Uncertainty in Neural Networks, Out of Distribution Detection

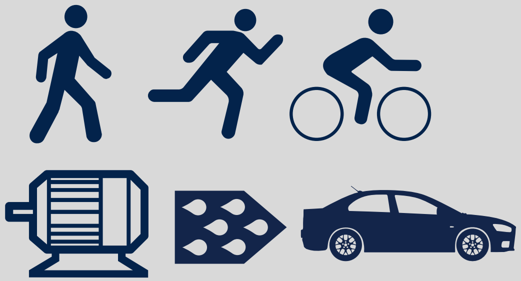
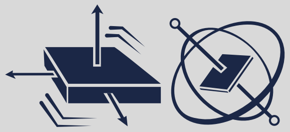
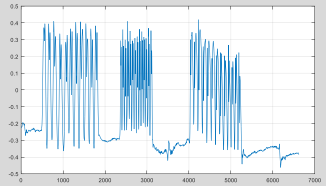
Angelo Bosco
STMicroelectronics - Artificial Intelligence Software & Tools Group
Seminar at Catania University - June 13, 2024

life.augmented

Introduction:
Anomaly Detection

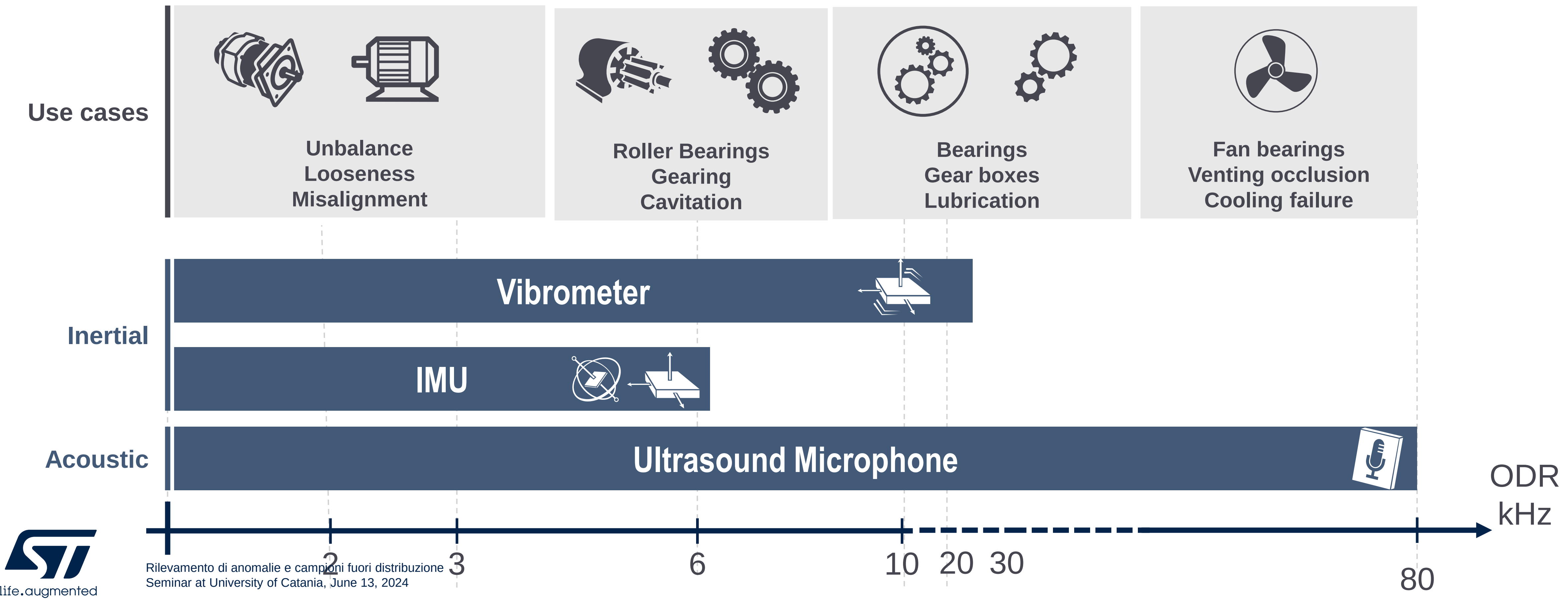
life.augmented

Dynamical Systems

Dynamical System	A system that evolves over time. It consists of: **State space**: which represents all the possible states of the system. State vectors can be high dimensional. **Rule:** that describes how the system's state changes with time.	
Time Scales of a DS	Rate at which significant changes occur within the system	…, minutes, seconds, ms,…
Fast Dynamics	Changes happen over short time intervals. In a mechanical system: vibrations or rapid oscillations.	 6 axes IMU (Inertial Measurement Unit) 
Slow Dynamics	Changes over long time intervals: gradual wear and tear of mechanical components.	Choose appropriate acquisition settings

Choosing the appropriate sensor

- The appropriate (set of) sensor(s) must be chosen depending on the equipment being monitored, acquisition parameters depend on the system being monitored.



Accelerometer Settings

Accelerometer LSM6DSM (for smart phones with OIS / EIS and AR/VR systems)

Key Settings ODR (Output Data Rate); Full Scale Range; Power Mode (Low, Normal, High Perf.)

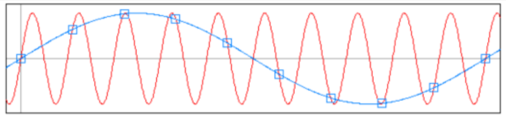
ODR

Table 42. FIFO ODR selection

ODR_FIFO_[3:0]	Configuration[1]
0000	FIFO disabled
0001	FIFO ODR is set to 12.5 Hz
0010	FIFO ODR is set to 26 Hz
0011	FIFO ODR is set to 52 Hz
0100	FIFO ODR is set to 104 Hz
0101	FIFO ODR is set to 208 Hz
0110	FIFO ODR is set to 416 Hz
0111	FIFO ODR is set to 833 Hz
1000	FIFO ODR is set to 1.66 kHz
1001	FIFO ODR is set to 3.33 kHz
1010	FIFO ODR is set to 6.66 kHz

1. If the device is working at an ODR slower than the one selected, FIFO ODR is limited to that ODR value. Moreover, these bits are effective if both the DATA_VALID_SEL FIFO bit of *MASTER_CONFIG (1Ah)* and the TIMER_PEDO_FIFO_DRDY bit of *FIFO_CTRL2 (07h)* are set to 0.

- ODR is the Sampling Rate of the accelerometer
- A digital signal must be sampled at least twice as its original bandwith (Nyquist Theorem).
- **In industrial applications 2.56x is often used.**



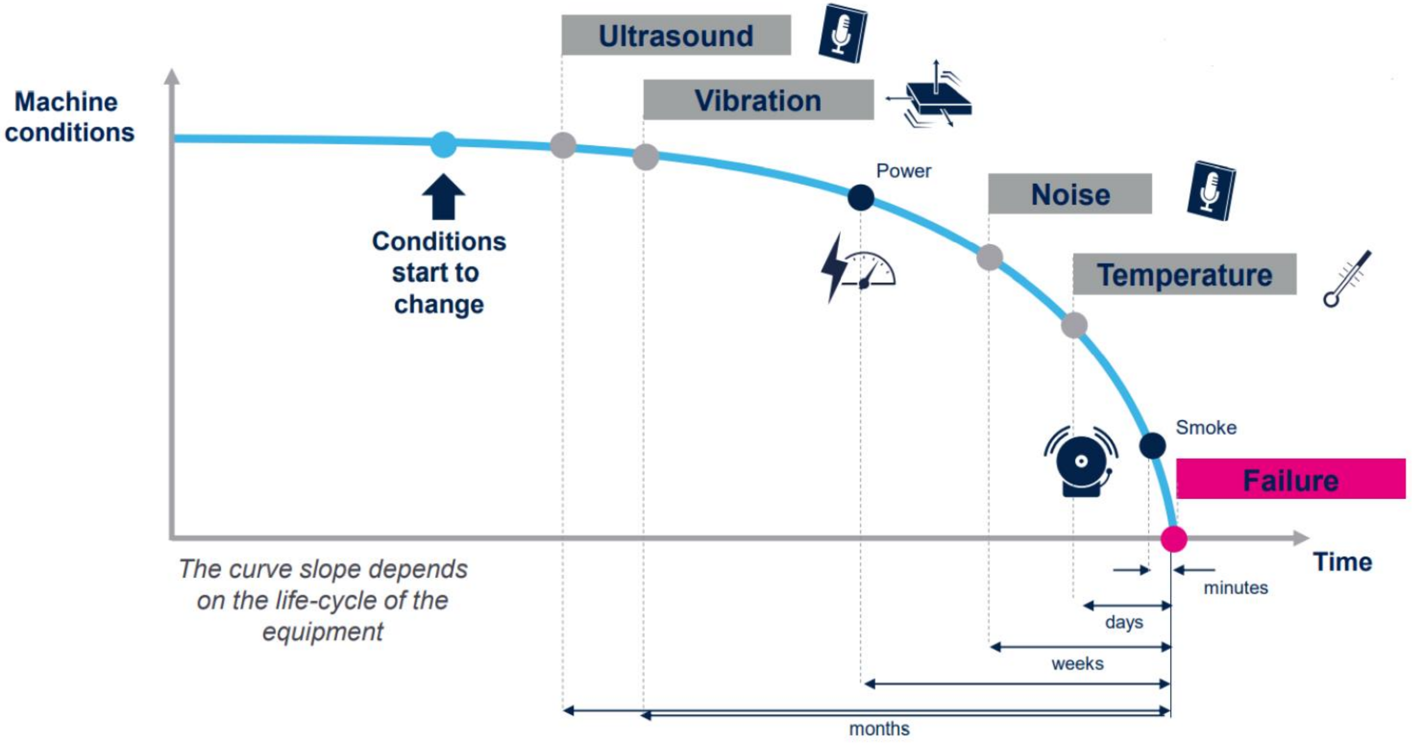
Full Scale Range (FS)

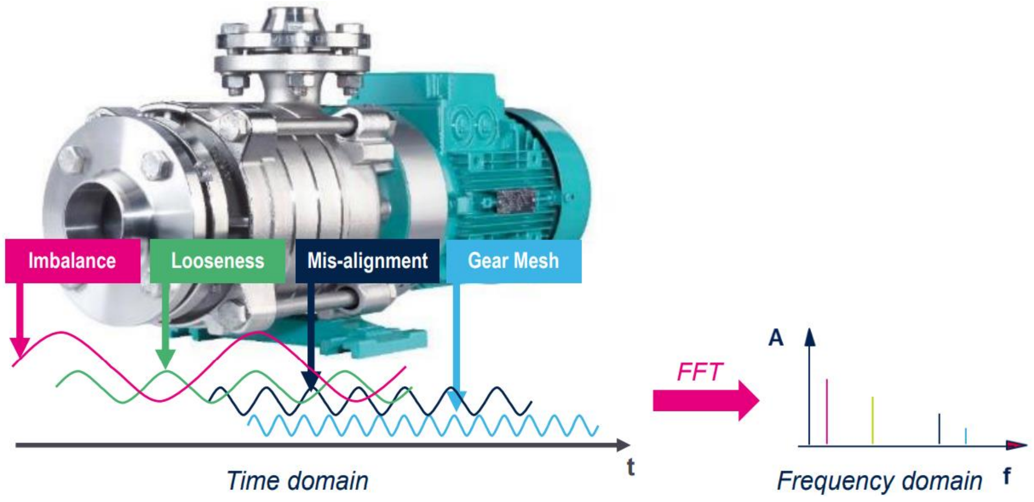
LA_So	Linear acceleration sensitivity[2]	FS = ±2		0.061		mg/LSB
		FS = ±4		0.122		
		FS = ±8		0.244		
		FS = ±16		0.488		
G_So	Angular rate sensitivity[2]	FS = ±125		4.375		mdps/LSB
		FS = ±250		8.75		
		FS = ±500		17.50		
		FS = ±1000		35		
		FS = ±2000		70		

- Maximum range of acceleration it can measure in any given axis
- expressed in units of 'g'

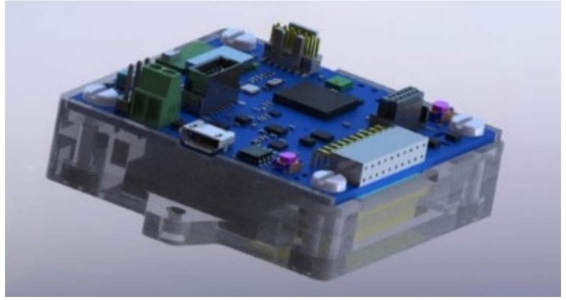
Increasing FS lowers the sensitivity; reduces signal clipping in case of strong accelerations

