntok	Run: 1ac5633	bleu	subset:val	28.829999923706055			21	0	11:50
	None	GMA-syntok	Run: 1ac5633	bleu	subset:test	30.90999984741211			21	0	11:50
	None	GMA-syntok	Run: 93074ef	bleu	subset:val	28.520000457763672			21	0	11:50
	None	GMA-syntok	Run: 93074ef	bleu	subset:test	31.3799991607666			21	0	11:50
	None	GMA-syntok	Run: e5744da	bleu	subset:val	28.450000762939453			21	0	11:50

Tooling



Base stack: pytorch, lightning, hydra (template)

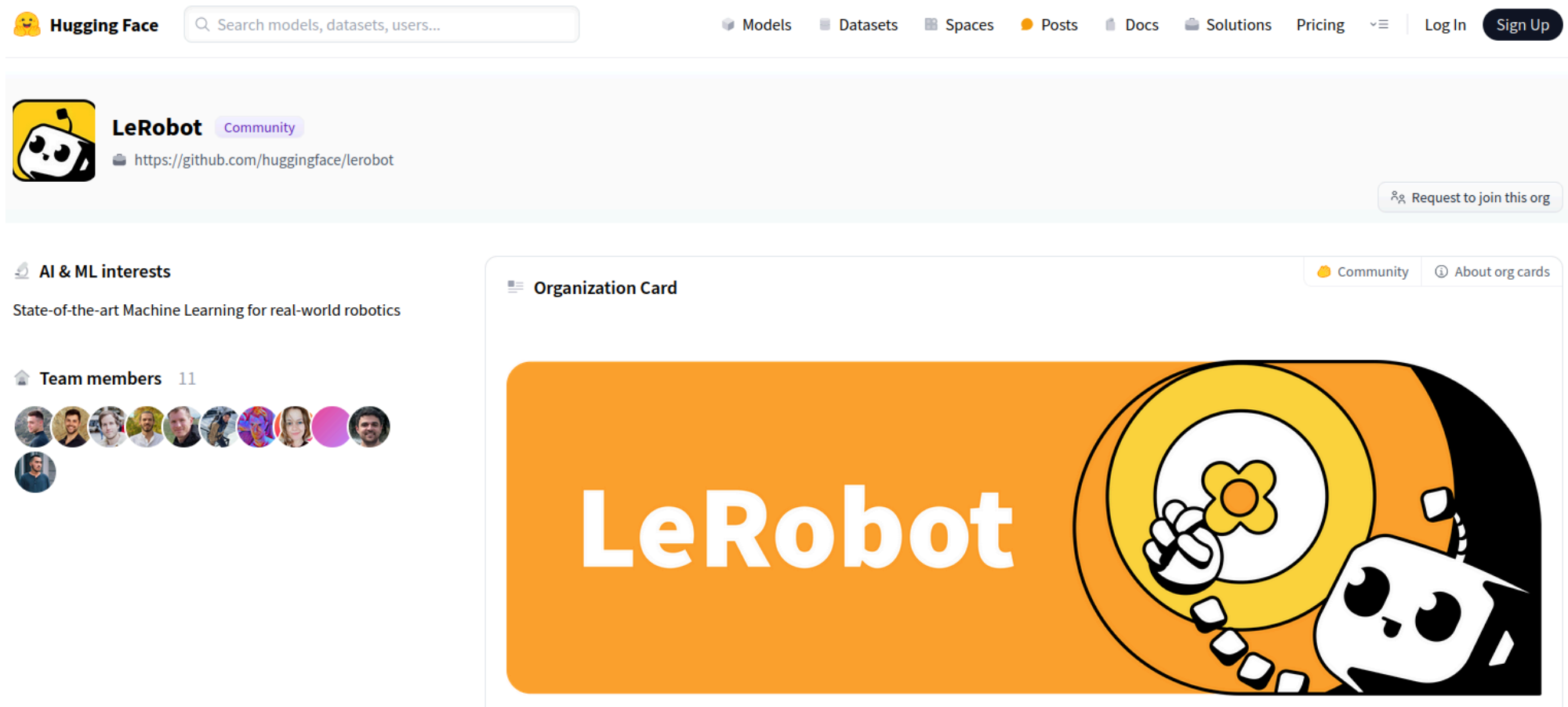
Additional packages: torchmetrics, optuna, captum, mapie, nni
grafana, prometheus, albumentations, onnx

