CONVOLUTIONAL NEURAL NETWORKS FOR ANOMALY DETECTION

Giacomo Boracchi,
Politecnico di Milano, DEIB.
https://boracchi.faculty.polimi.it/

February 7th, 2022
Advanced Deep Learning Models and Methods
PhD course

GIACOMO BORACCHI

Mathematician (Università Statale degli Studi di Milano 2004),

PhD in Information Technology (DEIB, Politecnico di Milano 2008)

Associate Professor since 2019 at DEIB (Computer Science), Polimi

Research Interests are mathematical and statistical methods for:

- *Image / Signal analysis and processing*
- *Unsupervised learning, change / anomaly detection*

ADVANCED DEEP LEARNING MODELS AND METHODS

Course Logistics

CALENDAR

The course will be taught in a blended mode: in presence for those who are allowed to join, with remote connections for the other students.

Lectures will be streamed on Giacomo Boracchi webex room

- February 7th 14:15 - 18:30 : *Deep Unsupervised Learning in Images*, Giacomo Boracchi
- February 9th 9:00 – 13:00: *Learning with Limited Supervision*, Alessandro Giusti (USI-SUPSI) **Sala Conferenze**,
- February 10th 14 – 18: *Deep Reinforcement Learning* Alessandro Lazaric (Facebook Paris). **This will be held online, connect to my webex room**
- February 16th, 14:15 - 18:30 *Deep Learning on Graphs and Structured Computation Models*, Jonathan Masci (NNAISENSE). **Sala Conferenze**
- February 17th, 14:15 - 18:30 *Variational AutoEncoders with Applications to Anomaly Detection,* Luigi Malagò (TINS, RM) ***Room 25.01***
- February 18th, 14:15 - 18:30 *Privacy preserving learning*, Matteo Matteucci (Polimi) ***Room 3.0.2***

REFERENCES: HTTPS://BORACCHI.FACULTY.POLIMI.IT/TEACHING/ADVANCEDDLMM.HTM

I will publish all the recordings, as long as these are fine for the speakers.

GIACOMO BORACCHI - TEACHING

Advanced Deep Learning Models and Methods
AA 2021/2022, PhD Course, Politecnico di Milano

Overview:
The course presents advanced learning problems and how deep learning models have been applied to successfully solve them. In particular, models are considered advanced either in terms of non-conventional data-type being handled (e.g. graphs), the conditions where the learner has to operate (lack / shortage of supervision), or the problem is extremely challenging by itself (e.g. generating a natural image).

More information on the Course program page

Organizers:
Giacomo Boracchi, Matteo Matteucci. Politecnico di Milano.

Dates:
From February 7th 2022 to February 18th 2022, 6 seminars of 4 hours each.

Enrollment:
Students from other universities are welcome to attend the lectures, but in general cannot take the exam nor receive an official attendance certificate from our secretariat. Students that need an official attendance certificate have to perform an official registration following the procedure on the PhD website . In case this case, an administrative fee (32E) is requested.

Blended Teaching Modality:
Either in presence or in my webex room. Please check details below, a lecture will be held entirely online

Schedule and Abstracts:

Deep Unsupervised Learning in Images
***Giacomo Boracchi** Professor at Politecnico di Milano*
February 7th, 14:15 - 18:30 Sala Conferenze, DEIB, Building 20
Early deep learning models were primarily solving supervised visual recognition problems such as classification, segmentation, detection. More recently, there has been a surge of deep learning methods addressing unsupervised vision tasks, including image restoration, enhancement and anomaly detection. As for their supervised counterparts, deep neural networks (and sometimes even famous pre-trained architectures) turned out to be more effective than traditional model-based solutions. During this lecture I will provide an overview of the image restoration (and in particular image denoising) and anomaly detection problems, describe the most relevant deep learning solutions in the literature, and discuss the most relevant challenges to be addressed by deep learning.
Slides , Video Recording .

COURSE DETAILS FOR PHD STUDENTS:

- Minimum 70% attendance requried. In practice you can skip a single lecture
- We will circulate a link to a MS form right after the break, please connect to webex to get it
- Assessment is pass-or-fail
- The exam consists in giving a talk presenting one paper (or two related papers) concerning the lecture topics
- You have to propose one/two papers that you are interested in. We will review your suggestions and define, possibly together with the speakers, which ones to present. Papers can be selected among those referenced in our slides.
- You can gather in groups of 2 students. The presentation should be around 10' per students + discussion
- We will arrange a few presentation days, where you can book a slot. We will gather papers related to the same topic in the same session, like a conference. Everybody is welcome to attend those presentation as well
- There will be a Q&A at the end of your presentation. Questions are welcome from the audience as well.

COURSE DETAILS FOR MSC STUDENTS:

- No attendance requirements, but please fill the form
- Exam grading is 18 – 30L as usual
- You can recommend a paper related to the course topics which you are interested in
- Together with the paper, you need to propose a project where you implement and use the model presented in the paper. The simplest option is to reproduce the results, but you can go further proposing variants / studies / new applications.
- The detailed project will then need to be agreed with us.
- You can gather in groups of 2 students (or 3 when thoroguhly justified due to an excessive work load).
- You need to write
  - Either a report presenting the work done to reproduce the model and experiments.
  - Either a paper-like resume presenting your solution / application, if this is possible.

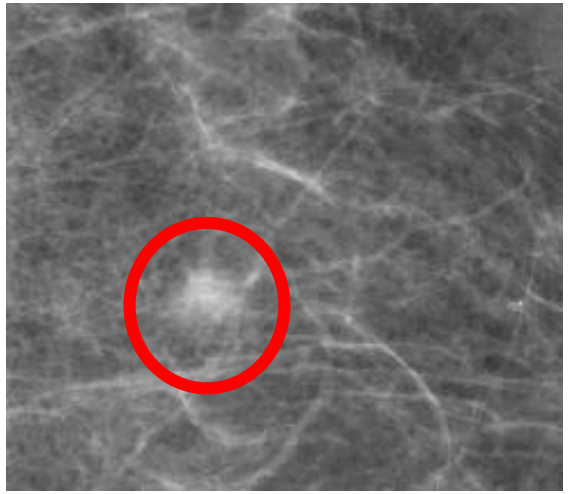
  Templates will be provided
- You need to prepare a 15' presentation
- There is no exam schedule for PhD courses. We will organize a few presentation days where everybody is welcome to attend

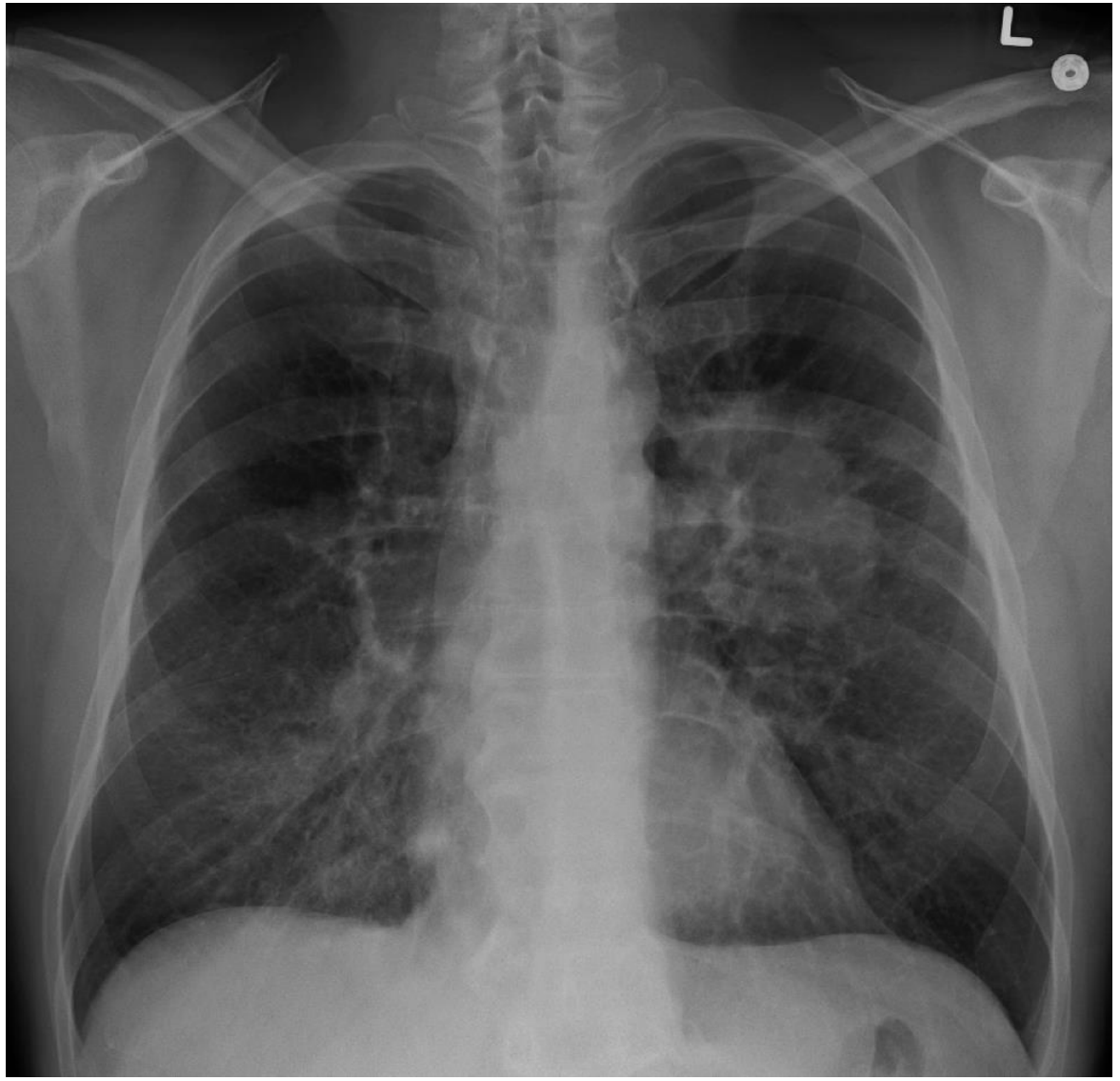
DEEP UNSUPERVISED LEARNING IN IMAGES

Anomaly Detection

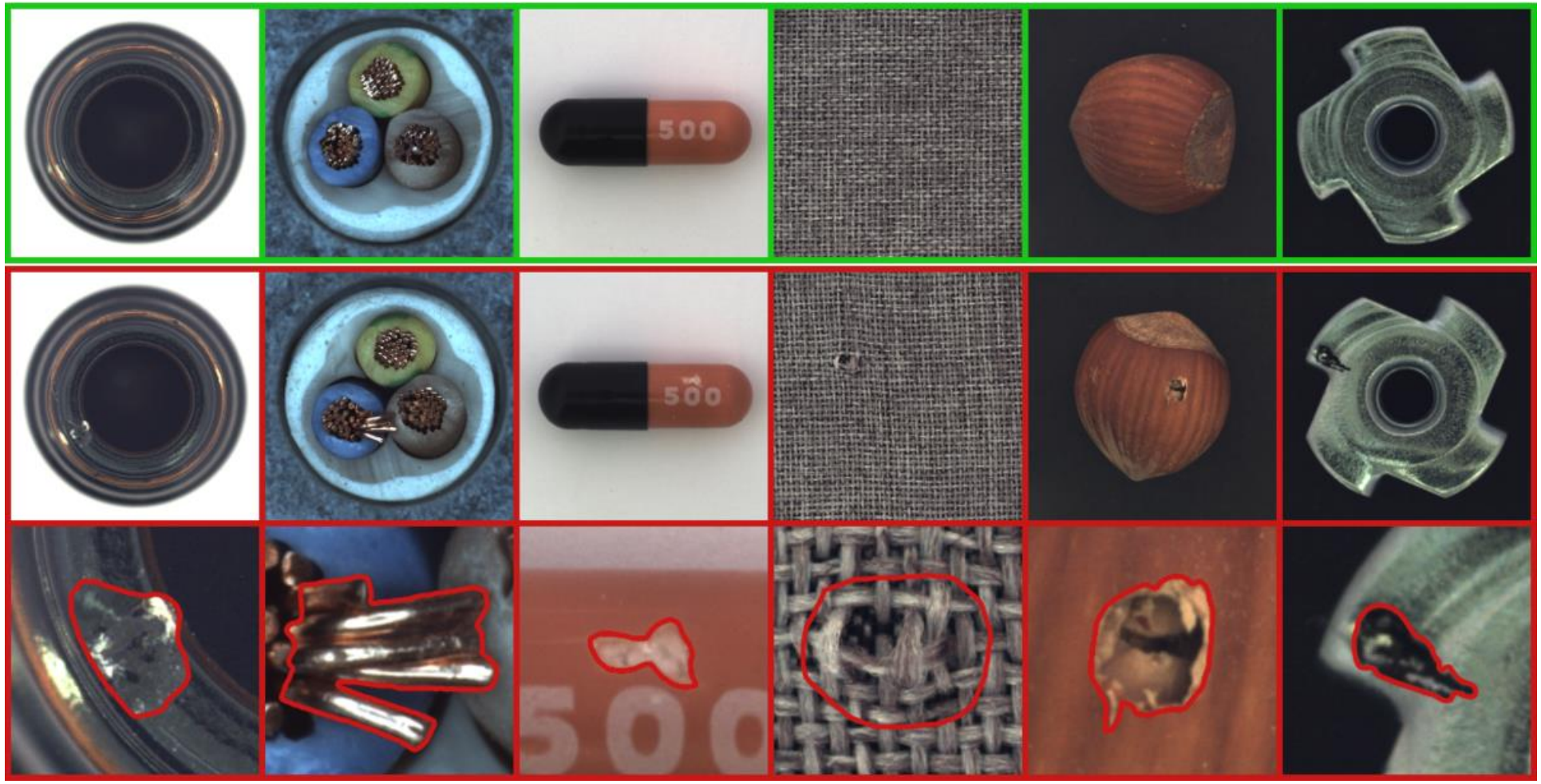
ANOMALY DETECTION IN HEALTH

Mammograms



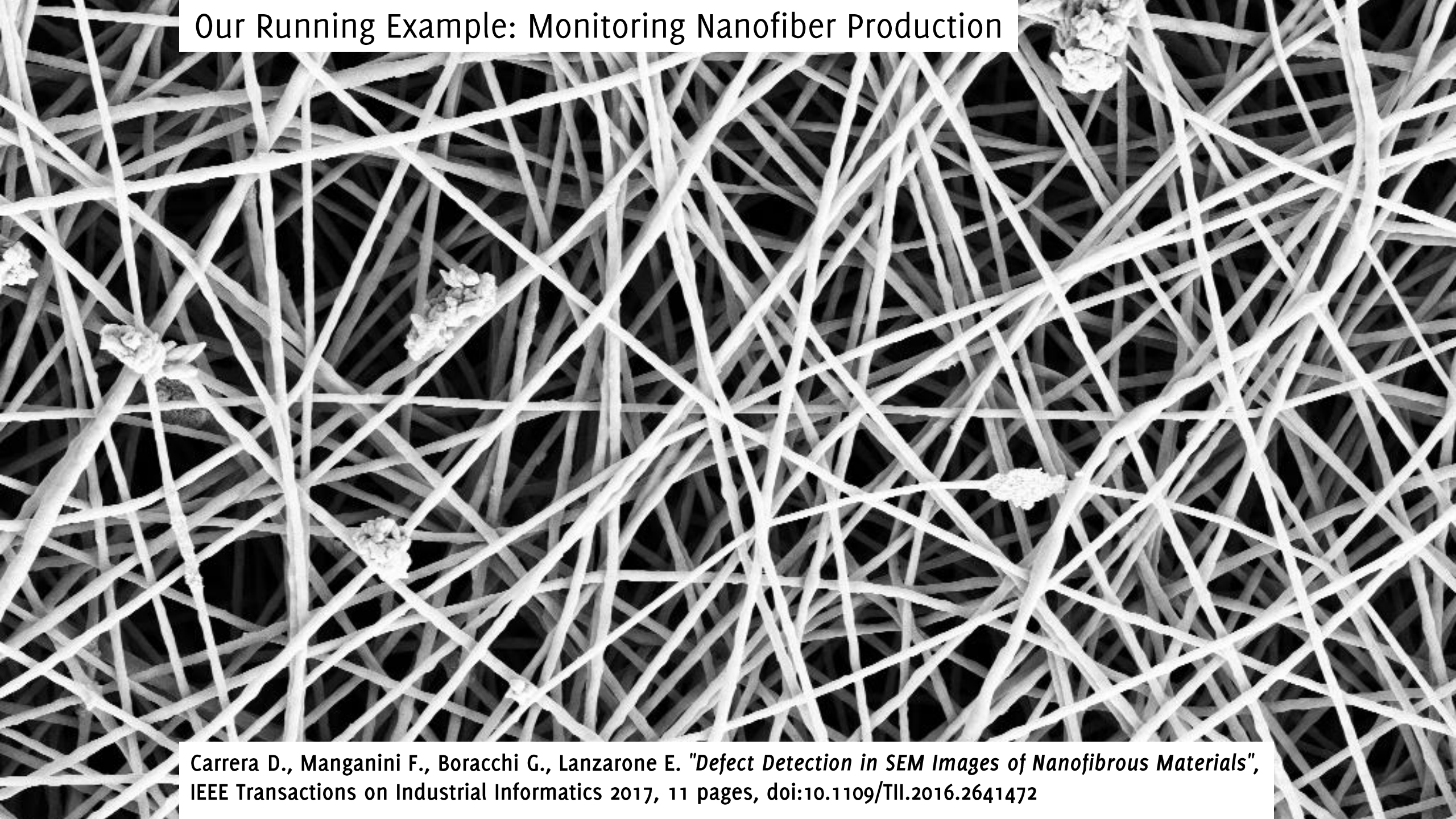


ANOMALY DETECTION FOR AUTOMATIC QUALITY CONTROL



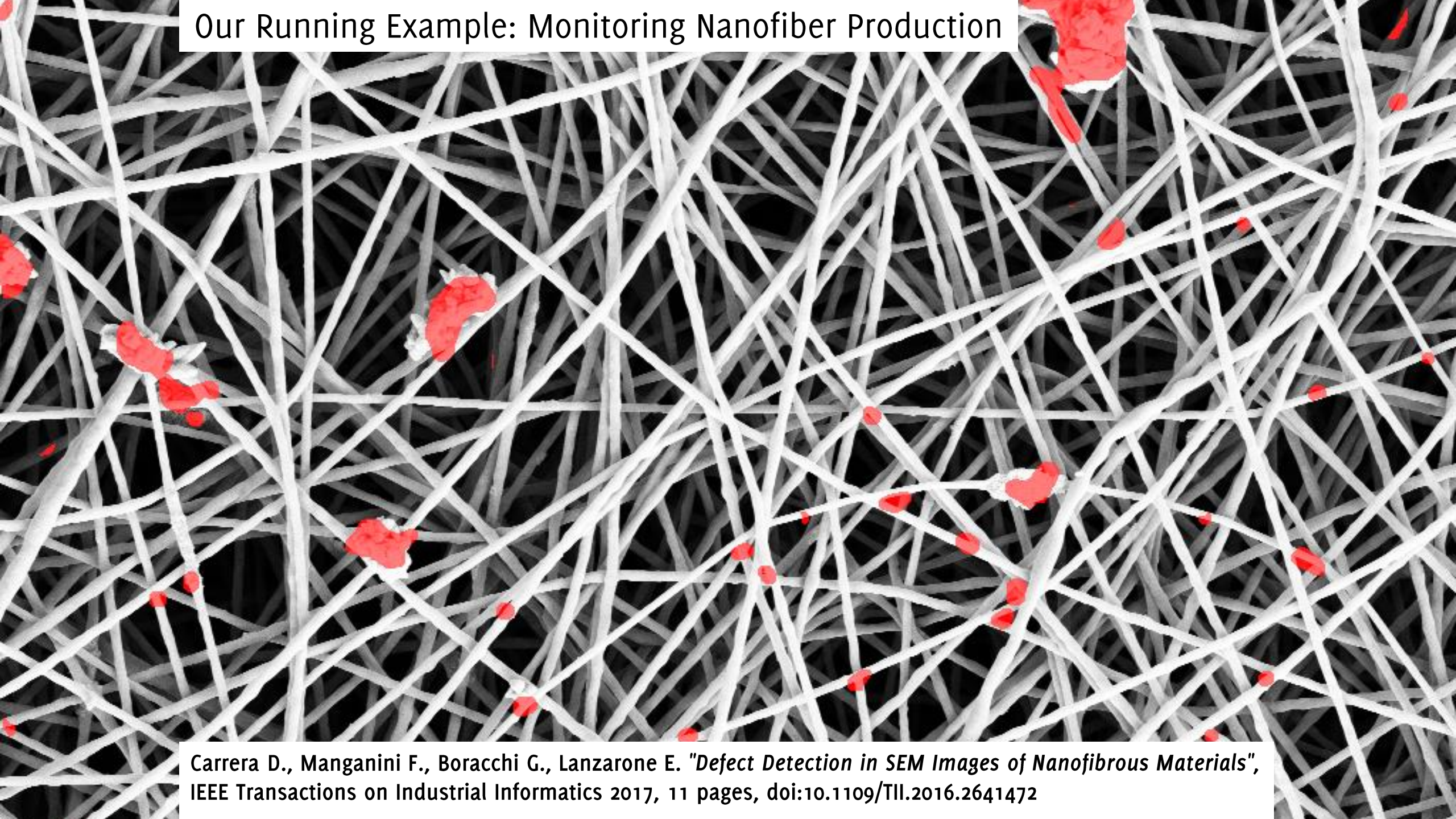
https://www.mvtec.com/company/research/datasets/mvtec-ad/

Our Running Example: Monitoring Nanofiber Production



Carrera D., Manganini F., Boracchi G., Lanzarone E. *"Defect Detection in SEM Images of Nanofibrous Materials"*, IEEE Transactions on Industrial Informatics 2017, 11 pages, doi:10.1109/TII.2016.2641472

Our Running Example: Monitoring Nanofiber Production



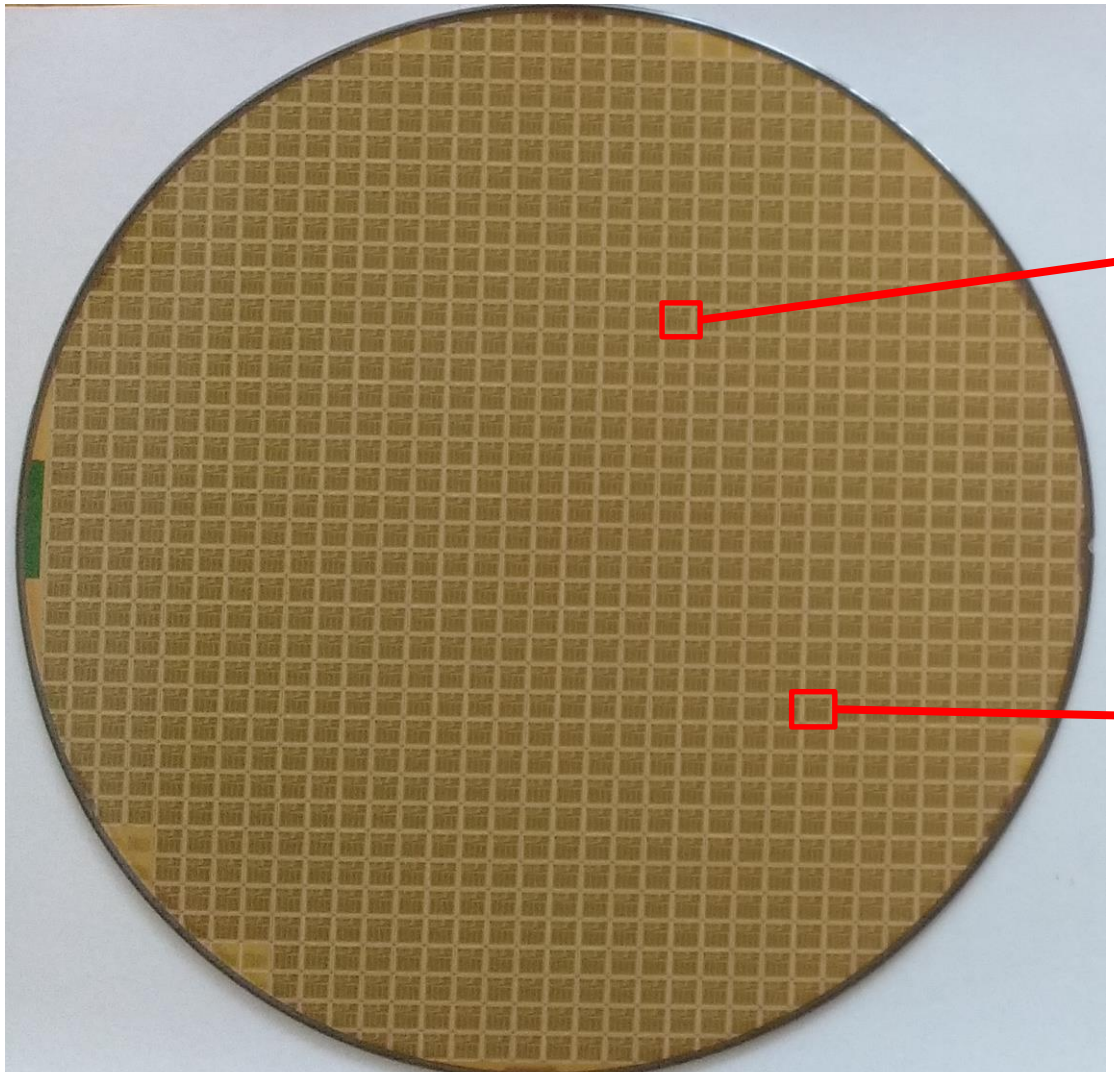
Carrera D., Manganini F., Boracchi G., Lanzarone E. *"Defect Detection in SEM Images of Nanofibrous Materials"*, IEEE Transactions on Industrial Informatics 2017, 11 pages, doi:10.1109/TII.2016.2641472

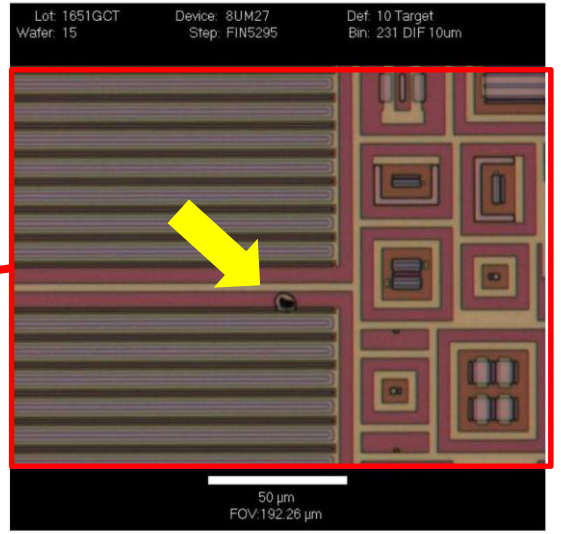
SYLICON WAFER MANUFACTURING

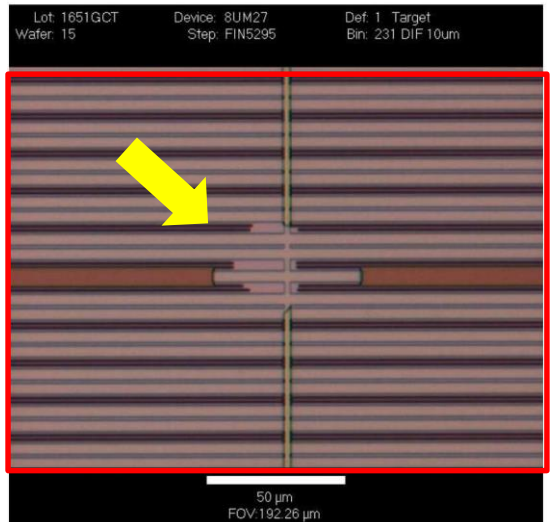
In collaboration with

life.augmented

Defects detected as anomalies in microscope images







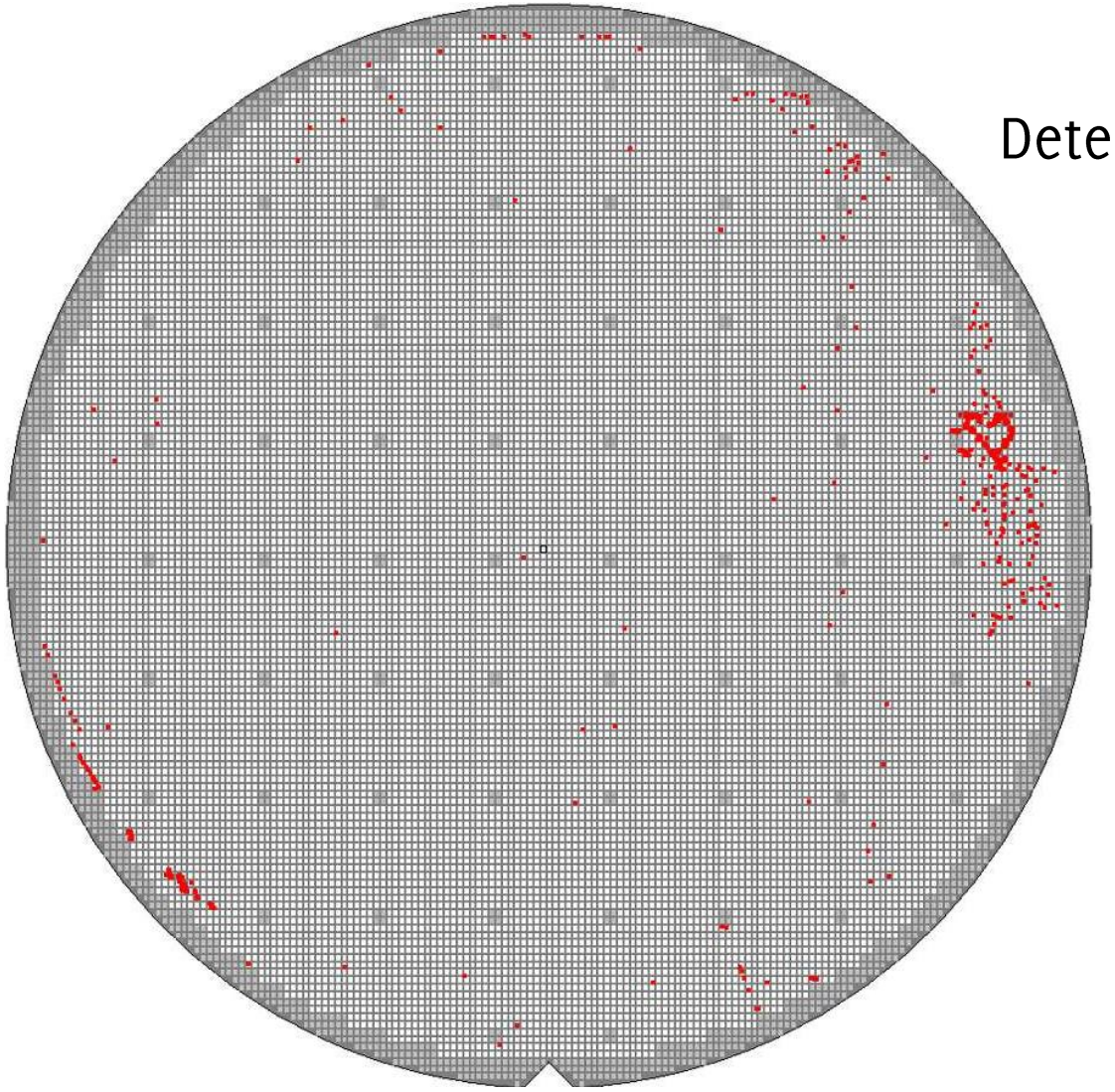
Boracchi, 2022

Not only images

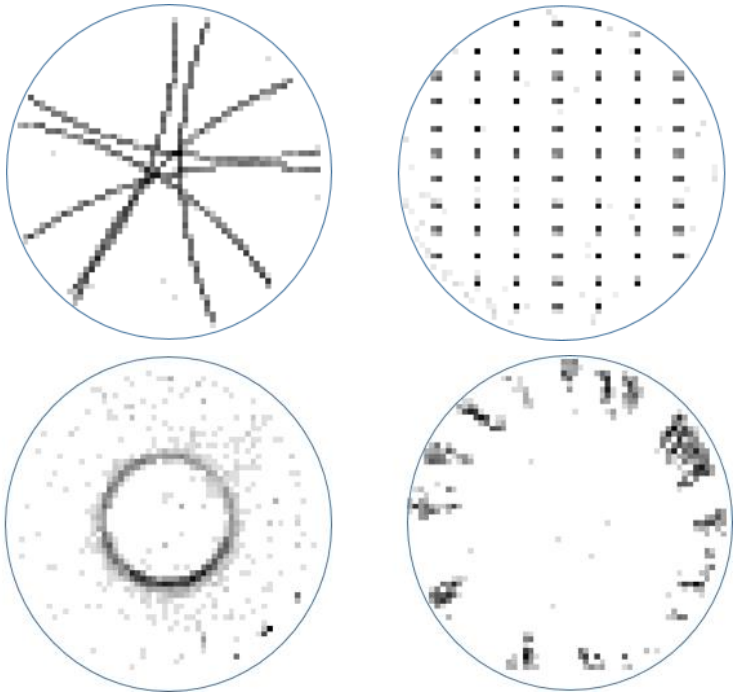
DETECTION OF ANOMALOUS PATTERNS

In collaboration with

life.augmented



Detect/Identify **patterns** in **wafer defect maps**
These might indicate faults,
problems or malfunctioning
in the chip production.

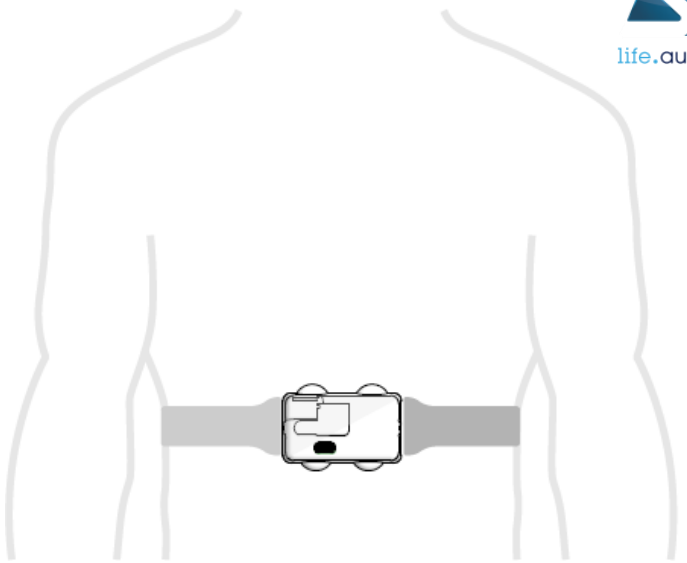


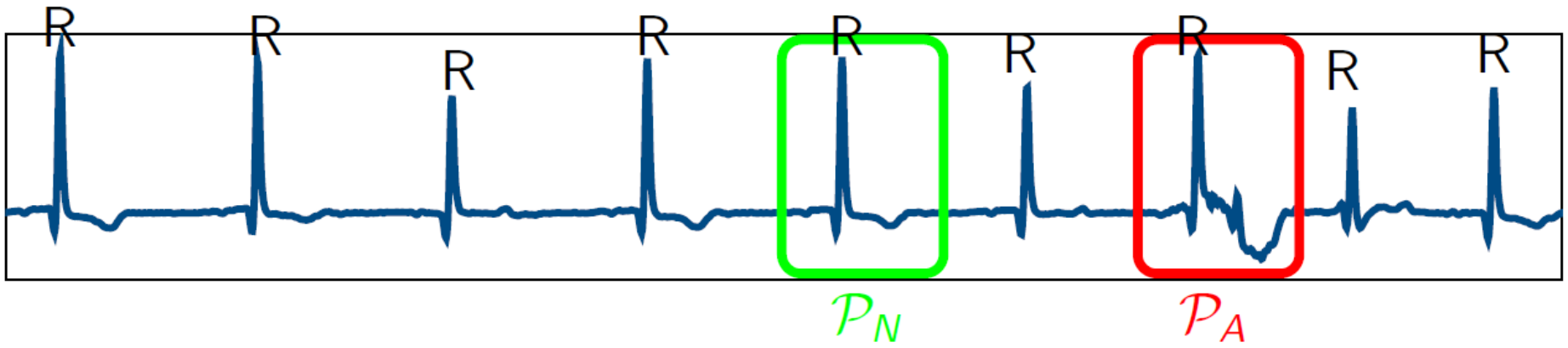
Di Bella, Carrera, Rossi, Fragneto, Boracchi *Wafer Defect Map Classification Using Sparse Convolutional Neural Networks* ICIAP 2009
Frittoli, Carrera, Rossi, Fragneto, Boracchi, *Deep Open-Set Recognition for Silicon Wafer Production Monitoring* Pattern Recognition 2021

AUTOMATIC AND LONG TERM EGC MONITORING

Health monitoring / wearable devices:

Automatically analyze EGC tracings to detect arrhythmias or incorrect device positioning





Carrera D., Rossi B., Zambon D., Fragneto P., Boracchi G. "*ECG Monitoring in Wearable Devices by Sparse Models*" in Proceedings of ECML-PKDD 2016, 16 pages

Longoni M., Carrera D., Rossi B., Fragneto P., Pessione M., Boracchi G. *"A Wearable Device for Online and Long-Term ECG Monitoring"* IJCAI 2018 (Demo track)

Carrera D., Rossi B., Fragneto P., Boracchi G. "*Online Anomaly Detection for Long-Term ECG Monitoring using Wearable Devices*", Pattern Recognition, Elsevier 2019

ANOMALOUS ACTIVITIES DETECTION IN VIDEOS

USCD Anomaly Dataset http://www.svcl.ucsd.edu/projects/anomaly/dataset.html

DEEP LEARNING MODELS FOR ANOMALY DETECTION

PRESENTATION OUTLINE

Part 1, Problem Formulation and Statistical Approaches:

- Problem formulation
- Performance measures
- Most relevant approaches

Part 2, Anomaly detection in images:

- The general approach
- Subspace / Feature extraction methods
- Reference-based methods

Part 3, Anomaly detection by deep learning models:

- Transfer Learning / Self-supervised
- Autoencoders
- Domain based, Generative models

DISCLAIMERS

We will refer to **either anomalies/changes** according to **our personal experience** with a primary focus on the applications we have been addressing.