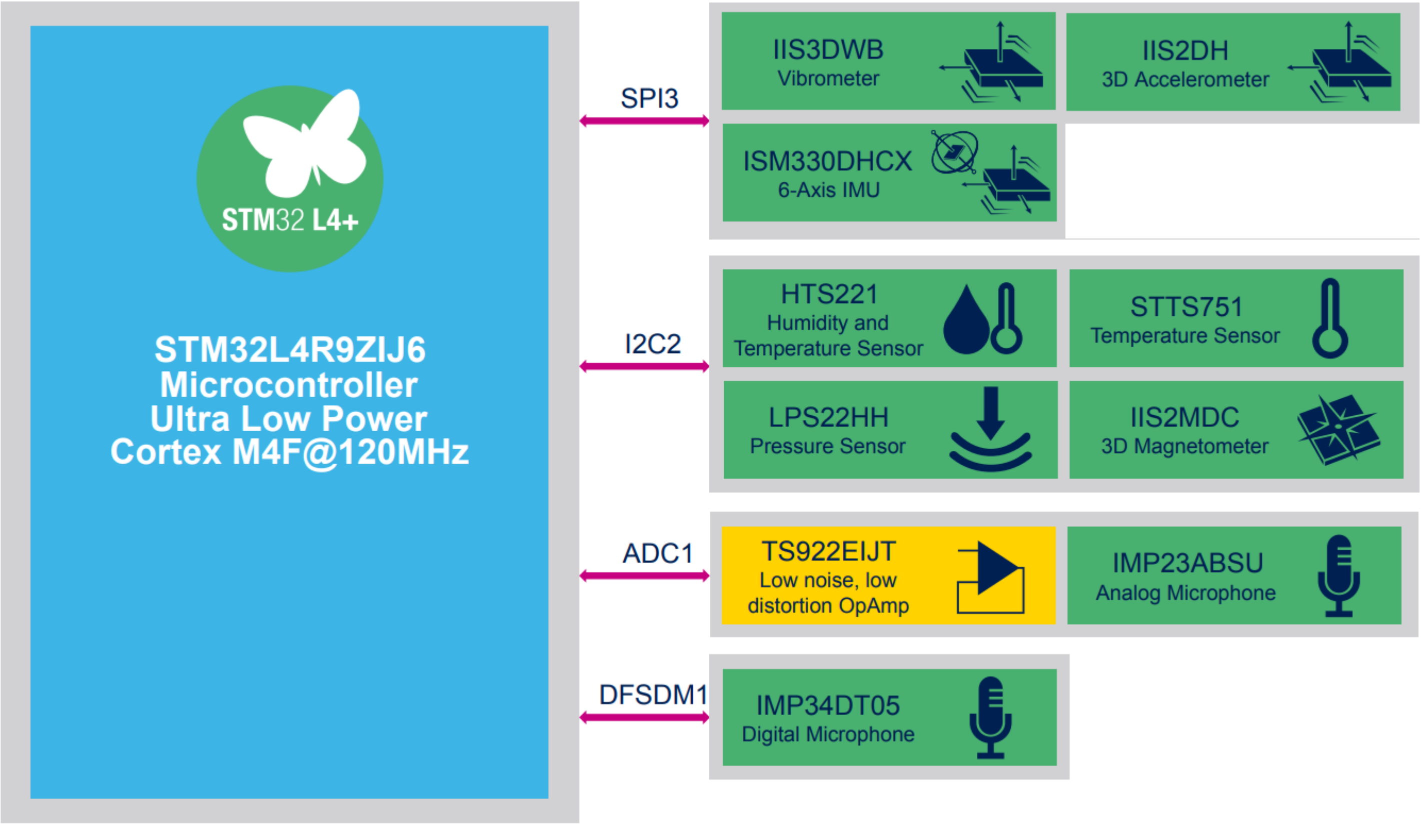
Machine Condition and Sensing





SensorTile Wireless Industrial Node

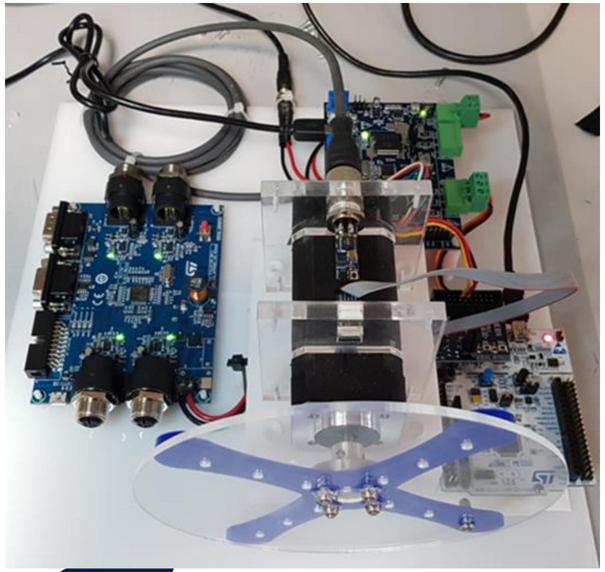


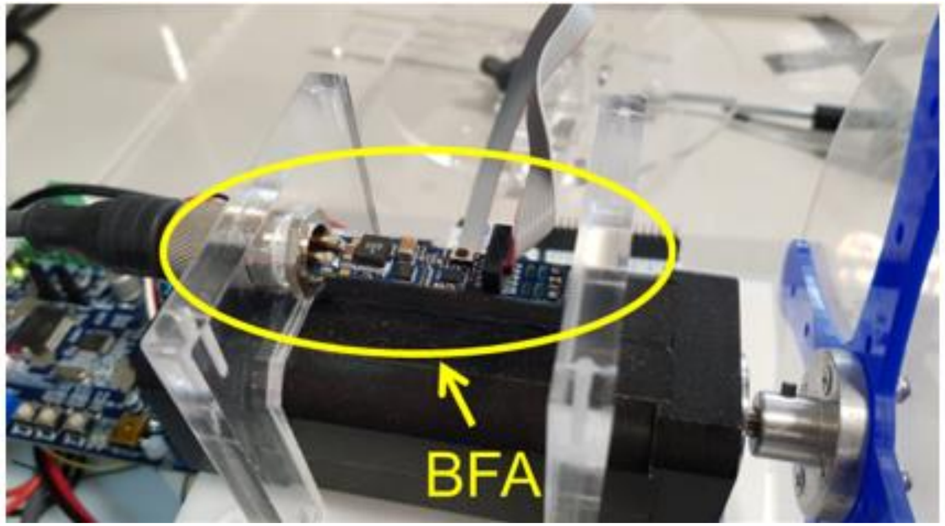


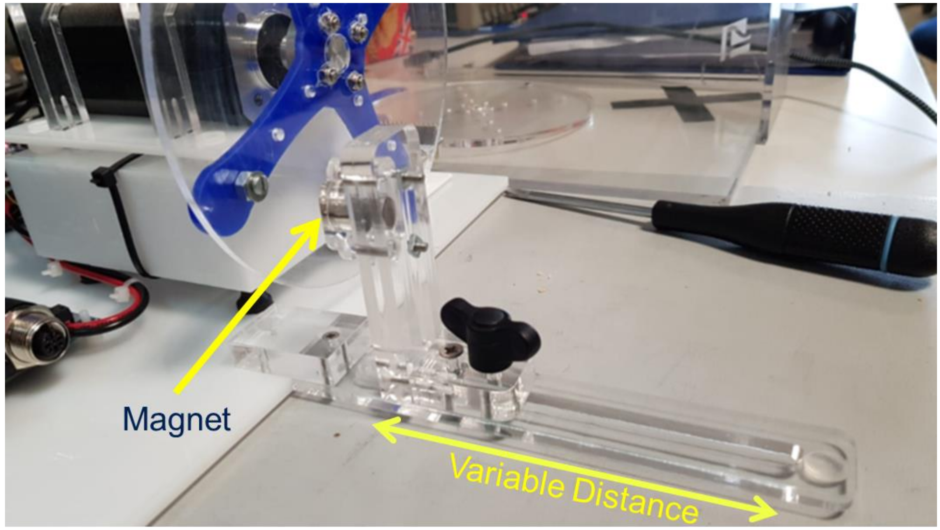
[STEVAL-STWINKT1B - STWIN SensorTile Wireless Industrial Node development kit and reference design for industrial IoT applications - STMicroelectronics]

Brief description of the use case: testbench

- **Goal of the demo**: detect anomalies on a motor by processing vibration data on an embedded device.
- **Injected Anomalies**:
  - Unbalancing (by inserting screws in the disc)
  - Misalignment (by using a magnet that can be positioned close to the disc)
- **Testing Conditions and Acquisition Parameters**:
  - 1800, 2160, 2520, 2880, 3240, 3600, 3960 RPMs (i.e. from 30Hz to 66Hz, step 6Hz)
  - Accelerometer ODR=1,6KHz ; high pass filter to remove DC
  - FFT 1024 points, overlap 75%, FFT averaging (9 averages per signature, Tacq = 1400ms)



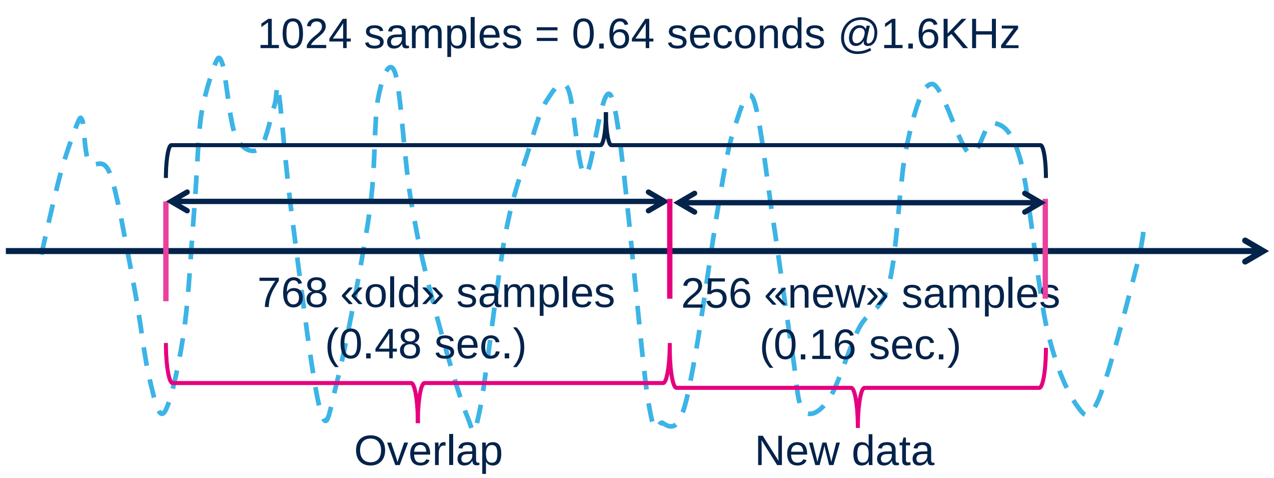




[STEVAL-BFA001V2B - Multi-sensor predictive maintenance kit with IO-Link stack v.1.1 - STMicroelectronics]

Overlapping Time Windows

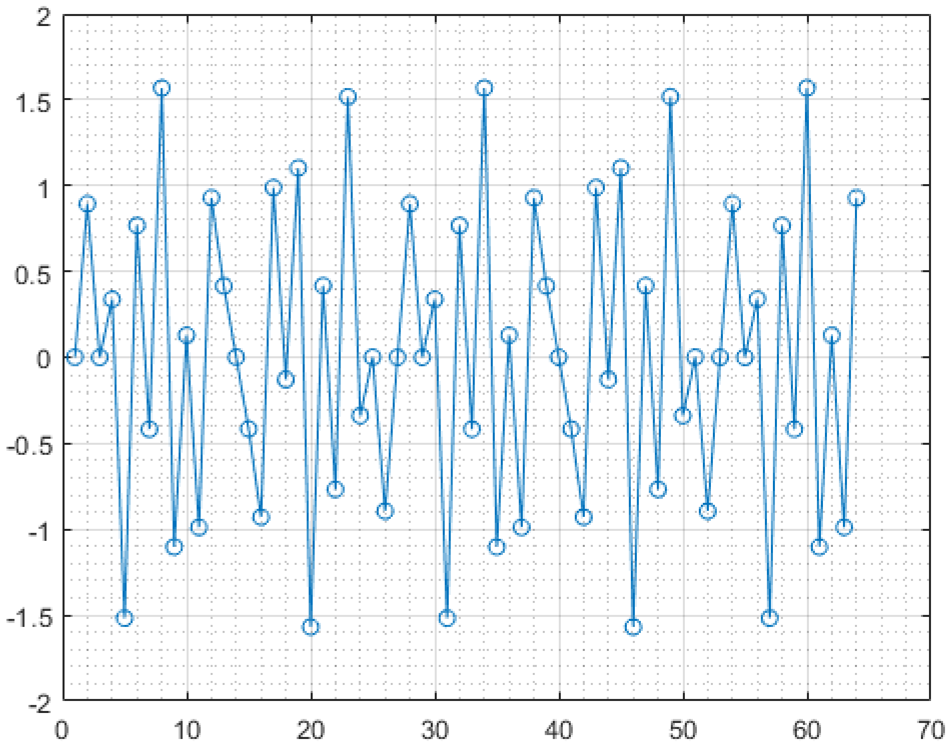
- A 3-axes accelerometer attached to a dynamical system records the evolution of motion over time and generates three 1D time series.
- One to 3 processing time windows with 1024 samples each and with overlapping.
- Length is power of 2 to allow efficient FFT.