AI Lifecycle, Troubleshooting AI

Miscellaneous

Resources: datasets, models, infrastructures

Hugging Face

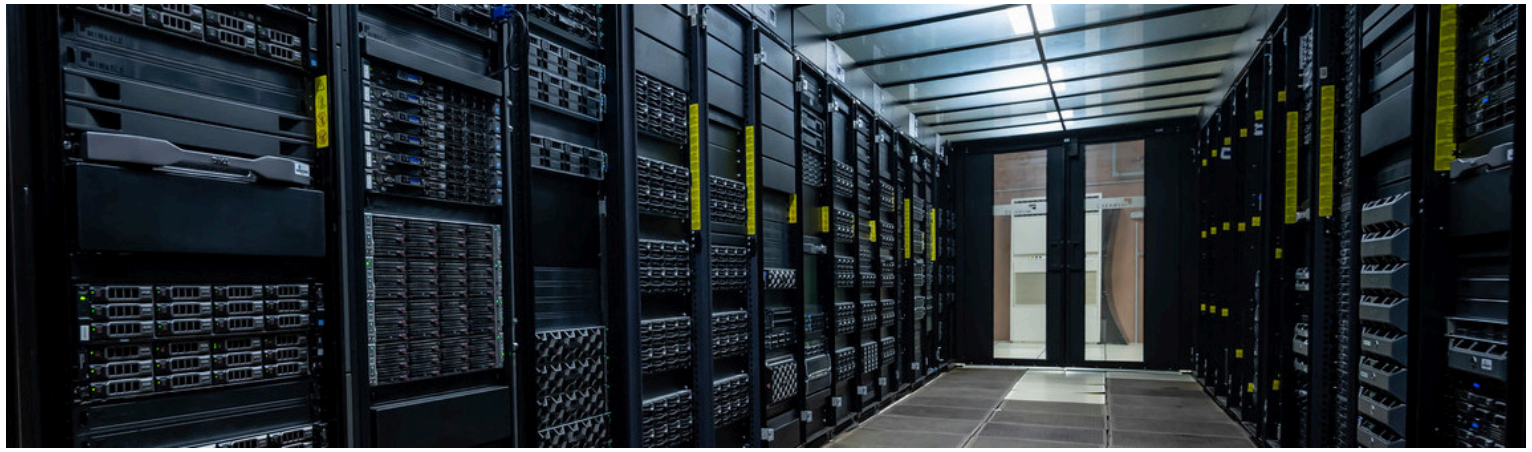


Also: [Papers with Code]

HPC/AI Resources

Hybrid Cluster (local)

- Access: Email
- Documentation



Grid'5000 (national)

- Access: Form
- Documentation

Jean Zay (national)

- Access: Form (academic and industry)
- Documentation

Usage

- Jupyter notebooks via a web interface or traditional command-line tools
- Submit jobs using SLURM, specifying required resources

Slurm

Manages a queue and launches jobs submitted on dedicated compute servers when the requested resources are available.

```
#!/bin/bash

#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --time=00:05:00
#SBATCH --job-name=GPU
#SBATCH --mem=1G
#SBATCH --gres=gpu:1

./program
```

> sbatch job.slurm

Miscellaneous

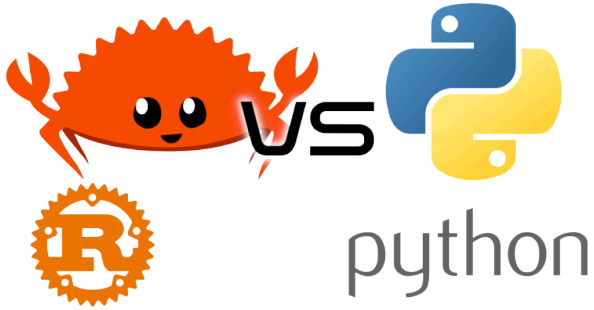
Hardware: SBC, sensors, neuromorphic, deployment