For a **complete overview** on anomaly algorithms please refer to surveys below.

We will consider **unsupervised and semi-supervised approaches**, as these better conform with anomaly detection settings

Ruff, Lukas, et al. "A unifying review of deep and shallow anomaly detection." Proceedings of the IEEE (2021).

T. Ehret, A. Davy, JM Morel, M. Delbracio *"Image Anomalies: A Review and Synthesis of Detection Methods"*, Journal of Mathematical Imaging and Vision, 1-34

V. Chandola, A. Banerjee, V. Kumar. *"Anomaly detection: A survey"*. ACM Comput. Surv. 41, 3, Article 15 (July 2009), 58 pages.

Pimentel, M. A., Clifton, D. A., Clifton, L., Tarassenko, L. *"A review of novelty detection"* Signal Processing, 99, 215-249 (2014)

A. Zimek, E. Schubert, H.P. Kriegel. *"A survey on unsupervised outlier detection in high-dimensional numerical data"* Statistical Analysis and Data Mining: The ASA Data Science Journal, 5(5), 2012.

THE PROBLEM FORMULATION

Anomaly Detection in Images

ANOMALIES

"Anomalies are patterns in data that do not conform to a well defined notion of normal behavior"

Thus:

- **Normal data** are generated from a **stationary process** $\mathcal{P}_N$
- **Anomalies** are from a **different process** $\mathcal{P}_A \neq \mathcal{P}_N$

Examples:

- **Frauds** in the stream of all the credit card transactions
- **Arrhythmias** in ECG tracings
- **Defective regions in an image**, which do not conform a reference pattern

Anomalies might appear as **spurious** elements, and are typically the **most informative** samples in the stream

V. Chandola, A. Banerjee, V. Kumar. *"Anomaly detection: A survey"*. ACM Comput. Surv. 41, 3, Article 15 (July 2009), 58 pages.

PROBLEM FORMULATION: ANOMALY DETECTION IN IMAGES

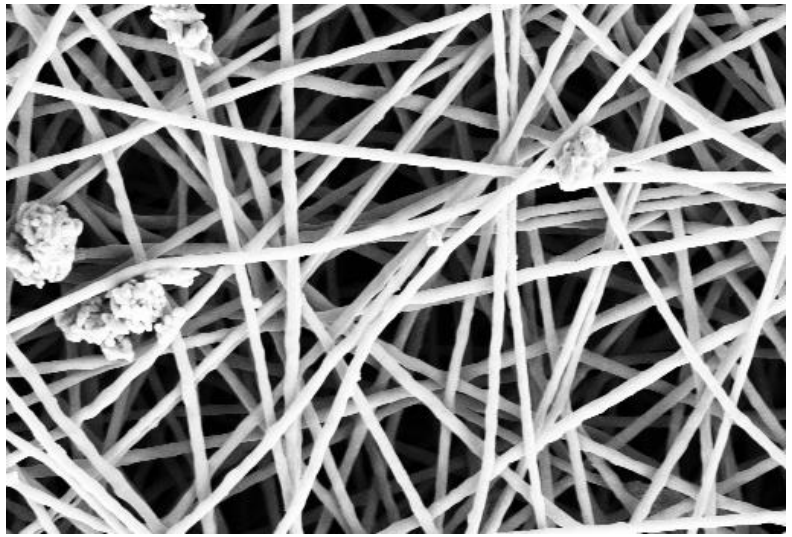
Let s be an image defined over the pixel domain $\mathcal{X} \subset \mathbb{Z}^2$,

let $c \in \mathcal{X}$ be a pixel and $s(c)$ the corresponding intensity.

Our goal is to **locate any anomalous region** in s, i.e. **estimating the unknown anomaly mask $\boldsymbol{\Omega}$** defined as

$$\Omega(c) = \begin{cases} 0 & \text{if } c \text{ falls in a normal region} \\ 1 & \text{if } c \text{ falls in an anomalous region} \end{cases}$$

s



PROBLEM FORMULATION: ANOMALY DETECTION IN IMAGES

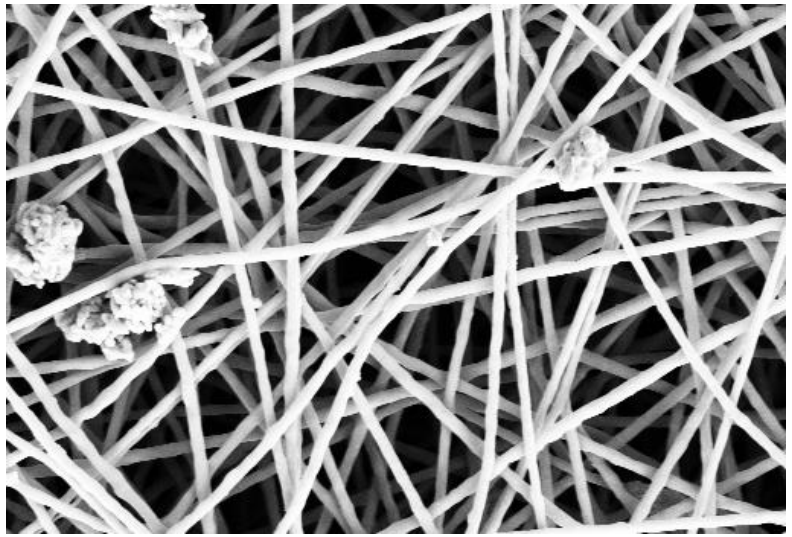
Let s be an image defined over the pixel domain $\mathcal{X} \subset \mathbb{Z}^2$,

let $c \in \mathcal{X}$ be a pixel and $s(c)$ the corresponding intensity.

Our goal is to **locate any anomalous region** in s, i.e. **estimating the unknown anomaly mask $\boldsymbol{\Omega}$** defined as

$$\Omega(c) = \begin{cases} 0 & \text{if } c \text{ falls in a normal region} \\ 1 & \text{if } c \text{ falls in an anomalous region} \end{cases}$$

s



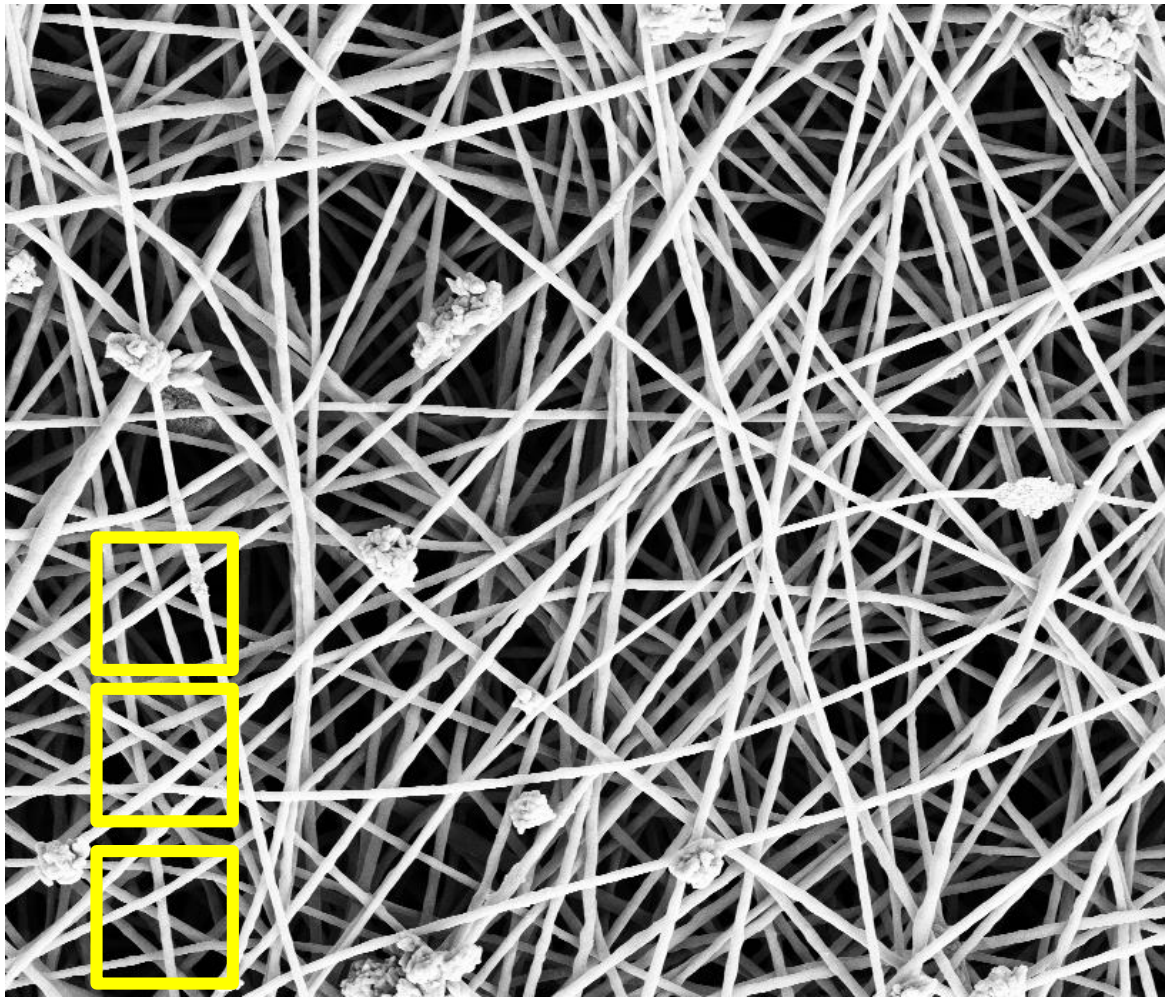
Ω

PATCH-WISE ANOMALY DETECTION

The goal is not determining whether the whole image is normal or anomalous, but **locate/segment possible anomalies**

Therefore, it is convenient to

1. **Analyze the image patch-wise**
2. Isolate regions containing patches that are detected as as anomalies

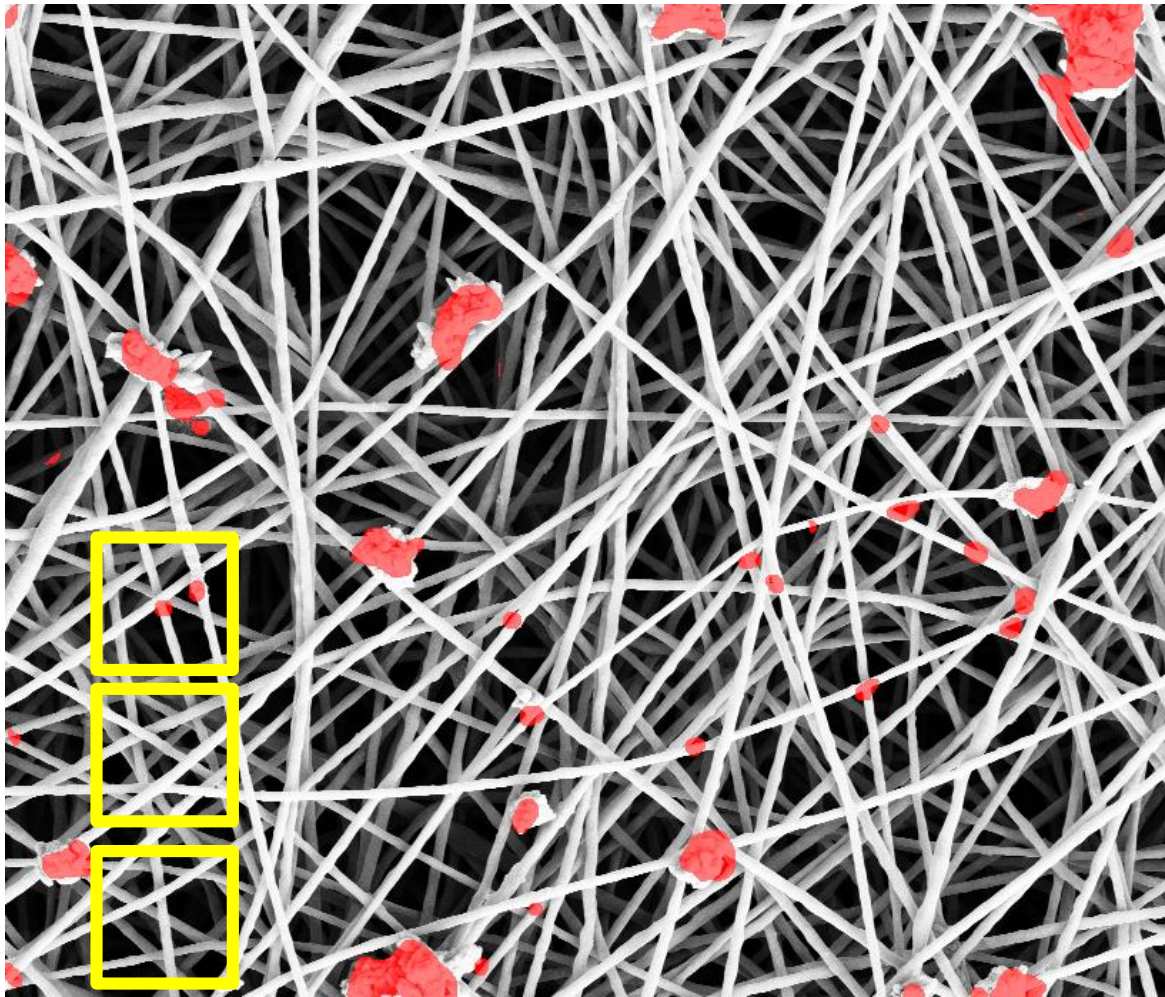


PATCH-WISE ANOMALY DETECTION

The goal is not determining whether the whole image is normal or anomalous, but **locate/segment possible anomalies**

Therefore, it is convenient to

1. **Analyze the image patch-wise**
2. Isolate regions containing patches that are detected as as anomalies

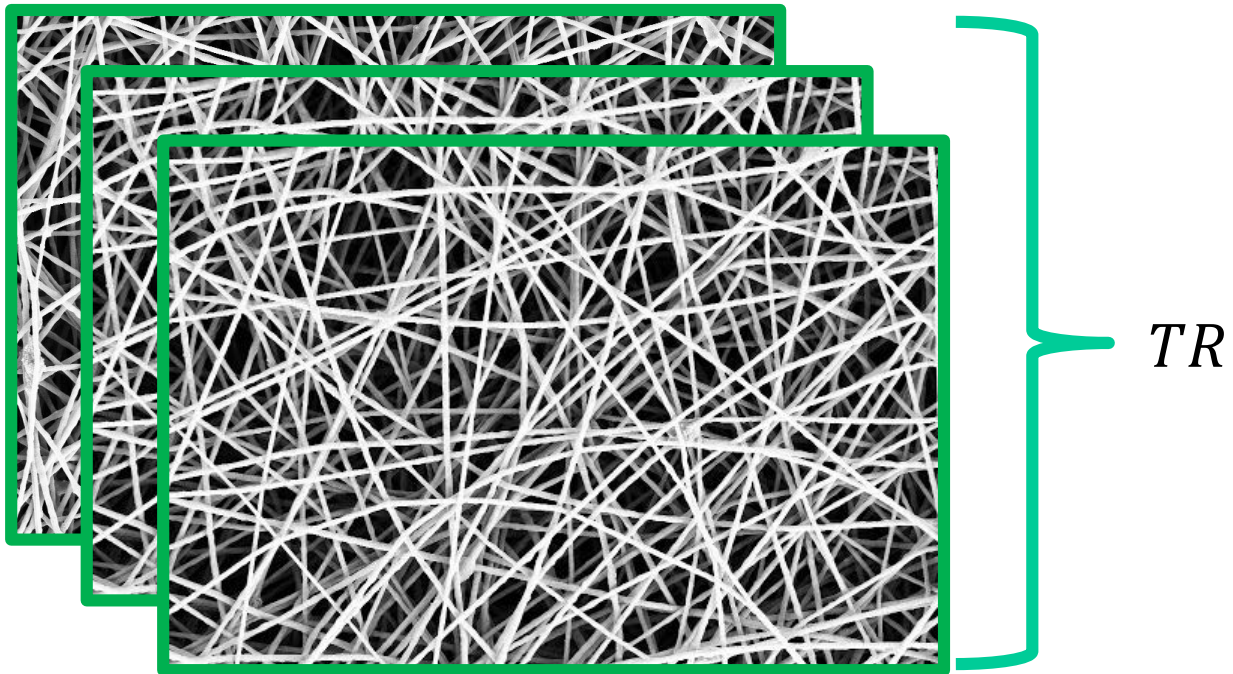


TYPICAL ASSUMPTIONS

A training set TR is provided for configuring the AD algorithm

Depending on the algorithm

- Only normal images -> semi-supervised methods
- Unlabeled images -> unsupervised methods
- Annotated images -> supervised methods



WHY SHOULDN'T WE GO FOR SUPERVISED LEARNING?

… indeed, labels are provided at least for testing….

In standard AD settings, ***labeled anomalous*** *data are often nonexistent. When available, it is usually* ***insufficient to fully characterize all notions of anomalousness****. This typically makes a supervised approach ineffective. Instead, a central idea in AD is to* ***learn a model of normality*** *from normal data in an unsupervised manner so that* ***anomalies become detectable through deviations from the model****.*

Ruff, Lukas, et al. "A unifying review of deep and shallow anomaly detection." *Proceedings of the IEEE* (2021).

ANOMALY DETECTION IN STATISTICS

Anomaly Detection Problem where observations are i.i.d. realizations of a random variable

Forget about images and deep learning for a while

Let us address the fundamental problem of detecting anomalies in a set of random vectors

… these techniques will come **very handy also for images**

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

ANOMALY-DETECTION IN A STATISTICAL FRAMEWORK

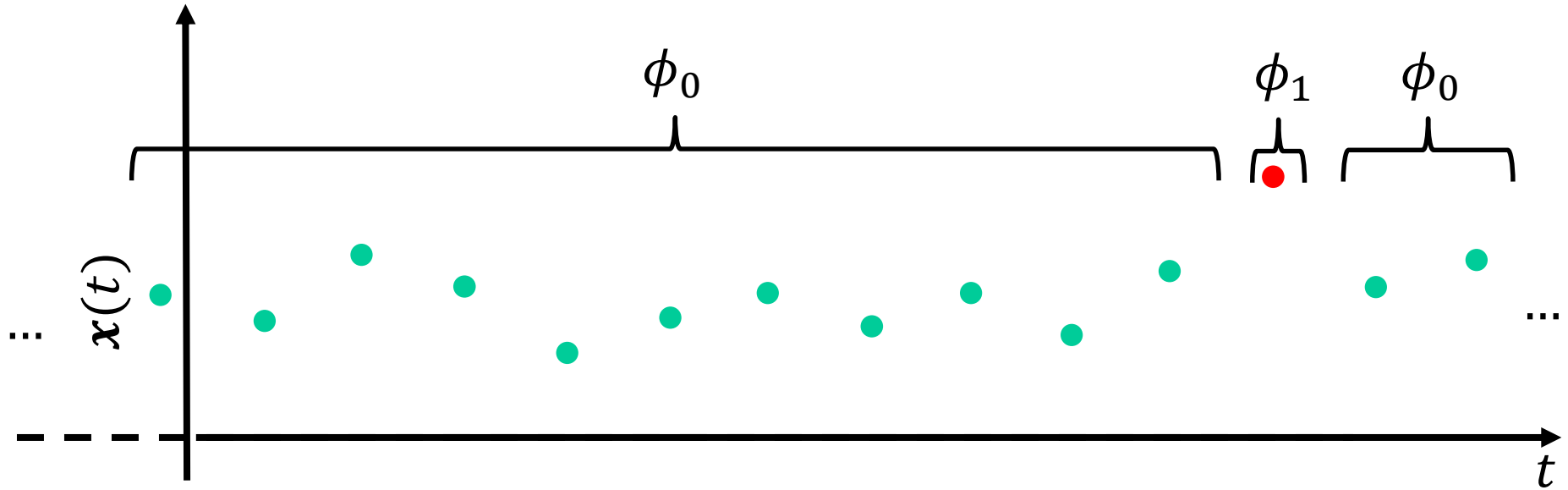
Often, the anomaly-detection problem **boils down to:**

Monitor a set of data (not necessarily a stream)

$$\{\boldsymbol{x}(t),\ t = t_0, \dots\},\quad \boldsymbol{x}(t) \in \mathbb{R}^d$$

where $x(t)$ are realizations of a **random variable having pdf** ϕ_o, and detect **outliers** i.e., those points that do not conform with ϕ_o

$$\boldsymbol{x}(t) \sim \begin{cases} \phi_0 & \text{normal data} \\ \phi_1 & \text{anomalies} \end{cases},$$



ANOMALY-DETECTION IN A STATISTICAL FRAMEWORK

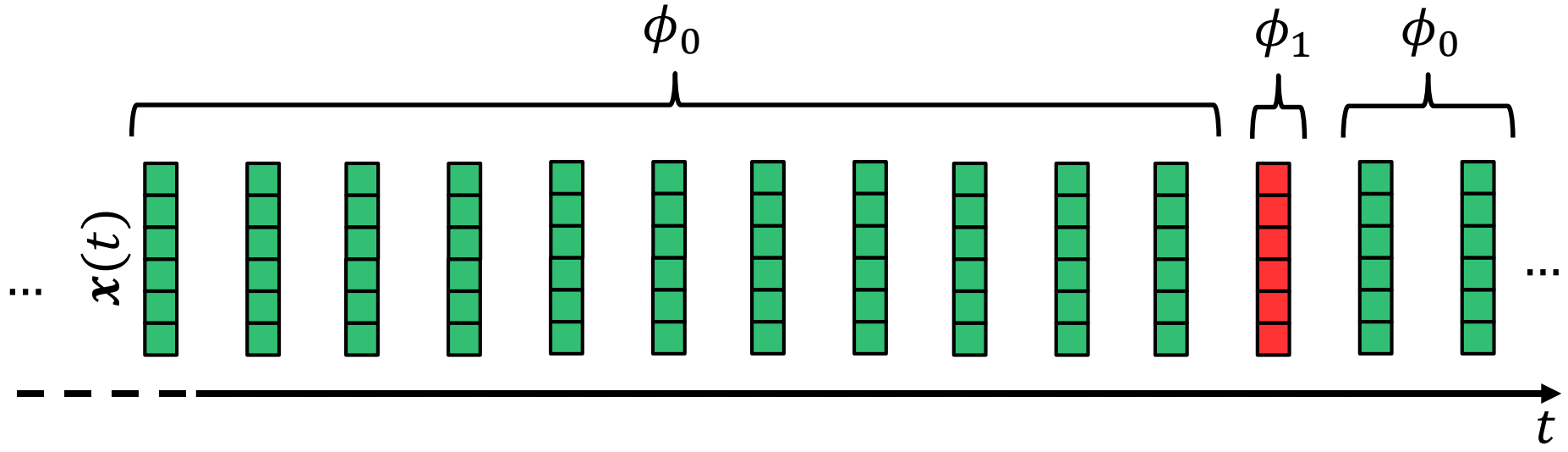
Often, the anomaly-detection problem **boils down to:**

Monitor a set of data (not necessarily a stream)

$$\{\boldsymbol{x}(t),\ t = t_0, \dots\},\ \ \boldsymbol{x}(t) \in \mathbb{R}^d$$

where $x(t)$ are realizations of a **random variable having pdf** ϕ_o, and detect **outliers** i.e., those points that do not conform with ϕ_o

$$\boldsymbol{x}(t) \sim \begin{cases} \phi_0 & \text{normal data} \\ \phi_1 & \text{anomalies} \end{cases},$$



ANOMALY-DETECTION IN A STATISTICAL FRAMEWORK

Often, the anomaly-detection problem **boils down to**:

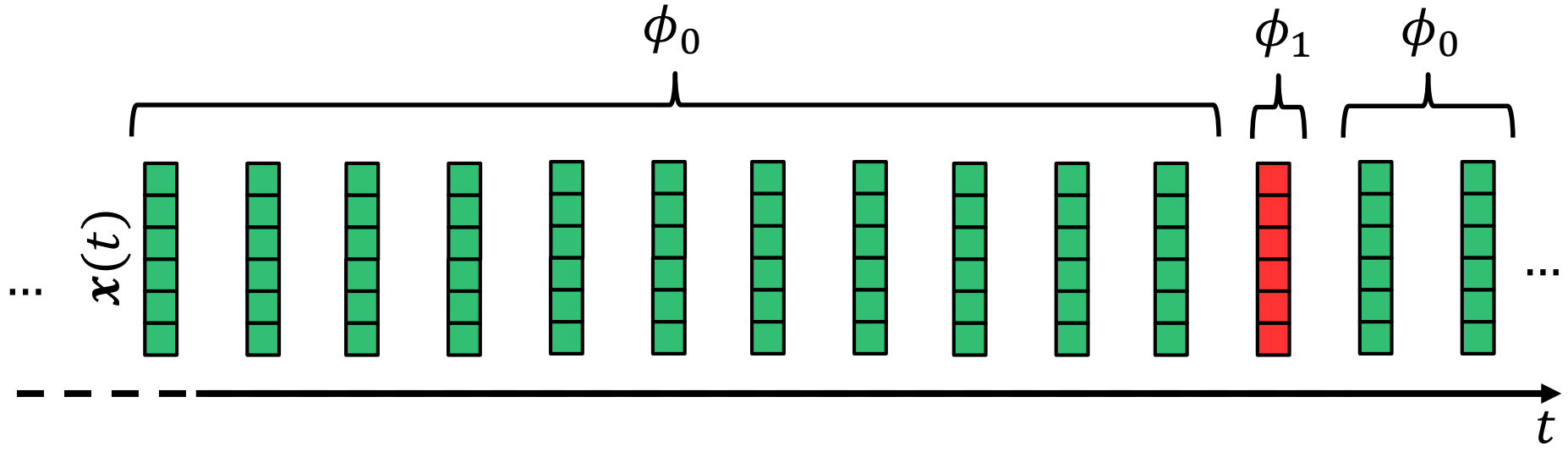
Monitor a set of data (not necessarily a stream)

$$\{\boldsymbol{x}(t),\ t = t_0, \dots\},\quad \boldsymbol{x}(t) \in \mathbb{R}^d$$

where $\boldsymbol{x}(t)$ are realizations of a random variable having pdf $f_{\boldsymbol{x}}$ and detect **outliers** i.e., those

*Locate those **samples that do not conform the normal ones** or a **model explaining normal ones***

$$\boldsymbol{x}(t) \sim \begin{cases} \phi_0 \\ \phi_1 \end{cases}$$ anomalies ,



STATISTICAL APPROACHES

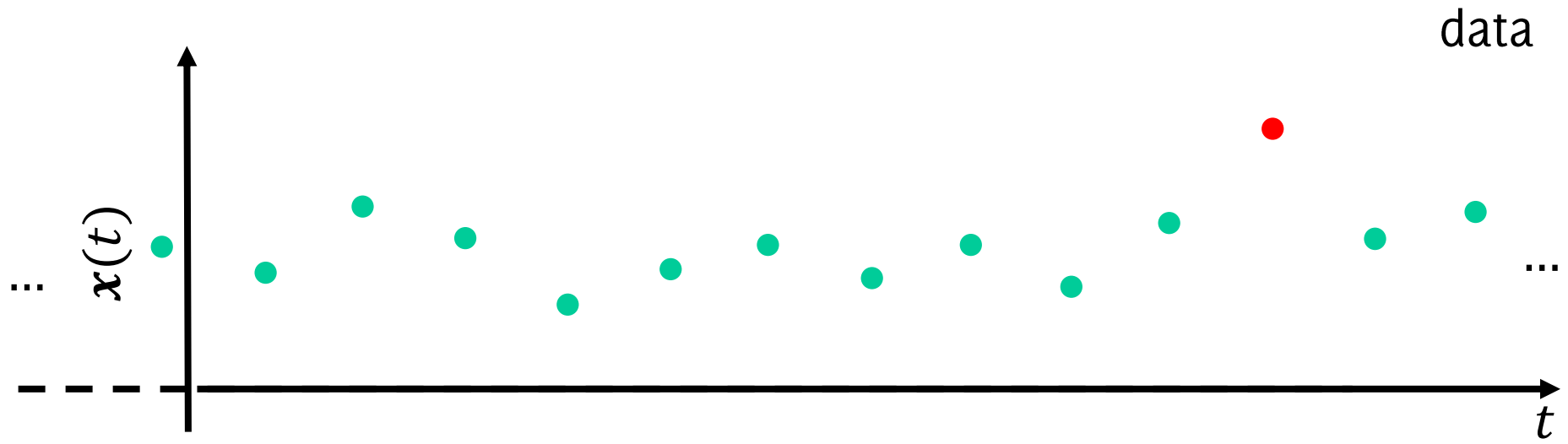
..to detect anomalies

THE TYPICAL SOLUTIONS

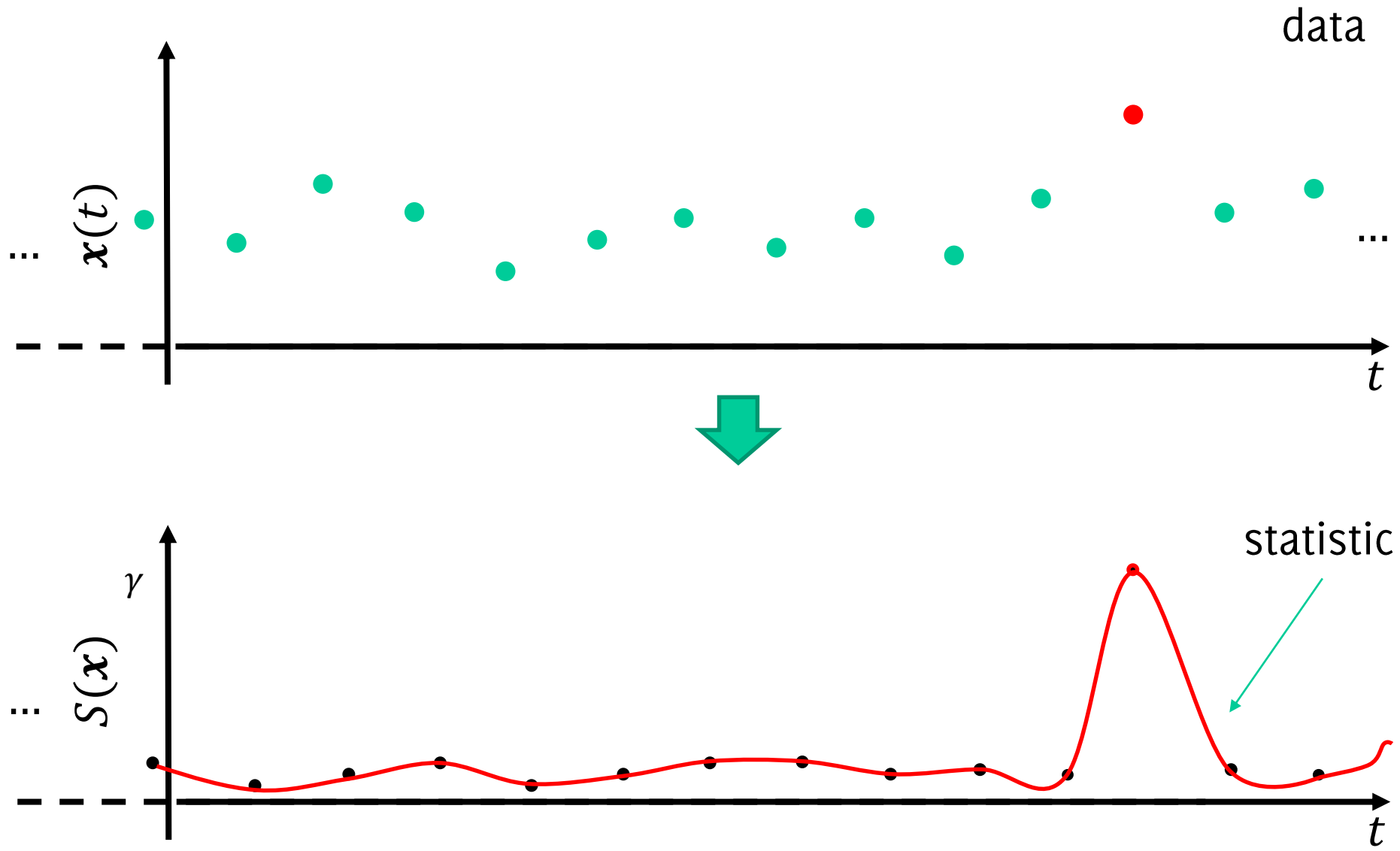
Most algorithms are composed of:

- A **statistic** that has a known response to normal data (e.g., the average, the sample variance, the log-likelihood, the confidence of a classifier, an “anomaly score”…)
- A **decision rule** to analyze the statistic (e.g., an adaptive threshold, a confidence region)