To further reduce noise, N noise signatures are averaged, e.g. N=8

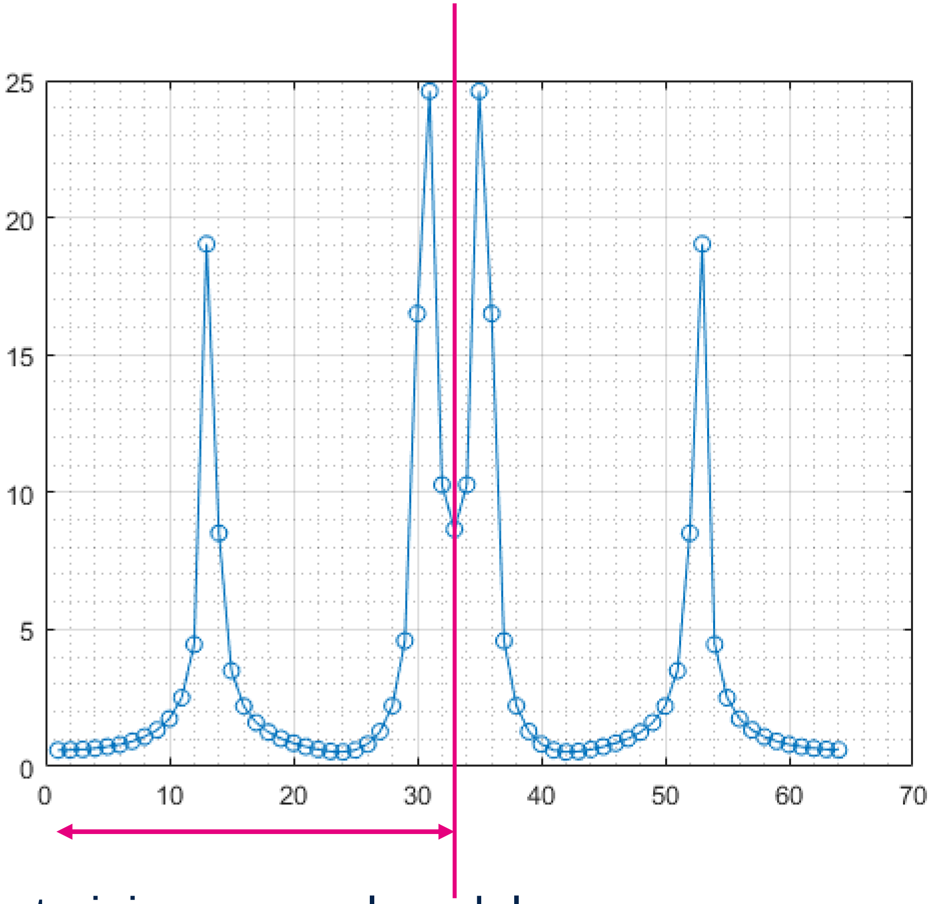
Short Time Fourier Transform

- We want to discover the frequencies of the signals that compose our time domain signal S:



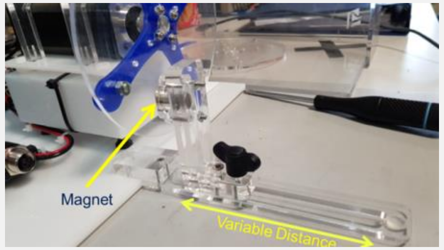
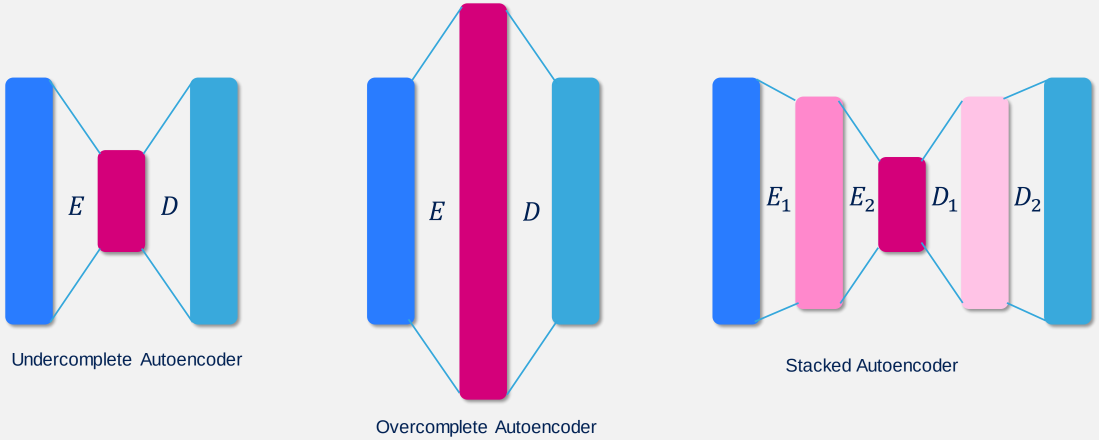
STFT

STFT: Short Time Fourier Transform



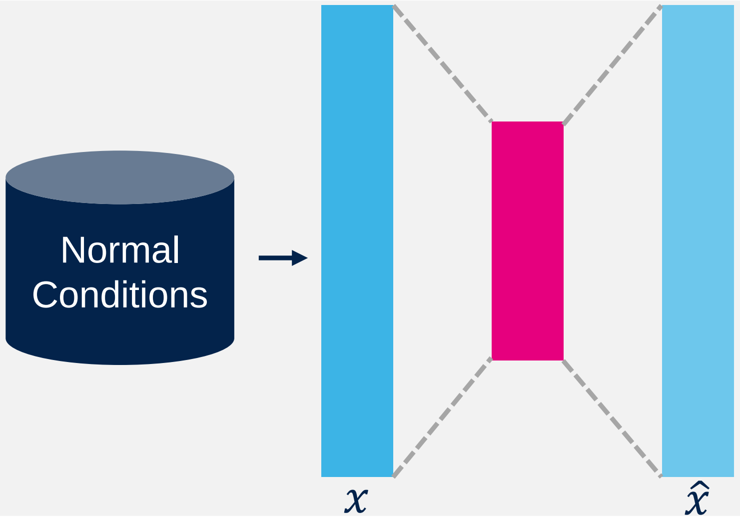
FFT is symmetric, we only use the first half when training a neural model

Autoencoders for Anomaly Detection

Final Dataset	STFT 512-dimensional vectors for each of the 7 speeds
Anomaly Detection	Detect abnormal vibrations w.r.t. the known speeds. 
Autoencoders for Anomaly Detection	 • Autoencoders reconstruct their inputs. • An autoencoder with *normal* data, (*should*) fail to reconstruct anomalous data because it has not seen it during the training phase.

Autoencoders for Anomaly Detection

Autoencoder



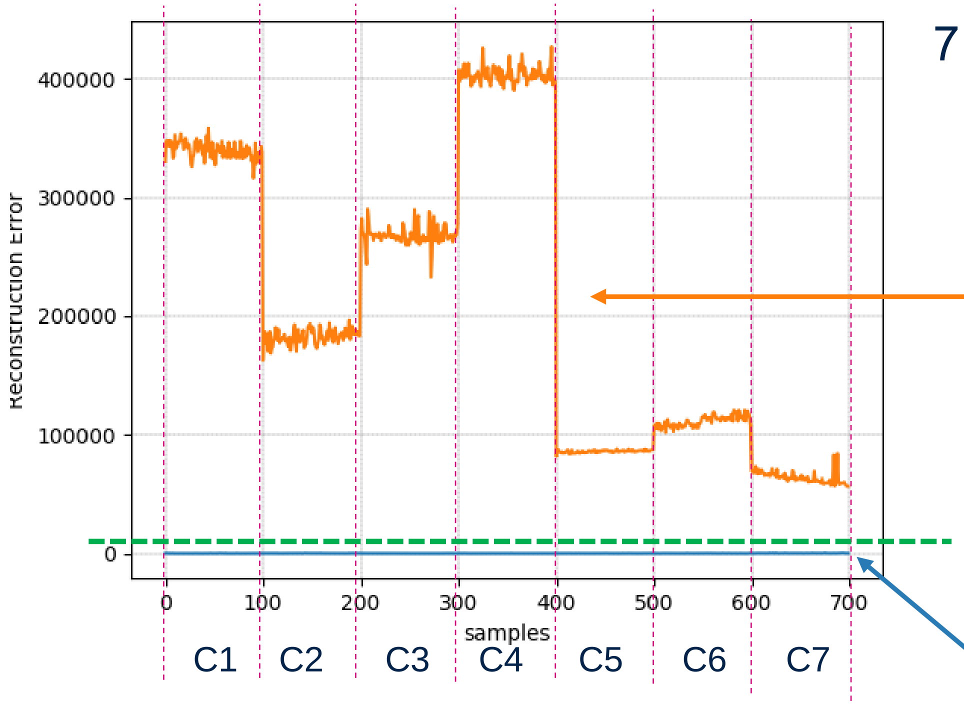
$$Error = \|x - \hat{x}\|$$

AE for feature extraction:
Only the Encoder is necessary

AE for anomaly detection:
Encoder and Decoder are necessary

- **Train the Autoencoder on normal data only**. No need to acquire anomalous data.
- Compute the difference between the decoded vector $\hat{x}$ and the original input x according to a chosen metric
- If the reconstruction error E is «high» then:
  - The autoencoder received anomalous data that it is not able to reconstruct → Raise Anomaly Flag

Anomaly Detection using Autoencoder Reconstrucion Error



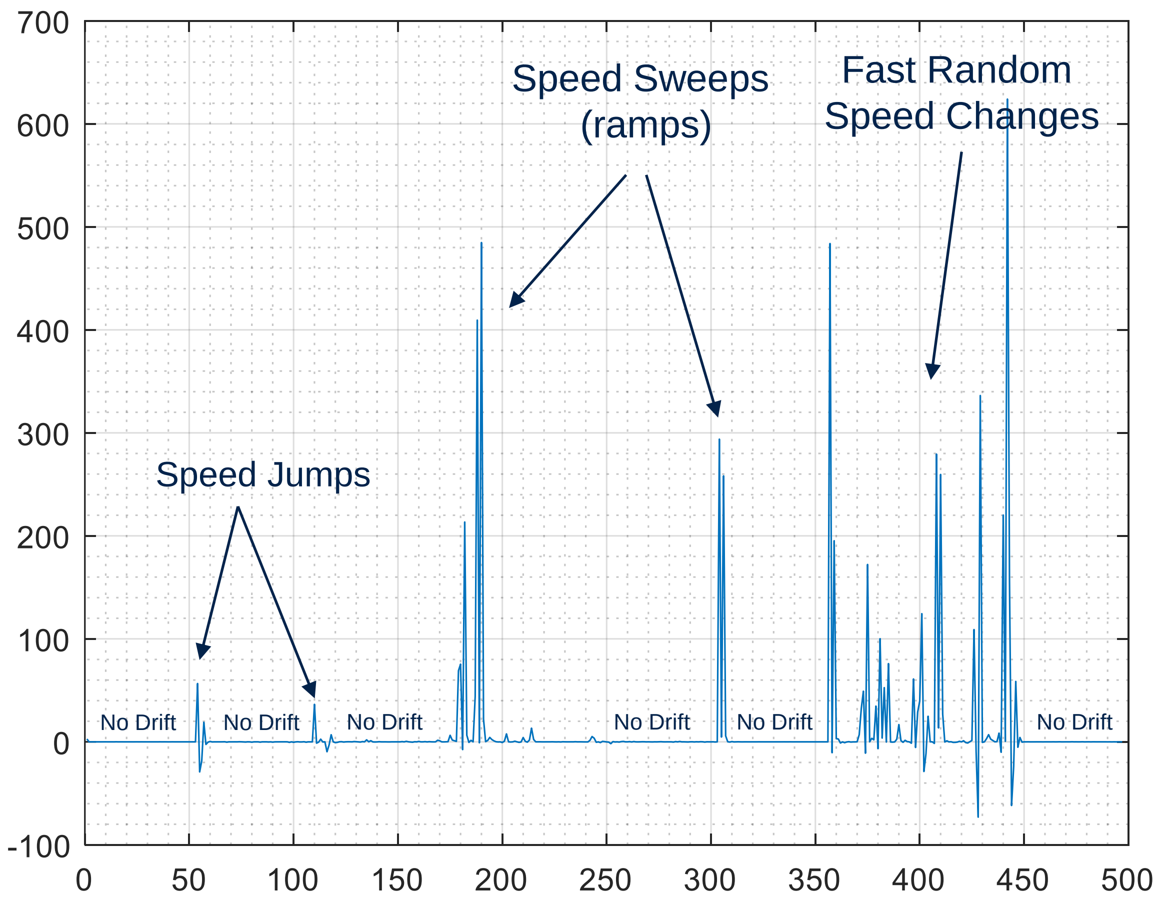
7 motor speeds, C1,…C7, ranging from 1800RPM to 3960RPM

Reconstruction Error of the autoencoder when input data contain anomalies (disc unbalancing).

Possible threshold for detecting anomalies (in this example there is a wide gap between normal and anomalous conditions).

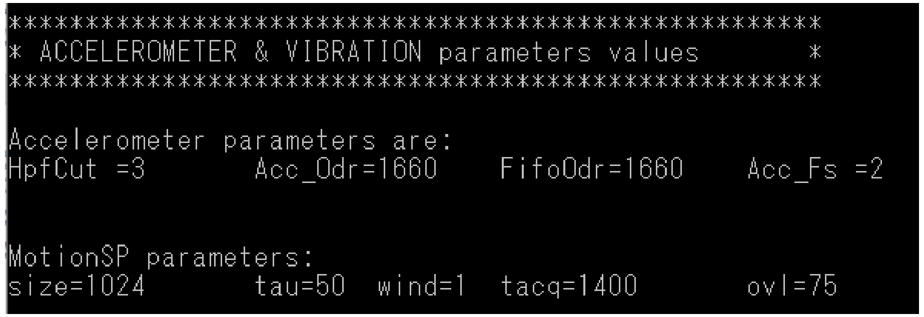
Reconstruction Error of the autoencoder when input data is normal (no disc unbalancing).