Embedded AI

- SBC: Nvidia Jetson, Raspberry Pi
- SoC: Qualcomm Snapdragon 8cx Gen 3, Google Tensor, Apple M-series
- MPU: Intel Core i7/i9/Xeon (AVX-512, DL Boost), AMD Ryzen 9 (Threadripper)
- ASIC/NPU/LPU: ARM Cortex Ethos, Apple Neural Engine, Samsung Exynos, Huawei Ascend, Intel Gaudi, Groq, cerebras
- MCU: STMicroelectronics STM32 AI, Arduino
- FPGA: AMD/Xilinx Kria, Versal AI
- Sensor: Luxonis OAK, event-based cameras
- Neuromorphic: Intel Loihi
- Co-inference

Libraries: EdgeImpulse, NanoEdgeAIStudio

Deployment



- Python overhead can seriously hurt performance
- The GIL is a notorious source of headaches
- Make serverless inference possible with lightweight binaries
- Remove Python from production workloads

Why Rust: safety and speed, borrow checker, scoped resource management, error handling, wasm, tooling, ...

Libraries: Candle, Luminal

Deployment

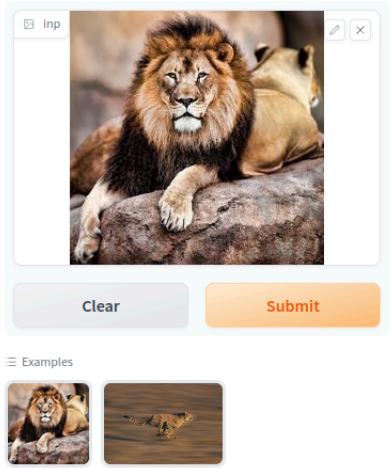
```python
import torch
import torchvision

model = torchvision.models.resnet18()
traced_script_module = torch.jit.trace(model, torch.rand(1, 3, 224, 224))
traced_script_module.save("traced_resnet_model.pt")
```

```cpp
#include <torch/script.h>

#include <iostream>
#include <memory>

torch::jit::script::Module module;
module = torch::jit::load(argv[1]);

std::vector<torch::jit::IValue> inputs;
inputs.push_back(torch::ones({1, 3, 224, 224}));

at::Tensor output = module.forward(inputs).toTensor();
std::cout << output.slice(/*dim=*/1, /*start=*/0, /*end=*/5) << '\n';
```

CPP export, Pytorch Edge

Deployment

Gradio: easily create interactive web-based user interfaces for ML models




```
def predict(inp):
        pass

gr.Interface(fn=predict, inputs=gr.Image(type="pil"), outputs=gr.Label(num_top_classes=3),
                examples=["lion.jpg", "cheetah.jpg"]).launch()
```