THE TYPICAL SOLUTIONS

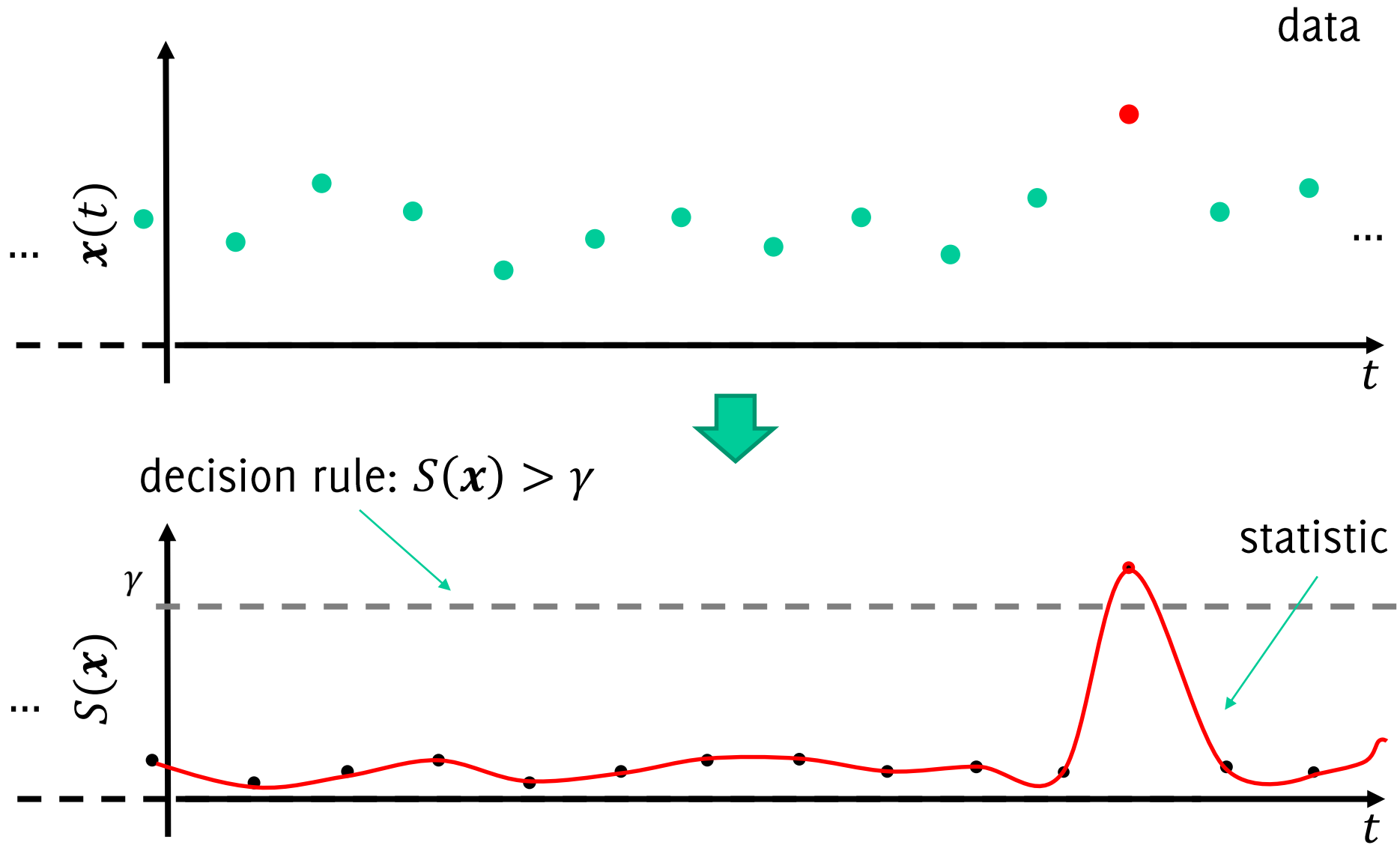


THE TYPICAL SOLUTIONS

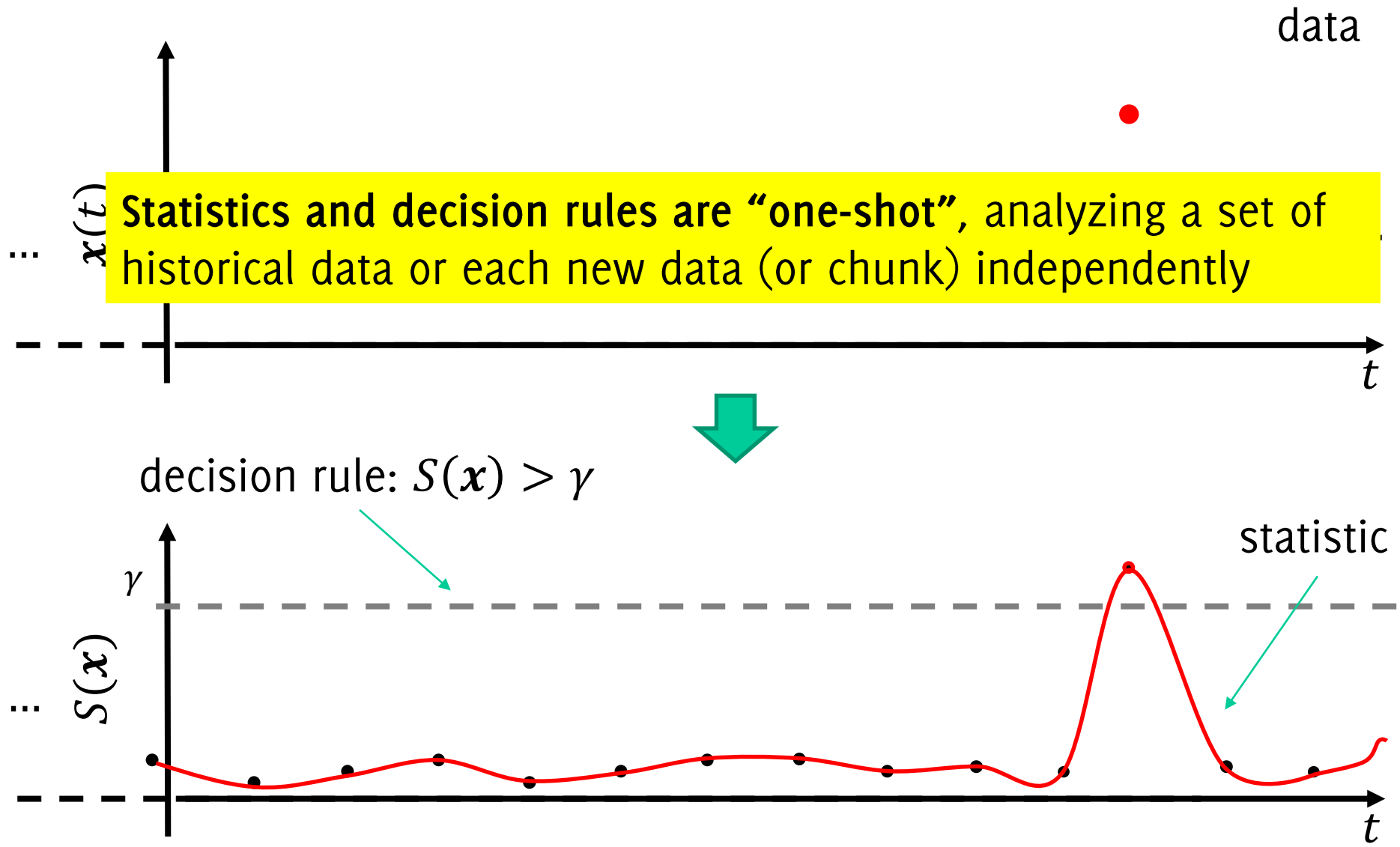


Boracchi, 2022

THE TYPICAL SOLUTIONS



THE TYPICAL SOLUTIONS



Statistics and decision rules are "one-shot", analyzing a set of historical data or each new data (or chunk) independently

PERFORMANCE MEASURES

Assessing performance of anomaly detection algorithms

ANOMALY-DETECTION PERFORMANCE

Anomaly detection performance:

- True positive rate: $TPR = \frac{\#\{\text{anomalies detected}\}}{\#\{\text{anomalies}\}}$
- False positive rate: $FPR = \frac{\#\{\text{normal samples detected}\}}{\#\{\text{normal samples}\}}$

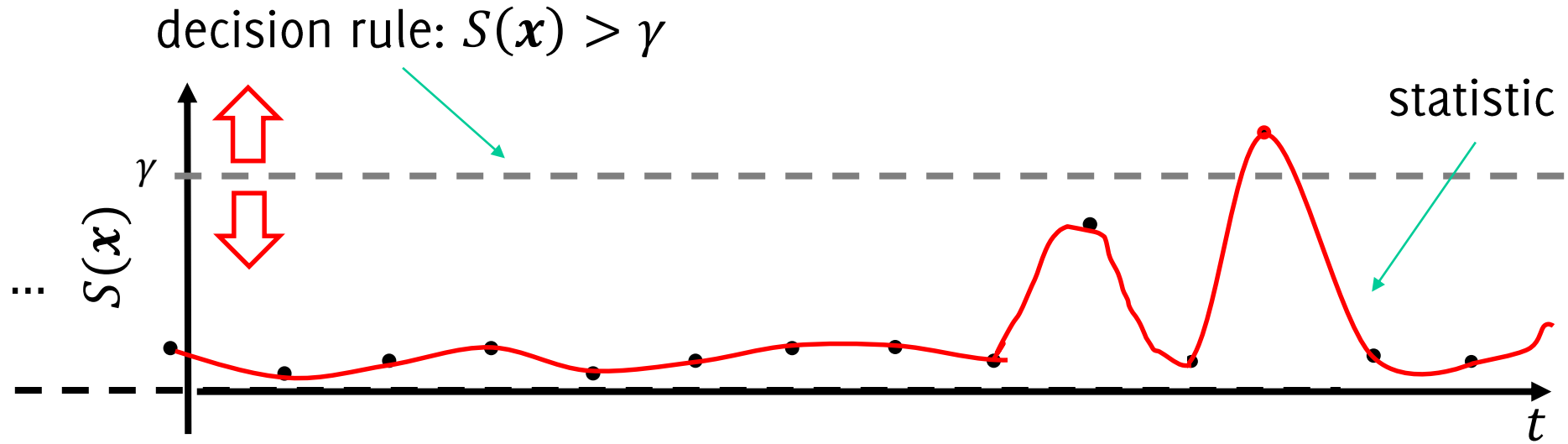
You have probably also heard of

- False negative rate (or miss-rate): $FNR = 1 - TPR$
- True negative rate (or specificity): $TNR = 1 - FPR$
- Precision on anomalies: $\frac{\#\{\text{anomalies detected}\}}{\#\{\text{detections}\}}$
- Recall on anomalies (or sensitivity, hit-rate): TPR

TPR / FPR TRADE-OFF

There is always a **trade-off between TPR and FPR** (and similarly for derived quantities), which is ruled by algorithm parameters

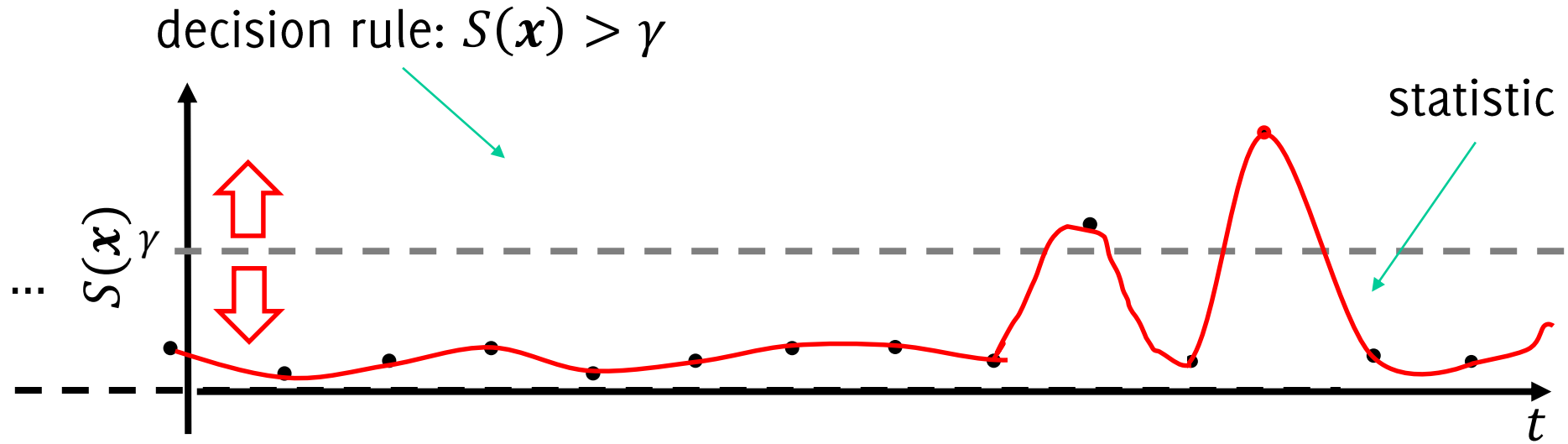
By changing γ performance changes (e.g. true positive increases but also false positives do)



TPR / FPR TRADE-OFF

There is always a **trade-off between TPR and FPR** (and similarly for derived quantities), which is ruled by algorithm parameters

By changing γ performance changes (e.g. true positive increases but also false positives do)



ANOMALY-DETECTION PERFORMANCE

There is always a **trade-off between TPR and FPR** (and similarly for derived quantities), which is ruled by algorithm parameters

Thus, to correctly assess performance it is necessary to consider at least **two indicators** (e.g., TPR, FPR)

Indicators combining both TPR and FPR:

$$\text{Accuracy} = \frac{\#\{\text{anomalies detected}\} + \#\{\text{normal samples not detected}\}}{\#\{\text{samples}\}}$$

$$\text{F1 score} = \frac{2\#\{\text{anomalies detected}\}}{\#\{\text{detections}\} + \#\{\text{anomalies}\}}$$

These equal 1 in case of “ideal detector” which detects all the anomalies and has no false positives

ANOMALY-DETECTION PERFORMANCE

Comparing different methods might be tricky since we have to make sure that both have been configured in their best conditions

Testing a large number of parameters lead to the **ROC** (receiver operating characteristic) **curve**

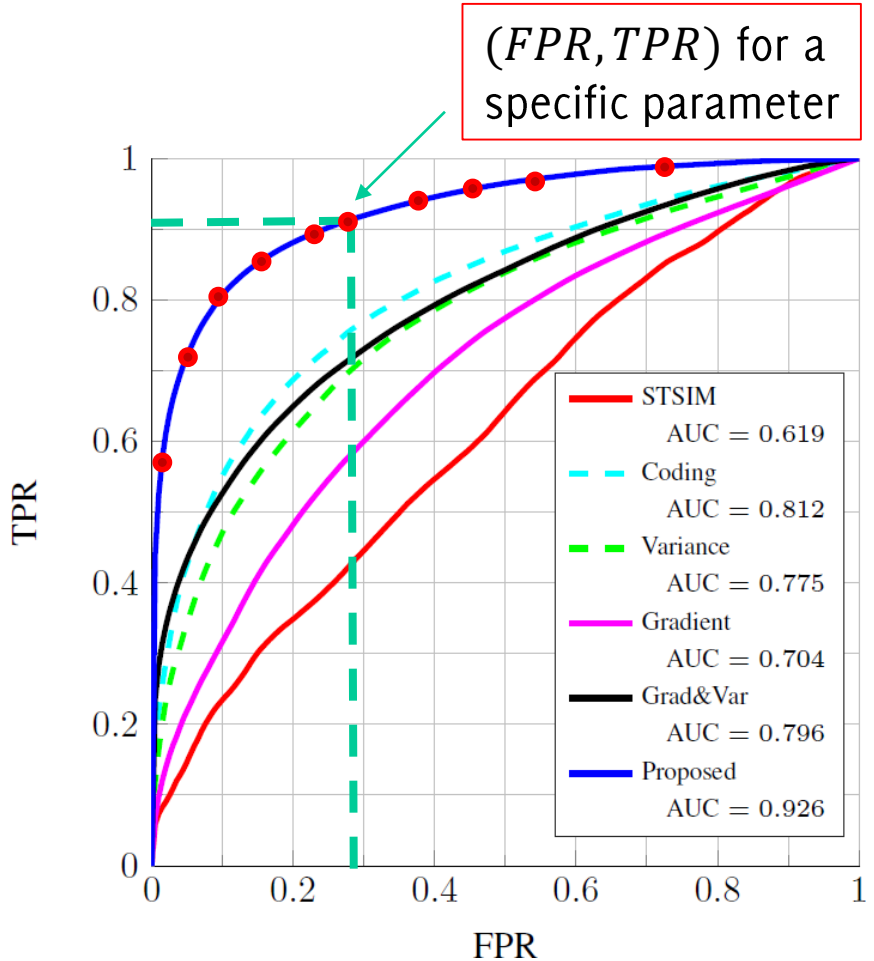
The ideal detector would achieve:

- $FPR = 0\%$,
- $TPR = 100\%$

Thus, the closer to (0,1) the better

The largest the **Area Under the Curve** (AUC), the better

The optimal parameter is the one yielding the point closest to (0,1)



Boracchi, 2022

STATISICAL APPROACHES TO DETECT ANOMALIES

...when ϕ_0 and ϕ_1 are unknown

ANOMALY DETECTION WHEN ϕ_0 AND ϕ_1 ARE UNKNOWN

Most often, **only a training set** TR is **provided**:

There are three scenarios:

- **Semi-Supervised:** Only normal training data are provided, i.e. no anomalies in TR.
- **Unsupervised:** TR is provided without label.
- **Supervised:** Both normal and anomalous training data are provided in TR.

ANOMALY DETECTION WHEN ϕ_0 AND ϕ_1 ARE UNKNOWN

Most often, **only a training set** TR is **provided**:

There are three scenarios:

- **Semi-Supervised:** Only normal training data are provided, i.e. no anomalies in TR.
- **Unsupervised:** TR is provided without label.
- **Supervised:** Both normal and anomalous training data are provided in TR.

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

SEMI-SUPERVISED ANOMALY DETECTION

In semi-supervised methods the TR is composed of normal data

$$TR = \{x(t), x \sim \phi_0 \text{ and } t < t_0, \}$$

Very practical assumptions:

- **Normal data** are **easy to gather** and the vast majority
- **Anomalous data** are **difficult**/costly **to collect**/select and it would be **difficult to gather a representative training set**
- Training examples in TR might not be **representative of all the possible anomalies** that can occur

All in all, it is often **safer** to **detect any data departing from** the **normal conditions**

Semi-supervised anomaly-detection methods are also referred to as **novelty-detection methods**

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

Pimentel, M. A., Clifton, D. A., Clifton, L., Tarassenko, L. *"A review of novelty detection"* Signal Processing, 99, 215-249 (2014)

DENSITY-BASED METHODS

Density-Based Methods: ***Normal*** *data occur in* ***high probability regions*** *of a stochastic model, while* ***anomalies*** *occur in the* ***low probability regions*** *of the model*

During training: $\hat{\phi}_0$ can be **estimated** from the training set

$$TR = \{x(t), x \sim \phi_0 \text{ and } t < t_0, \}$$

- parametric models (e.g., Gaussian mixture models)
- nonparametric models (e.g. KDE, histograms)

During testing:

- Anomalies are detected as data yielding $\hat{\phi}_0(\boldsymbol{x}) < \eta$

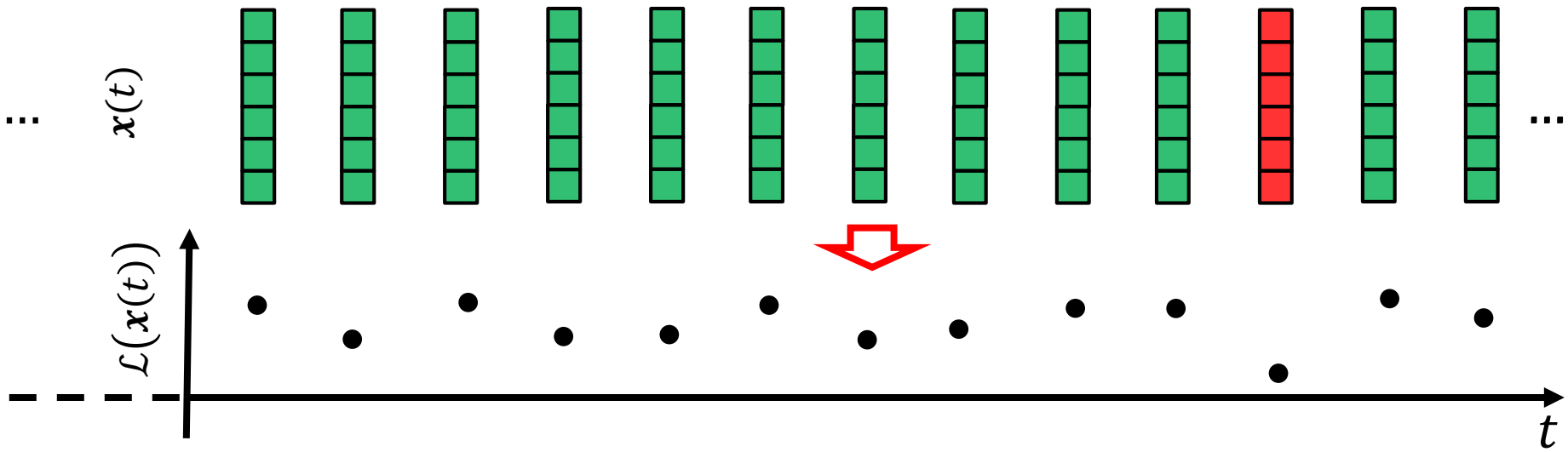
DENSITY-BASED METHODS: MONITORING THE LOG-LIKELIHOOD

Monitoring the log-likelihood of data w.r.t $\hat{\phi}_0$ allow to address anomaly-detection problems in multivariate data

1. During training, estimate $\hat{\phi}_0$ from TR
2. During testing, compute

$$\mathcal{L}\big(\boldsymbol{x}(t)\big) = \log(\hat{\phi}_0(\boldsymbol{x}(t)))$$

3. Monitor $\big\{\mathcal{L}\big(\boldsymbol{x}(t)\big),\ t = 1, \ldots\big\}$



DENSITY-BASED METHODS: MONITORING THE LOG-LIKELIHOOD

Monitoring the log-likelihood of data w.r.t $\hat{\phi}_0$ allow to address anomaly-detection problems in multivariate data

1. During training, estimate $\hat{\phi}_0$ from TR
2. During testing, compute

$$\mathcal{L}\big(\boldsymbol{x}(t)\big) = \log(\hat{\phi}_0(\boldsymbol{x}(t)))$$

3. Monitor $\big\{\mathcal{L}\big(\boldsymbol{x}(t)\big),\ t = 1, \ldots\big\}$

This is quite a popular approach in either anomaly and change detection algorithms

L. I. Kuncheva, *"Change detection in streaming multivariate data using likelihood detectors,"* IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 5, 2013.
X. Song, M. Wu, C. Jermaine, and S. Ranka, *"Statistical change detection for multidimensional data,"* in Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD), 2007.
J. H. Sullivan and W. H. Woodall, *"Change-point detection of mean vector or covariance matrix shifts using multivariate individual observations,"* IIE transactions, vol. 32, no. 6, 2000.
C. Alippi, G. Boracchi, D. Carrera, M. Roveri, "*Change Detection in Multivariate Datastreams: Likelihood and Detectability Loss*" IJCAI 2016, New York, USA, July 9 - 13

DENSITY-BASED METHODS

Advantages:

- $\hat{\phi}_0(\boldsymbol{x})$ indicates how safe a detection is (like a p-value)
- If the density estimation process is robust to outliers, it is possible to tolerate few anomalous samples in TR
- in relatively small dimensions, you might use non-parametric models like histograms

Challenges:

- **It is challenging to fit models for high-dimensional data**
- Histograms traditionally suffer of **curse of dimensionality** when d increases
- Often the **1D histograms** of the marginals are monitored, **ignoring the correlations** among components

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

DOMAIN-BASED METHODS

Domain-based methods: ***Estimate a boundary around normal data****, rather than the density of normal data.*

A **drawback of density-estimation methods** is that they are meant to be accurate in high-density regions, while anomalies live in low-density ones.

One-Class SVM are domain-based methods defined by **the normal samples at the periphery of the distribution**.

Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., Platt, J. C. "*Support Vector Method for Novelty Detection*". In NIPS 1999 (Vol. 12, pp. 582-588).
Tax, D. M., Duin, R. P. "*Support vector domain description*". Pattern recognition letters, 20(11), 1191-1199 (1999)
Pimentel, M. A., Clifton, D. A., Clifton, L., Tarassenko, L. *"A review of novelty detection"* Signal Processing, 99, 215-249 (2014)

ONE-CLASS SVM (SCHÖLKOPF ET AL. 1999)

Idea: define boundaries by estimating a **binary function** f that **captures regions of the input space where density is higher**.

As in support vector methods, f is **defined in the feature space** F and **decision boundaries are defined by a few support vectors** (i.e., a few normal data).

Let $\psi(\boldsymbol{x})$ the feature associated to $\boldsymbol{x}$, f is defined as

$$f(\boldsymbol{x}) = \text{sign}(< w, \psi(\boldsymbol{x}) > -\rho)$$

Where the hyperplane parameters w, ρ are optimized to yield a **function that is positive on most training samples.** Thus in the feature space, normal points can be separated from the origin.

A linear separation in the feature space corresponds to a **variety of nonlinear boundaries in the space of $\boldsymbol{x}$**.

Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., Platt, J. C. "*Support Vector Method for Novelty Detection*". In NIPS 1999 (Vol. 12, pp. 582-588).

ONE-CLASS SVM (TAX AND DUIN 1999)

Boundaries of normal region can be also defined by an **hypersphere that, in the feature space, encloses most of the normal data** i.e., $\psi(\boldsymbol{x})$ for $\boldsymbol{x} \in TR$.

Similar detection formulas hold, measuring the **distance in the feature space from the sphere center**.

The sphere center can be defined in terms of support vectors.

Remarks: In both one-class approaches, the amount of samples that falls within the margin (outliers) is controlled by **regularization parameters**.

This parameter regulates the number of outliers in the training set and the detector sensitivity.

Tax, D. M., Duin, R. P. "*Support vector domain description*". Pattern recognition letters, 20(11), 1191-1199 (1999)

ANOMALY DETECTION WHEN ϕ_0 AND ϕ_1 ARE UNKNOWN

Most often, **only a training set** TR is **provided**:

There are three scenarios:

- **Semi-Supervised:** Only normal training data are provided, i.e. no anomalies in TR.
- **Unsupervised:** TR is provided without label.
- **Supervised:** Both normal and anomalous training data are provided in TR.

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

UNSUPERVISED ANOMALY-DETECTION

The training set TR might contain **both normal and anomalous data**. However, **no labels** are provided

$$TR = \{x(t), t < t_0\}$$

Underlying assumption: ***anomalies are rare*** *w.r.t. normal data* TR

In principle:

- Density/Domain based methods that are **robust to outliers** can be applied in an unsupervised scenario
- Unsupervised methods can be **improved whenever labels are available**

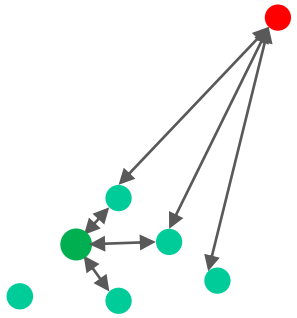
DISTANCE-BASED METHODS

Distance-based methods: ***normal*** *data fall in* ***dense neighborhoods****, while* ***anomalies*** *are* ***far from their closest neighbors****.*

A critical aspect is the **choice of the similarity measure** to use.

Anomalies are detected by **monitoring**:

- **distance** between each data and its k **–nearest neighbor**
- the **density** of each data **relatively to its neighbors**



V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages
Zhao, M., Saligrama, V. *"Anomaly detection with score functions based on nearest neighbor graphs"*. NIPS 2009
A. Zimek, E. Schubert, H. Kriegel. *"A survey on unsupervised outlier detection in high-dimensional numerical data"* SADM 2012

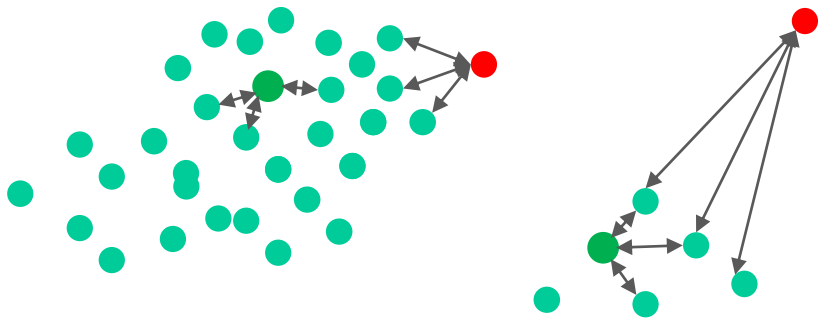
DISTANCE-BASED METHODS

Distance-based methods: ***normal*** *data fall in* ***dense neighborhoods****, while* ***anomalies*** *are* ***far from their closest neighbors****.*

A critical aspect is the **choice of the similarity measure** to use.

Anomalies are detected by **monitoring**:

- **distance** between each data and its k **–nearest neighbor**
- the **above distance** considered **relatively to neighbors**



V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

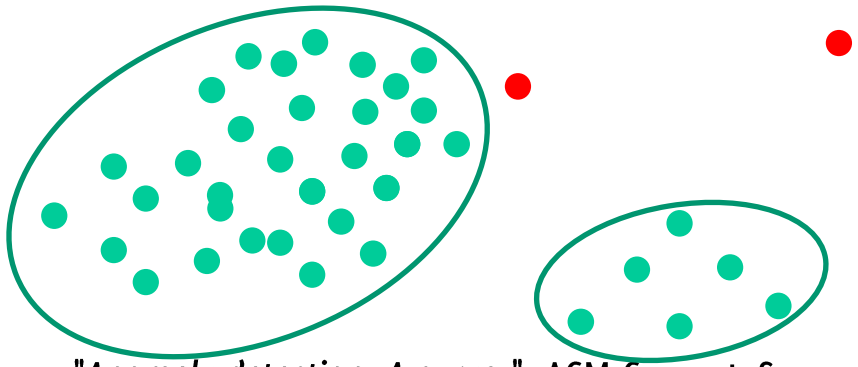
DISTANCE-BASED METHODS

Distance-based methods: ***normal*** *data fall in* ***dense neighborhoods****, while* ***anomalies*** *are* ***far from their closest neighbors****.*

A critical aspect is the **choice of the similarity measure** to use.

Anomalies are detected by **monitoring**:

- **distance** between each data and its k **–nearest neighbor**
- the **above distance** considered **relatively to neighbors**
- whether they do not belong to **clusters**, or are at the cluster periphery, or belong to small and sparse clusters

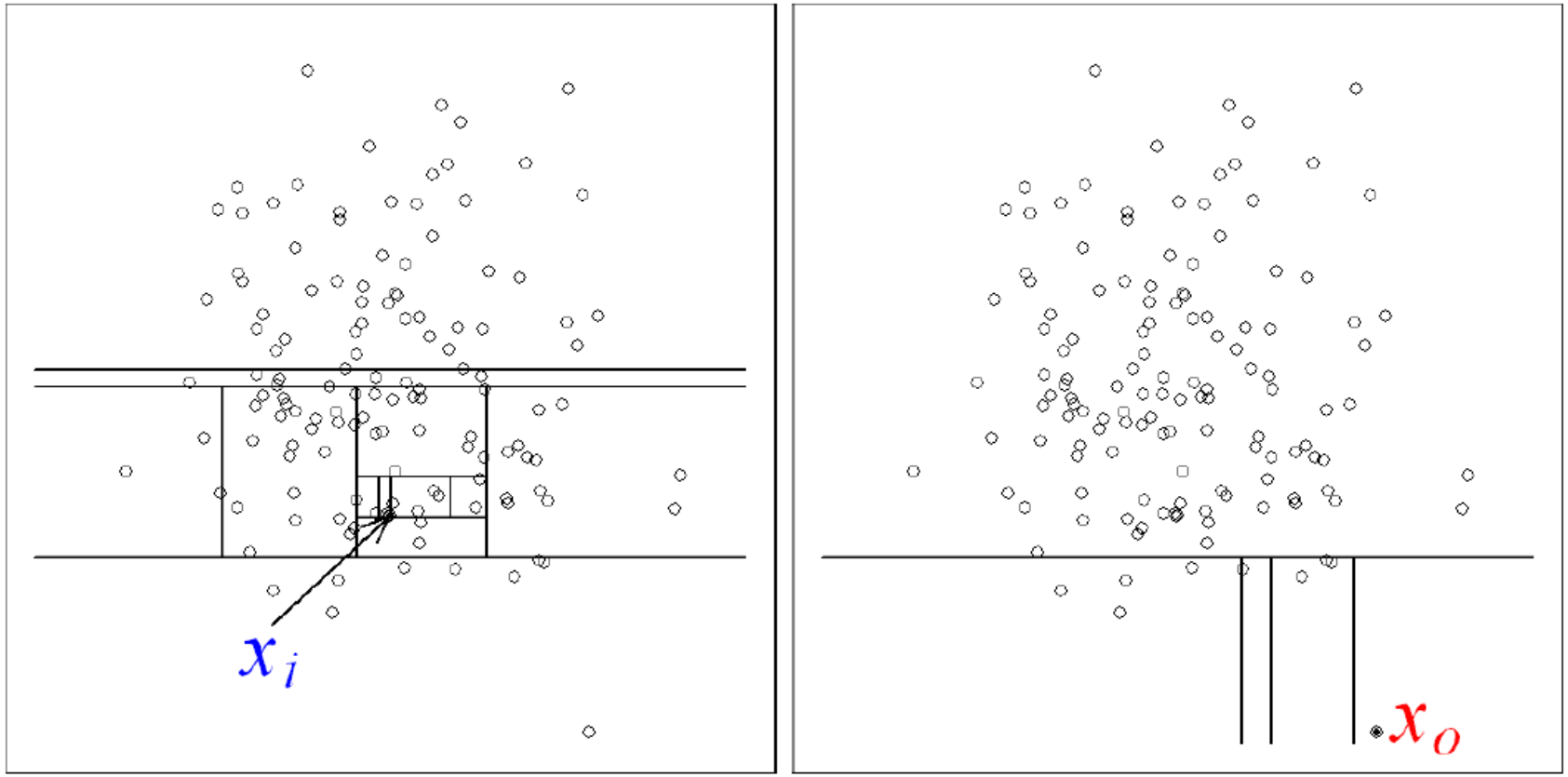


V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

ISOLATION FOREST

An anomalous point (x_0) can be easily isolated

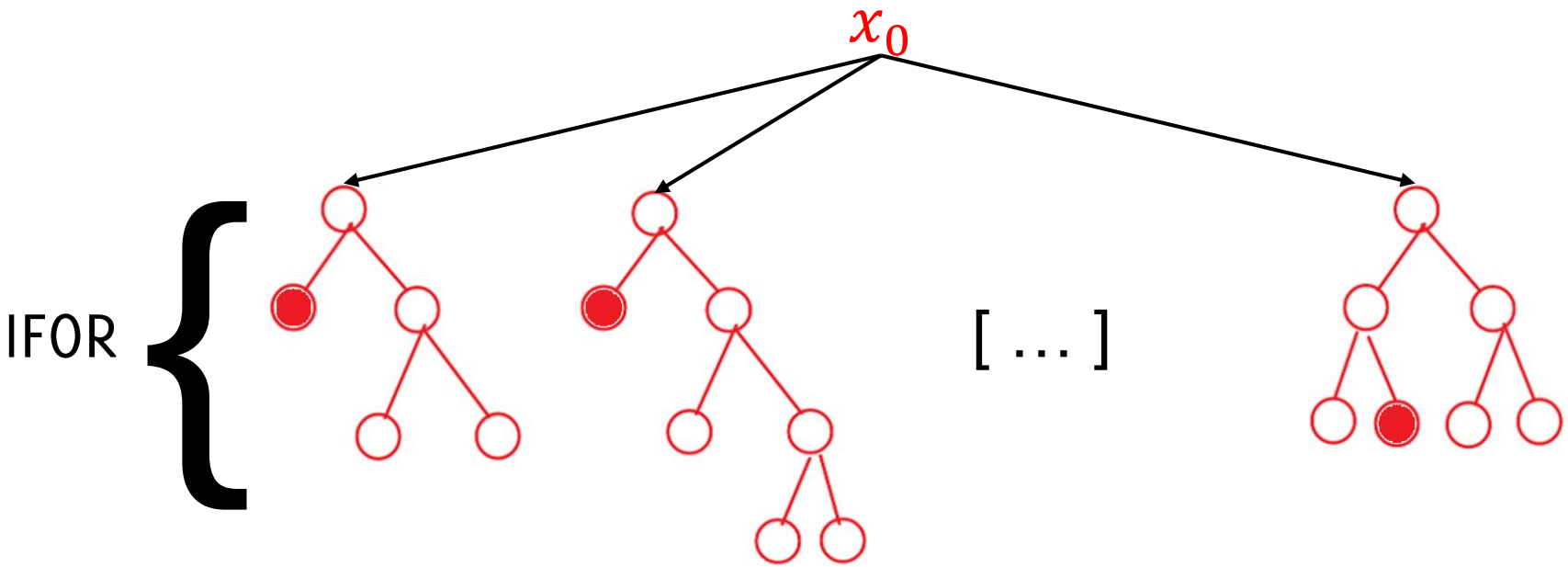
Genuine points (x_i) are instead difficult to isolate.



Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest*, *ICDM 2008*

ISOLATION FOREST

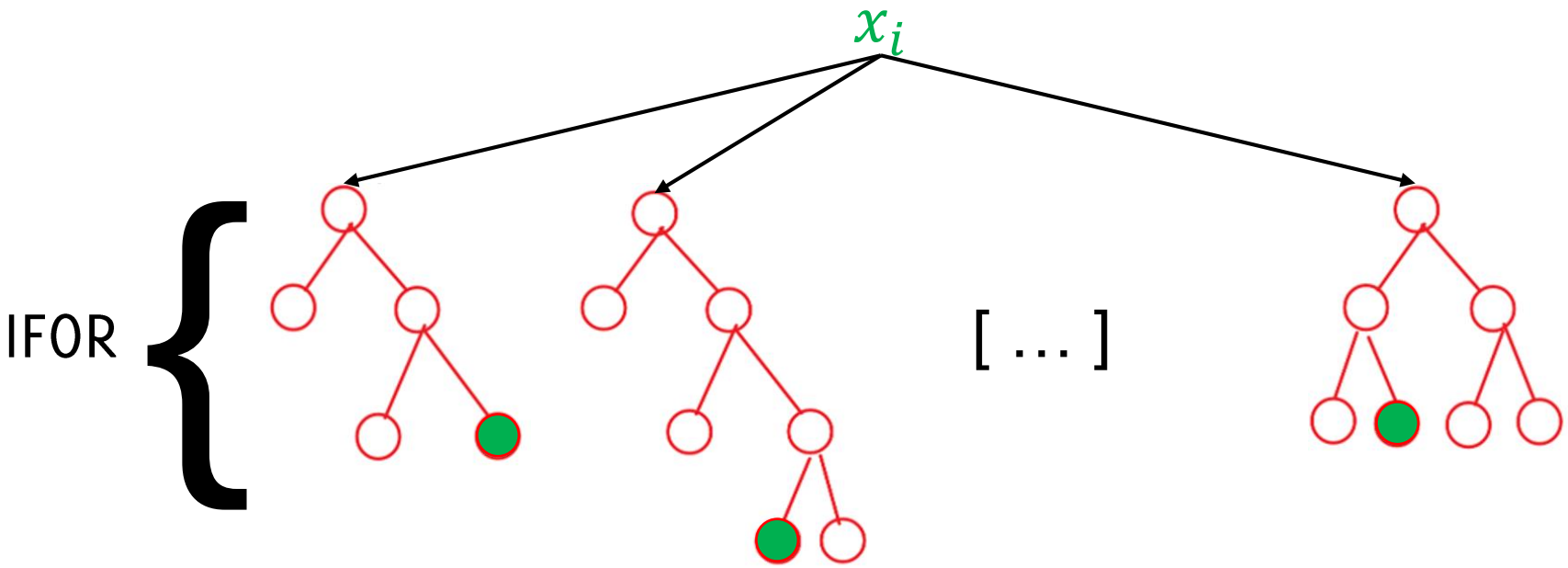
Anomalies



F. T. Liu, K. M. Ting and Z. Zhou, *Isolation Forest, ICDM 2008*

ISOLATION FOREST

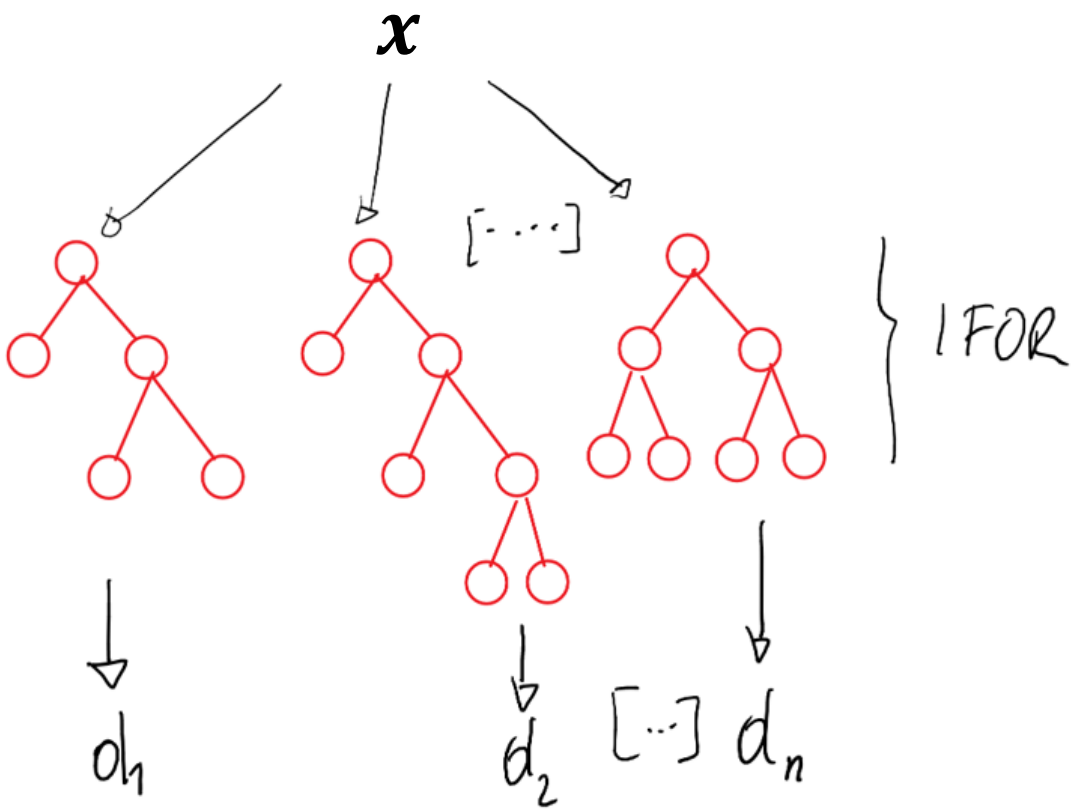
Normal data



F. T. Liu, K. M. Ting and Z. Zhou, *Isolation Forest, ICDM 2008*

ISOLATION FOREST: TESTING

Compute $E\big(h(\boldsymbol{x})\big)$, the **average path length** among all the trees in the forest, of a test sample $\boldsymbol{x}$



F. T. Liu, K. M. Ting and Z. Zhou, *Isolation Forest, ICDM 2008*

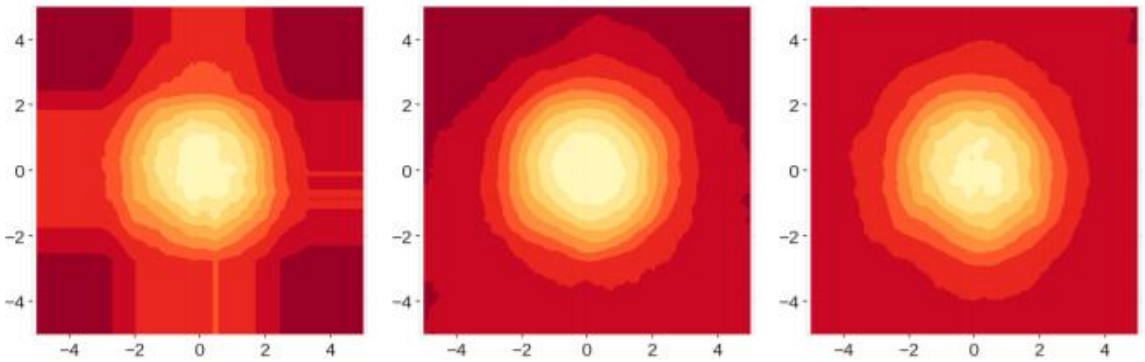
ISOLATION FOREST: TESTING

A test sample is identified as **anomalous** when:

$$\mathcal{A}(\boldsymbol{x}) = 2^{-\frac{E(h(\boldsymbol{x}))}{c(n)}} > \gamma$$

- n : number of sessions in TR
- $c(n)$: average path length of unsuccessful search in Binary

Several extensions including **EIF** (Extendend Isolation Forest) modify the splitting criteria to yield anomaly scores map that better conform to normal data



Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*
S. Hariri, M. C. Kind, and R. J. Brunner, "Extended isolation forest", TKDE, 2019.

ANOMALY DETECTION WHEN ϕ_0 AND ϕ_1 ARE UNKNOWN

Most often, **only a training set** TR is **provided**:

There are three scenarios:

- **Semi-Supervised:** Only normal training data are provided, i.e. no anomalies in TR.
- **Unsupervised:** TR is provided without label.
- **Supervised:** Both normal and anomalous training data are provided in TR.

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

SUPERVISED ANOMALY DETECTION – DISCLAIMER

Most papers and reviews agree that **supervised methods have not to be considered part of anomaly detection,** because:

- Anomalies in general lacks of a statistical coherence
- Not (enough) training samples are provided for anomalies

However,

- Some supervised problems are often referred to as «detection», in case of **severe class imbalance** (e.g. fraud detection)
- **Supervised models can be transferred** in unsupervised settings, in particular for deep learning

T. Ehret, A. Davy, JM Morel, M. Delbracio *"Image Anomalies: A Review and Synthesis of Detection Methods"*, Journal of Mathematical Imaging and Vision, 1-34

SUPERVISED ANOMALY DETECTION - SOLUTIONS

In **supervised methods** training data are annotated and divided in normal (+) and anomalies (−) :

$$TR = \{(\boldsymbol{x}(t), y(t)), \boldsymbol{x} \in \mathbb{R}^d, y \in \{+, -\} \text{ and } t < t_0\}$$

Solution:

- **Train a two-class classifier** to distinguish normal vs anomalous data.

During training:

- Train a classifier $\mathcal{K}$ from TR.

During testing:

- Compute the classifier output $\mathcal{K}(\boldsymbol{x})$, or
- Set a threshold on the posterior $p_{\mathcal{K}}(-|\boldsymbol{x})$, or
- Select the k −most likely anomalies

SUPERVISED ANOMALY DETECTION – CHALLENGES

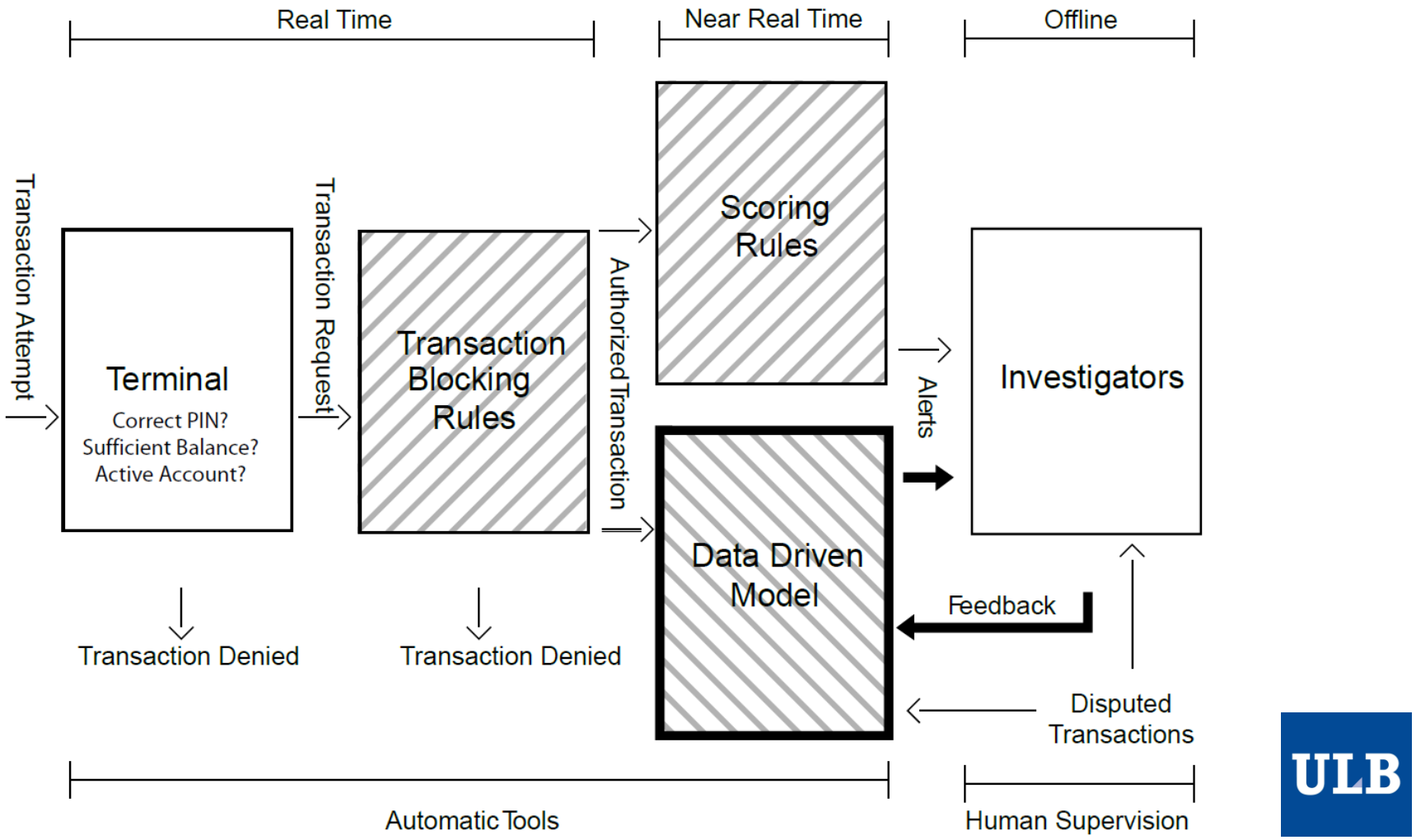
These **classification problems are challenging** because these anomaly-detection settings typically imply:

- **Class Imbalance:** Normal data far outnumber anomalies
- **Concept Drift:** Anomalies might **evolve** over time, thus the few annotated anomalies might not be representative of anomalies occurring during operations
- **Selection Bias:** Training samples are typically selected through a **closed-loop and biased procedure**. Often **only detected anomalies are annotated**, and the vast majority of the stream remain unsupervised. This biases the selection of training samples.

Dal Pozzolo A., Boracchi G., Caelen O., Alippi C. and Bontempi G., *"Credit Card Fraud Detection: a Realistic Modeling and a Novel Learning Strategy"* , IEEE TNNL 2017,

… ANOMALY DETECTION IN CREDIT CARD TRANSACTIONS

Fraud detection in credit card transactions/web sessions



Dal Pozzolo A., Boracchi G., Caelen O., Alippi C. and Bontempi G., *"Credit Card Fraud Detection: a Realistic Modeling and a Novel Learning Strategy"* , IEEE TNNL 2017, 14 pages

SUPERVISED ANOMALY DETECTION – AN EXAMPLE

This is **what typically happens in fraud detection**.

Class Imbalance:

- Frauds are typically less than 1% of genuine transactions

Concept Drift:

- Fraudster always implement new strategies

Sampling Selection Bias:

- Only alerted / reported transactions are controlled and annotated
- Old transactions that have not been disputed are considered genuine transactions

Dal Pozzolo A., Boracchi G., Caelen O., Alippi C. and Bontempi G., *"Credit Card Fraud Detection: a Realistic Modeling and a Novel Learning Strategy"* , IEEE TNNL 2017, 14 pages

A RELATED PROBLEM: OPEN SET RECOGNITION

Closed Set Classification Settings (standard)

You are given a training set

$$TR = \{(\boldsymbol{x}(t), y(t)), \boldsymbol{x} \in \mathbb{R}^d, y \in \Lambda\}$$

You train a classifier

$$\mathcal{K}: \mathbb{R}^d \rightarrow \Lambda$$

Which has to classify instances having label Λ.

Open Set Recognition Settings

The classifier $\mathcal{K}$ at test time need to recognize possible input samples that do not belong to any class in Λ

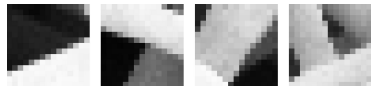
$$\mathcal{K}: \mathbb{R}^d \rightarrow \{\Lambda, \text{"unknown"}\}$$

These are more general settings and require changing $\mathcal{K}$ after training or adopting specific training procedures

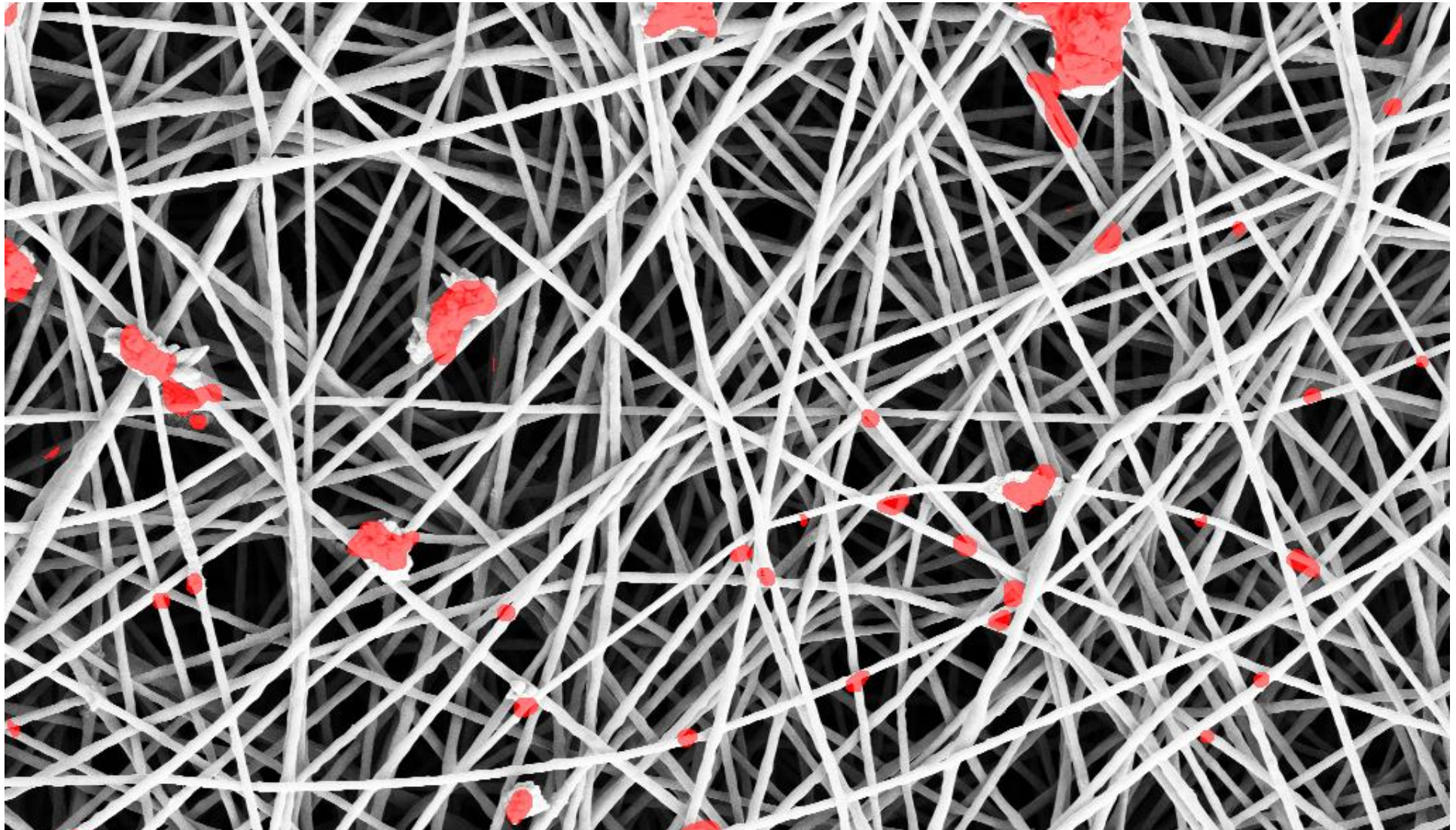
LET'S GO BACK TO IMAGES NOW…

Applying statistical methods to image patches

OUR RUNNING EXAMPLE



Goal: Automatically measure area covered by defects

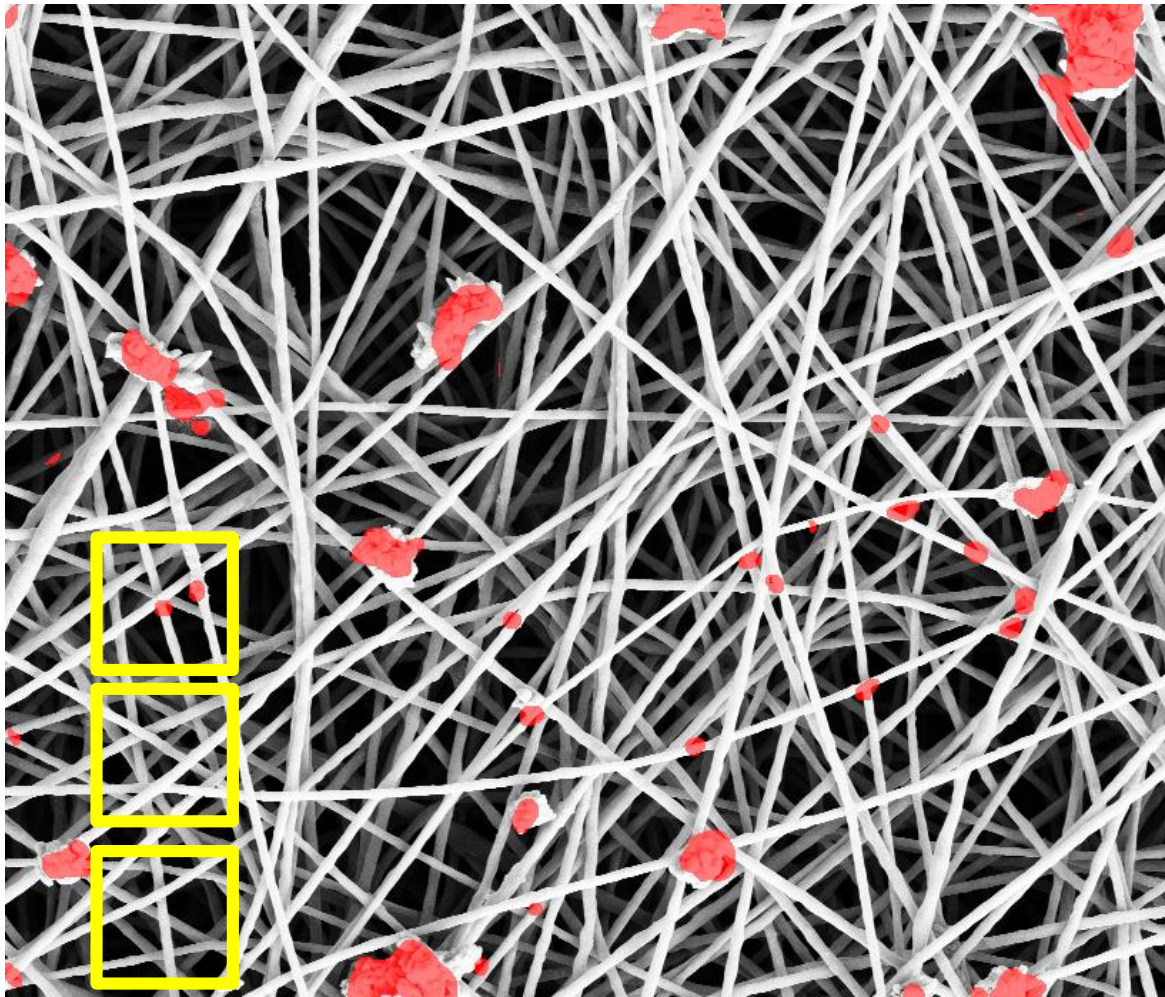


ANOMALY DETECTION IN IMAGES

The goal not determining whether the whole image is normal or anomalous, but **locate/segment possible anomalies**

Therefore, it is convenient to

1. **Analyze the image patch-wise**
2. Isolate regions containing patches that are detected as as anomalies



Can we pursue approaches designed for random variables on image patches?

DENSITY-BASED APPROACH ON IMAGE PATCHES

A density-based approach to AD would be:

Training

i. Split the normal image in patches $\boldsymbol{s}$

ii. Fit a statistical model $\hat{\phi}_0 = \mathcal{N}(\mu, \Sigma)$ describing normal patches.

Testing

i. Split the test image in patches

ii. Compute $\hat{\phi}_0(\boldsymbol{s})$ the likelihood of each test patch $\boldsymbol{s}$

iii. Detect anomalies by thresholding the likelihood

Du, B., Zhang, L.: *Random-selection-based anomaly detector for hyperspectral imagery*. IEEE Transactions on Geoscience and Remote sensing

DENSITY-BASED APPROACH ON IMAGE PATCHES

A density-based approach to AD would be:

Training

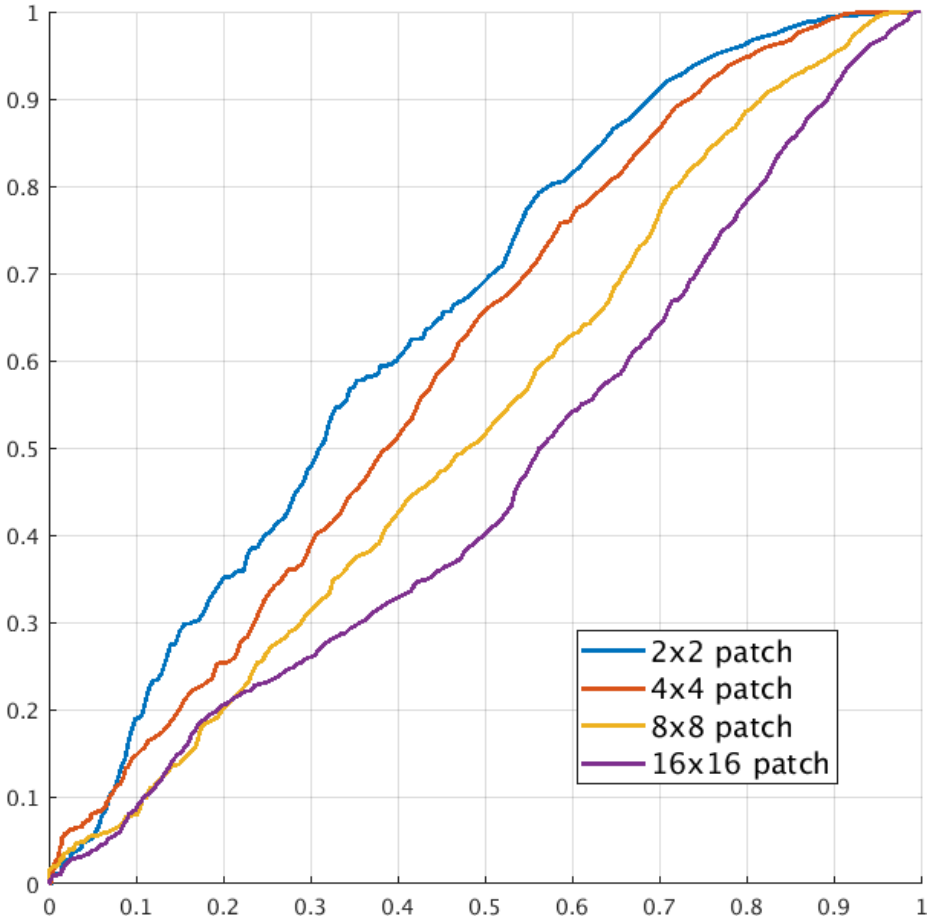
i. Split the normal image in patches $\boldsymbol{s}$

ii. Fit a statistical model $\hat{\phi}_0 = \mathcal{N}(\mu, \Sigma)$ describing normal patches.

This model is rarely accurate on natural images. Small patches (e.g. 2×2 or 5×5) are typically preferred

Du, B., Zhang, L.: *Random-selection-based anomaly detector for hyperspectral imagery*. IEEE Transactions on Geoscience and Remote sensing
X Xie, M Mirmehdi *"Texture exemplars for defect detection on random textures"* – ICPR 2005

THE LIMITATIONS OF THE RANDOM VARIABLE MODEL



This model is rarely accurate on natural images. Small patches (e.g. 2×2 or 5×5) are typically preferred

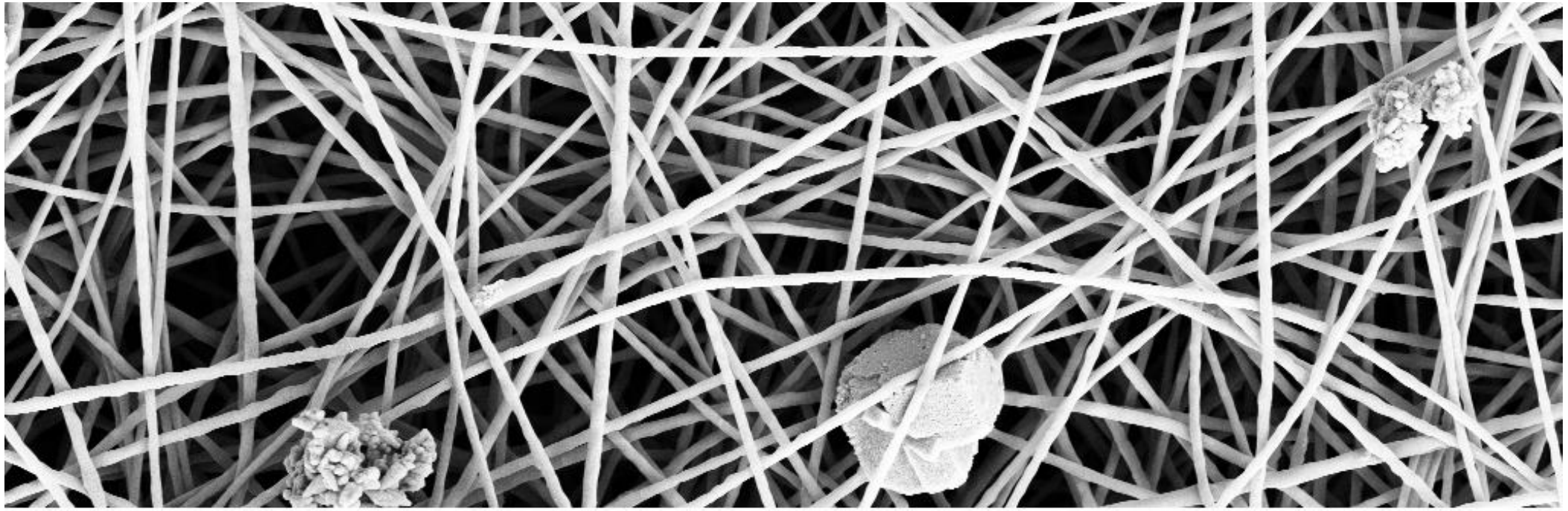
The model becomes even more unfit as the patch size increases

Du, B., Zhang, L.: *Random-selection-based anomaly detector for hyperspectral imagery*. IEEE Transactions on Geoscience and Remote sensing

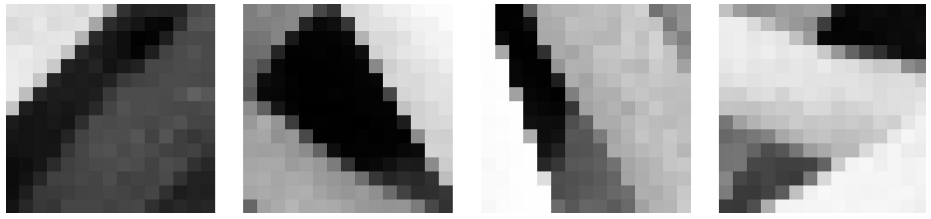
X Xie, M Mirmehdi *"Texture exemplars for defect detection on random textures"* – ICPR 2005

THE LIMITATIONS OF THE RANDOM VARIABLE MODEL

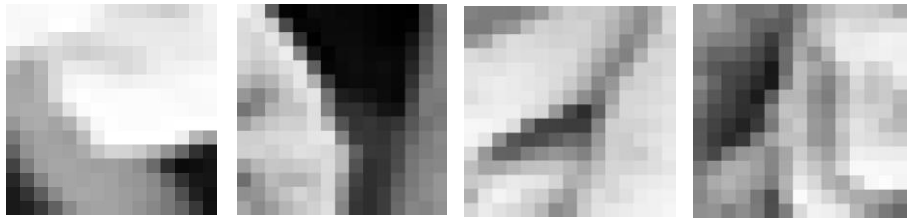
In many anomaly-detection problems in imaging, **normal regions exhibit peculiar structures and spatial correlation**



Normal patches



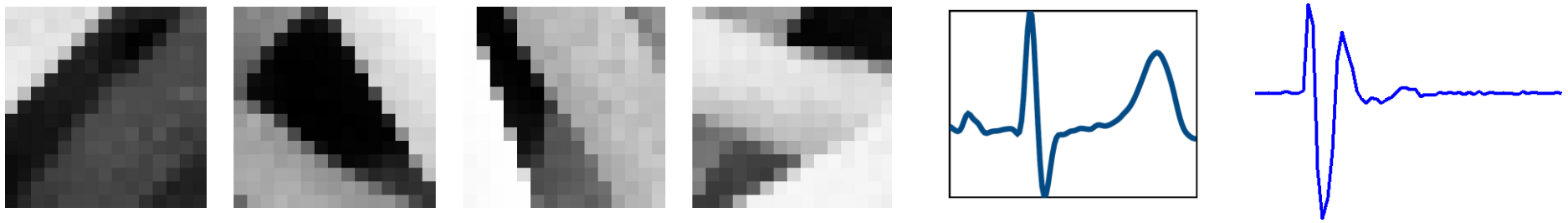
Anomalous patches



Boracchi, 2022

REAL WORLD DETECTION PROBLEMS

Random variable model **does not successfully apply to signals or images** (not even small portions)



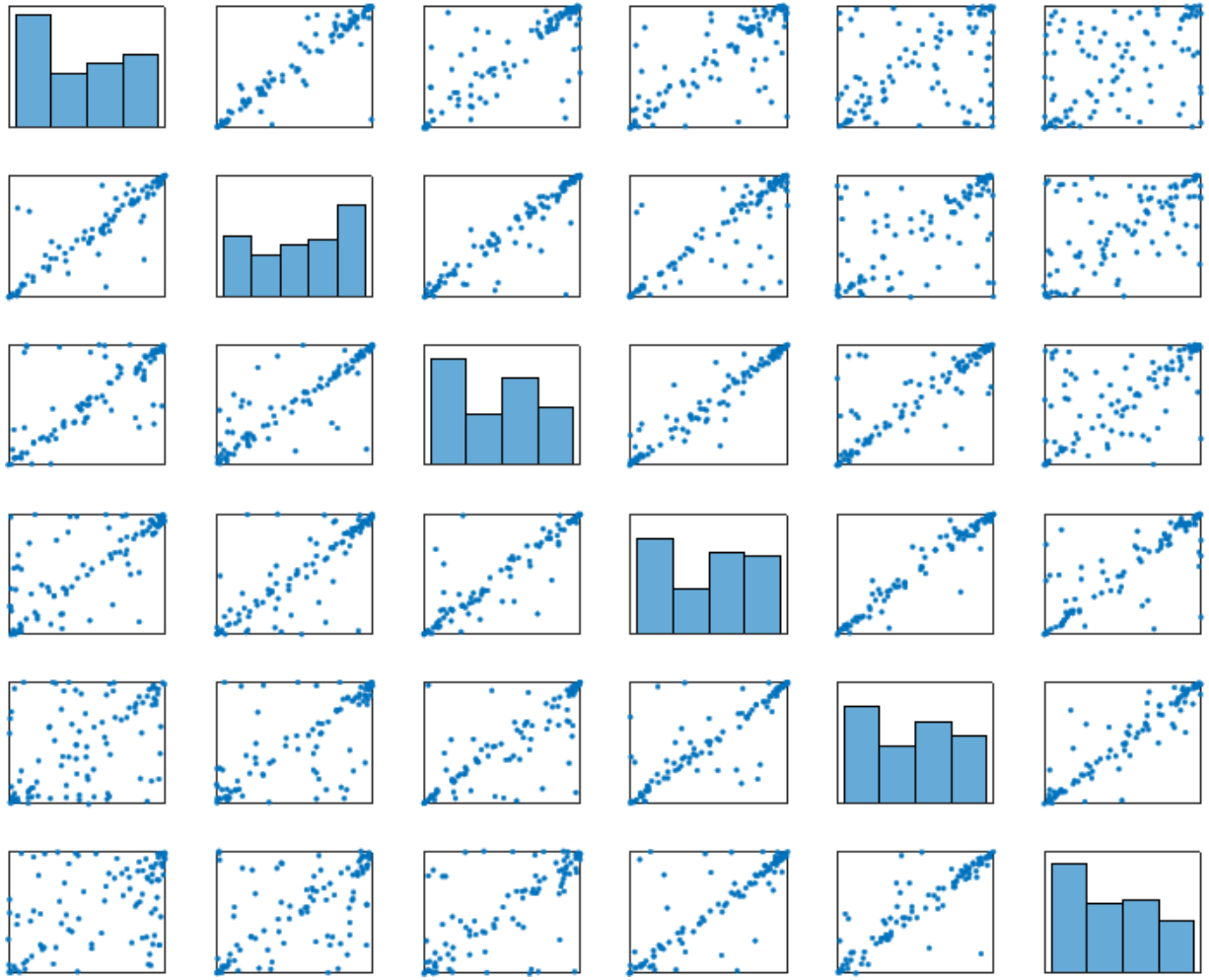
Stacking each patch/signal $\boldsymbol{s} \in \mathbb{R}^d$ in a vector $\boldsymbol{x}$ is not convenient:

- **Data dimension** d becomes **huge**
- **Strong correlations** among components, **difficult to directly model** by a probability density function ϕ_0

Specific models have to be used to **approximate** the **manifold** where **signals and images live**