Drift Detector

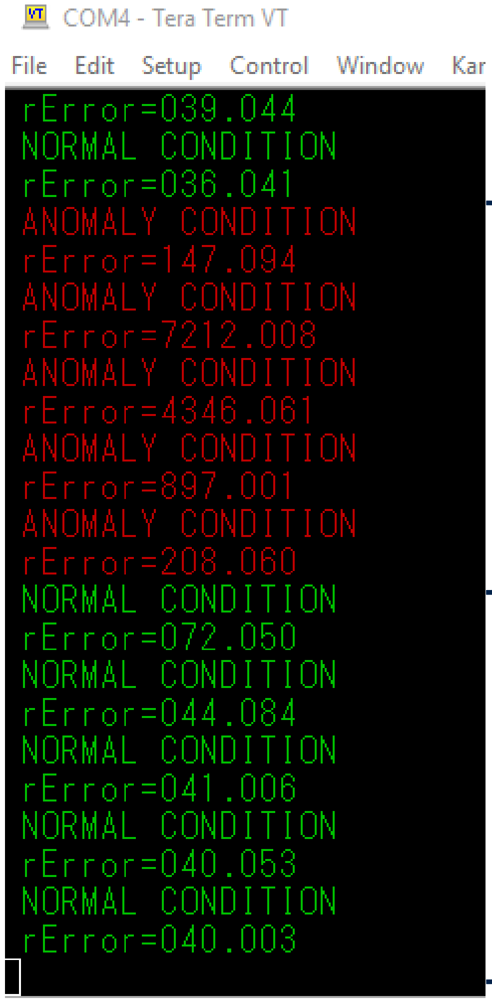


If the autoencoder is not trained with transients that occur during drift, it will signal them as anomalies.

Anomaly Detection Demo



Accelerometer settings



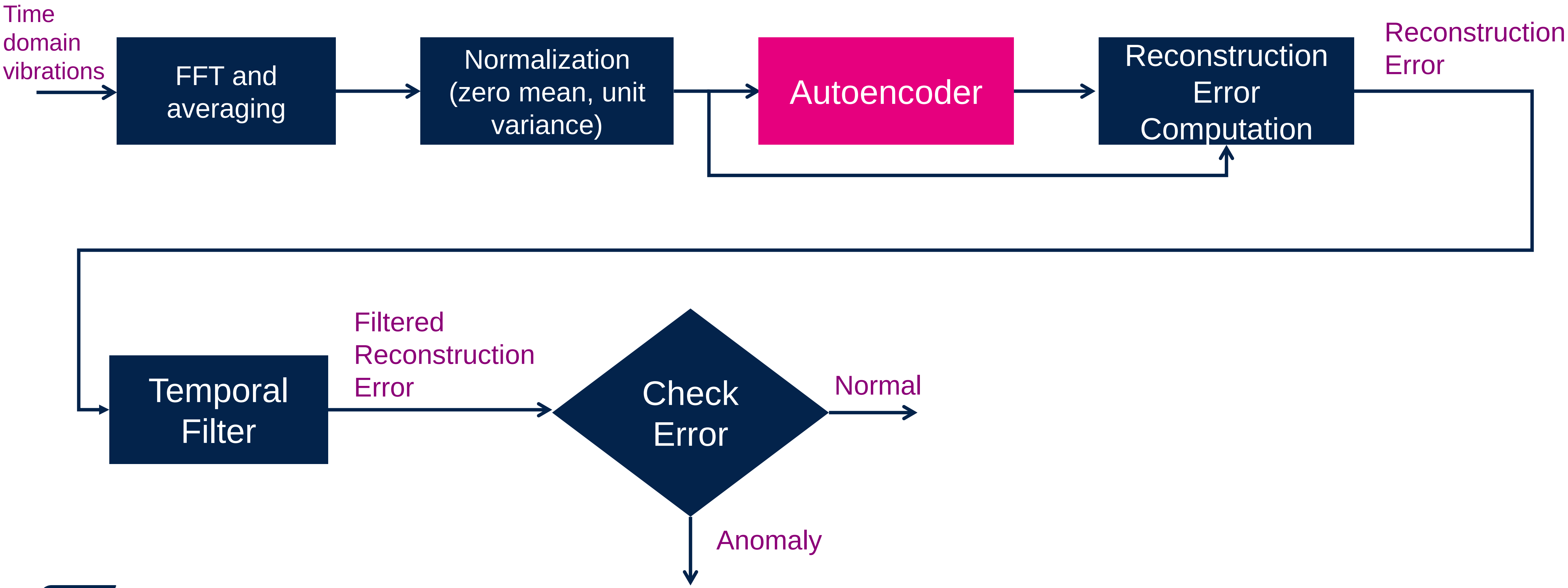
Anomaly Detector output during a speed sweep

The systems passes through unknown speeds (transients)

The systems is at a known speed. No disc unbalancing

life.augmented

Brief description of the use case: processing pipeline



Uncertainty and Out of Distribution Detection in Deep Neural Networks

life.augmented

Introduction

- Deep Neural Networks (DNNs) can make incorrect and overconfident predictions.
- This is a problem when DNNs must be deployed in real-world safety-critical applications, such as

Autonomous Driving

Medical Diagnosis

Predictive Maintenance

- Let us explore why

Let’s start with an example

Sources of Uncertainty

Real World Data Uncertainty

Measurement Choices and Errors

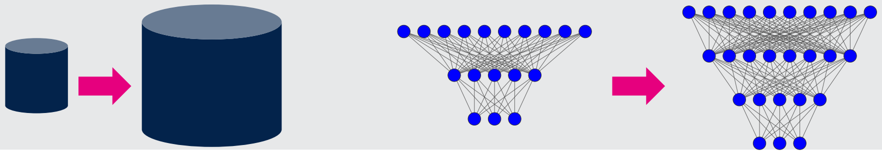
Choice of the Model Architecture

Choice of Training Hyperparameters

Inference Process

Predictive Uncertainty

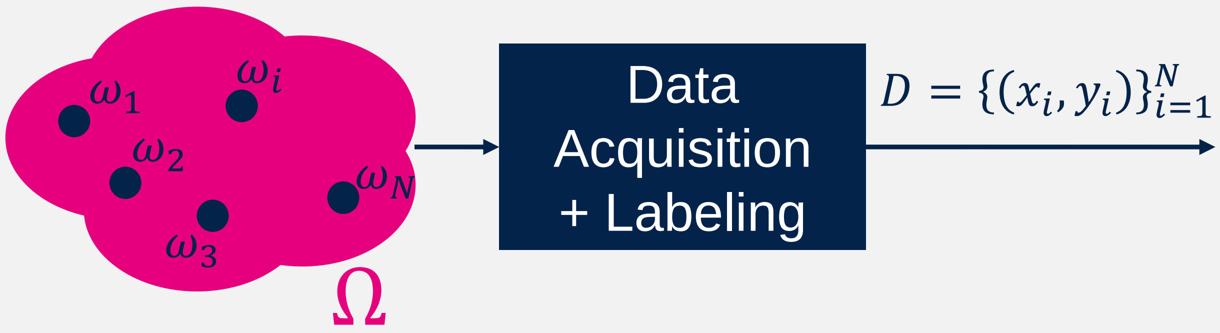
Aleatoric and Epistemic Uncertainty

	Root Cause	Can be reduced?
Aleatoric Uncertainty	Intrinsic Randomness in the data generation process (e.g. sensor noise, stochastic processes)	**Not possible** even acquiring new data
Epistemic Uncertainty	**Data scarcity, weak model** 	**Yes:** by increasing quality/quantity of the data and/or by refining the model 

Out of Distribution	• Even with *enough* data and a *good* model, you still face the OOD problem. • Why? Because of new **semantic classes**, **anomalies**, **data drift**, **extrapolation** regime, **adversarial samples**, ...

Uncertainty - Data Acquisition Process

How does uncertainty propagate from the real world to a prediction y*?



Each x is not a perfect representation of the corresponding ω

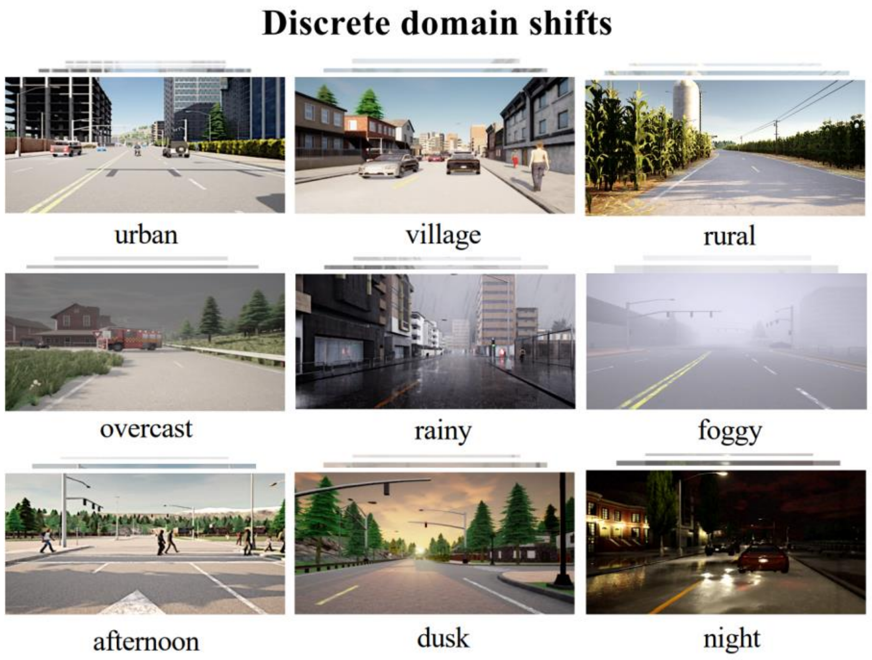
We choose a domain in which samples x are acquired. Multimodal is also possible.

- **Variability of the Real World**
- **Error and Noise in Measurement Systems**

$x \mid \omega \sim p_{x\mid\omega}$	$y \mid \omega \sim p_{y\mid\omega}$
Data Space	Label Space

Data Acquisition Process
Real World Variability

- Real-world conditions constantly change, leading to **distribution shift**.
- The current environment no longer aligns with the data known to the DNN.
- This makes DNNs less reliable/accurate when dealing with new, unseen scenarios.



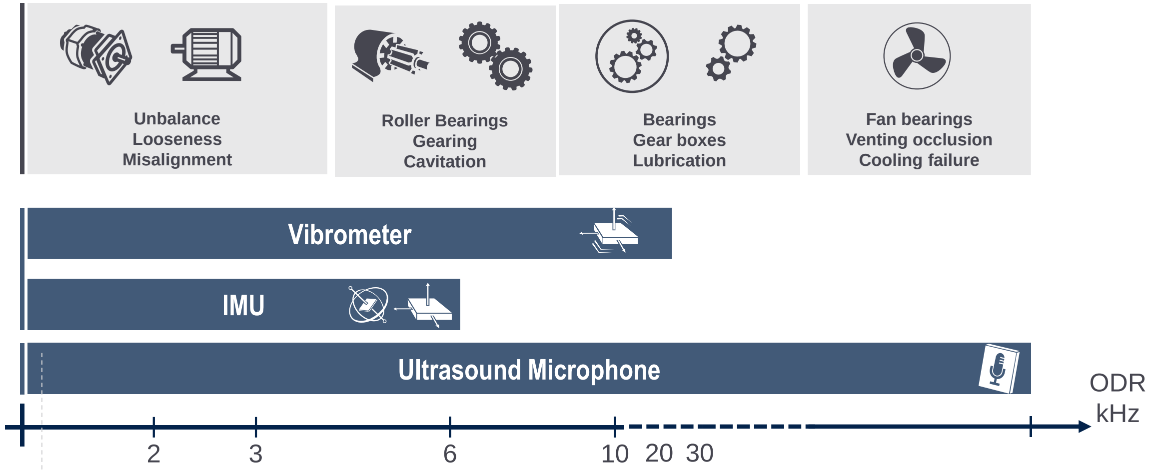
Source: [2206.08367] SHIFT: A Synthetic Driving Dataset for Continuous Multi-Task Domain Adaptation (arxiv.org)

Unknown variations of known classes

Data Acquisition Process
Measurement Errors

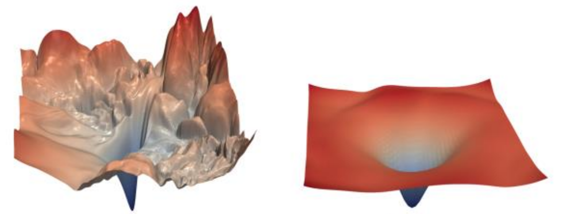
Combination of: hardware faults, complex real world domain, weak/bad acquisition choices:

- Sensor Faults
- Sensor Noise
- Low Resolution
- Missing Data
- Label Noise



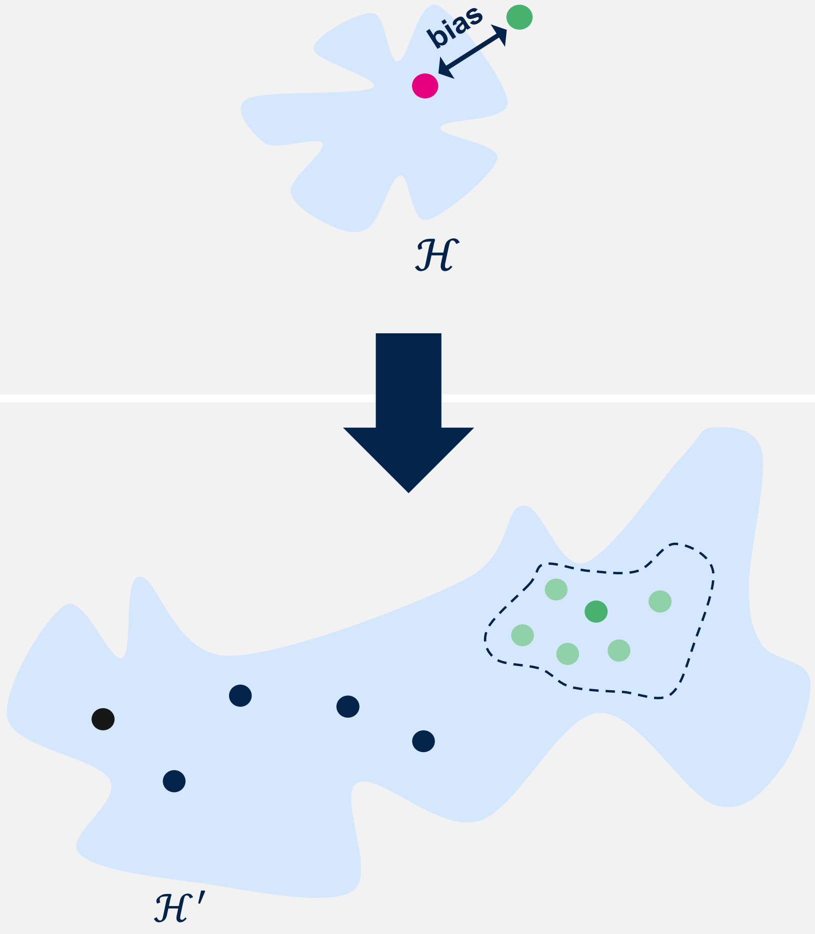
Uncertainty - Architecture of the Model : DNN Training