Also: FastAPI, TorchServe

Miscellaneous

Advances: AI and robotics

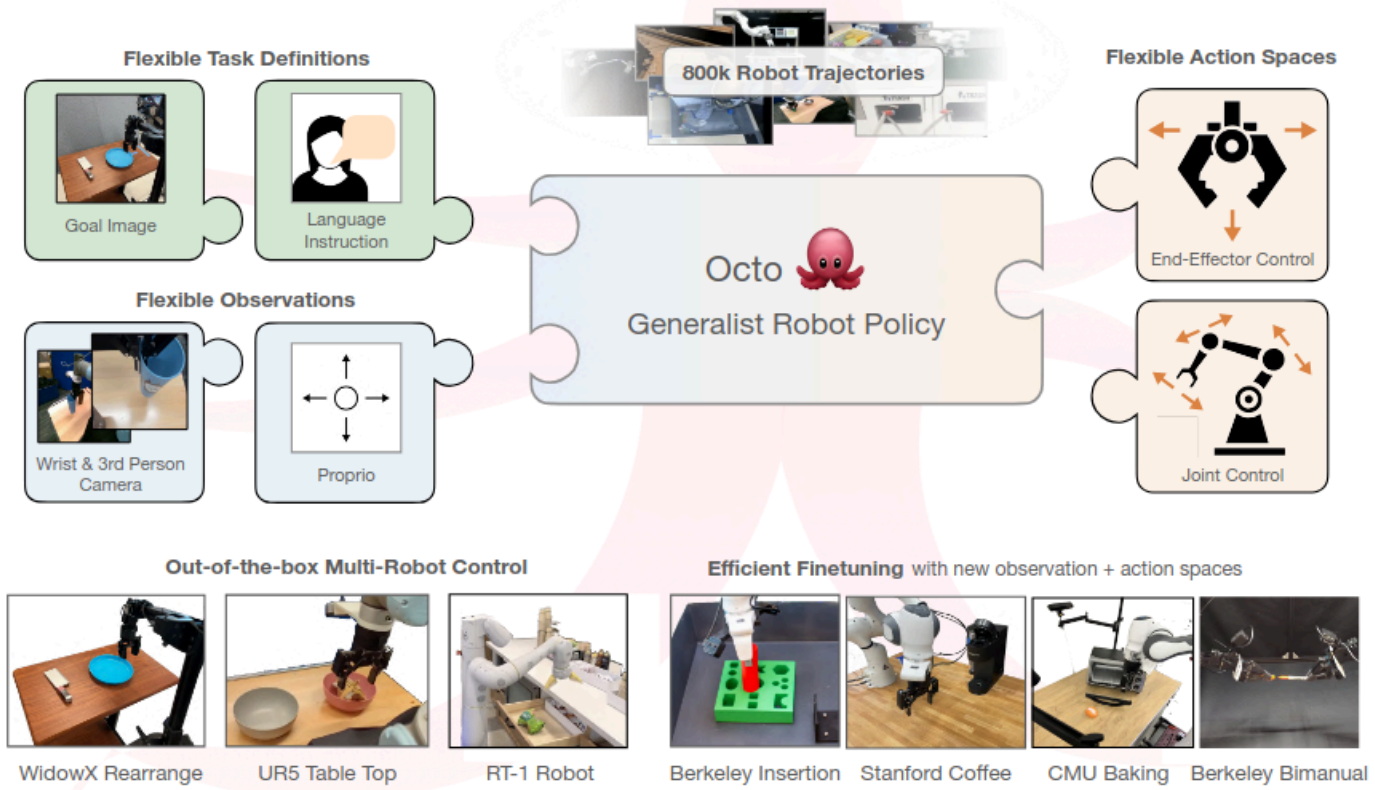
GRUtopia

Dream General Robots in a City at Scale

[35]