THE LIMITATIONS OF THE RANDOM VARIABLE MODEL

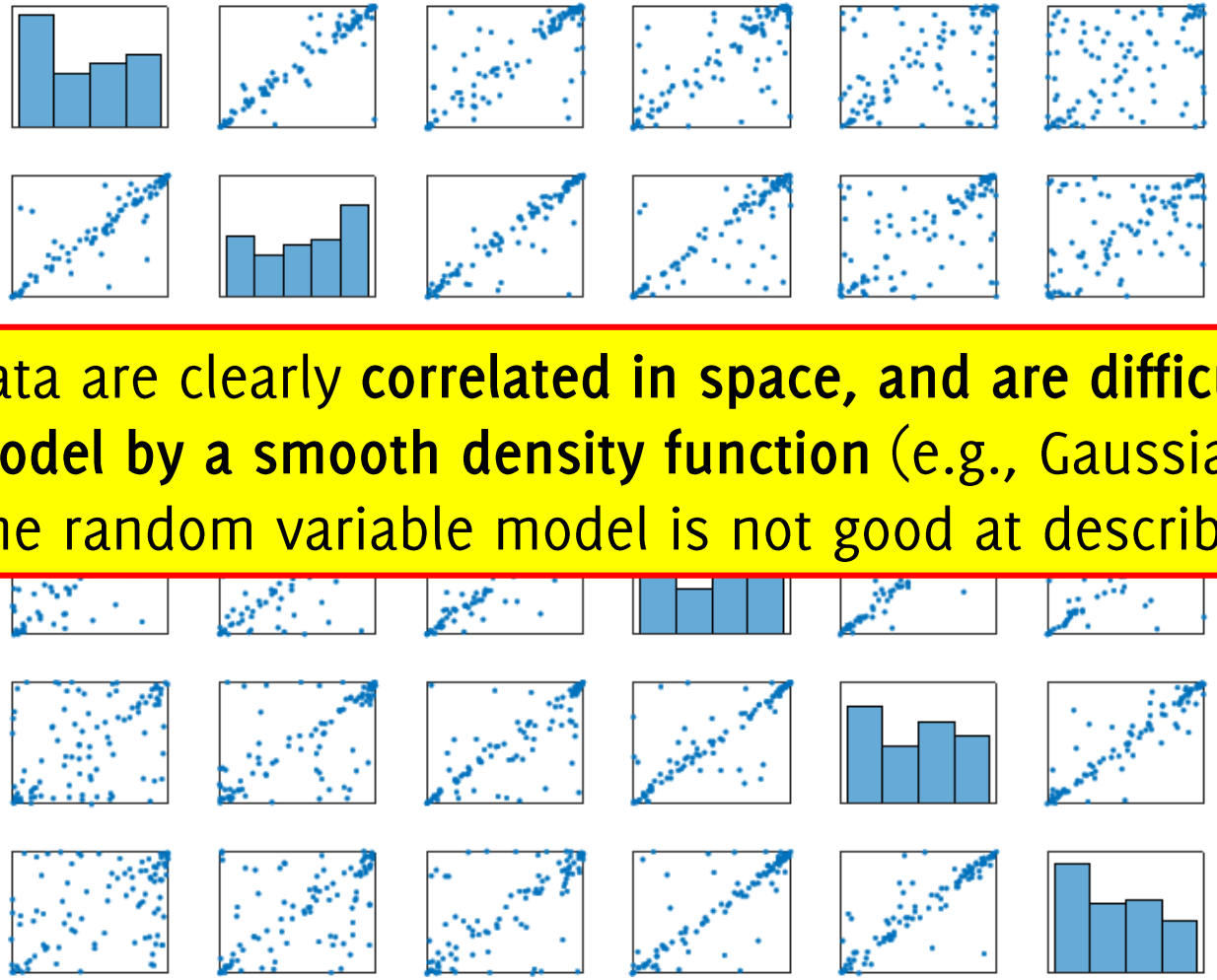
Distribution of adjacent pixel values inside a patch:



Boracchi, 2022

THE LIMITATIONS OF THE RANDOM VARIABLE MODEL

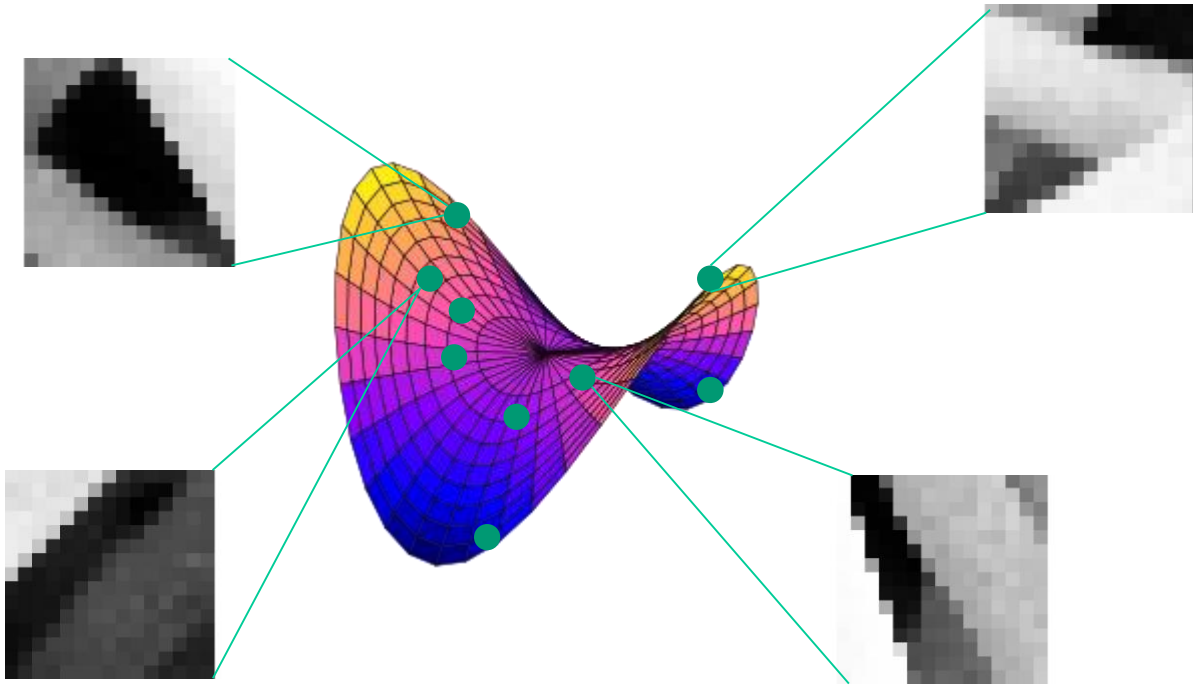
Distribution of adjacent pixel values inside a patch:



Data are clearly **correlated in space, and are difficult to model by a smooth density function** (e.g., Gaussians)
The random variable model is not good at describing images

THE LIMITATIONS OF THE RANDOM VARIABLE MODEL

Patches from natural images live close to a low dimensional manifold

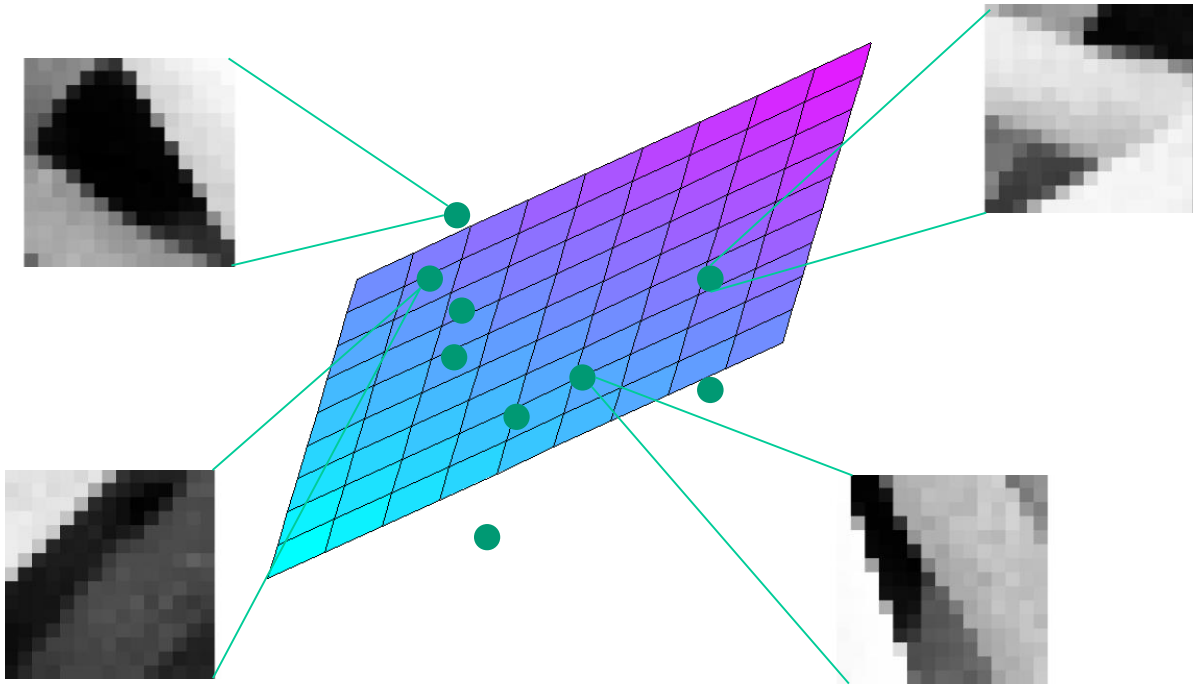


These means that patches can be well described by few latent variables

Bengio, Courville, Vincent, Representation Learning: A Review and New Perspectives, IEEE Pattern Analysis and Machine Intelligence ' 2013

A SIMPLE EXPERIMENT

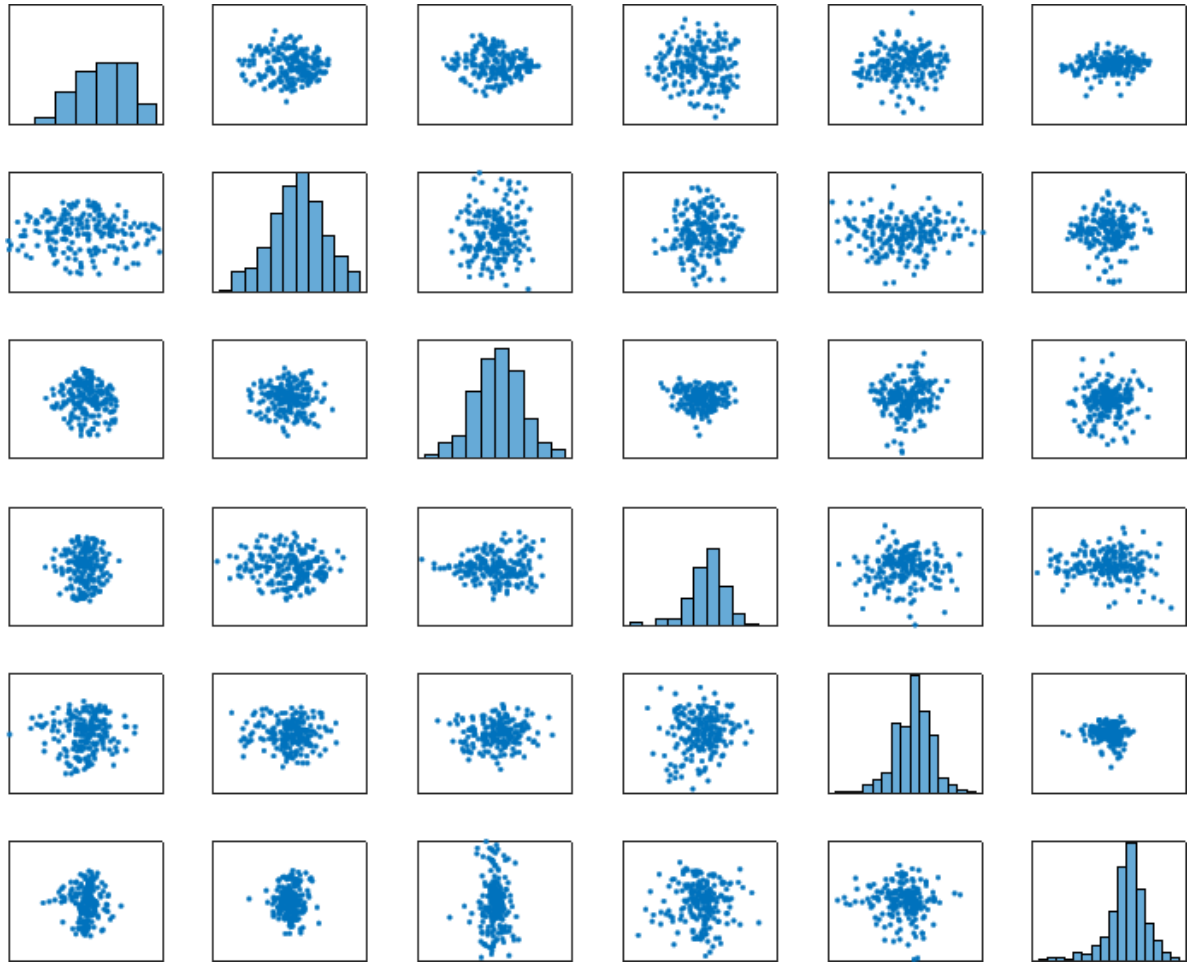
Let’s approximate this manifold with the simplest one: a linear subspace



In practice, we compute the PCA of training patches and consider the PCA score as latent variables

A SIMPLE EXPERIMENT

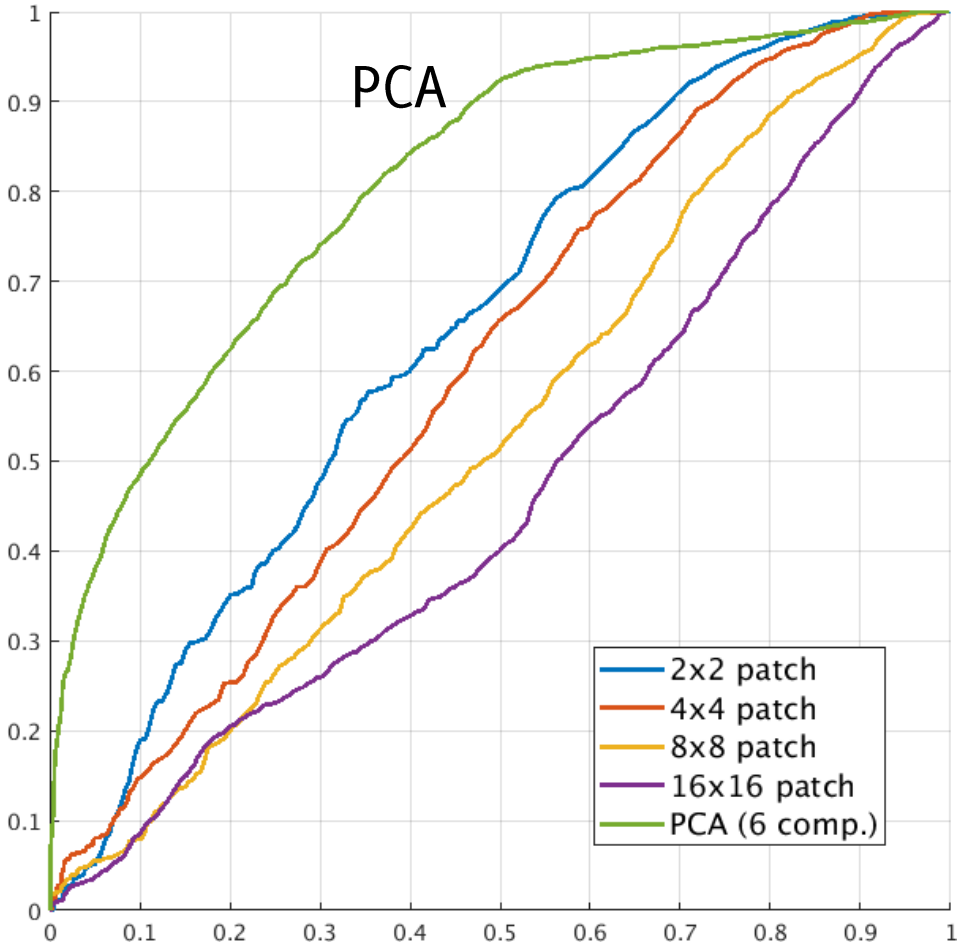
Distribution of first 6 PCA coefficients:



Boracchi, 2022

A SIMPLE EXPERIMENT

We fit a $\hat{\phi}_0 = \mathcal{N}(\mu, \Sigma)$ on PCA components to describe normal patches, and perform anomaly detection



Boracchi, 2022

PART 2: ANOMALY DETECTION IN IMAGES

Out of the “Random Variable” World:
signal-based models for images

THE THREE MAJOR INGREDIENTS

Most detection algorithms have three major ingredients:

- The **background model** $\mathcal{M}$, learned from normal data
- The **statistic / anomaly score**: $\mathrm{err}(\boldsymbol{s}), \mathcal{L}(\boldsymbol{s}), \mathcal{A}(\boldsymbol{s}), \ldots$
- **Decision rule** to detect, e.g. $\mathrm{err}(\boldsymbol{s}) \gtrless \gamma$ possibly controlling the FPR, as in other statistical detection methods

THE TYPICAL APPROACH

Most of the considered methods

1. **Estimate** a **model** describing **normal data** (background model)
2. Use the background model to provide, for each test signal/patch, an **anomaly score,** or measure of rareness
3. Apply a **decision rule** to the anomaly score to detect anomalies (typically thresholding)
4. **[optional]** Perform **post-processing** operations to enforce smooth detections and avoid isolated pixels that are not consistent with neighbourhoods

Remark: Statistical-based approaches seen before uses as background model the statistical distribution $\hat{\phi}_0$ and a statistic as anomaly score

THE TYPICAL APPROACH

Most of the considered methods

1. **Estimate** a **model** describing **normal data** (background model)
2. Use the background model to provide, for each test signal/patch, an **anomaly score,** or measure of rareness
3. Apply a **decision rule** to the anomaly score to detect anomalies (typically thresholding)
4. **[optional]** Perform **post-processing** operations to enforce smooth detections and avoid isolated pixels that are not consistent with neighbourhoods

Remark: Statistical-based approaches seen before uses as background model the statistical distribution $\hat{\phi}_0$ and a statistic as anomaly score

The background model is used to **bring an image patch into the "random variable world"**

THE TYPICAL APPROACH

Most of the considered methods

1. **Estimate** a **model** describing **normal data** (background model)
2. Use the background model to provide, for each test signal/patch, an **anomaly score,** or measure of rareness
3. Apply a **decision rule** to the anomaly score to detect anomalies (typically thresholding)
4. **[optional]** Perform **post-processing** operations to enforce smooth detections and avoid isolated pixels that are not consistent with neighbourhoods

Remark: Statistical-based approaches seen before uses as background model the statistical distribution $\hat{\phi}_0$ and a statistic as anomaly score

Once "having applied" the background model, one can use **anomaly detection methods for the "random variable world".**
This might require **fitting an additional (density) model** in the random variable world

THE TYPICAL APPROACH

Most of the considered methods

1. **Estimate** a **model** describing **normal data** (background model)
2. Use the background model to provide, for each test signal/patch, an **anomaly score,** or measure of rareness
3. Apply a **decision rule** to the anomaly score to detect anomalies (typically thresholding)
4. **[optional]** Perform **post-processing** operations to enforce smooth detections and avoid isolated pixels that are not consistent with neighbourhoods

Remark: Statistical-based approaches seen before uses as background model the statistical distribution $\hat{\phi}_0$ and a statistic as anomaly score

And it is important to control the False Positive Rate or the ARL_0 of the overall monitoring scheme

SEMI-SUPERVISED ANOMALY-DETECTION IN IMAGES

Out of the "Random Variable" world

- Reconstruction-based methods
    - Subspace methods
- Feature-based monitoring
    - Expert-driven Features
    - Data-driven Features

SEMI-SUPERVISED ANOMALY-DETECTION IN IMAGES

Out of the "Random Variable" world

- Reconstruction-based methods
  - Subspace methods
- Feature-based monitoring
  - Expert-driven Features
  - Data-driven Features

RECONSTRUCTION-BASED METHODS

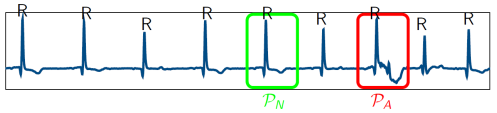
Fit a statistical model to the observation to ***describe dependence, apply anomaly detection*** *on the independent* ***residuals****.*

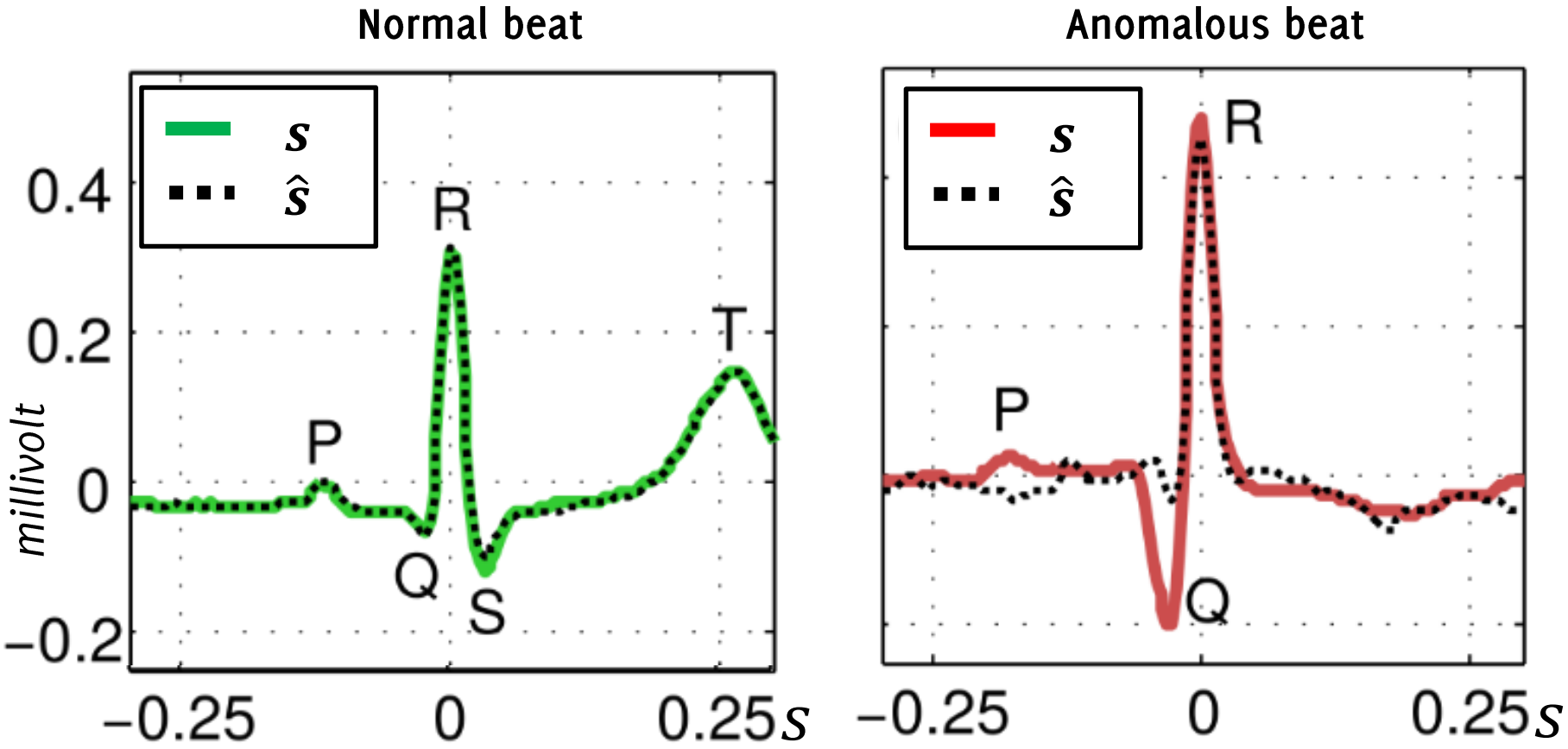
Detection is performed by using **a model** $\mathcal{M}$ which **can encode and reconstruct normal data**:

- **During training:** learn the model $\mathcal{M}$ from training set S
- **During testing:**
  - Encode and reconstruct each test signal $\boldsymbol{s}$ through $\mathcal{M}$.
  - Assess $\mathrm{err}(\boldsymbol{s})$, namely the **residual** between $\boldsymbol{s}$ and its reconstruction through $\mathcal{M}$

The rationale is that $\mathcal{M}$ **can reconstruct only normal data**, thus anomalies are expected to yield large reconstruction errors.

RECONSTRUCTION-BASED MONITRING



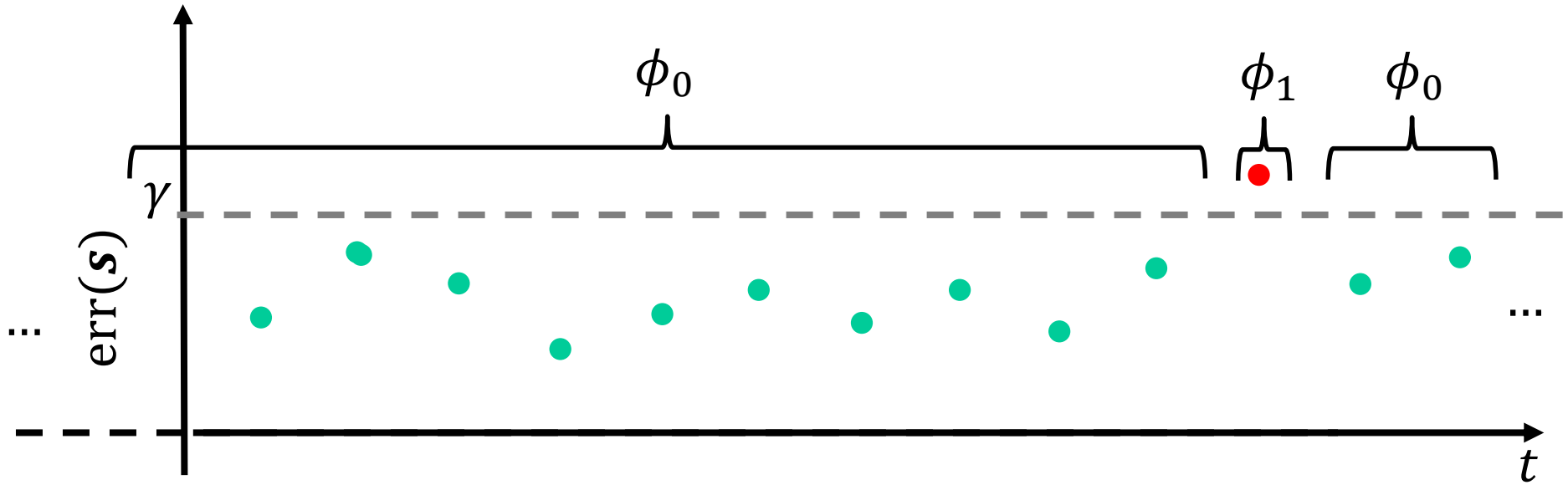


Carrera D., Rossi B., Fragneto P., Boracchi G. "*Online Anomaly Detection for Long-Term ECG Monitoring using Wearable Devices*", Pattern Recognition, Elsevier 2019

MONITORING THE RECONSTRUCTION ERROR

Normal data are expected to yield values of $\mathrm{err}(\boldsymbol{s})$ that **are low**, while anomalies do not. This holds when the model $\mathcal{M}$ was specifically learned to describe normal data

Outliers can be detected by thresholding $\mathrm{err}(\boldsymbol{s})$



RECONSTRUCTION-BASED METHODS

Popular models are:

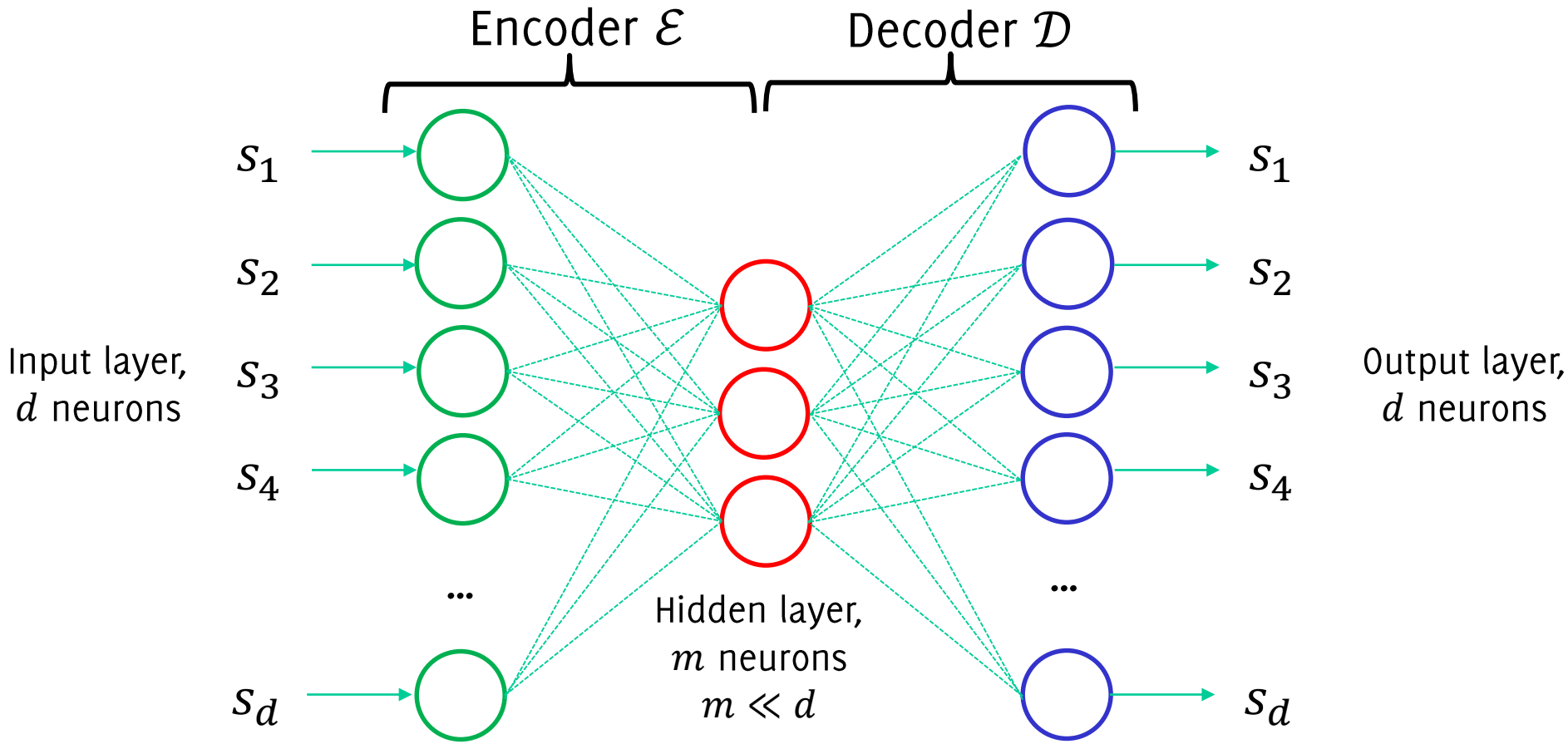
- **neural networks**, in particular auto-encoders (and convolutional auto-encodrs), for images and high-dimensional data
- **projection on subspaces** / manifolds
- **dictionaries yielding sparse-representations**
- **autoregressive models** for time series (ARMA, ARIMA…)

Methods based on projections and dictionaries can be also interpreted as subspace methods

RECONSTRUCTION-BASED METHODS

Autoencoders are neural networks used for data reconstruction (they learn the identity function)

The typical structure of an autoencoder is:



RECONSTRUCTION-BASED METHODS

Autoencoders are trained to reconstruct all the samples in the training set. The reconstruction loss is

$$\sum_{\boldsymbol{s}\in S} \left\| \boldsymbol{s} - \mathcal{D}\big(\mathcal{E}(\boldsymbol{s})\big) \right\|_2 + \lambda \mathcal{R}(\mathcal{E}(\boldsymbol{s}))$$

and $\mathcal{D}\big(\mathcal{E}(\cdot)\big)$ is trained via backpropagation algorithm, $\lambda > 0$ and $\mathcal{R}(\cdot)$ is a recularization term

Remarks

- Typically $\mathcal{D}\big(\mathcal{E}(\cdot)\big)$ does not provide perfect reconstruction, since $m \ll d$.
- **Regularization terms** $\mathcal{R}(\cdot)$ might be included in the loss function for the latent representation $\mathcal{E}(\boldsymbol{s})$ to feature specific properties

Bengio, Y., Courville, A., Vincent, P. *"Representation learning: A review and new perspectives"*. IEEE TPAMI 2013
Mishne, G., Shaham, U., Cloninger, A., & Cohen, I. *Diffusion nets*. Applied and Computational Harmonic Analysis (2017).

MONITORING THE RECONSTRUCTION ERROR

Detection by reconstruction error monitoring (AE notation)

Training (Monitoring the Reconstruction Error):

1. Train the model $\mathcal{D}(\mathcal{E}(\cdot))$ from the training set S
2. Learn the distribution of reconstruction errors

$$\mathrm{err}(\boldsymbol{s}) = \left\| \boldsymbol{s} - \mathcal{D}\big(\mathcal{E}(\boldsymbol{s})\big) \right\|_2, \qquad \boldsymbol{s} \in V$$

over a validation set V, such that $V \cap S = \emptyset$, and define a suitable threshold γ

Testing (Monitoring the Reconstruction Error):

1. Perform encoding and compute the reconstruction error

$$\mathrm{err}(\boldsymbol{s}) = \left\| \boldsymbol{s} - \mathcal{D}\big(\mathcal{E}(\boldsymbol{s})\big) \right\|_2$$

2. Consider $\boldsymbol{s}$ anomalous when $\mathrm{err}(\boldsymbol{s}) > \gamma$

OUTLINE ON SEMI-SUPERVISED APPROACHES

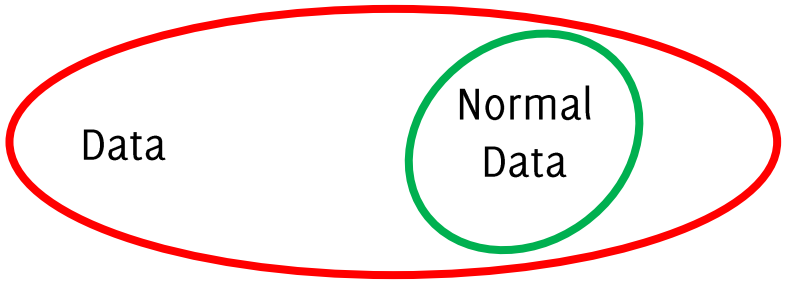
Out of the "Random Variable" world

- Reconstruction-based methods
  - Subspace methods
- Feature-based monitoring
  - Expert-driven Features
  - Data-driven Features
  - Extended models

SUBSPACE METHODS

The underlying assumption is that:

- **Normal patches live in a subspace** that can be identified by S
- **Anomalies can be detected by projecting test patches** in such subspace and by monitoring the reconstruction error (distance with the projection)



V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

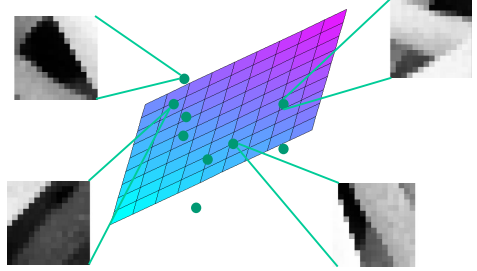
SUBSPACE METHODS

A few example of **models used for describing normal patches:**

- Orthogonal basis: normal patches can be expressed by **a few selected basis elements** (Fourier, Wavelets..)
- PCA: normal patches live in the **linear subspace of the first components**
- Robust PCA: defined on the ℓ^1 distance to be **insensitive to outliers in normal data**
- Kernel PCA: normal patches live in **a non-linear manifold**
- **Dictionaries** yielding **sparse representations**
- Random projections
- .. Latent representations from an autoencoder

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

SUBSPACE METHODS: PCA-BASED MONITORING

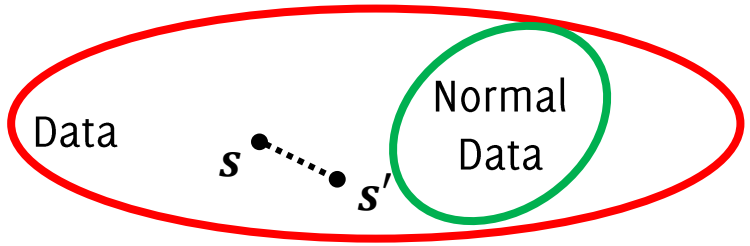


Anomaly detection based on PCA (and similar techniques):

1. Compute the **projection on the subspace**,
$$\boldsymbol{s}' = P^T \boldsymbol{s}, \qquad P \in \mathbb{R}^{d \times m}, \qquad m \ll d$$
which is the projection over the first m principal components and a way to reduce data-dimensionality.

2. Monitor the **reconstruction error**:
$$\text{err}(\mathbf{s}) = \|\boldsymbol{s} - PP^T \boldsymbol{s}\|_2$$
which is the distance between $\boldsymbol{s}$ and its projection $PP^T \boldsymbol{s}$ over the subspace of normal patches

2. [bis] The **projection along the last principal component**, is also a good anomaly score, as it becomes large at anomalies.