Architecture of the DNN $\boldsymbol{f_\theta}$	f_θ (# layers, # params $\boldsymbol{\theta}$,…) is trained on **finite** set of (x, y) pairs D Stochastic Process depends on a random variable θ (weights): $\theta \mid D, f \sim p_{\theta \mid D, f}$
Inference	applied on unseen samples $x^* \neq x_i$: $f_\theta(x^*) = y^*$ is y^* right, wrong or n.a.?
Loss Landscape	highly non linear leads to different local minima f_{θ^*}, yielding different models


(a) without skip connections (b) with skip connections

Source: [Visualizing the Loss Landscape of Neural Nets (neurips.cc)]

Uncertainty - Architecture of the Model : Bias and Variance Tradeoff

Small Model	smaller hyphothesis space, higher bias, lower variance, → risk of underfitting, lack of generalization, the model is too simple	
Larger Model	larger hyphotesis space, lower bias, higher variance (the model adapts to data and noise), → risk of overfitting	

Uncertainty - Architecture of the Model:
Bias and Variance Tradeoff

$$Total\ Error = Bias^2 + Variance + Irreducible\ Error$$

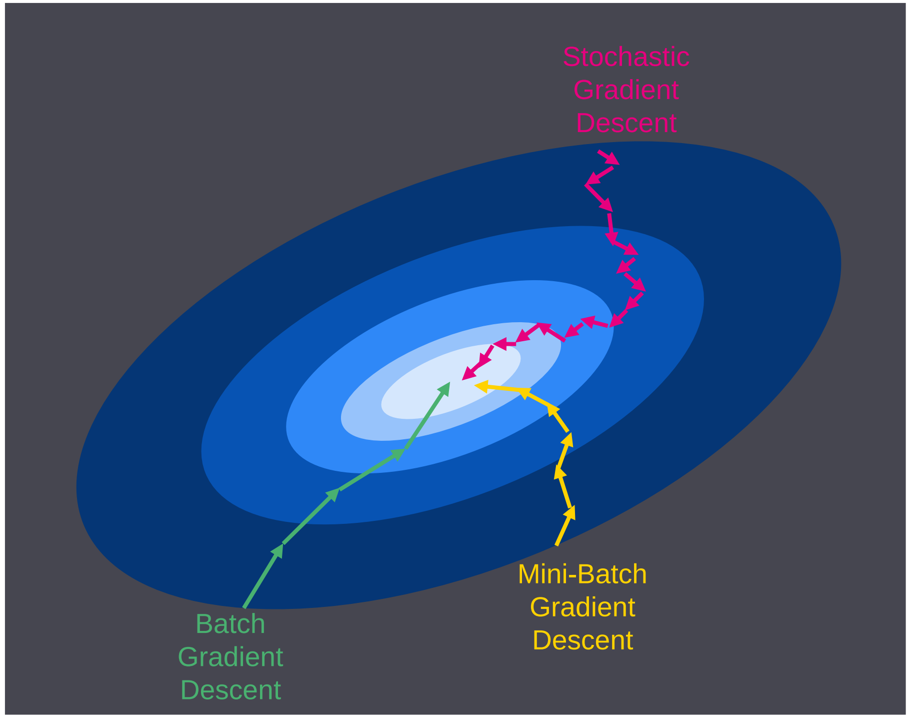
Error introduced by approximating a real-world problem, using by overly simple model

Error that cannot be reduced by any model. It's inherent in the problem itself and represents the noise or the randomness in the data.

Error introduced by the model's complexity. A model with high variance has too high "capacitance", it performs well on the training data but not on unseen data. It overfits: "memorizes" the training data, potentially fitting also random noise.

Uncertainty – DNN Training: Gradient Descent

Batch Gradient Descent	The **entire training** set is **shown** to the optimizer **before** triggering a **backpropagation** step (i.e. update of the weights)
Mini-Batch Gradient Descent	The training set is divided into **smaller subsets** (mini-batches). A **backpropagation** step is triggered **every** time a **mini-batch** has been shown to the optimizer. Trade-off between Batch and Stochastic Gradient Descent.
Stochastic Gradient Descent	The size of the **mini-batches** is reduced to **1**. A **backpropagation** step is triggered every time a **new sample** is shown to the optimizer.



Training is a Stochastic Process

Architectural Hyperparameters	• Convolutional Network, MLP, Skip Connections, • Activation Functions, Number of neurons, #filters • Hyperparameters either manually chosen, or automatically determined via grid search, …
Training Hyperparameters	• Weights Initialization, Loss Function, Loss Regularization Terms, Batch Size, Learning Rate, Number of Epochs, Stopping Criteria, …
Input Data	• Balanced or unbalanced dataset and related weighting coefficients • Data Augmentation • Dropout, …
Consequences	• Different **hyperparameters** choices lead to different model accuracies/uncertainty. • **Unbalanced** training sets skew the classifier if appropriate compensation is not applied (e.g. **SMOTE**: Synthetic Minority Over-sampling Technique).

Uncertainty – DNN Training & Inference
Unknown Data

Unknown Data

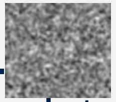
- In classification tasks, a DNN trained on samples from domain P, could receive as input samples from an unknown domain N, or *uncovered subspaces* of P

Source of Uncertainty in case of Unknown Data

In this case the source of uncertainty does not lie in the data acquisition process

How does this data look like?

Unknown data might resemble too much noise on a sensor or complete failure, but actually it is not:

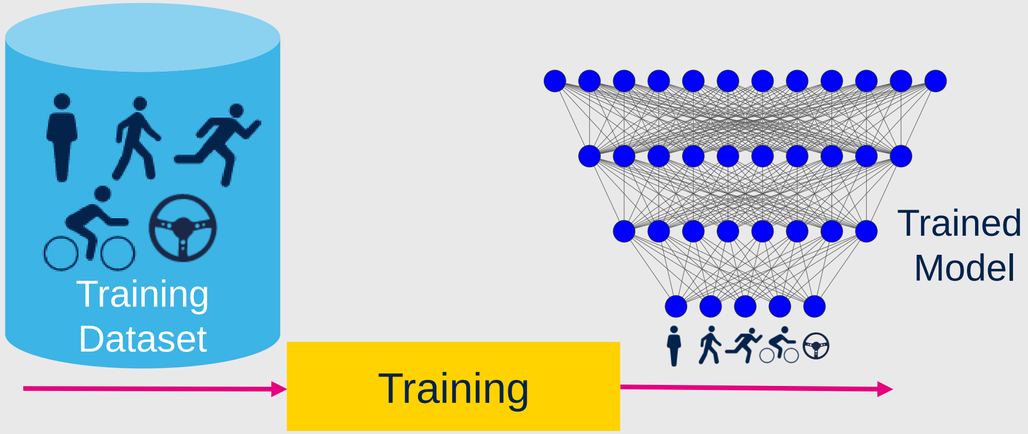
- Pure noise is not the only definition of OOD. 
- Samples that form novel classes wrt training data are also OOD.
- Anomalies are OOD.

How do we detect this?

Closed Set vs Open Set Classification

Closed Set Classification

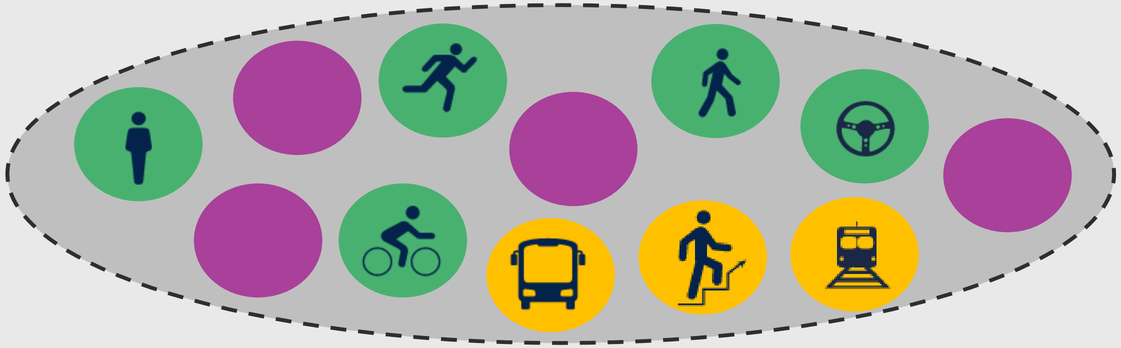
Discriminate between a given set of classes



Standard Evaluation Protocol:
Confusion Matrix

Open Set Classification

Discriminate classes in a larger space we do not know about

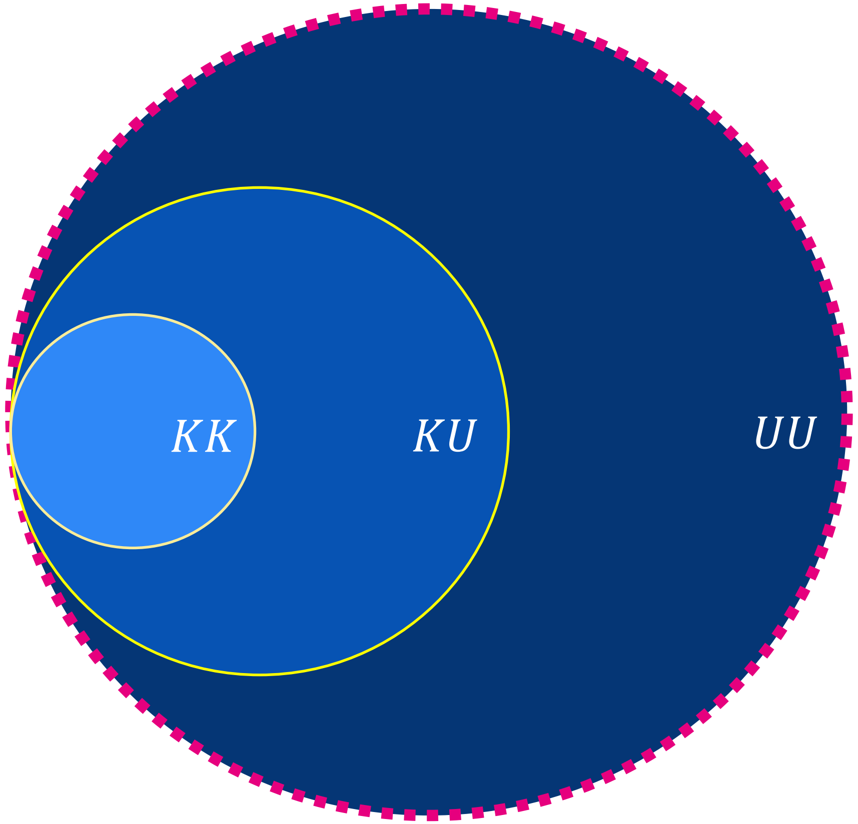


Known Known : In-Distribution

Unknown Unknown : Out-of-Distribution

Known Unknown : labeled OOD Samples

Openness Measure of a Classification Task



$$O^* = 1 - \sqrt{\frac{KK + KU}{KK + KU + UU}}$$ [Battaglino et al. 2016]

In the AED use case:

$KK = 4, UU = 46, KU = 0$

$$O^* = 1 - \sqrt{\frac{4 + 0}{4 + 0 + 46}} = 1 - \sqrt{\frac{4}{50}} = 1 - 0{,}28 = 0{,}72$$