Octo

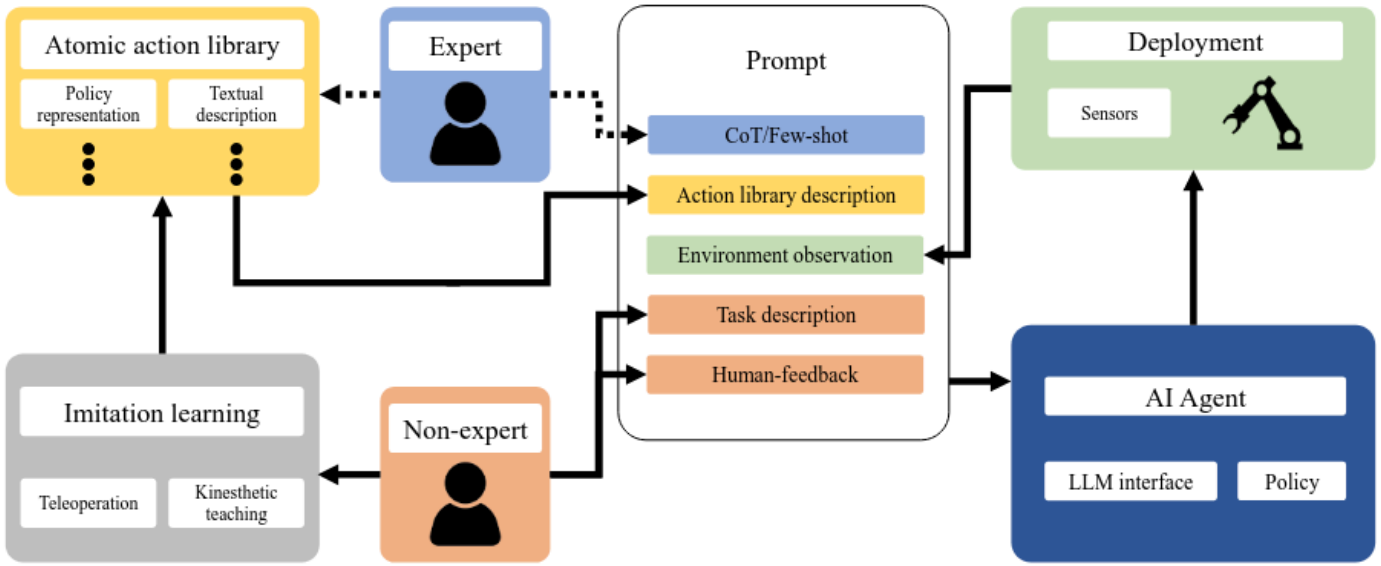
An Open-Source Generalist Robot Policy



[36]

ROS-LLM

A ROS framework for embodied AI with task feedback and structured reasoning



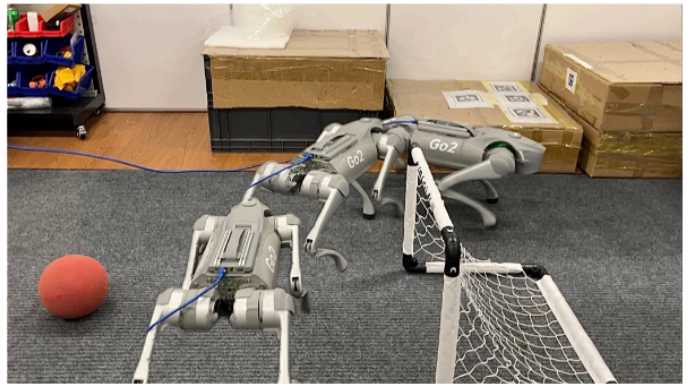
[37]

Make An Agent

Generalizable Policy Network Generator with Behavior-Prompted Diffusion

Making agile turns to avoid stepping on a bouquet while moving across a mat

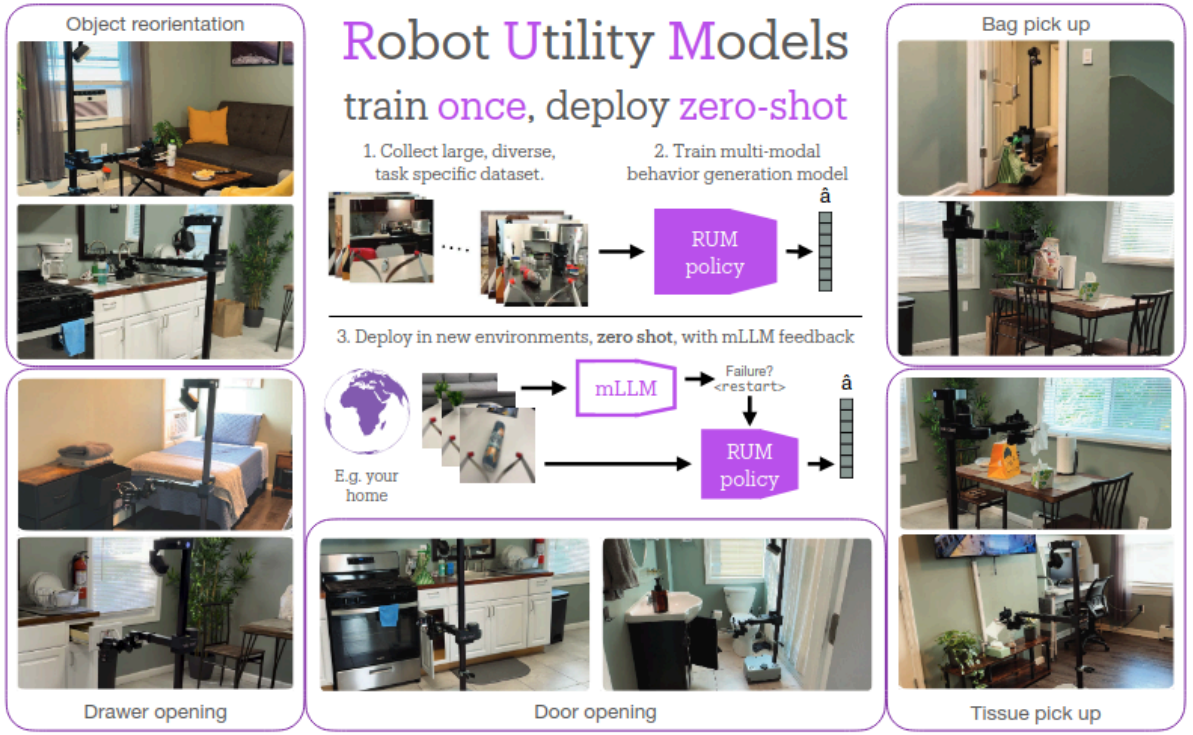
Navigating to circumvent the goal and ball while swiftly moving backward.