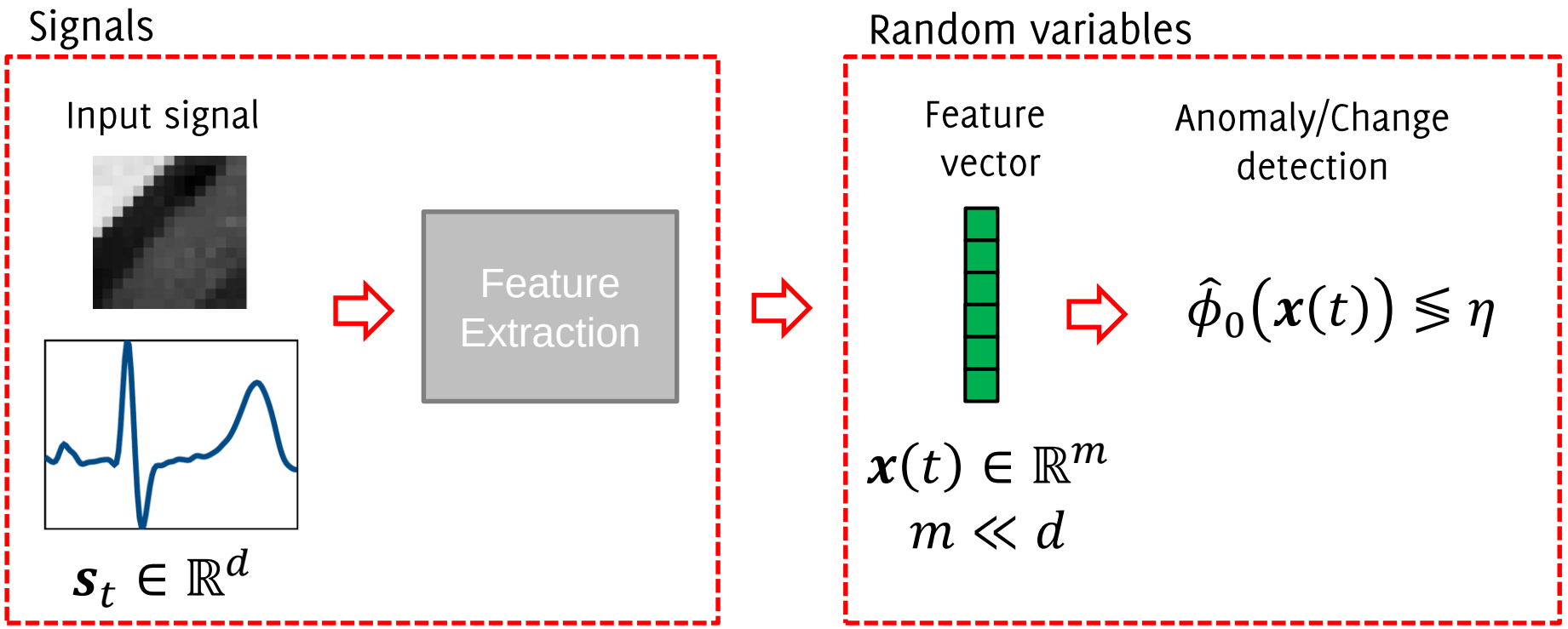


OUTLINE ON SEMI-SUPERVISED APPROACHES

- Reconstruction-based methods
  - Subspace methods
- Feature-based monitoring
  - Expert-driven Features
  - Data-driven Features
  - Extended models

TYPICAL APPROACH: MONITORING FEATURES

Feature extraction: meaningful indicators to be monitored which have a known / controlled response w.r.t. normal data



Feature Extraction: signal processing, a priori information, learning methods

The customary statistical framework for anomaly detection

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

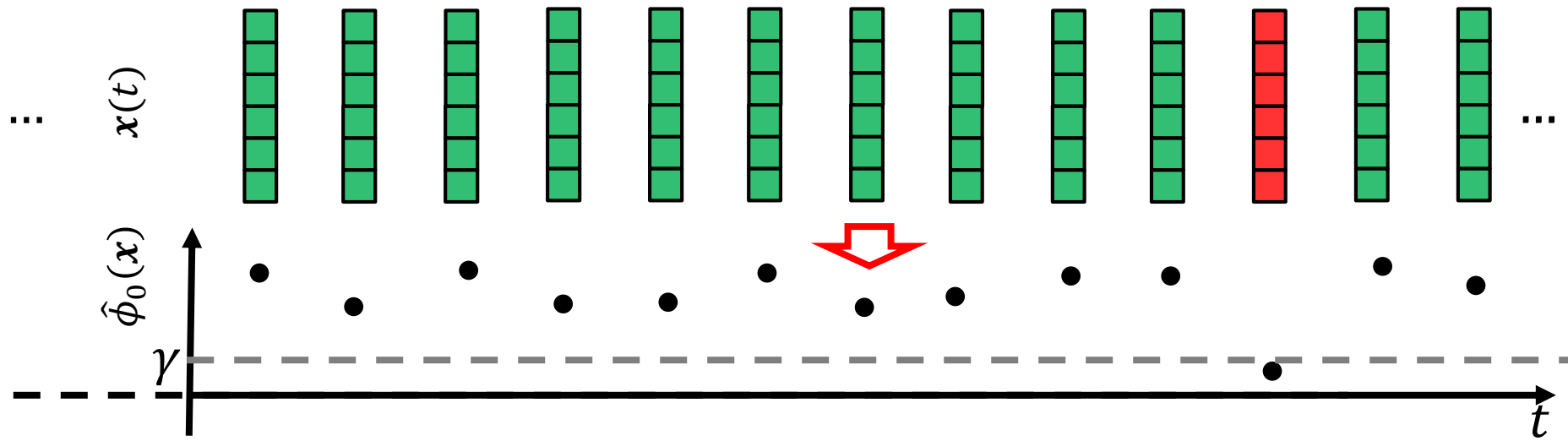
MONITORING FEATURE DISTRIBUTION

Normal data are expected **to yield features** $\boldsymbol{x}$ that are **i.i.d.** and **follow an** unknown **distribution** ϕ_0.

Anomalous data do not, as they follow $\phi_1 \neq \phi_0$.

We are **back to our statistical framework** and we can

- **learn** $\hat{\phi}_0$ from a set features extracted from normal data
- detect anomalous data by **extracting features** $\boldsymbol{x}$ associated to each input s, and then **testing whether** $\hat{\phi}_0(x) < \gamma$



MONITORING FEATURE DISTRIBUTION

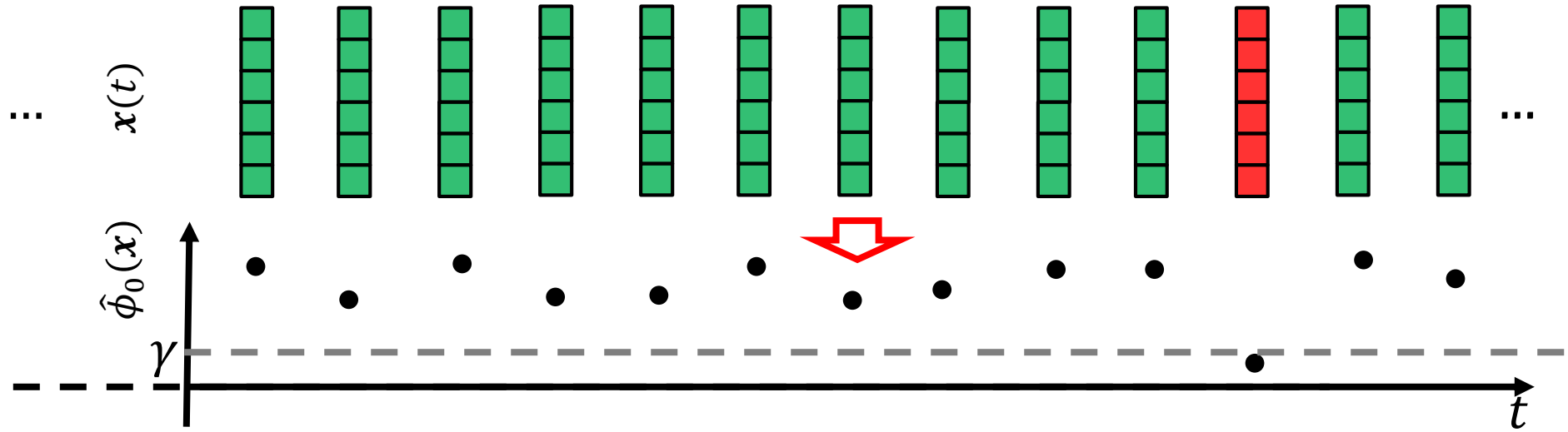
Normal data are expected **to yield features** $\boldsymbol{x}$ that are **i.i.d.** and **follow an** unknown **distribution** ϕ_0.

Anomalous data do not, as they follow $\phi_1 \neq \phi_0$.

We are **back to our statistical framework** and we can

- **learn** $\hat{\phi}$ from a set features extracted from normal data
- detect [illegible] out s, and then **testing whether** $\phi_0(x) < \gamma$

Or by adopting any other statistical tool to detect anomalies in $\boldsymbol{x}$



FEATURE EXTRACTION

Data dimensionality can be reduced by extracting features

Good features should:

- Yield a **stable response** w.r.t. **normal data**
- Yield **unusual response** on **anomalies**

Examples of features seen so far:

- Reconstruction error $\text{err}(\boldsymbol{s})$
- representation coefficients $(P^T\boldsymbol{s}, \boldsymbol{\alpha}, \dots)$

… but these are not the only

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

FEATURE EXTRACTION APPROACHES

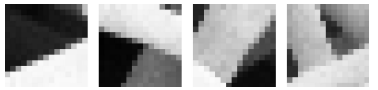
There are two major approaches for extracting features:

Expert-driven (hand-crafted) features: computational expressions that are **manually designed by experts** to distinguish between normal and anomalous data

Data-driven features: **features** characterizing normal data are automatically **learned from training set of normal samples** S

OUTLINE ON SEMI-SUPERVISED APPROACHES

- Detrending/Filtering for time-series
- Reconstruction-based methods
  - Subspace methods
- Feature-based monitoring
  - Expert-driven Features
  - Data-driven Features
  - Data-driven Features: extended models



EXAMPLES OF EXPERT-DRIVEN FEATURES

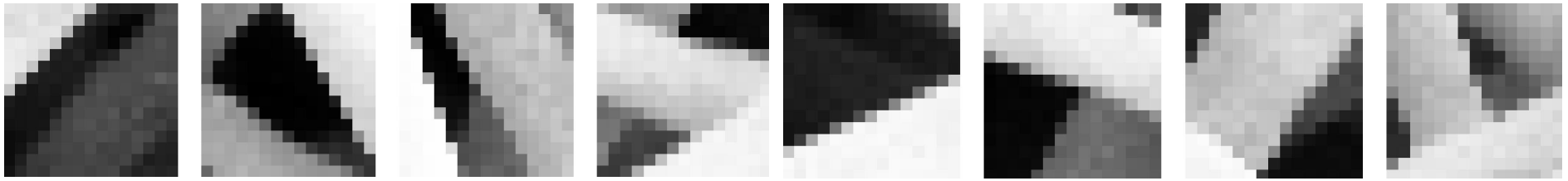
Expert-driven features: each patch of an image s

$$\mathbf{s}_c = \{s(c+u), u \in \mathcal{U}\}$$

Example of features are:

- the average,
- the variance,
- the total variation (the energy of gradients)

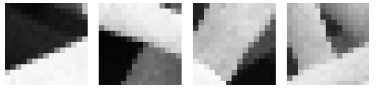
These can hopefully **distinguish normal** and **anomalous** patches, considering also how **anomalous regions will be** (e.g. flat or without edges)



Carrera D., Manganini F., Boracchi G., Lanzarone E. *"Defect Detection in SEM Images of Nanofibrous Materials"*, IEEE Transactions on Industrial Informatics 2017, 11 pages, doi:10.1109/TII.2016.2641472

OUTLINE ON SEMI-SUPERVISED APPROACHES

- Detrending/Filtering for time-series
- Reconstruction-based methods
  - Subspace methods
- Feature-based monitoring
  - Expert-driven Features
  - Data-driven Features
  - Data-driven Features: extended models



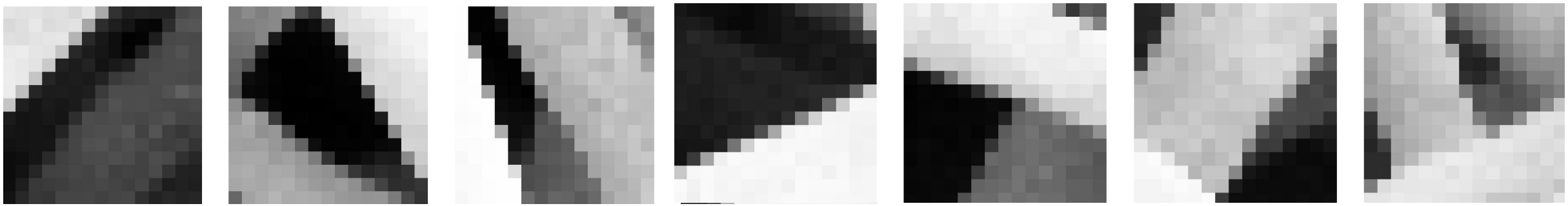
EXAMPLES OF DATA-DRIVEN FEATURES

Analyze each patch of an image s

$$\mathbf{s}_c = \{s(c+u), u \in \mathcal{U}\}$$

and determine whether it is normal or anomalous.

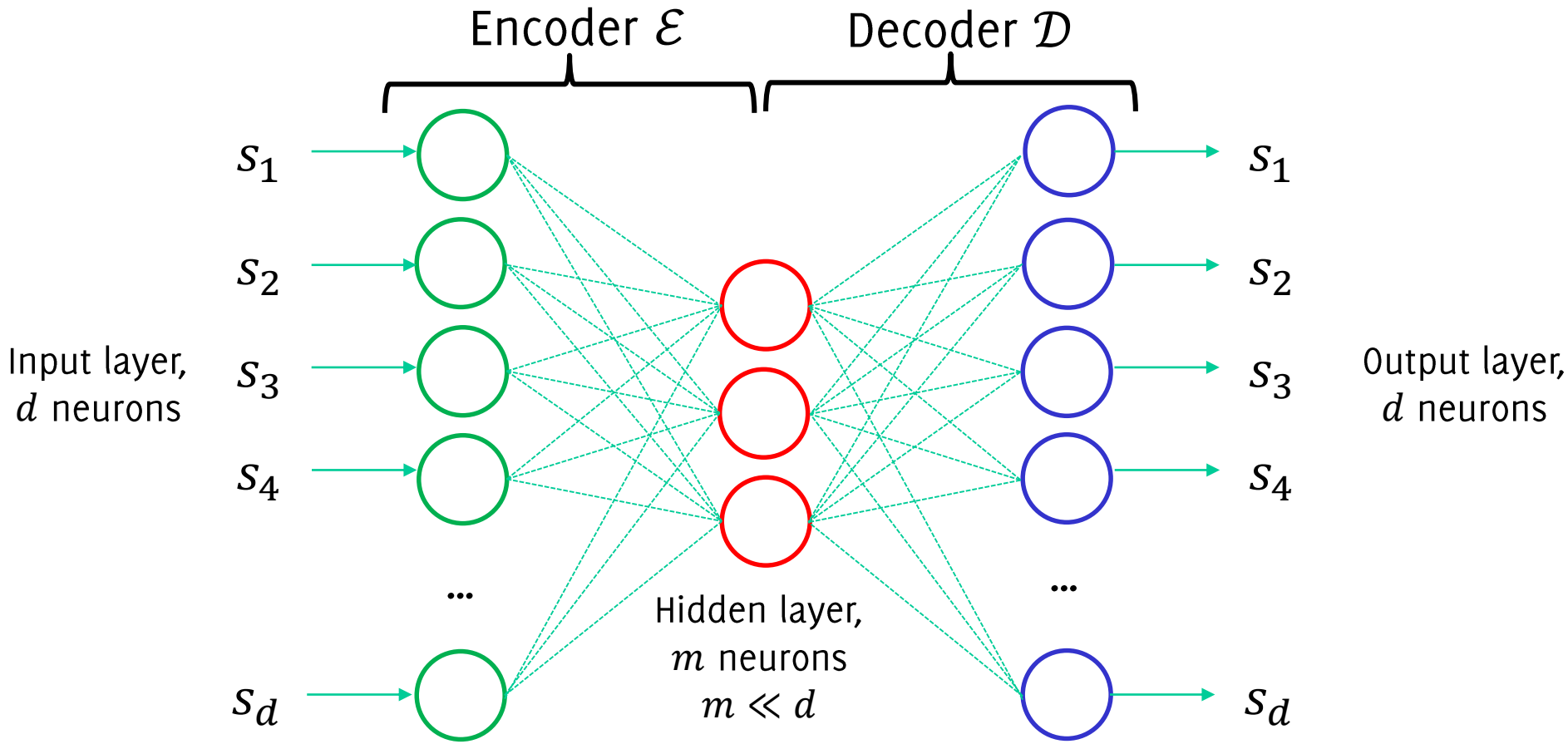
Data driven features: expressions to **quantitatively assess whether test patches conform** or not **with the model**, learned from normal data.



Carrera D., Manganini F., Boracchi G., Lanzarone E. *"Defect Detection in SEM Images of Nanofibrous Materials"*, IEEE Transactions on Industrial Informatics 2017, 11 pages, doi:10.1109/TII.2016.2641472

AUTOENCODERS AS FEATURE EXTRACTORS

Autoencoders can be also used in feature-based monitoring schemes, monitoring as **feature** the hidden/**latent representation** of the input signal



ANOMALY DETECTION BY MONITORING FEATURE DISTRIBUTION

Detection by **feature monitoring** (AE notation)

Training (Monitoring Feature Distribution):

- Learn the autoencoder $\mathcal{D}(\mathcal{E}(\cdot))$ from the training set S
- Fit a density model $\hat{\phi}_0$ to the encoded features

$$\{\mathcal{E}(\boldsymbol{s}), \boldsymbol{s} \in V\}$$

  over a validation set V, such that $V \cap S = \emptyset$
- Define a suitable threshold γ for $\hat{\phi}_0(\boldsymbol{s})$

Testing (Monitoring Feature Distribution):

- Encode each incoming signal $\boldsymbol{s}$ through $\mathcal{E}$
- Detect anomalies if $\hat{\phi}_0(\mathcal{E}(\boldsymbol{s})) < \gamma$

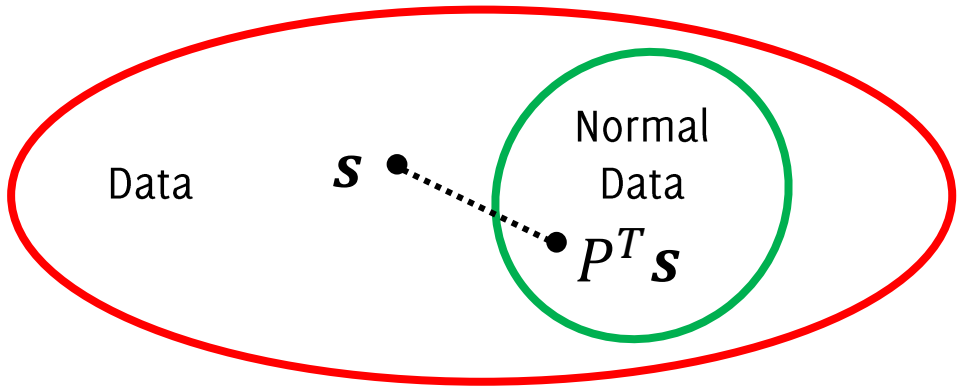
ANOMALY DETECTION BY MONITORING PCA PROJECTIONS

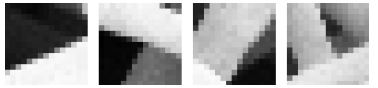
Compute the **projection on the subspace**,

$$\boldsymbol{s}' = P^T \boldsymbol{s}, \qquad P \in \mathbb{R}^{d \times m}, \qquad m \ll d$$

which is the projection over the first m principal components and a way to reduce data-dimensionality.

Monitor the projections $P^T \boldsymbol{s}$ by a suitable statistical technique (e.g. density based), as when monitoring $\mathcal{E}(\boldsymbol{s})$





SPARSE REPRESENTATIONS AS FEATURE EXTRACTORS

To assess the conformance of $\boldsymbol{s}_c$ with D we solve the following

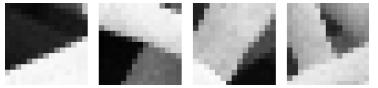
Sparse coding:

$$\boldsymbol{\alpha} = \underset{\boldsymbol{a}\in\mathbb{R}^n}{\operatorname{argmin}} \|D\boldsymbol{a} - \mathbf{s}\|_2^2 + \lambda\|\boldsymbol{a}\|_1, \qquad \lambda > 0$$

which is the BPDN formulation and we solve using ADMM.

The penalized ℓ^1 formulation has more degrees of freedom in the reconstruction, **the conformance of $\boldsymbol{s}$ with $\boldsymbol{D}$ have to be assessed monitoring both terms of the functional**

Boyd S., Parikh N., Chu E., Peleato B., Eckstein J., *"Distributed optimization and statistical learning via the alternating direction method of multipliers"* 2011



FEATURES EXTRACTED FROM SPARSE CODING

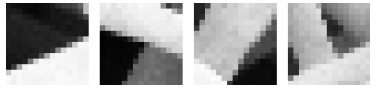
Features then include both the **reconstruction error**

$$\mathrm{err}(\boldsymbol{s}) = \|D\boldsymbol{\alpha} - \mathbf{s}\|_2^2$$

and **the sparsity** of the representation

$$\|\boldsymbol{\alpha}\|_1$$

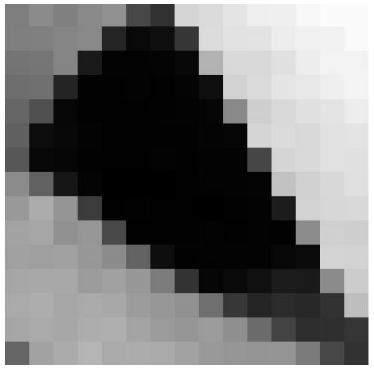
Thus obtaining **a data-driven feature vector** $\boldsymbol{x} = \begin{bmatrix} \|D\boldsymbol{\alpha} - \mathbf{s}\|_2^2 \\ \|\boldsymbol{\alpha}\|_1 \end{bmatrix}$

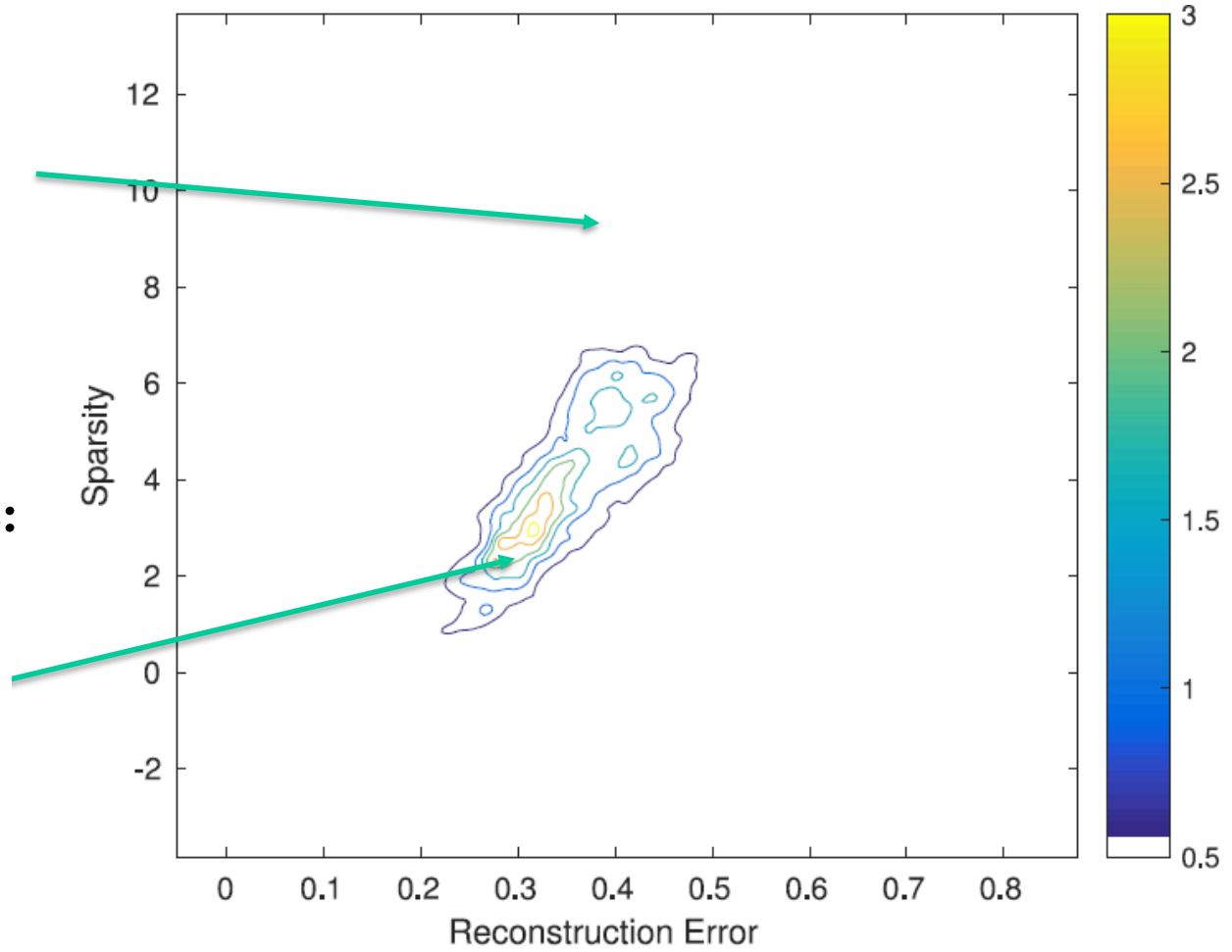


DENSITY-BASED MONITORING

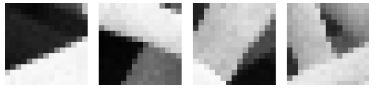
Anomalies

Normal patches:





Boracchi, 2022



FEATURES EXTRACTED FROM SPARSE CODING

Training:

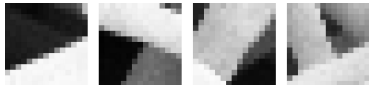
- **Learn** from S the dictionary D
- **Compute** the sparse representation w.r.t. D, thus features $\boldsymbol{x}$ over the validation set V, such that $V \cap S = \emptyset$
- **Learn** from V, the distribution $\hat{\phi}_0$ of normal features vectors $\boldsymbol{x}$ and the threshold γ.

The model for anomaly detection is $(D, \hat{\phi}_0, \gamma)$

Testing:

- Perform sparse coding of a test signal $\boldsymbol{s}$, thus get the feature vector $\boldsymbol{x}$
- Detect anomalies when $\hat{\phi}_0(\boldsymbol{x}) < \gamma$

Carrera D., Manganini F., Boracchi G., Lanzarone E. *"Defect Detection in SEM Images of Nanofibrous Materials"*, IEEE Transactions on Industrial Informatics 2017, 11 pages, doi:10.1109/TII.2016.2641472



FEATURES EXTRACTED FROM SPARSE CODING

Training:

- **Learn** from S the dictionary D
- **Compute** the sparse representation w.r.t. D, thus features $\boldsymbol{x}$ over the validation set V, such that $V \cap S = \emptyset$
- **Learn** from V, the distribution $\hat{\phi}_0$ of normal features vectors $\boldsymbol{x}$ and the threshold γ.

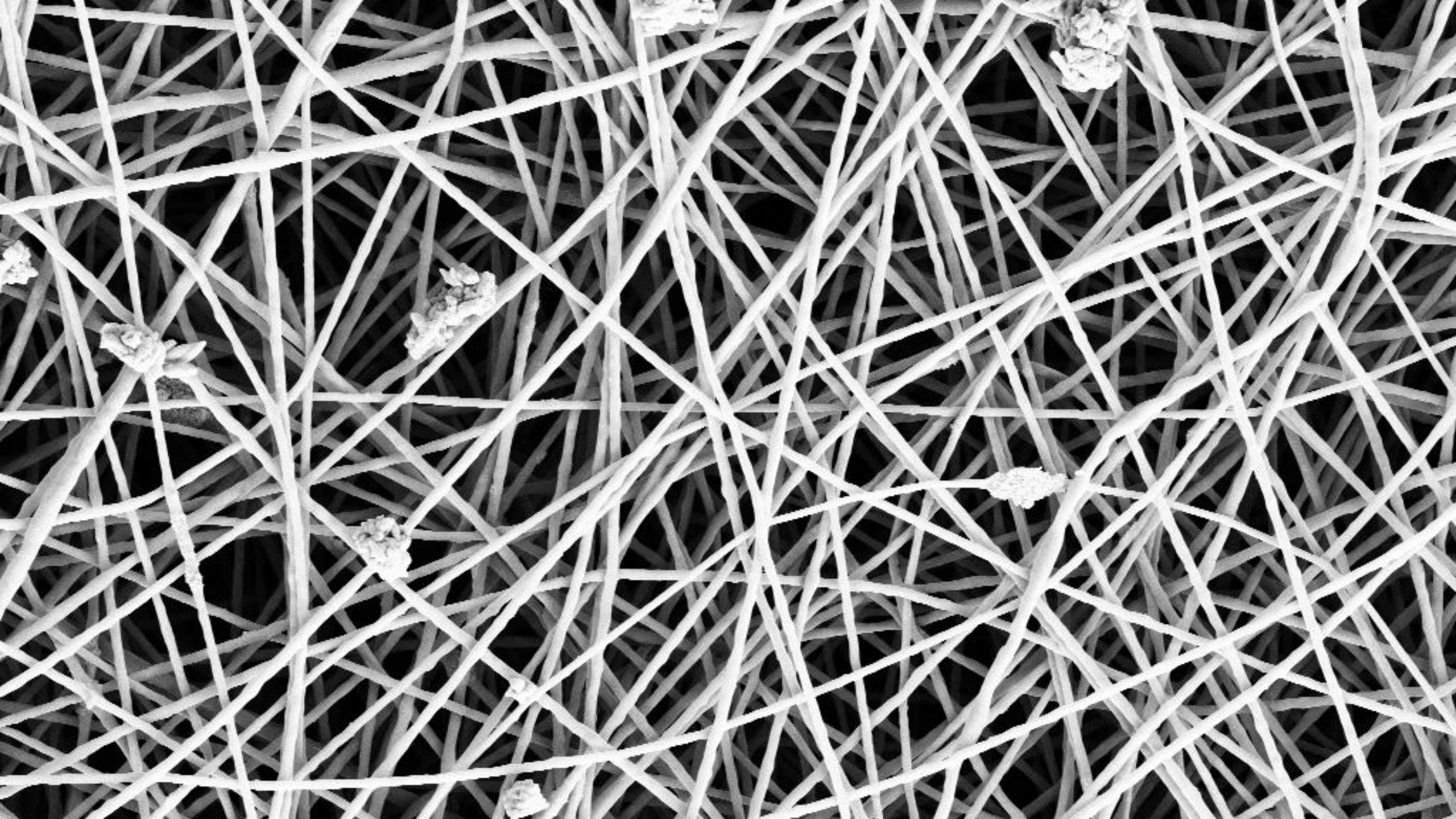
The model for anomaly detection is $(D, \hat{\phi}_0, \gamma)$

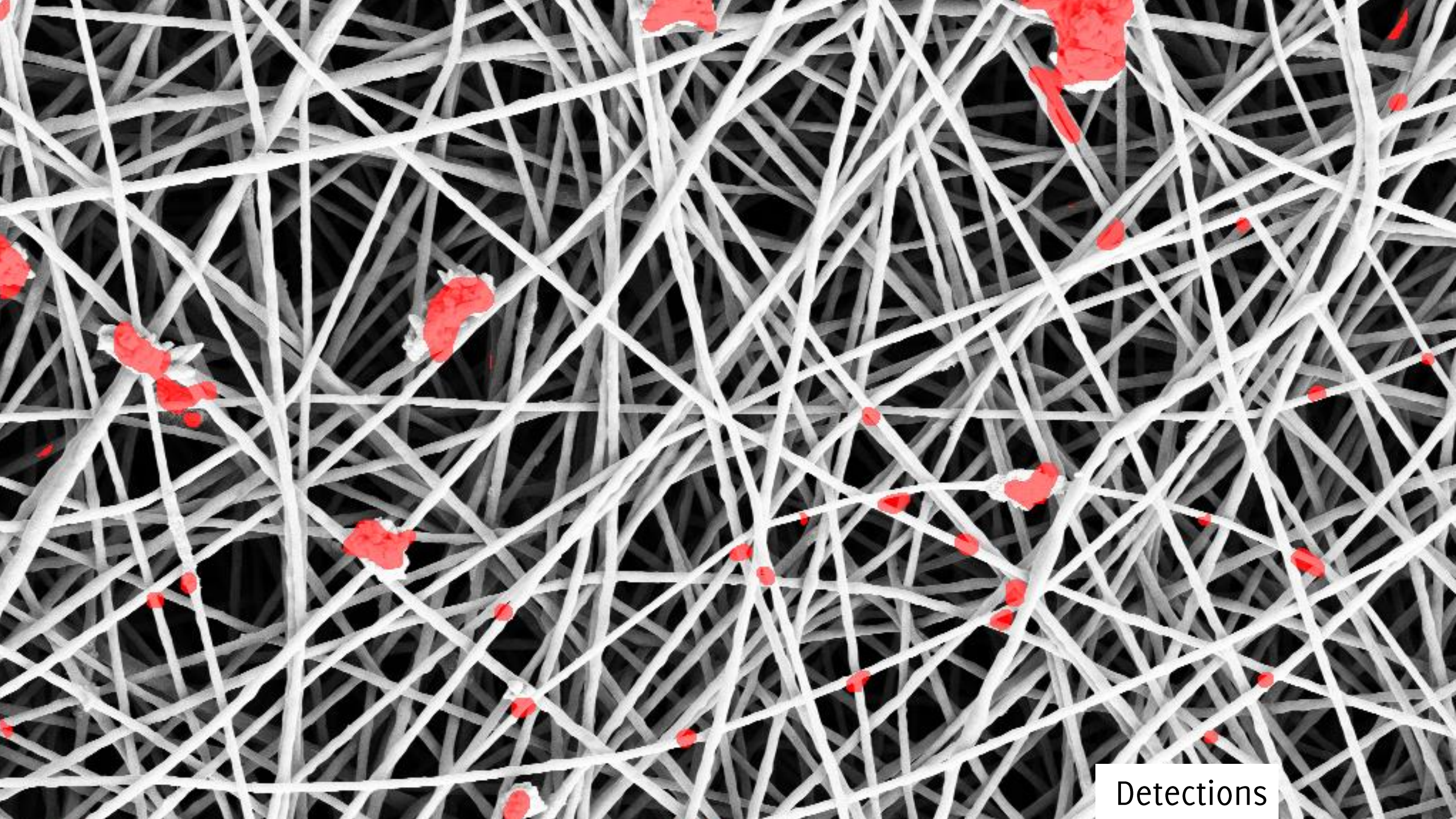
Testing:

- Perform sparse coding of a test signal $\boldsymbol{s}$, thus get the feature vector $\boldsymbol{x}$
- Detect anomalies when $\hat{\phi}_0(\boldsymbol{x}) < \gamma$

This is rather a flexible solution and can be adapted when operating conditions changes (e.g. different zooming level)

Carrera D., Manganini F., Boracchi G., Lanzarone E. *"Defect Detection in SEM Images of Nanofibrous Materials"*, IEEE Transactions on Industrial Informatics 2017, 11 pages, doi:10.1109/TII.2016.2641472




Detections

PART3: DEEP LEARNING FOR ANOMALY DETECTION IN IMAGES

An overview

THE THREE MAJOR INGREDIENTS

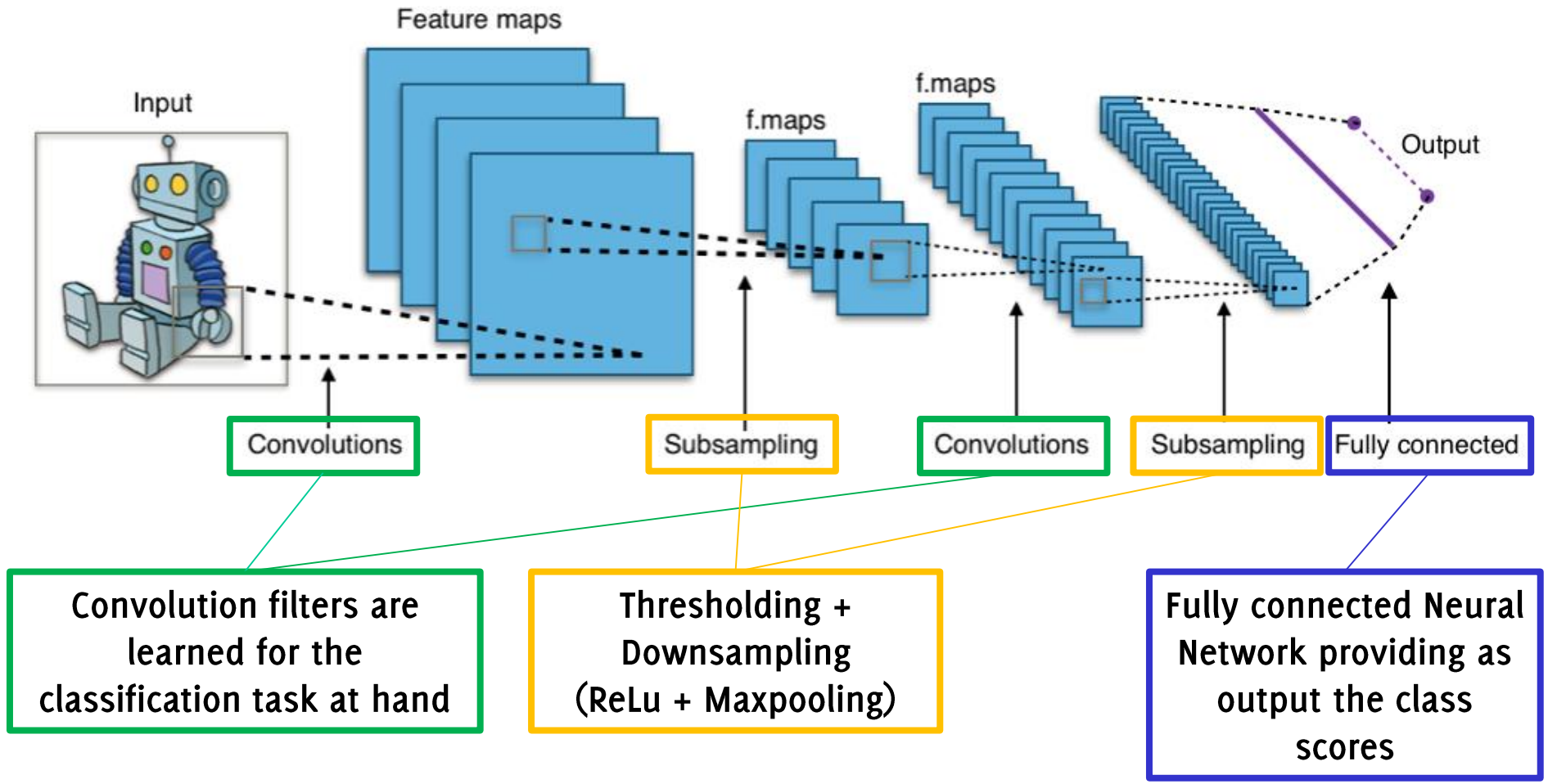
Most detection algorithms have three major ingredients:

- The **background model** $\mathcal{M}$, learned from normal data
- The **statistic / anomaly score**: $\mathrm{err}(\boldsymbol{s}), \mathcal{L}(\boldsymbol{s}), \mathcal{A}(\boldsymbol{s}), \ldots$
- **Decision rule** to detect, e.g. $\mathrm{err}(\boldsymbol{s}) \gtrless \gamma$ possibly controlling the FPR, as in other statistical detection methods

The background model becomes a Deep Neural Network.
In case of images, it is a Deep CNN

RECAP: CONVOLUTIONAL NEURAL NETWORKS (CNN)

The typical architecture of a convolutional neural network



WHY DEEP NN FOR ANOMALY DETECTION?