In training we can use data from KK and KU

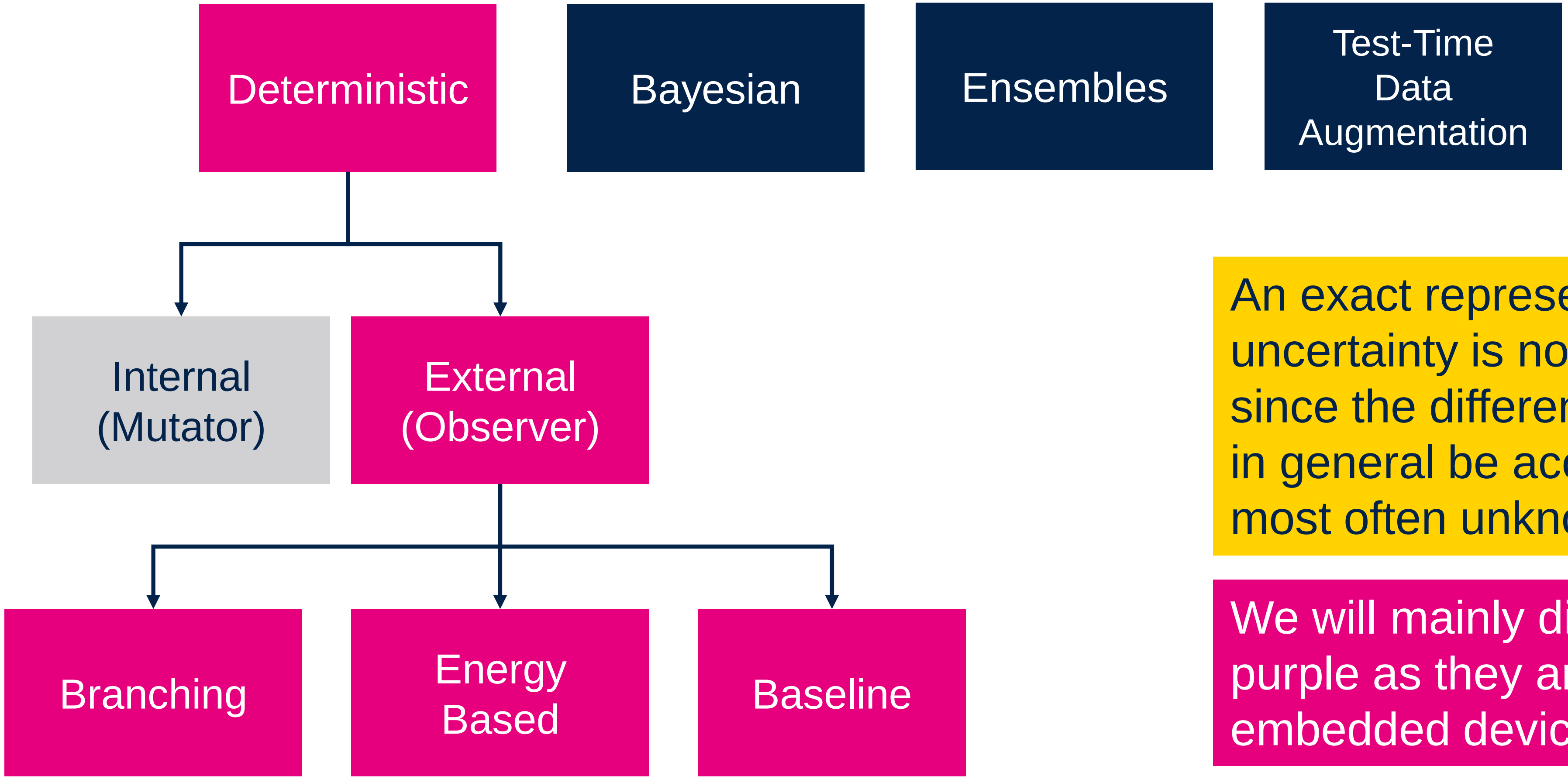
UU>0 only in openset classification (UU is used at test time only)

Original formulation:

$$O^* = 1 - \sqrt{\frac{2 \times T_{Tr}}{T_{Tr} + T_{Te}}}$$ [Scheirer et al. 2012]

- In general, the higher the openness, the more difficult for the classifier to be «always» accurate and reliable.
- This is a rough estimate that does not take into account the model, nor the complexity of the classes
- In the standard case, the classifier accuracy is estimated in a Closed Set scenario.

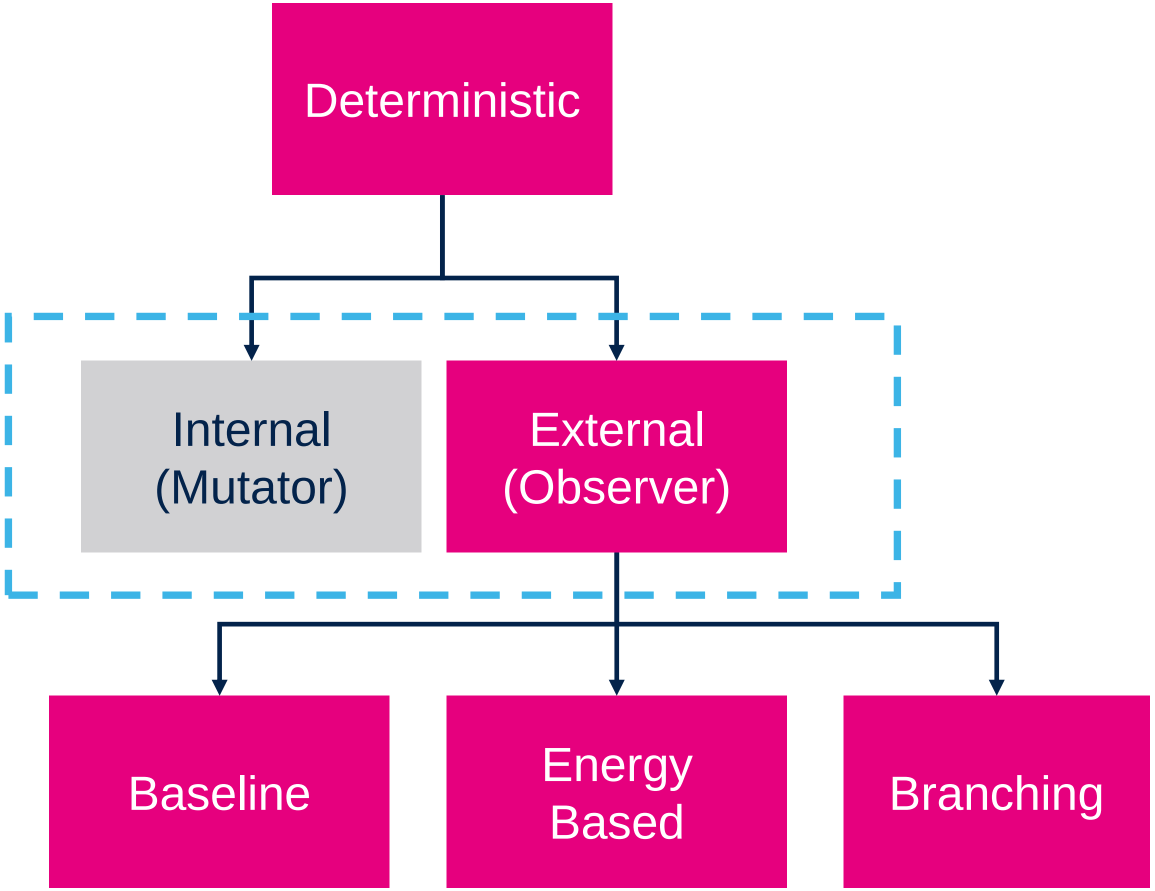
Uncertainty Estimation



An exact representation of DNN uncertainty is not possible to compute, since the different uncertainties cannot in general be accurately modeled and most often unknown.

We will mainly discuss the blocks in purple as they are suitable for embedded devices implementation.

Single Deterministic Methods (SDM)



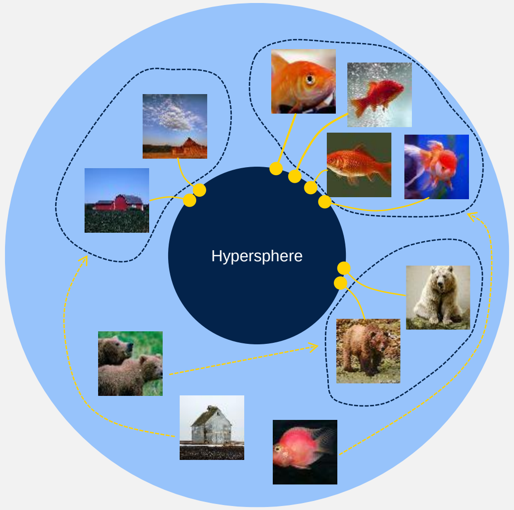
- DNNs parameters $\boldsymbol{\theta}$ are fixed, inference is deterministic.

SDMs are broadly categorized in two approaches:

- A single network is explicitly modeled and trained in order to «handle» uncertainties (**Internal Approach**, Mutator Approach)
- Additional components in order to give an uncertainty estimate on the prediction of a network (**External Approach**, Observer Approach)

Internal Uncertainty Estimation

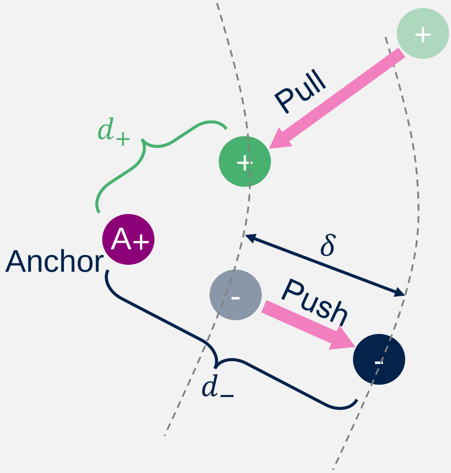
Contrastive Loss



- Contrastive loss minimizes the distance between the positive (similar) examples while increasing the distance between the negative (dissimilar) examples

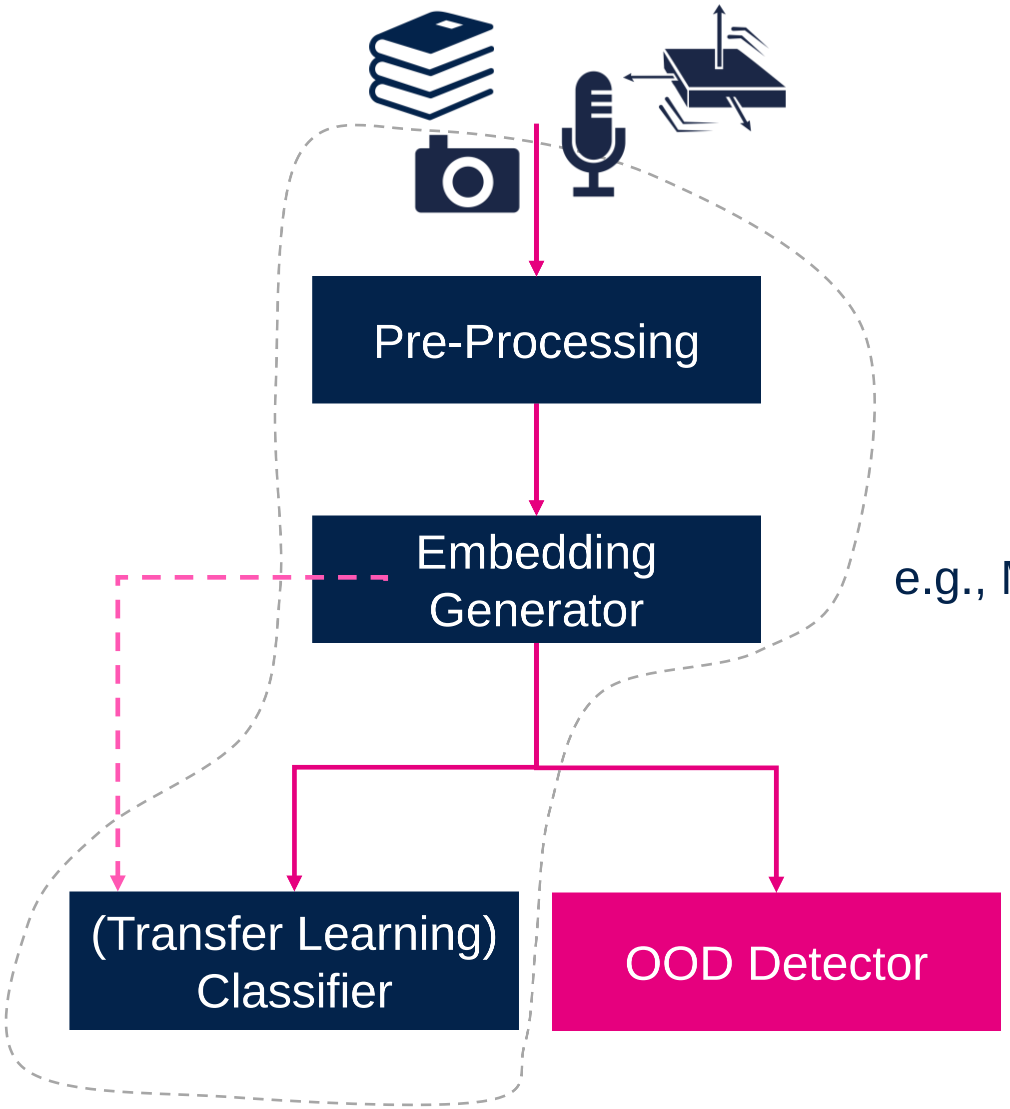
Inspired by: [reveiw-understanding contrastive loss (velog.io)]

Triplet Loss



- Negative samples are pushed away from the anchor
- Positive samples are pulled towards the anchor
- Distances are not necessarily Euclidean
- The triplet loss tries to preserve a minimum margin δ between positive and negative distances

External Uncertainty Estimation



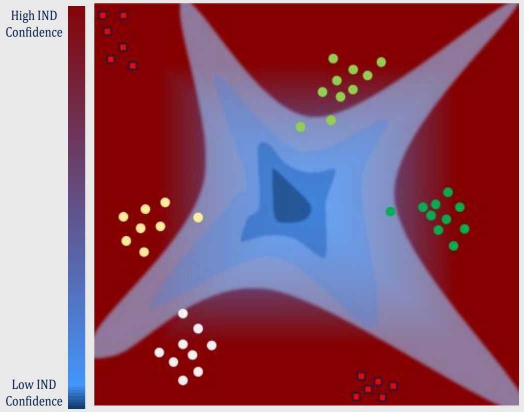
e.g., MobileNetV2 for embedded devices

Internal vs External Methods

	Internal Methods	External Methods
Description	• Requires ad-hoc training. • Single inference. • No external components.	• The network performs a single inference. • An additional component estimates uncertainty
Computational Complexity	• Usually relies on modification of the loss	• Depends on the complexity of the additional component
Can be applied on pre-trained networks	No	Yes
Separate prediction and uncertainty estimation	No	Yes
Requires negative data	Depends	Depends

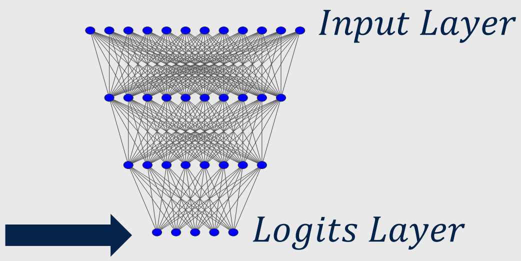
Max-Softmax Baseline Approach for OOD Detection

Overconfidence Issue



- NNs learned decision boundaries are reliable for in-distribution data only.
- The network could “fail silently” with high confidence on OOD data.