[38]

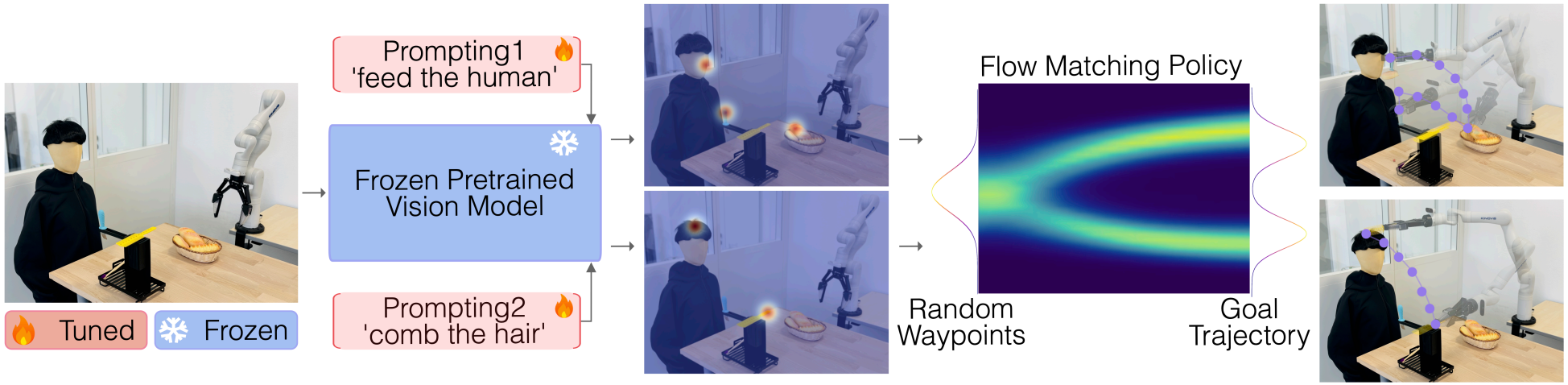
Robot Utility Models

General Policies for Zero-Shot Deployment in New Environments



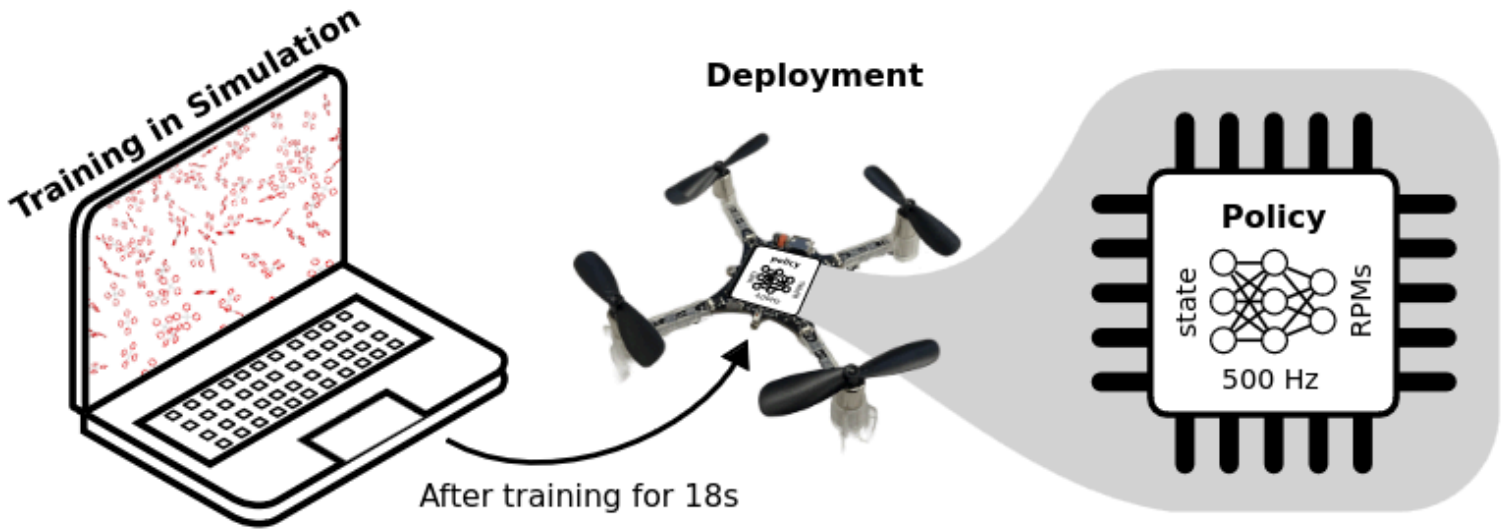