The rationale behind using deep NN for anomaly detection is that

- they are great at extracting **meaningful representations** from large collection of data.
- they can handle images, signals,

However, it is not clear how to learn representations can be used also for unsupervised learning problems….

SEMI-SUPERVISED APPROACHES

CNN as data-driven feature extractor

- Transfer learning
- Autoencoders
- Self-supervised learning
- Deep One-Class Classification

Generative models

- Variational Auto Encoders (VAEs)
- Generative Adversarial Networks (GANs)

SEMI-SUPERVISED APPROACHES

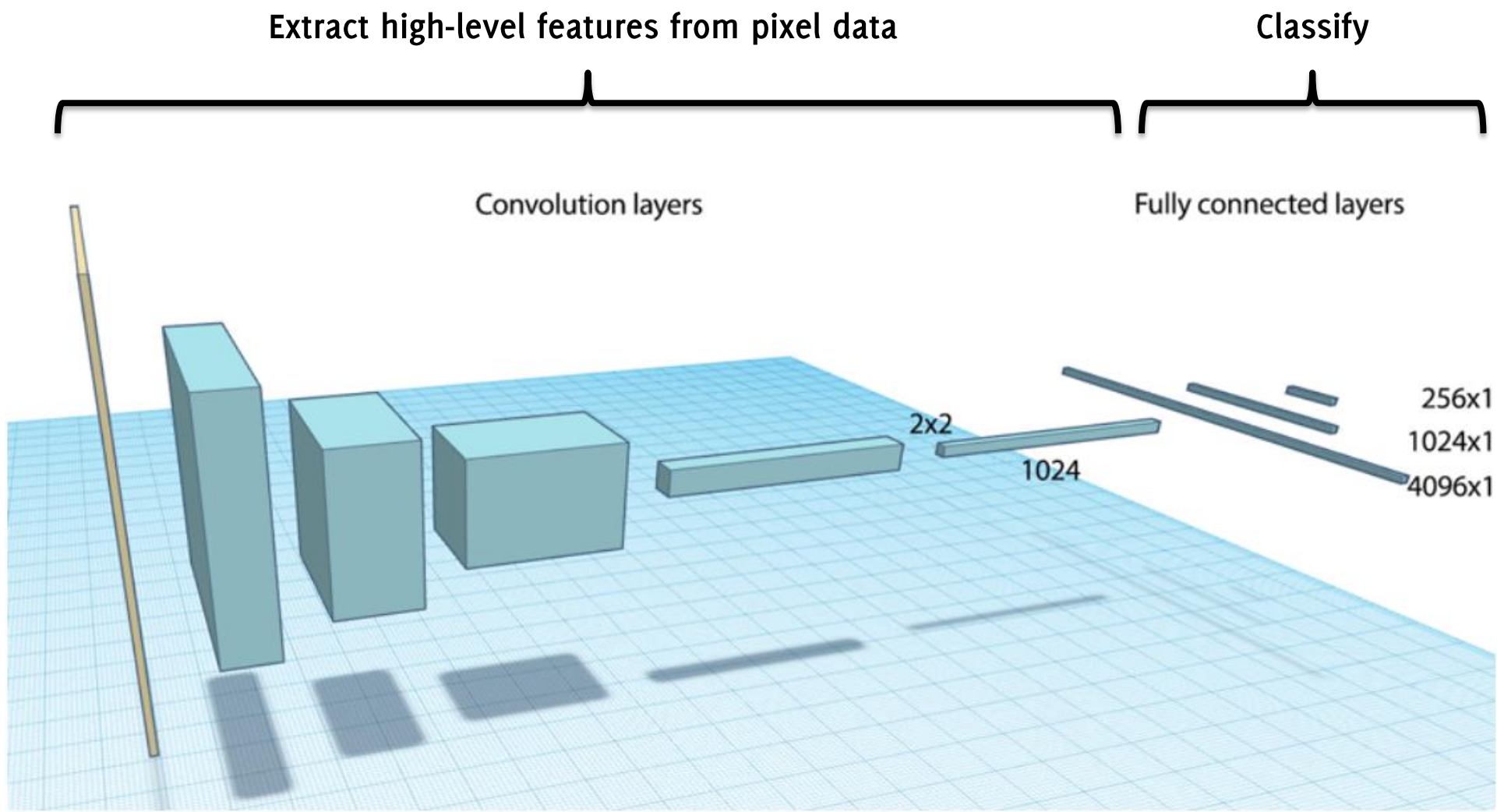
CNN as data-driven feature extractor

- Transfer learning
- Autoencoders
- Self-supervised learning
- Deep One-Class Classification

Generative models

- Variational Auto Encoders (VAEs)
- Generative Adversarial Networks (GANs)

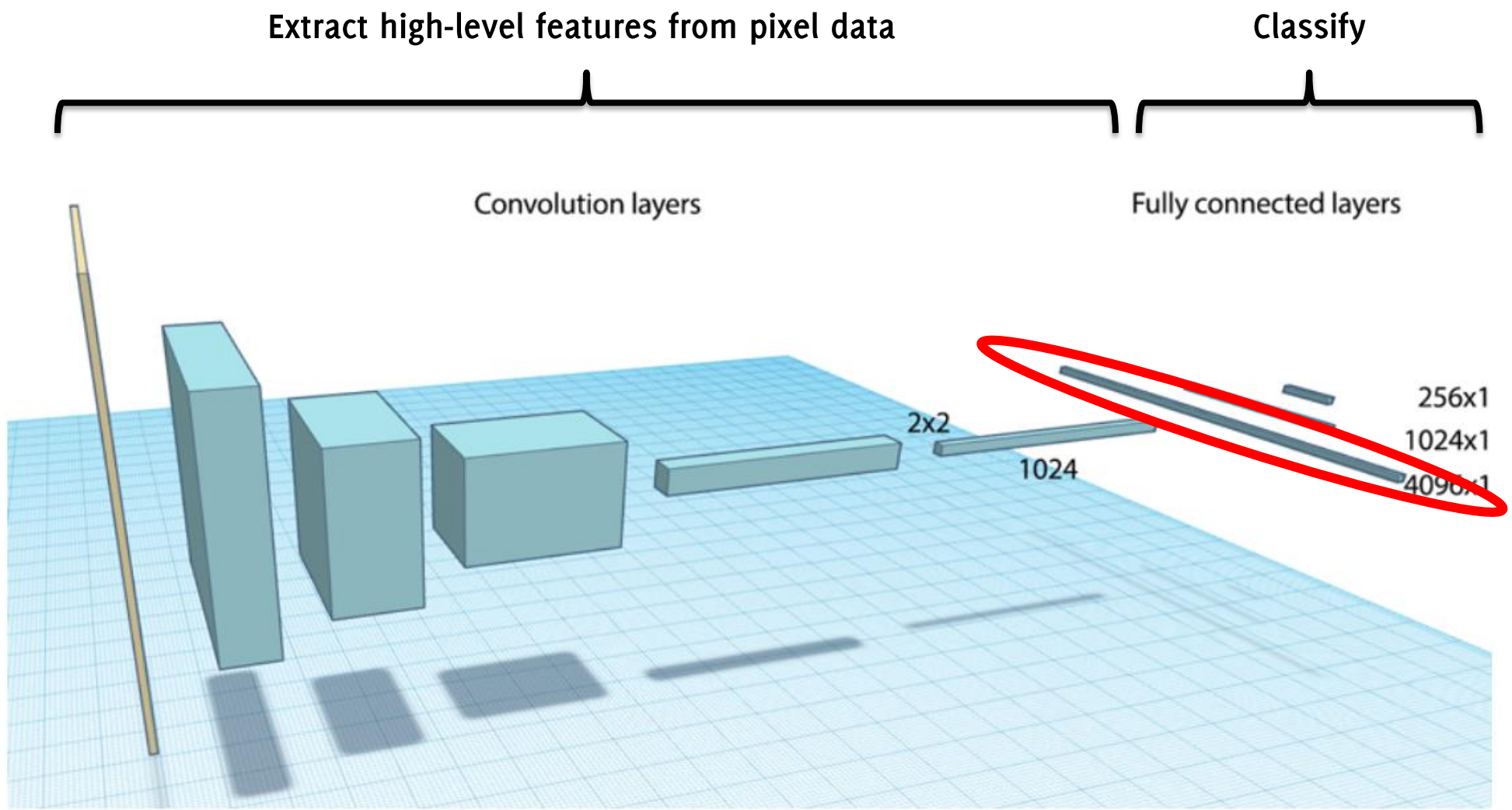
CNN AS DATA-DRIVEN FEATURE EXTRACTOR



Boracchi, 2022

CNN AS DATA-DRIVEN FEATURE EXTRACTOR

The feature vectors extracted from the last layers can be modeled as a random vector



Boracchi, 2022

TRANSFER LEARNING

Idea:

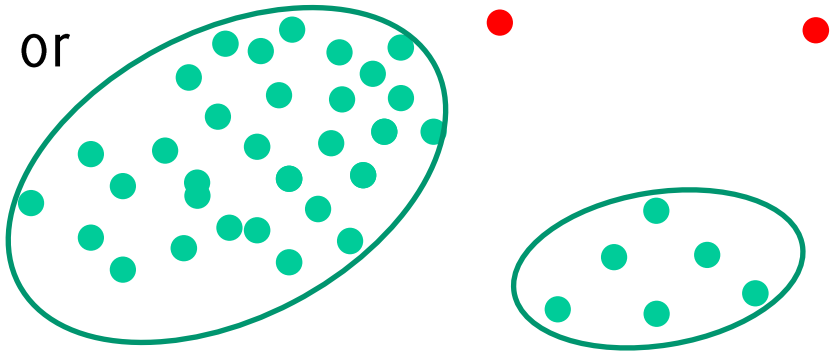
- Use a pretrained network CNN (e.g. AlexNet, VGG, Resnet, EfficientNet), that was trained for classification and on a different training set
- Throw away the last layer(s), these are **powerful discriminative features**
- Use the CNN to build a new dataset TR' from TR:

$$TR' = \{\psi(\boldsymbol{s_i}),\ \boldsymbol{s_i} \in TR\}$$

- Train your favorite anomaly detector on TR'

ANOMALY DETECTION IN THE LATENT SPACE

- Features extracted from a CNN, i.e., $\psi(\boldsymbol{s})$ is typically very large for deep networks (e.g. ResNET). **Reduce data-dimensionality** by PCA defined on a set of normal features
- **Anomalies** can be **detected by measuring distance w.r.t. normal features**, possibly using clustering to speed up performance.
- Thresholds can be computed by the three-sigma rule or bootstrap.



Thus,

- The background model $\mathcal{M}$: a (possibly pre-trianed) CNN
- The statistic / anomaly score: distance in the latent space
- Decision rule to detect, e.g. $\text{err}(\boldsymbol{s}) \gtrless \gamma$

CNN as feature extrctor

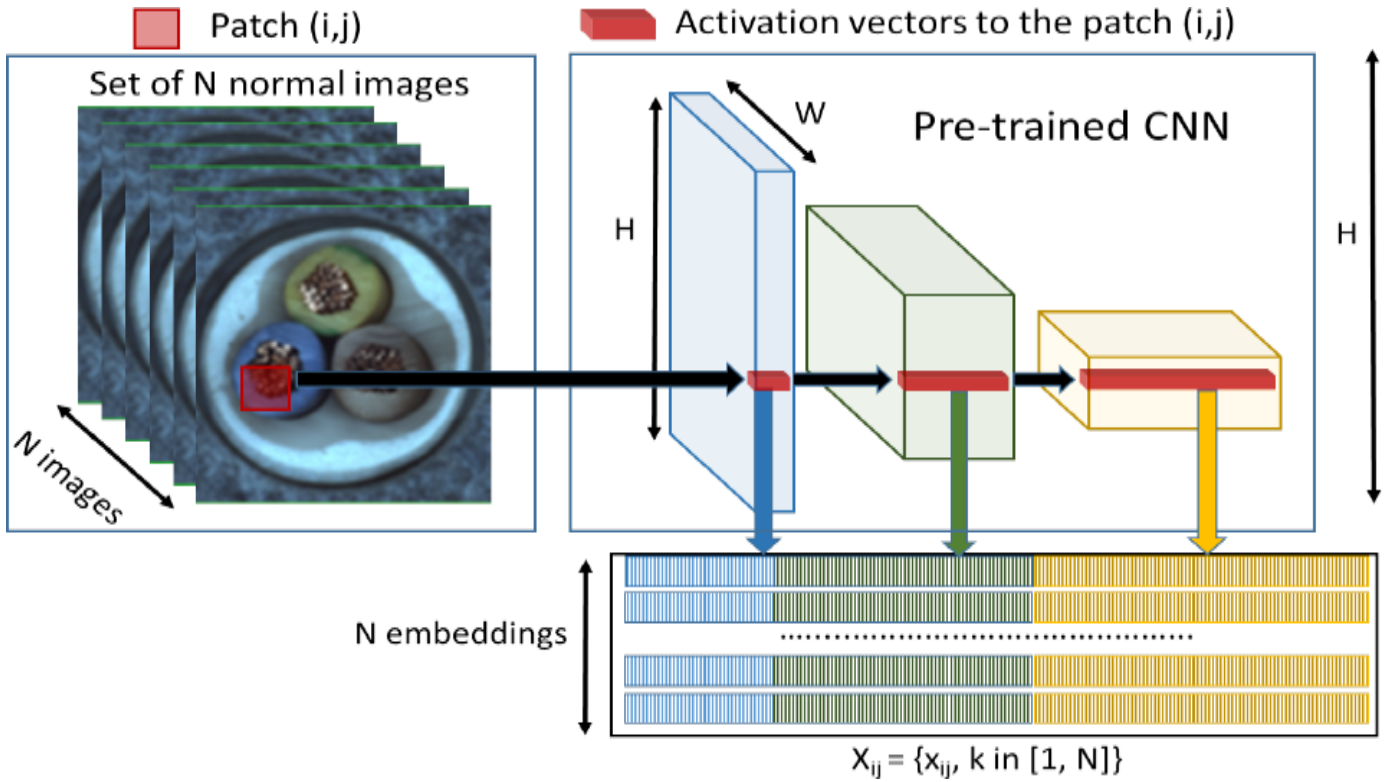
Unsupervised anomaly detection based on clustering

Napoletano P., Piccoli F., Schettini R., *"Anomaly Detection in Nanofibrous Materials by CNN-Based Self-Similarity"*, Sensors 2018

ANOMALY DETECTION IN THE LATENT SPACE

Uses a Pre-trained CNN to embed test patches in a latent space

- The **feature vector stacks activations from different layers**, to obtain a fine-grained and global representation of the input patch
- Possibly **reduce the dimension** of the latent representation by **PCA** or selection of layers for the activations



Defard, Thomas, et al. "PADIM: a patch distribution modeling framework for anomaly detection and localization." ICPR 2020

ANOMALY DETECTION IN THE LATENT SPACE

Uses a Pre-trained CNN to embed test patches in a latent space

- The **feature vector stacks activations from different layers**, to obtain a fine-grained and global representation of the input patch
- Possibly **reduce the dimension** of the latent representation by **PCA** or selection of layers for the activations

A Multivariate Gaussian is fit to describe the distribution of **normal feature vectors**

- Fit a **Multivariate Gaussian density $\hat{\phi}_{r,c}$ per each patch location** in the image (!)

The **anomaly score** $\mathcal{A}(\cdot)$ is the Mahalanobis distance w.r.t. the density model… that in case of Gaussians it corresponds to the likelihood

$$\mathcal{L}(\boldsymbol{x}_{r,c}) = \log(\hat{\phi}_{r,c}(\boldsymbol{x}_{r,c})) = (\boldsymbol{x}_{r,c} - \boldsymbol{\mu}_{r,c})' \, \Sigma_{r,c}^{-1} \, (\boldsymbol{x}_{r,c} - \boldsymbol{\mu}_{r,c})$$

Defard, Thomas, et al. "Padim: a patch distribution modeling framework for anomaly detection and localization.“ ICPR 2020

ANOMALY DETECTION IN THE LATENT SPACE

Uses a Pre-trained CNN to embed test patches in a latent space

- The **feature vector stacks activations from different layers**, to obtain a fine-grained and global representation of the input patch
- Possibly **reduce the dimension** of the latent representation by **PCA** or selection of layers for the activations

CNN as feature extrctor

A Multivariate Gaussian is fit to describe the distribution of **normal feature vectors**

- Fit a **Multivariate Gaussian density** $\hat{\phi}_{r,c}$ **per each patch location** in the image (!)

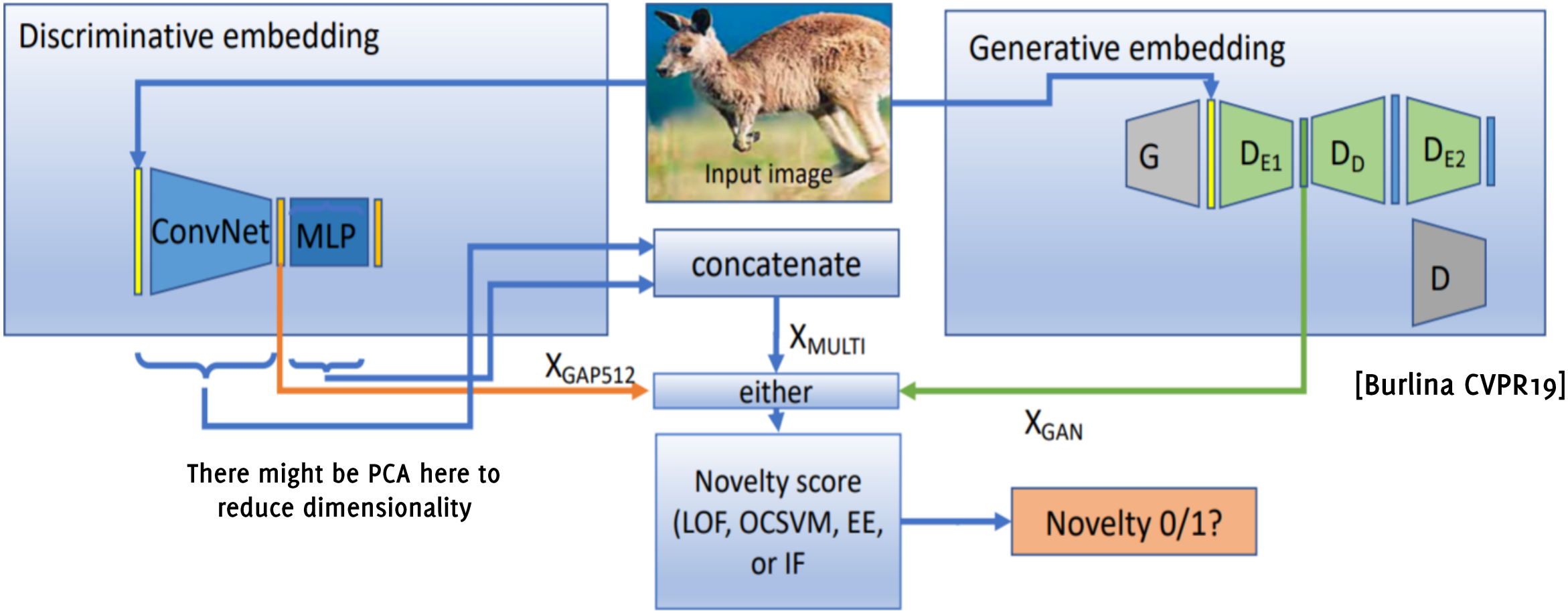
The **anomaly score** $\mathcal{A}(\cdot)$ is the Mahalanobis distance w.r.t. the density model… that in case of Gaussians it corresponds to the likelihood

$$\mathcal{L}(\boldsymbol{x}_{r,c}) = \log(\hat{\phi}_{r,c}(\boldsymbol{x}_{r,c})) = (\boldsymbol{x}_{r,c} - \boldsymbol{\mu}_{r,c})' \, \Sigma_{r,c}^{-1} \, (\boldsymbol{x}_{r,c} - \boldsymbol{\mu}_{r,c})$$

Anomaly Detection In the «random variable word» by denisty models

Defard, Thomas, et al. "Padim: a patch distribution modeling framework for anomaly detection and localization." ICPR 2020

…. AND MANY MORE APPROACHES ALONG THE SAME LINE



[Burlina CVPR19]

Burlina, Philippe, Neil Joshi, and I. Wang. "Where's Wally now? Deep generative and discriminative embeddings for novelty detection." *CVPR 2019*

Sabokrou, Mohammad, et al. "Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes." *Computer Vision and Image Understanding* 172 (2018): 88-97.

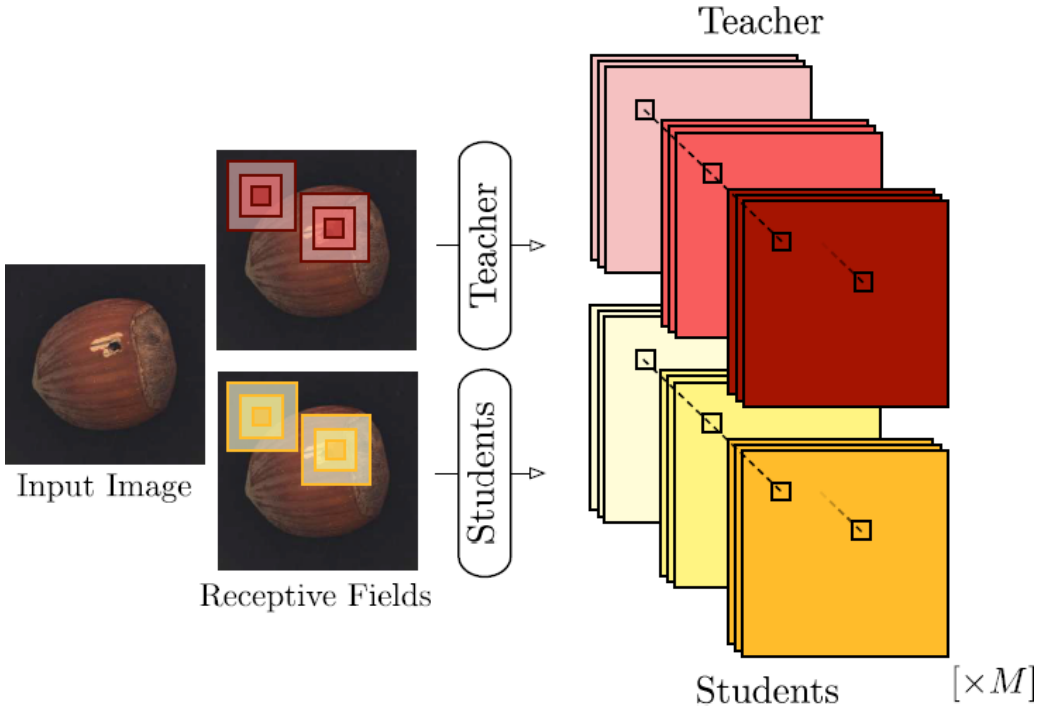
STUDENT – TEACHER MODEL FOR ANOMALY DETECTION

A teacher is a pre-trained classification CNN that outputs a feature map

- It is possibly distilled and made fully convolutional for speedup

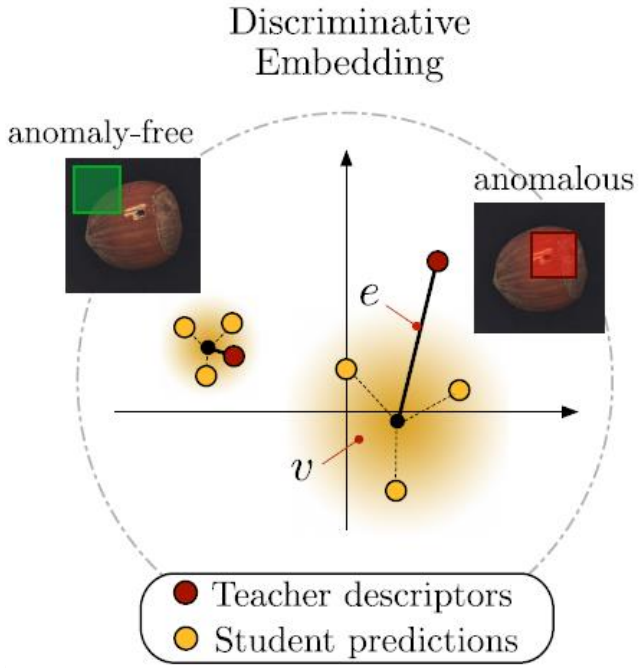
Students form an ensemble and have the same architecture a the teacher.

Students are trained to regress the output of the teacher ***on normal data*** (unsupervised, minimization of ℓ^2 distance in the feature space)



Bergmann, Paul, et al. "*Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings.*" CVPR 2020

STUDENT – TEACHER MODEL FOR ANOMALY DETECTION



Thus, during inference, for each pixel we measure:

- **The discrepancy** between the ensemble output $\boldsymbol{\mu}_{(r,c)}$, which corresponds to the average prediction of the students, and the teacher prediction $\boldsymbol{y}_{(r,c)}$

$$e_{(r,c)} = \left\| \boldsymbol{\mu}_{(r,c)} - \boldsymbol{y}_{(r,c)} \right\|^2$$

Students are not expected to predict features from anomalous samples.

- The **predictive uncertainty** of the ensemble, as the extent to which the average norm of the students predictoins departs from norm of the ensemble

$$v_{(r,c)} = \frac{1}{M} \sum_{S_i} \left\| \boldsymbol{\mu}_{(r,c)}^{S_i} \right\|^2 - \left\| \boldsymbol{\mu}_{(r,c)} \right\|^2$$

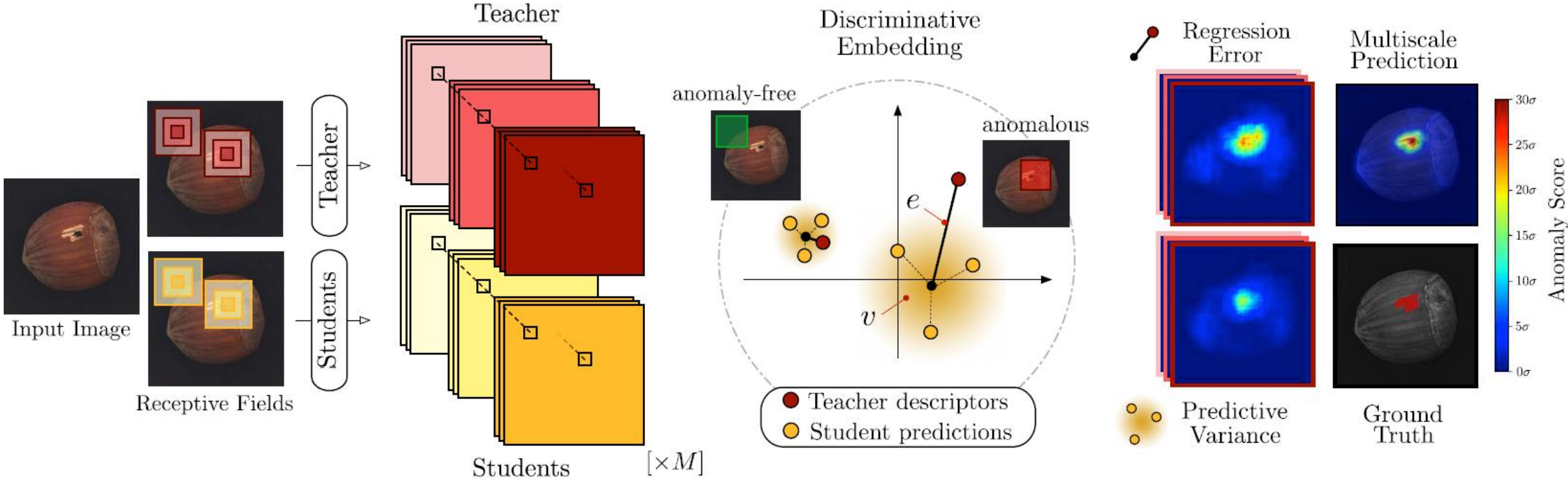
Students are less confident in their prediction, and different among them, on anomalous samples. They generalize more poorly.

Bergmann, Paul, et al. "*Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings.*" CVPR 2020

STUDENT – TEACHER MODEL FOR ANOMALY DETECTION

The image is processed in a fully-convolutional manner and the two scores (prediction error and predictive uncertainness) are then normalized over a validation set and summed to yield a unique anomaly score.

Multiscale variants of the scores are computed to improve detection performance



Bergmann, Paul, et al. "*Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings.*" CVPR 2020

TRANSFER LEARNING

Pros:

- Pre-trained networks are very powerful, since they were usually trained on datasets with million of images
- Very easy and practical to implement

Cons:

- the network is **not trained on normal** data. Meaningful structures in normal images might not be successfully captured by network trained on images from a different domain (e.g. medical vs natural images)
- The **anomaly detector and the CNN are not jointly learned**, while the end-to-end learning strategy is the key to achieve impressive results in supervised tasks.

Personal opinion: I am rather sceptical about these approaches as a pre-trained model might have seen anomalies during training. In this case, performance might be too optimistic compared to anomalies in general

Boracchi, 2022

SEMI-SUPERVISED APPROACHES

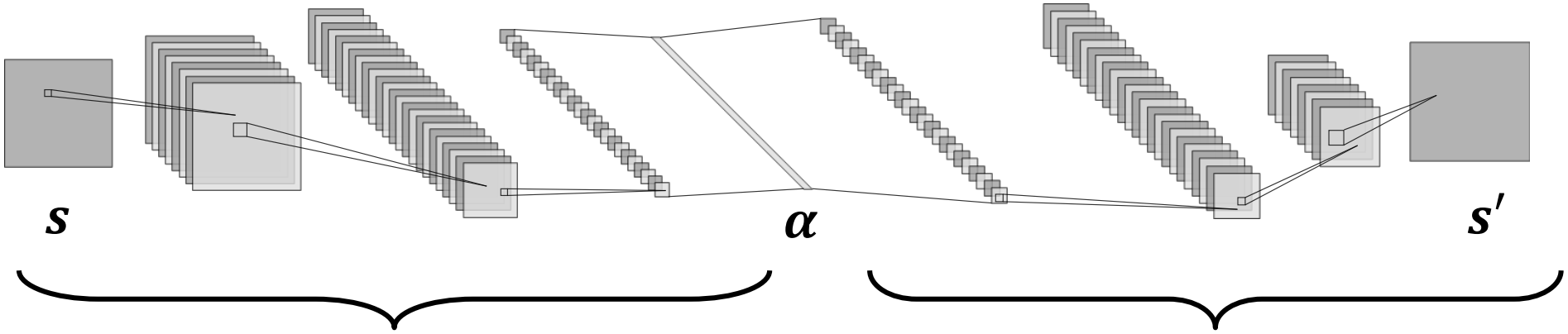
CNN as data-driven feature extractor

- Transfer learning
- Autoencoders
- Self-supervised learning
- Deep One-Class Classification

Generative models

- Variational Auto Encoders (VAEs)
- Generative Adversarial Networks (GANs)

AUTOENCODERS (REVISITED)



Autoencoders can be trained directly on normal data by minimizing the reconstruction loss measured in a least-square sense

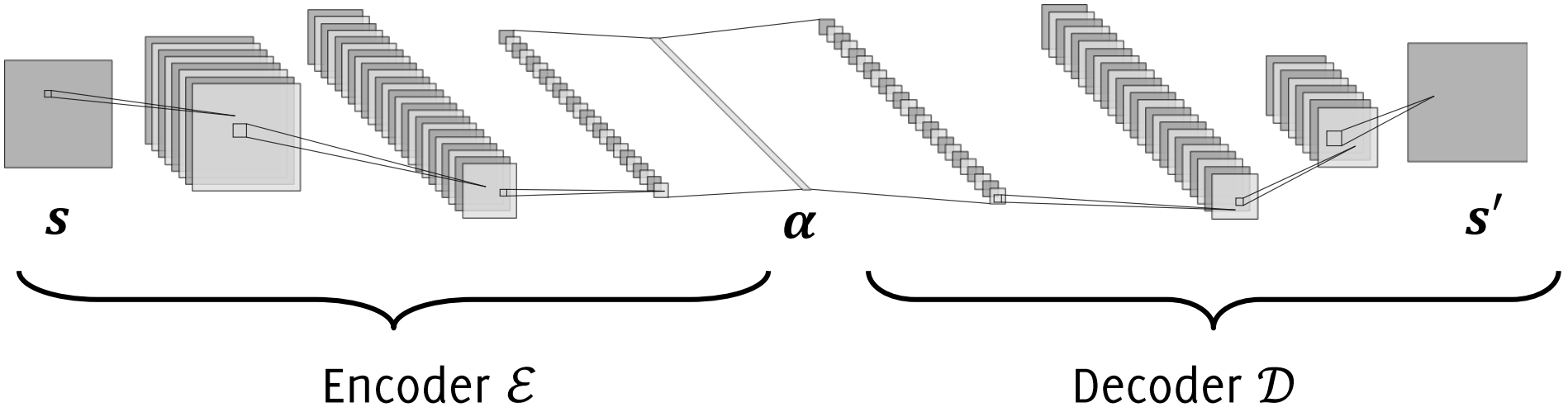
$$\ell^2(\boldsymbol{s} - \hat{\boldsymbol{s}}) = \sum_{\boldsymbol{s} \in TR} \left\| \boldsymbol{s} - \mathcal{D}\big(\mathcal{E}(\boldsymbol{s})\big) \right\|_2$$

Structural similarity measures (e.g. $SSIM(\boldsymbol{s}, \hat{\boldsymbol{s}})$) could also be used and possibly combined with the squared error in the loss used for training

$$\mathcal{L}(\boldsymbol{s}, \hat{\boldsymbol{s}}) = \ell^2(\boldsymbol{s} - \hat{\boldsymbol{s}}) + \lambda\, SSIM(\boldsymbol{s}, \hat{\boldsymbol{s}})$$

Bergmann et al. "Improving unsupervised defect segmentation by applying structural similarity to autoencoders" arXiv preprint 2018

AUTOENCODERS: ANOMALY DETECTION BY MONITORING THE RECONSTRUCTION ERROR



The autoencoder $\mathcal{D}(\mathcal{E}(\cdot))$ is trained TR containing exclusively normal instances.