Max-Softmax (MSM) Baseline Approach



- Given logits z_i, ***softmax*** computes uncalibrated probabilities p_i:

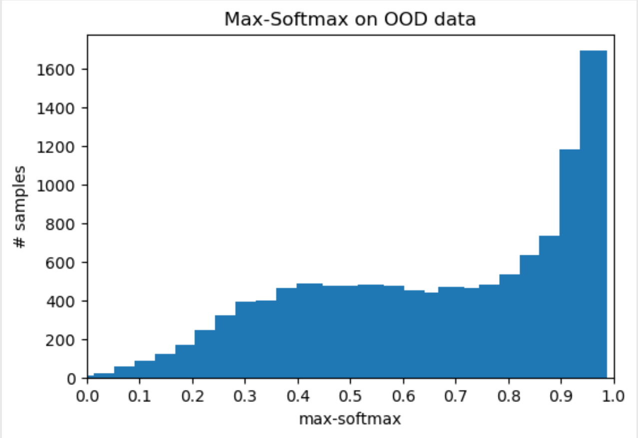
$$p_i = \frac{\exp(z_i/T)}{\sum_{j=1}^{L} \exp\left(z_j/T\right)},$$

$$T \geq 1, i = 1, \ldots, \#classes$$

$$such\ that \sum_i p_i = 1$$

- T increases the sensitivity to low probability candidates.

MSM on OOD Data

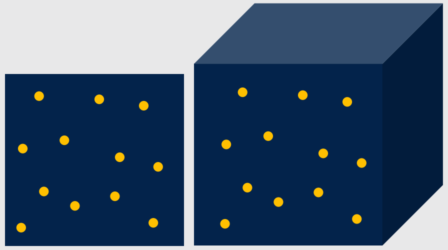


Ambient vs Embedding Space

1

Ambient Space

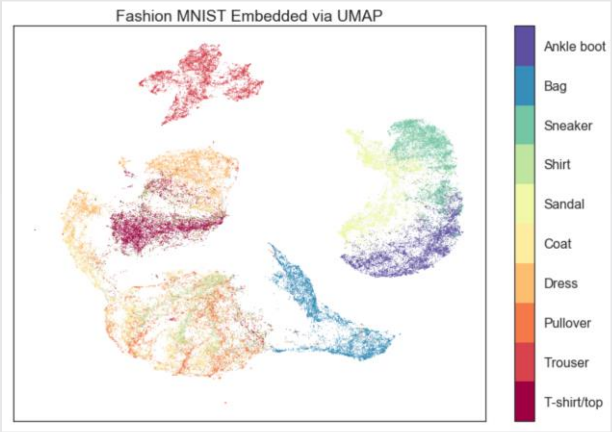
- High Dimensional
- Redundant Dimensions
- Sparsity Dominates
- Distance metrics lose meaning



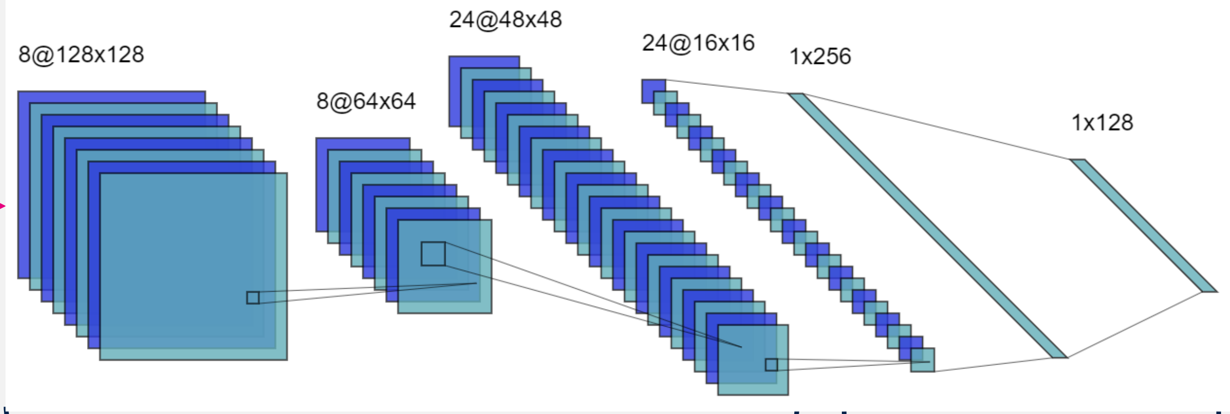
2

Embedding Space

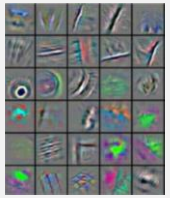
- “Compact” data representation
- “Less” Redundant Dimensions
- “Less” Sparse
- “HQ” embeddings



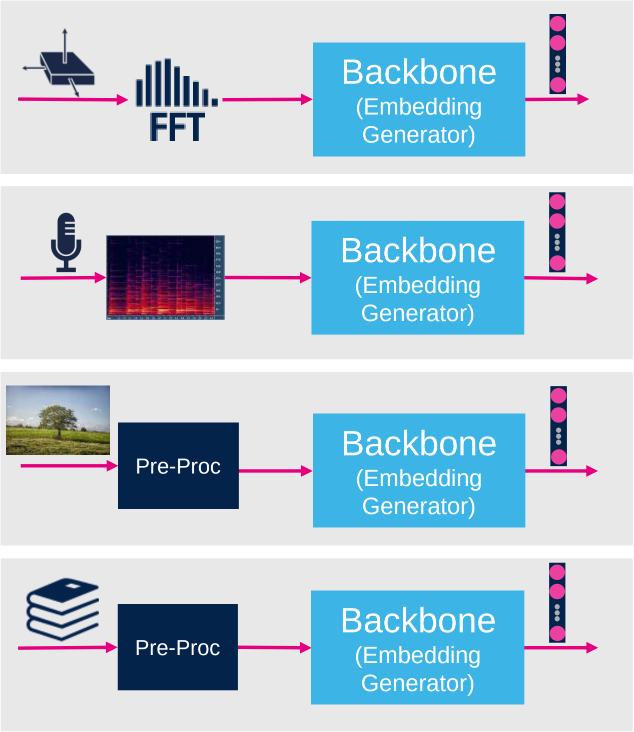
Deep Feature Vectors are Embeddings



- High Dimensionality
- Layers for **“Low” level features**, likely **shared** among **IND** and **OOD**

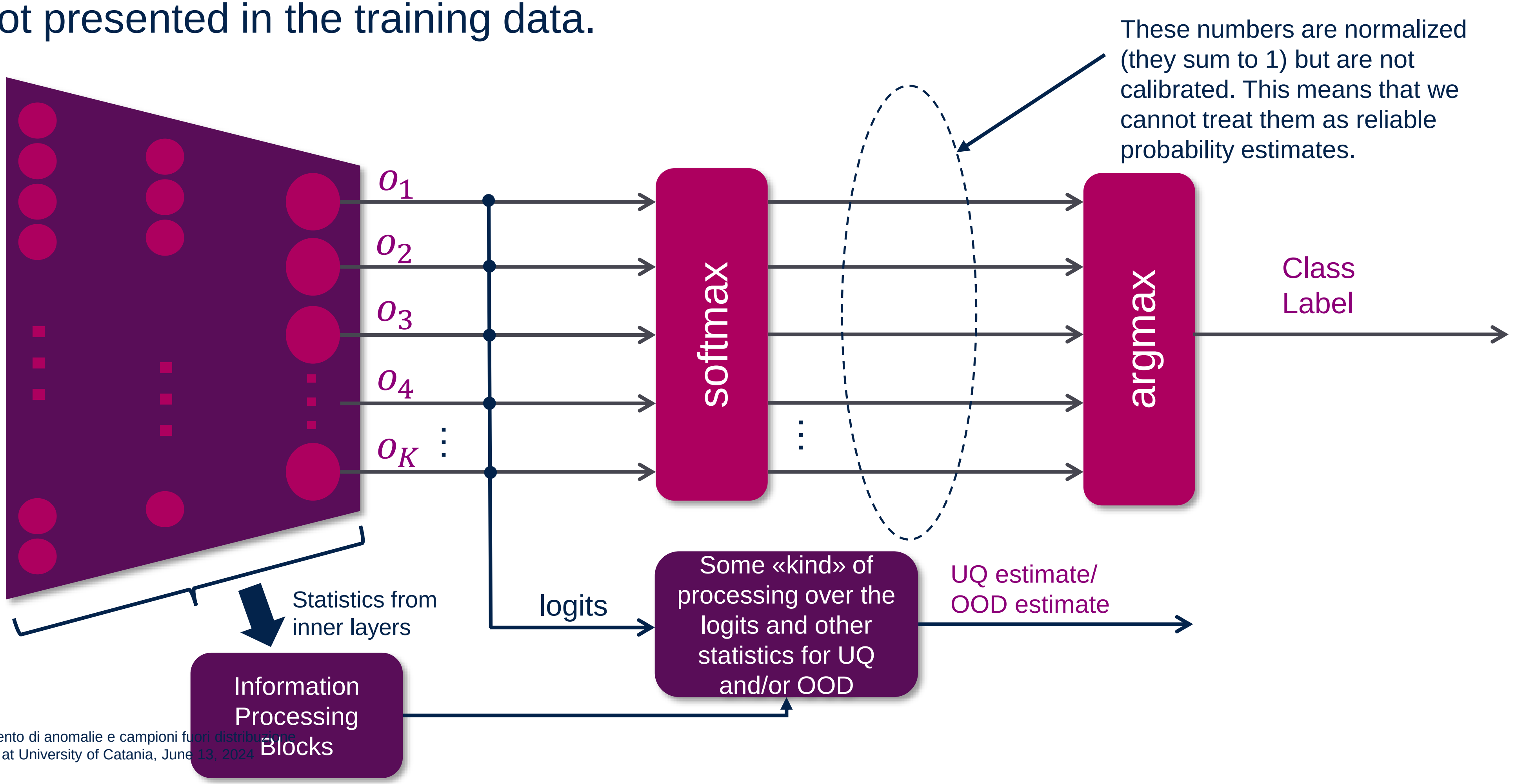


- **“Good” lower dimensional layers** conveying more information **peculiar to In-distribution data**.
- Thus, **useful** for **OOD** Detection



Uncertainty Quantification – Baseline (External Uncertainty Estimation Method)

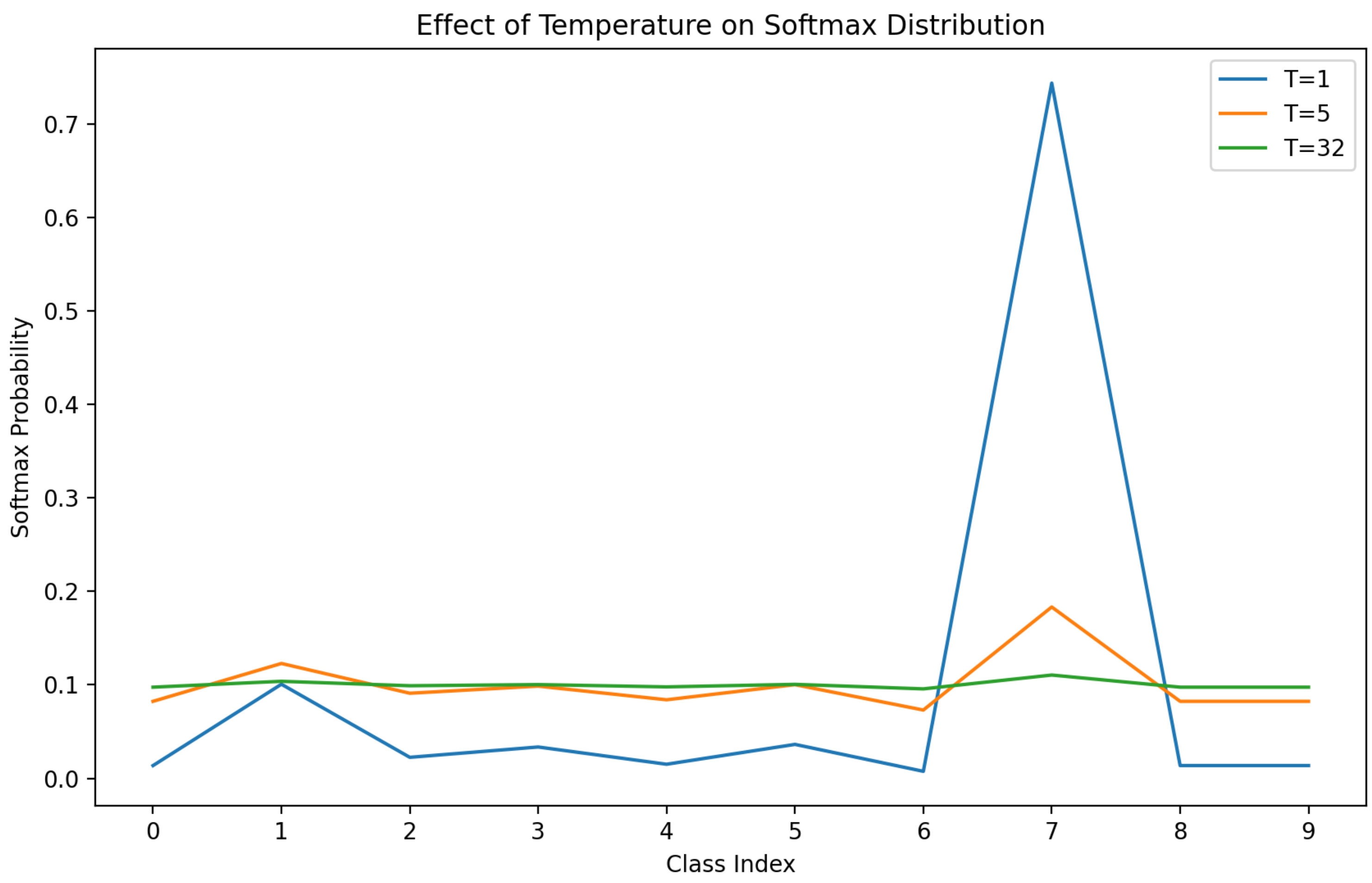
- A baseline approach for UQ is to analyze the activation outputs before the softmax layer : if unnormalized scores are ***relatively small*** for all classes, you are likely observing a novel class not presented in the training data.