Flow Matching Policy

Affordance-based Robot Manipulation with Flow Matching



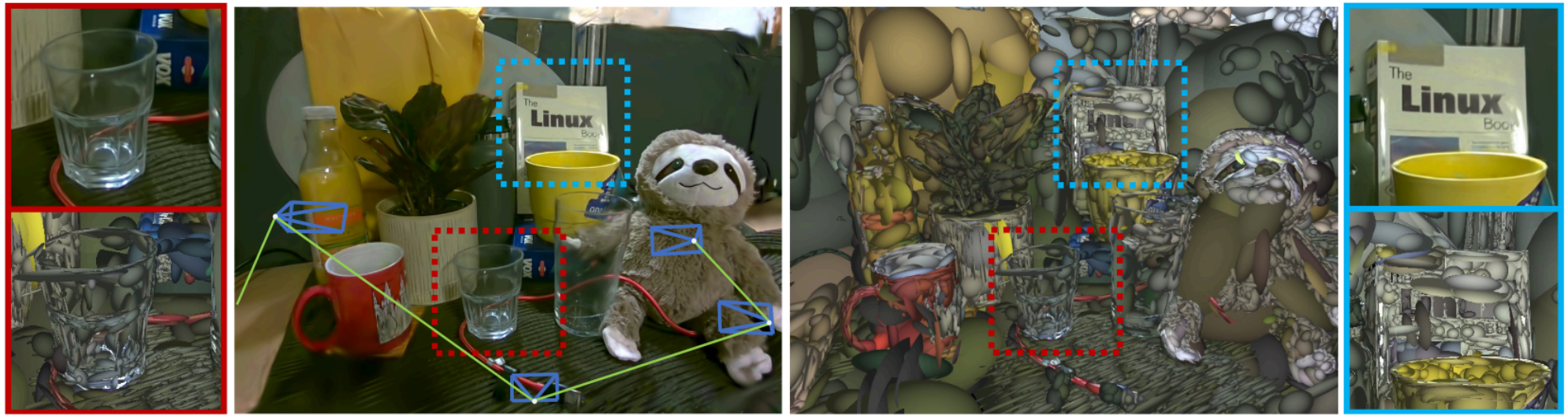
[40]

Learning to Fly in Seconds



[41]