Uncertainty Quantification – Baseline Softmax and Temperature Scaling

Softmax	In a classification problem with N classes, softmax normalizes a not normalized embedding vector into normalized vector of N real numbers such that it sums to 1.		
Standard Softmax	$p(i\|\mathbf{z}) = \frac{e^{z_c}}{\sum_{j=1}^{L} e^{z_j}}$, $j = 1, \dots, N$	**Softmax with Temperature Parameter**	$p(i\|\mathbf{z}) = \frac{e^{\frac{z_i}{T}}}{\sum_{j=1}^{L} e^{\frac{z_j}{T}}}$, $j = 1, \dots, N$
Softmax with Temperature Parameter	• The temperature T controls the randomness of the predictions by scaling the logits before applying the softmax. • Increasing the temperature T yields smoothed probability distributions, converging to uniform distribution as T increases and tends to infinity. • The temperature increases the sensitivity to low probability candidates.		

Softmax with higher Temperature

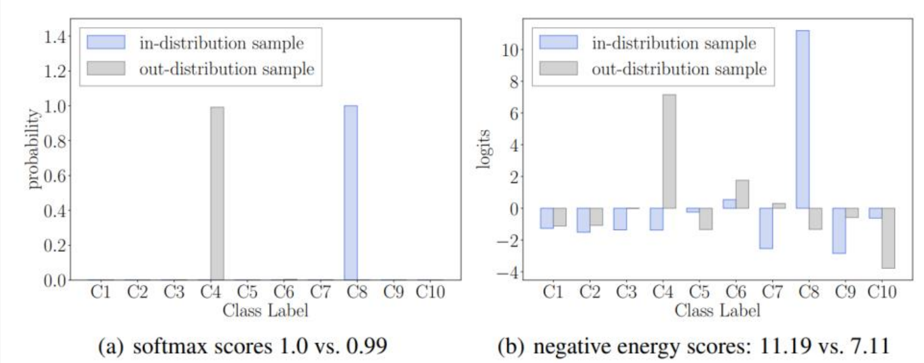


Increasing T, lowers the peak and increases smaller probabilities

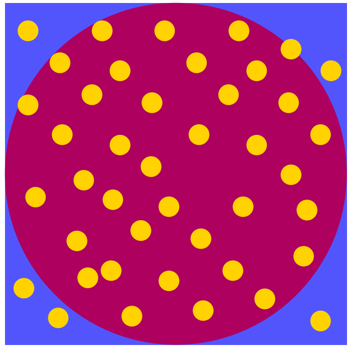
number one

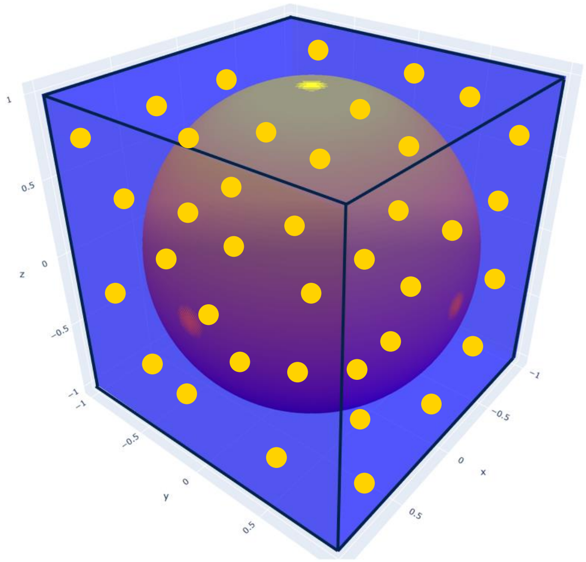
number seven..oh wait..or is it number 1 ?

Energy Based Models for OOD

Energy Based Models (EBMs)	Based on Gibbs (Boltzmann) distribution of statistical physics. A physical system tends to go towards state(s) of minimum energy. → Minimum energy states are the most probable.
Energy for OOD Detection	It is possible to derive the mathematical formulation from the Gibbs distribution to the Energy associated to a ML system (e.g., a neural network). $E(\boldsymbol{x}) = -log \sum_{i=1}^{L} e^{z_i}$ z_i is the i-th logit of the embedding vector produced by the classifier
Energy	We expect low energy for in-distribution samples and high energy for OOD samples  The lower the energy associated with a particular state of the physical or ML system, the higher the probability that the system will reach that state. [2010.03759] Energy-based Out-of-distribution Detection (arxiv.org)

HD Spaces are Counterintuitive





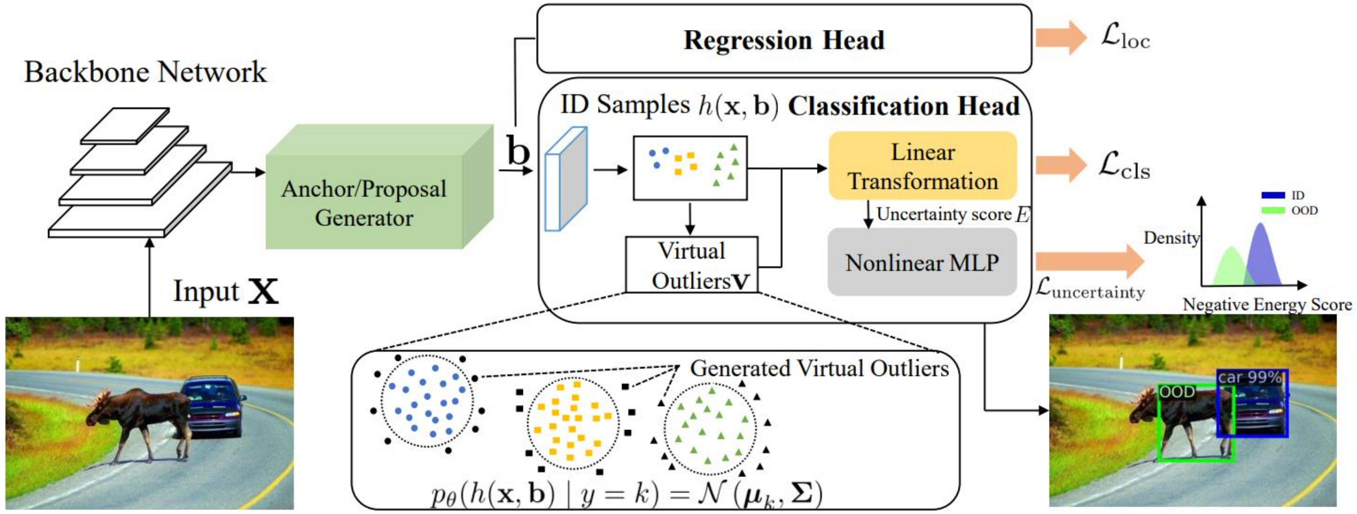
The **volume** in **HD spaces increases exponentially** as we add dimensions, and data becomes sparse.

A HD space is **mostly “empty”**, most of the points that could be sampled lie close to the boundary of the space.

In a **unit hypercube**, **almost all** of the **volume** is **near** the **edges**, thus **data points are likely to be found near the surface of the volume**.

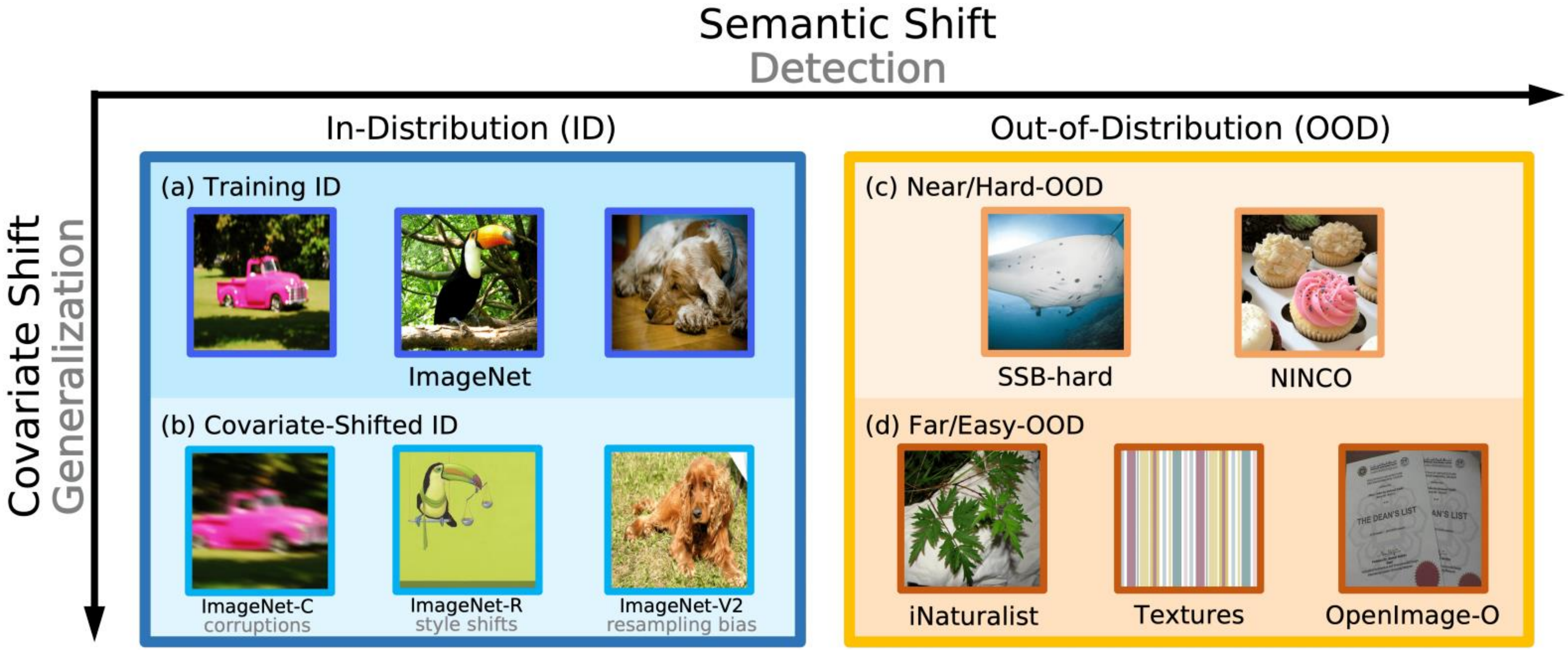
Distance Metrics tend to lose their meaning, as all samples are almost equidistant from each other.

Virtual Outlier Synthesis (in the latent space)



- [2202.01197] VOS: Learning What You Don't Know by Virtual Outlier Synthesis (arxiv.org)

Covariate Shift – Semantic Shift
Near and Far OOD



[2306.09301] OpenOOD v1.5: Enhanced Benchmark for Out-of-Distribution Detection (arxiv.org)

Probability Distributions & Uncertainty

In-Distribution Data	At inference time, the test distribution has the same distribution of the training data $P_{test}(x, y) = P_{train}(x, y)$
OOD Data / Anomalies	At inference time, the test distribution has different distribution w.r.t. the training data $P_{test}(x, y) \neq P_{train}(x, y)$
Covariate Shift	Distribution of p(x) changes (samples from different or shifted domain), while p(y|x) remains constant: $P_{test}(x) \neq P_{train}(x)$ The distribution of the data changes, … $P_{test}(y\|x)$ …, but the label remains the same
Label Shift	Distribution of labels p(y) changes while p(x|y) remains constant

OOD Benchmarking

		NEAR-OOD	FAR-OOD
MNIST	10-class handwriting digit dataset that contains 60k images	near-OOD includes NOTMNIST, FashionMNIST	Texture, CIFAR-10, TinyImageNet, Places-365
CIFAR-10	10 classes of natural images	CIFAR-100, TinyImageNet,	MNIST, FashionMNIST
CIFAR-100	100 classes of natural images	CIFAR-10, TinyImageNet	MNIST, FashionMNIST, Texture, Places365
ImageNet-1K	1000 classes of natural images	iNaturalist, ImageNet-O, OpenImage-O	Texture, xMNIST, SVHN

FPR@95 is the false positive rate when the true positive rate is set to 95%.
Lower scores indicate better performance.

Our technology starts with You

Find out more at www.st.com

life.augmented