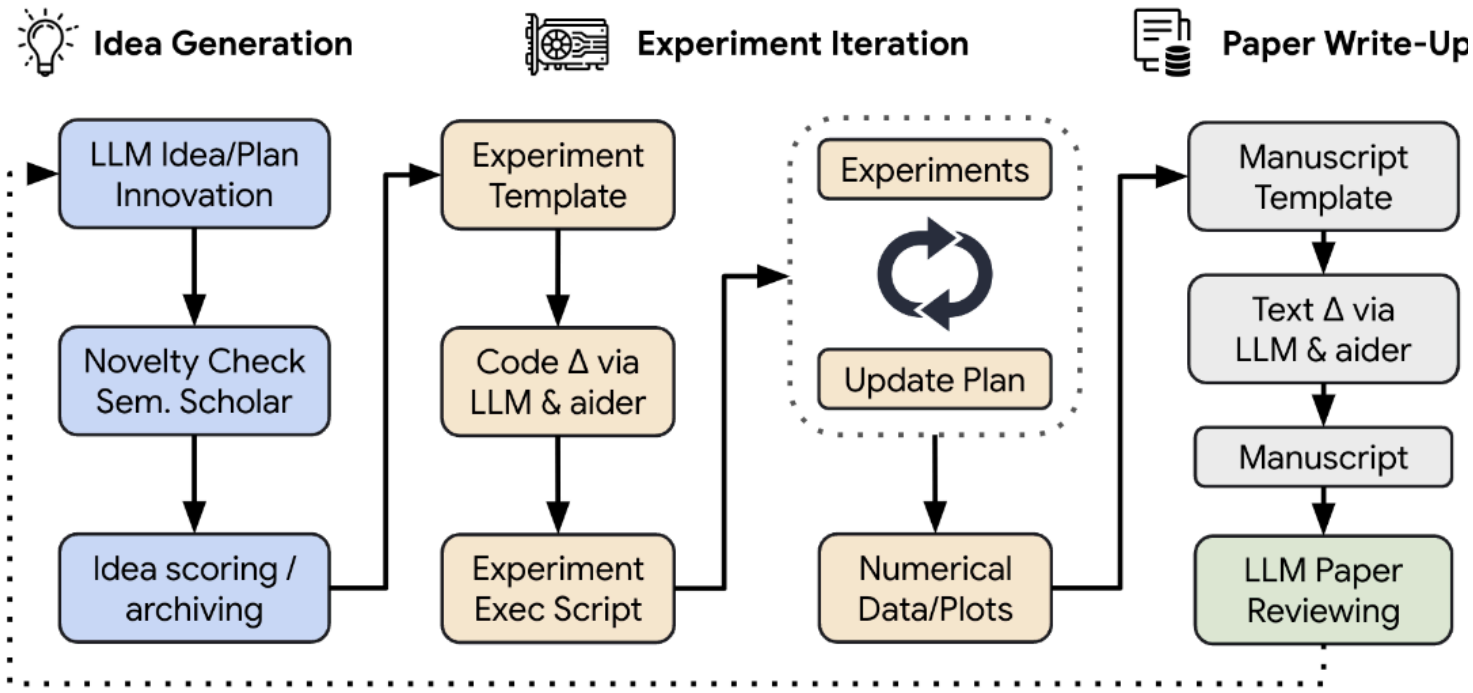
Gaussian Splatting SLAM



[42]

The AI Scientist

Towards Fully Automated Open-Ended Scientific Discovery



[43]

Conclusion

- Open source AI with HuggingFace
- Physics-informed AI and Multimodal Foundation Model
- Catastrophic forgetting, benchmarking
- TinyML Foundation

Responsible and Sustainable AI

FIDLE


Bases, Concepts et Enjeux
L'IA comme un outil,
Acteur de l'IA
1
History and Fundamental Concepts
2
New !
Data, models and representation's hell
Data and models
3
New !
Demonstration Illustration
LLM / Text to Image
4
AI, Law, Society and Ethics
5
Mathematics, gradients everywhere !
6
New !
Learning methodology
7
Convolutional models
CNN
8
Sparse (text) and sequences data
Embedding, RNN
9
«Attention is All You Need»
Transformers
10
Graph Neural Network
GNN
11
Autoencoder networks
AE
12
Variational Antoencoder
VAE
13
Generative Adversarial Networks
GAN
14
Diffusion Model
Text to image
15
Deep Reinforcement Learning
RL
16
Physics-Informed Neural Networks
PINNS
17
Learning faster and cheaper
Eco-Friendly
18
Jean-Zay
GPU acceleration
24 mai
the Event !
JDLS 2024
Deep Learning for Science
19
New !
New models
VLM, SM, Multimodal, ...
20
New !
Dreams come True !
Inference & Production

Any questions?