- The AE is expected to reconstruct well normal instances, and poorly anomalous ones
- The AE uses as **anomaly score the same loss used for training**

The lower the reconstruction quality, the higher the anomaly score.

To process the entire image efficiently, the AE can be made **fully-convolutional**

AUTOENCODERS: ANOMALY DETECTION BY MONITORING THE RECONSTRUCTION ERROR

In practice adopting a **multiscale loss function** during training can substantially improve AD performance.

We have experienced that simple AutoEncoders trained using multiscale losses can match the performance of deeper models, which however might not be always trainable on small TR

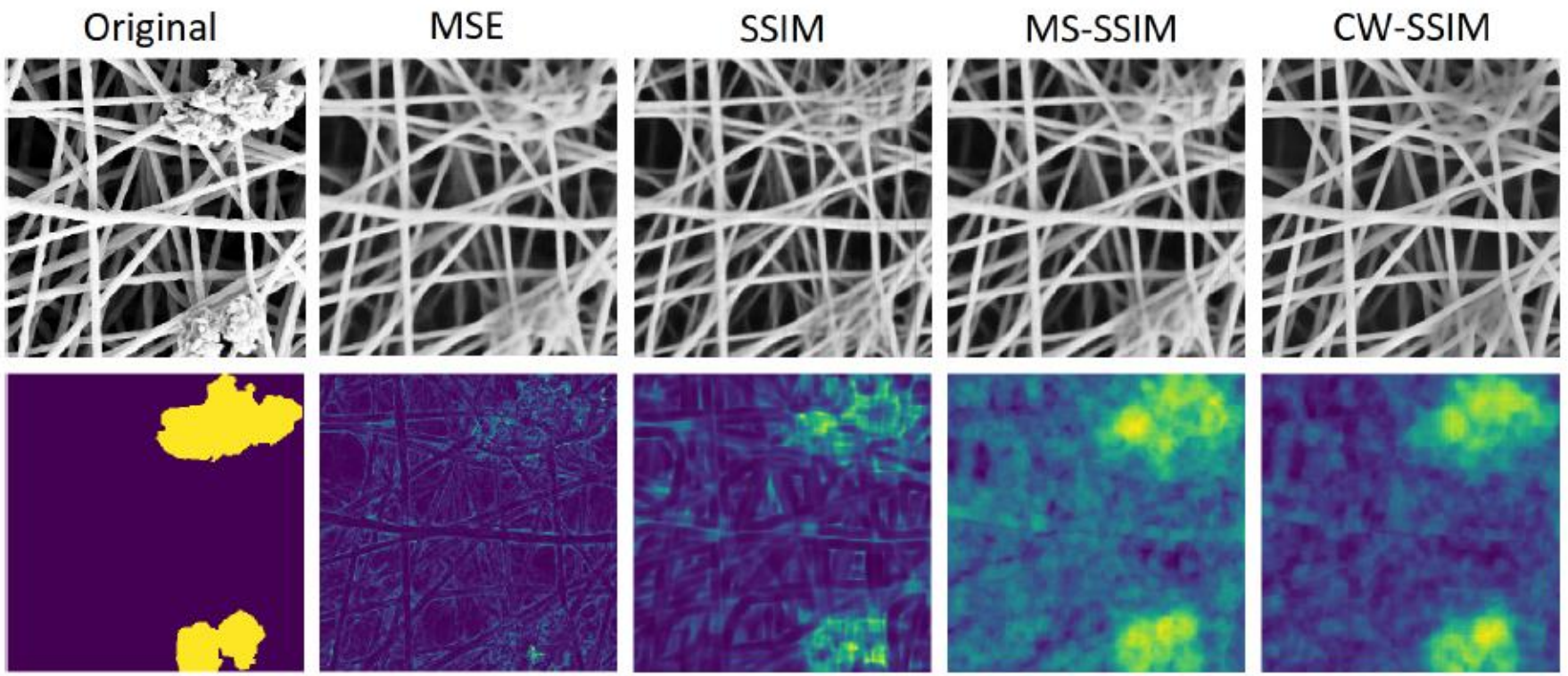
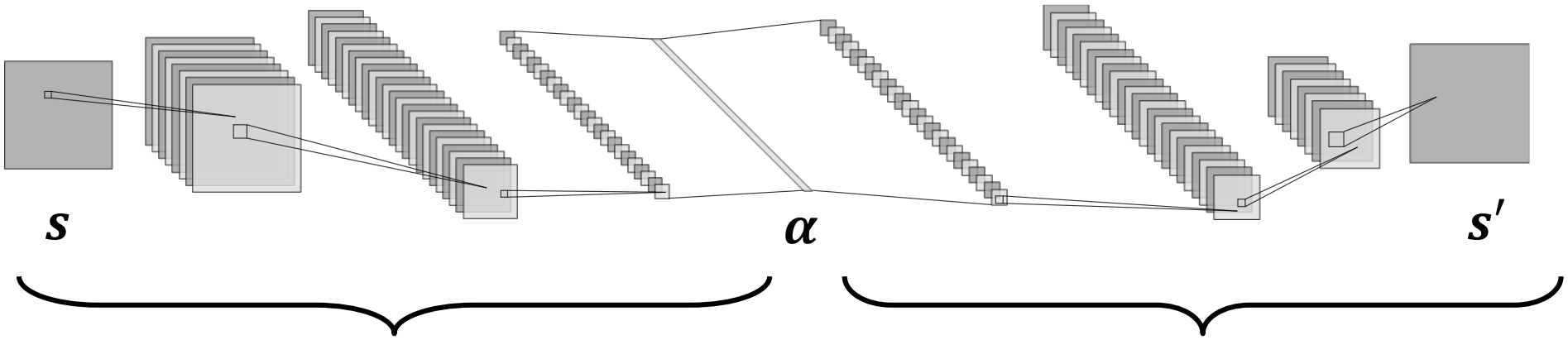


Fig. 2. An example of reconstruction and anomaly scores produced by autoencoders trained with different loss functions from a Nanofiber image. This image shows that autoencoders trained with structural similarity metrics, and CW-SSIM in particular, yield better reconstruction quality and superior anomaly detection performance than a traditional MSE autoencoder.

Credits Andrea Bionda Thesis

AUTOENCODERS: FEATURE BASED MONITORING



We can fit a **density model** (e.g. Gaussian Mixture) on $\boldsymbol{\alpha} = \mathcal{E}(\boldsymbol{s})$:

$$\boldsymbol{\alpha} \sim \sum_i \pi_i \varphi_{\boldsymbol{\mu}_i, \Sigma_i},$$

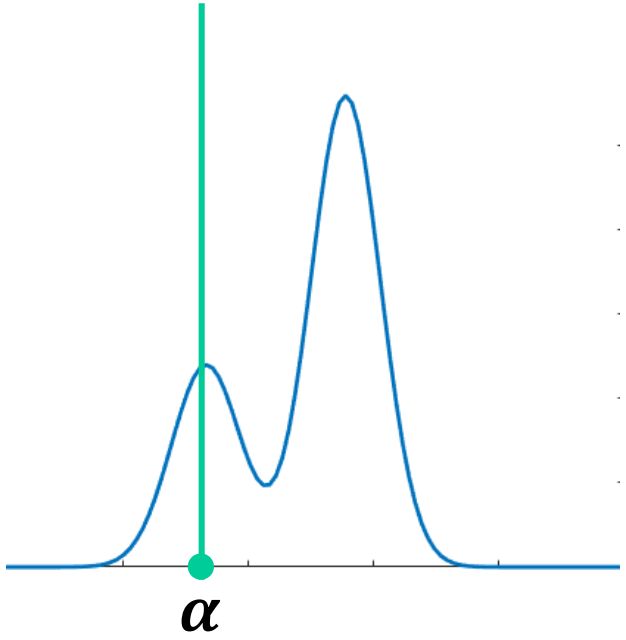
Where $\varphi_{\boldsymbol{\mu}_i, \Sigma_i}$ is the pdf of $\mathcal{N}(\boldsymbol{\mu_i}, \Sigma_i)$

EM-ALGORITHM FOR GAUSSIAN MIXTURES

Estimation of Gaussian Mixture parameters $\{\pi_i, \boldsymbol{\mu}_i, \Sigma_i\}$ from a training set $\{\boldsymbol{\alpha}_n\}_n$is typically performed via EM-algorithm, that iterates the E and M steps

- **E-step:** compute the membership weights $\gamma_{n,i}$ for each training sample $\boldsymbol{\alpha}_n$

$$\gamma_{n,i} = \frac{\pi_i \varphi_{\boldsymbol{\mu}_i, \Sigma_i}(\boldsymbol{\alpha}_n)}{\sum_k \pi_k \varphi_{\boldsymbol{\mu}_k, \Sigma_k}(\boldsymbol{\alpha}_n)}$$



Bishop, "Pattern recognition and machine learning"

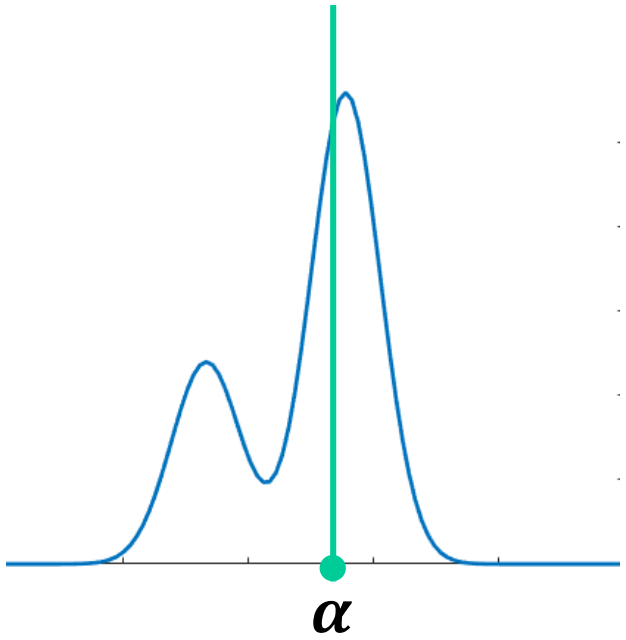
$$\gamma_1 \sim 1$$
$$\gamma_2 \sim 0$$

EM-ALGORITHM FOR GAUSSIAN MIXTURES

Estimation of Gaussian Mixture parameters $\{\pi_i, \boldsymbol{\mu}_i, \Sigma_i\}$ from a training set $\{\boldsymbol{\alpha}_n\}_n$is typically performed via EM-algorithm, that iterates the E and M steps

- **E-step:** compute the membership weights $\gamma_{n,i}$ for each training sample $\boldsymbol{\alpha}_n$

$$\gamma_{n,i} = \frac{\pi_i \varphi_{\boldsymbol{\mu}_i, \Sigma_i}(\boldsymbol{\alpha}_n)}{\sum_k \pi_k \varphi_{\boldsymbol{\mu}_k, \Sigma_k}(\boldsymbol{\alpha}_n)}$$



Bishop, "Pattern recognition and machine learning"

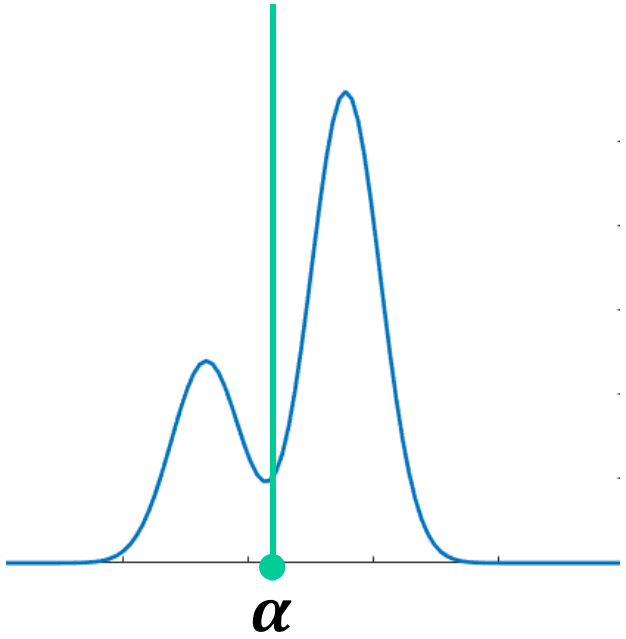
$$\gamma_1 \sim 0$$
$$\gamma_2 \sim 1$$

EM-ALGORITHM FOR GAUSSIAN MIXTURES

Estimation of Gaussian Mixture parameters $\{\pi_i, \boldsymbol{\mu}_i, \Sigma_i\}$ from a training set $\{\boldsymbol{\alpha}_n\}_n$is typically performed via EM-algorithm, that iterates the E and M steps

- **E-step:** compute the membership weights $\gamma_{n,i}$ for each training sample $\boldsymbol{\alpha}_n$

$$\gamma_{n,i} = \frac{\pi_i \varphi_{\boldsymbol{\mu}_i, \Sigma_i}(\boldsymbol{\alpha}_n)}{\sum_k \pi_k \varphi_{\boldsymbol{\mu}_k, \Sigma_k}(\boldsymbol{\alpha}_n)}$$



Bishop, "Pattern recognition and machine learning"

$$\gamma_1 \sim \frac{1}{2}$$
$$\gamma_2 \sim \frac{1}{2}$$

EM-ALGORITHM FOR GAUSSIAN MIXTURES

Estimation of Gaussian Mixture parameters $\{\pi_i, \boldsymbol{\mu}_i, \Sigma_i\}$ from a training set $\{\boldsymbol{\alpha}_n\}_n$is typically performed via EM-algorithm, that iterates the E and M steps

- **E-step:** compute the membership weights $\gamma_{n,i}$ for each training sample $\boldsymbol{\alpha}_n$

$$\gamma_{n,i} = \frac{\pi_i \varphi_{\boldsymbol{\mu}_i, \Sigma_i}(\boldsymbol{\alpha}_n)}{\sum_k \pi_k \varphi_{\boldsymbol{\mu}_k, \Sigma_k}(\boldsymbol{\alpha}_n)}$$

- **M-step:** update the parameters of the Gaussian Mixture

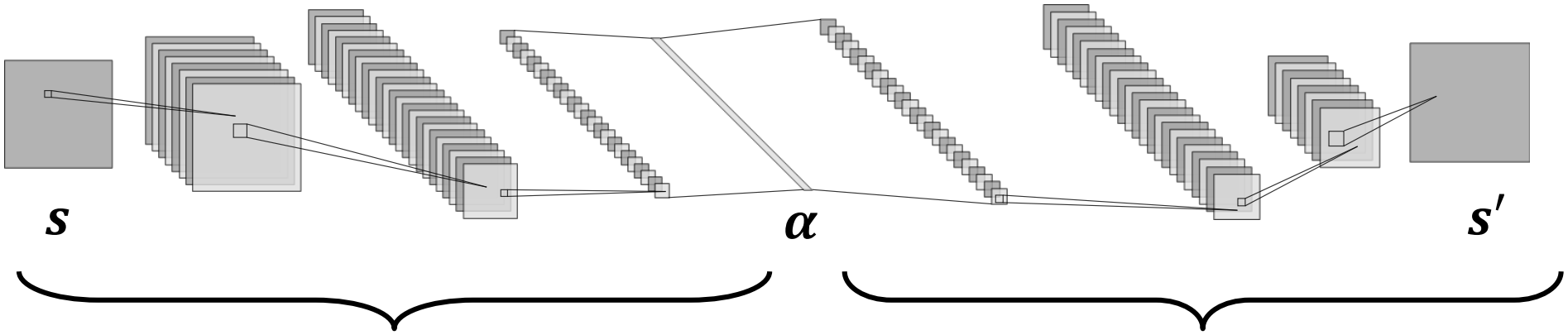
$$\pi_i = \frac{1}{N} \sum_n \gamma_{n,i}$$

$$\mu_i = \frac{\sum_n \gamma_{n,i} \boldsymbol{\alpha}_n}{\Sigma_n \gamma_{n,i}}$$

$$\Sigma_i = \frac{\sum_n \gamma_{n,i} (\boldsymbol{\alpha}_n - \boldsymbol{\mu}_i)(\boldsymbol{\alpha}_n - \boldsymbol{\mu}_i)^T}{\Sigma_n \gamma_{n,i}}$$

Bishop, “Pattern recognition and machine learning”

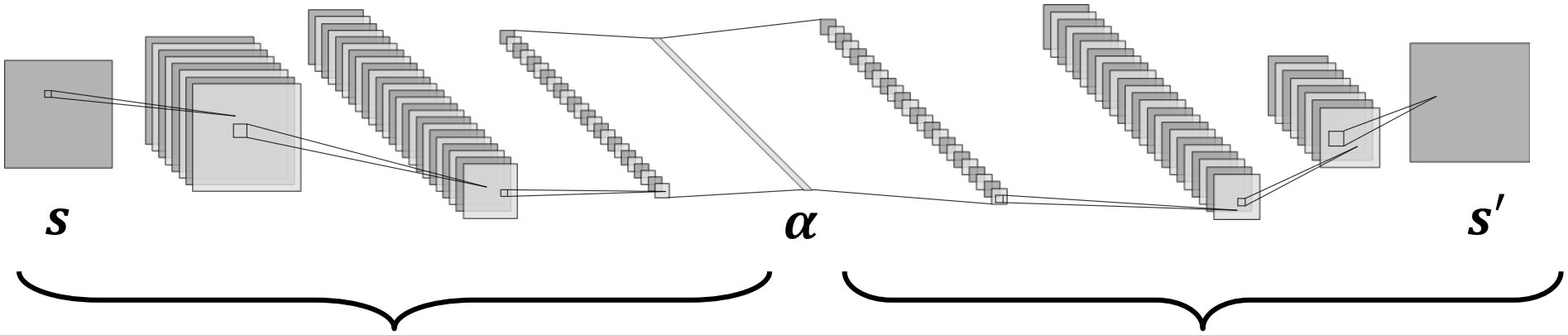
AUTOENCODERS (REVISITED)



We can compute the likelihood of a test sample $\boldsymbol{s}$ as:

$$\mathcal{L}(\boldsymbol{s}) = \sum_i \pi_i \varphi_{\boldsymbol{\mu}_i, \Sigma_i}(\mathcal{E}(\boldsymbol{s}))\,,$$

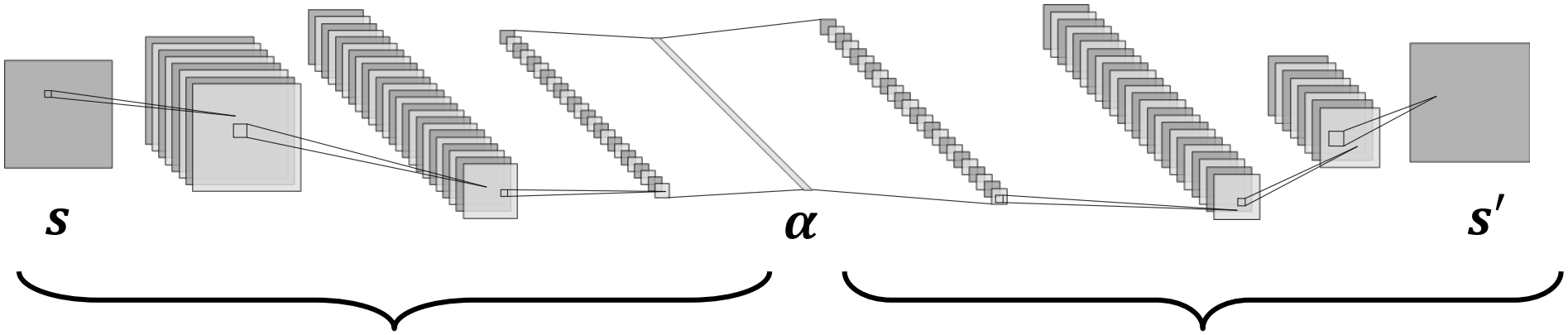
AUTOENCODERS (REVISITED)



We can compute the likelihood of a test sample $\boldsymbol{s}$ as:

$$\mathcal{L}(\boldsymbol{s}) = \sum_i \pi_i \varphi_{\boldsymbol{\mu}_i, \Sigma_i}(\mathcal{E}(\boldsymbol{s}))\,,$$

AUTOENCODERS (REVISITED)



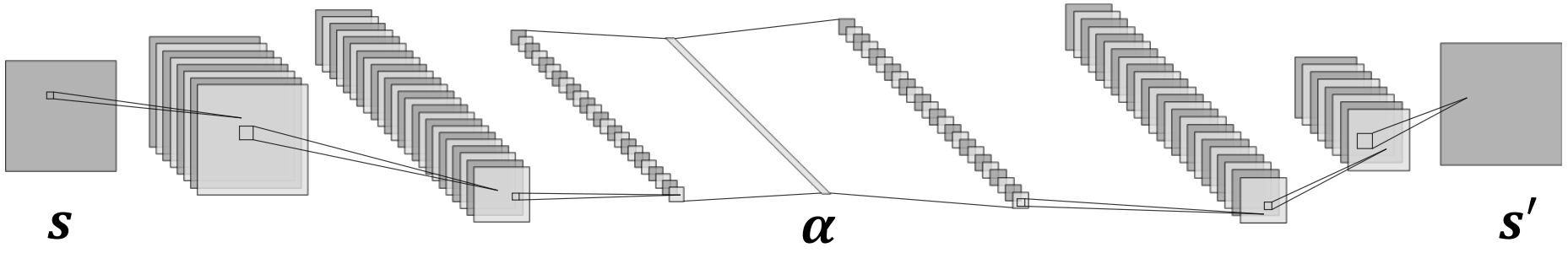
We can compute the likelihood of a test sample $\boldsymbol{s}$ as:

$$\mathcal{L}(\boldsymbol{s}) = \sum_i \pi_i \varphi_{\boldsymbol{\mu}_i, \Sigma_i}(\mathcal{E}(\boldsymbol{s}))\,,$$

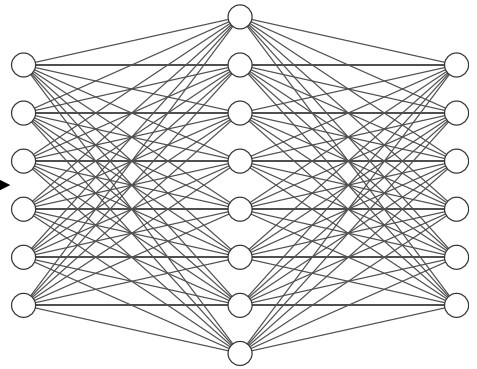
Limitation: The autoencoder and the Gaussian Mixture are not jointly learned!

DEEP AUTOENCODING GAUSSIAN MIXTURE MODEL

Idea: train a NN on TR to predict the membership weights of each normal sample



Estimation Network



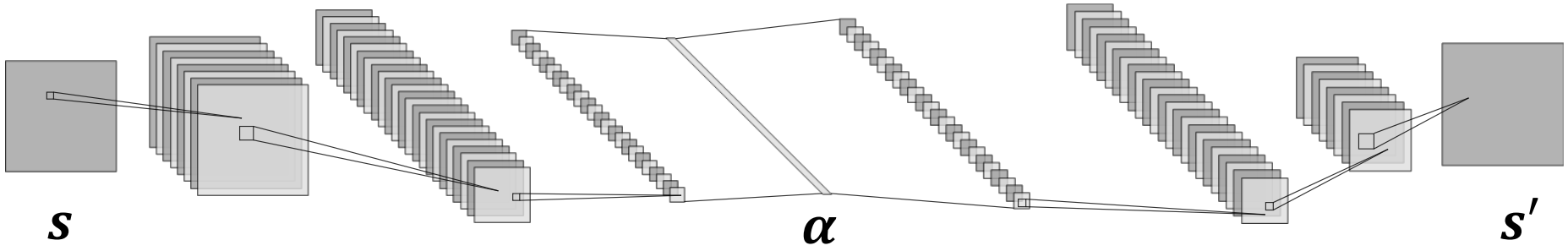
α γ

This can be optimized during AE training

Zong et al, *"Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection", ICLR 2018*

DEEP AUTOENCODING GAUSSIAN MIXTURE MODEL

Idea: Idea: train a NN on TR to predict the membership weights of each normal sample



Estimate the GM parameters as:

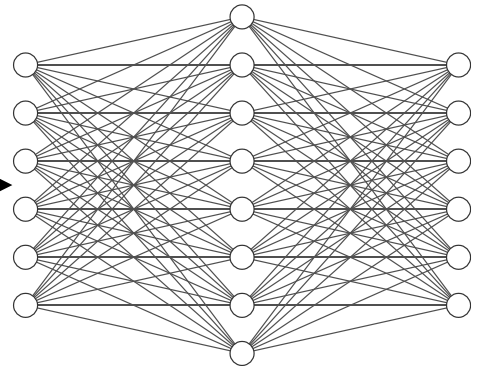
$$\pi_i = \frac{1}{N}\sum_n \gamma_{n,i}$$

$$\mu_i = \frac{\sum_n \gamma_{n,i}\boldsymbol{\alpha}_n}{\sum_n \gamma_{n,i}}$$

$$\Sigma_i = \frac{\sum_n \gamma_{n,i}(\boldsymbol{\alpha}_n - \boldsymbol{\mu}_i)(\boldsymbol{\alpha}_n - \boldsymbol{\mu}_i)^T}{\sum_n \gamma_{n,i}}$$

M-step

Estimation Network



E-step

Zong et al, *"Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection", ICLR 2018*

DEEP AUTOENCODING GAUSSIAN MIXTURE MODEL

Train the autoencoder minimizing the loss:

$$\min \sum_{\boldsymbol{s}} \left|\left|\boldsymbol{s} - \mathcal{D}\big(\mathcal{E}(\boldsymbol{s})\big)\right|\right|_2^2 + \lambda \mathcal{R}(\mathcal{E}(\boldsymbol{s}))$$

Where

- $\left|\left|\boldsymbol{s} - \mathcal{D}\big(\mathcal{E}(\boldsymbol{s})\big)\right|\right|_2^2$ is the reconstruction error
- $\mathcal{R}(\boldsymbol{\alpha}) = -\log \sum_i \pi_i \varphi_{\boldsymbol{\mu}_i, \Sigma_i}(\boldsymbol{\alpha})$ plays the role of the likelihood, as it models the probability that we could observe the input sample, according to the distribution of normal data. The higher, the more $\mathcal{E}(\boldsymbol{s})$ are likely to be generate from the GM

Additional regularizations has to be imposed on Σ_i to avoid trivial/degenerate solutions

$\mathcal{R}(\mathcal{E}(\boldsymbol{s}))$ can be used an anomaly score for a sample $\boldsymbol{s}$

Zong et al, *"Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection", ICLR 2018*

SOME REMARKS

- The **estimation network** introduces a **regularization** that helps to avoid local optima of the recontruction error
- The autoencoder is then able to extract meaningful feature from normal data
- Density estimation enables anomaly detection, but it is a **more complicated** task

SEMI-SUPERVISED APPROACHES

CNN as data-driven feature extractor

- Transfer learning
- Autoencoders
- Self-supervised learning
- Deep One-Class Classification

Generative models

- Variational Auto Encoders (VAEs)
- Generative Adversarial Networks (GANs)

SELF-SUPERVISED LEARNING

With transfer learning we use a model trained on a **different dataset** (e.g., Imagenet) to address a **different task** (e.g., classification).

The idea of self-supervised learning is to train a model on the **target dataset** but solving a **different task,** for which we can easily obtain the labels.

SELF-SUPERVISED LEARNING

We can **build** a **labeled dataset** for multiclass **classification** from normal data

- Consider a set of T transformation $\mathcal{T} = \{\tau_1, \dots, \tau_T\}$
- Apply each transformation τ_i to every $\mathbf{s} \in TR$:

$$TR_{new} = \{(\tau_i(\boldsymbol{s}), i) \mid \boldsymbol{s} \in TR, i = 1, \dots, T\}$$

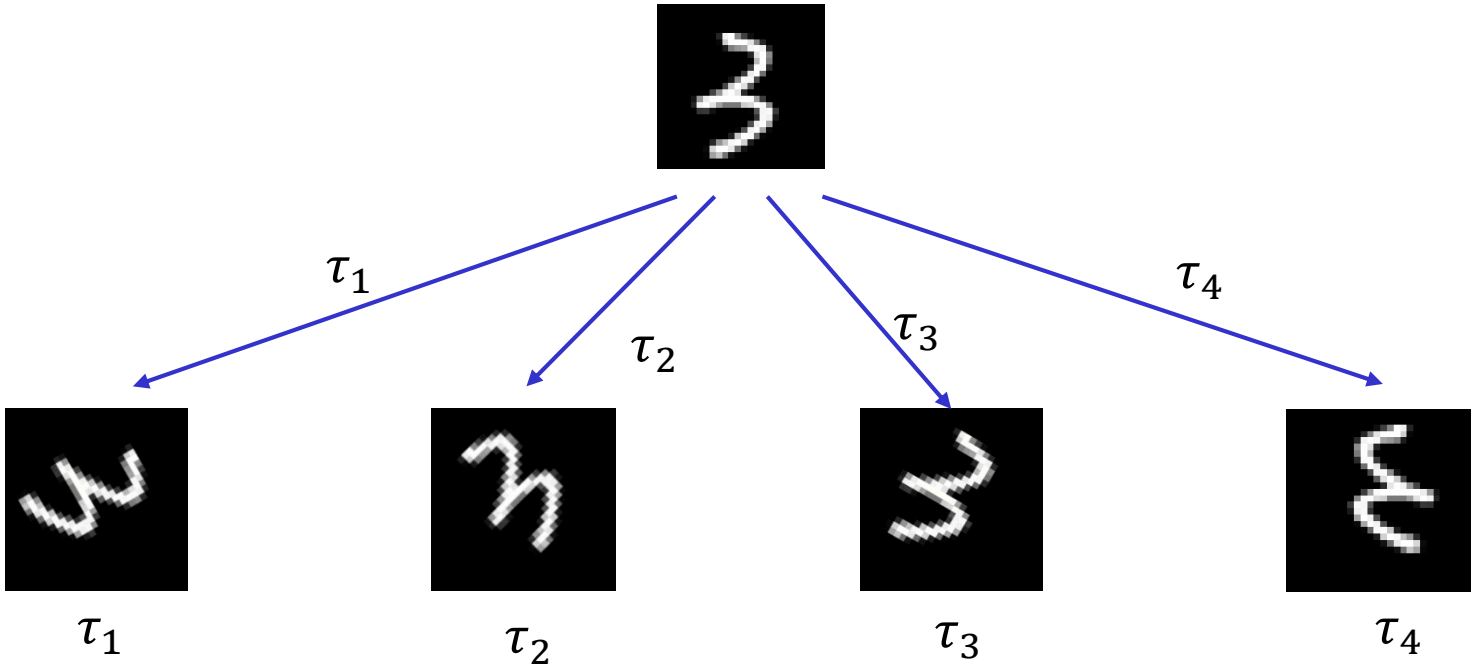
- Train a CNN on TR_{new} by categorical cross entropy over the T transformations
- The output of the last layer of the CNN is used as feature vector

We can avoid using a pre-trained model, and train a CNN directly for a supervised learning problem directly on TR

SELF-SUPERVISED LEARNING

Example:

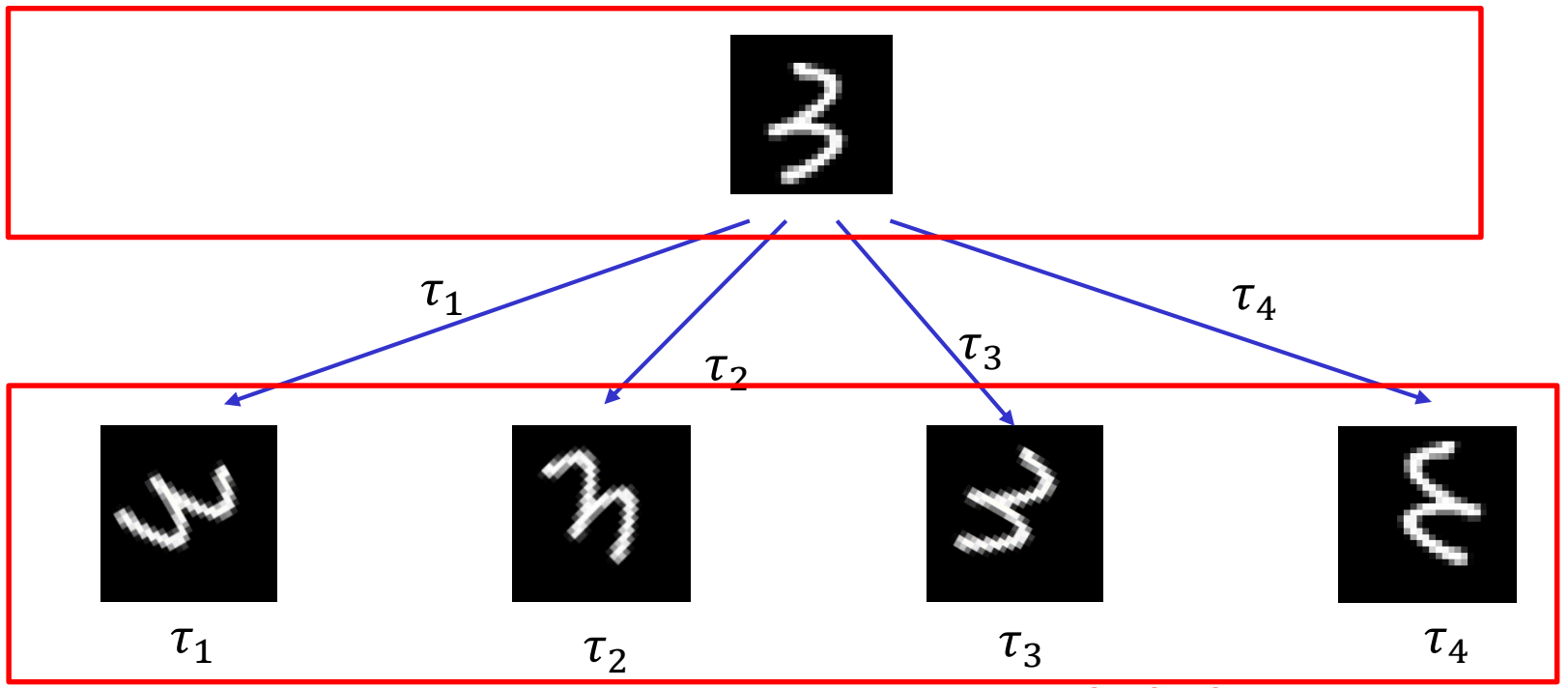
- TR contains only images representing digit 3
- $\mathcal{T}$ contains rotations and horizontal/vertical flips



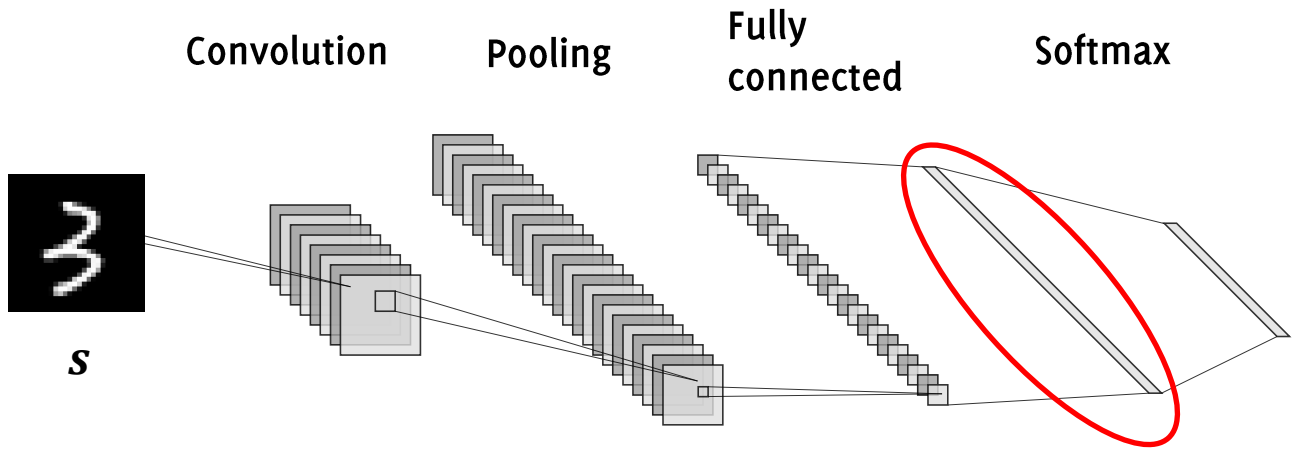
SELF-SUPERVISED LEARNING

Example:

- TR contains only images representing digit 3
- $\mathcal{T}$ contains rotations and horizontal/vertical flips



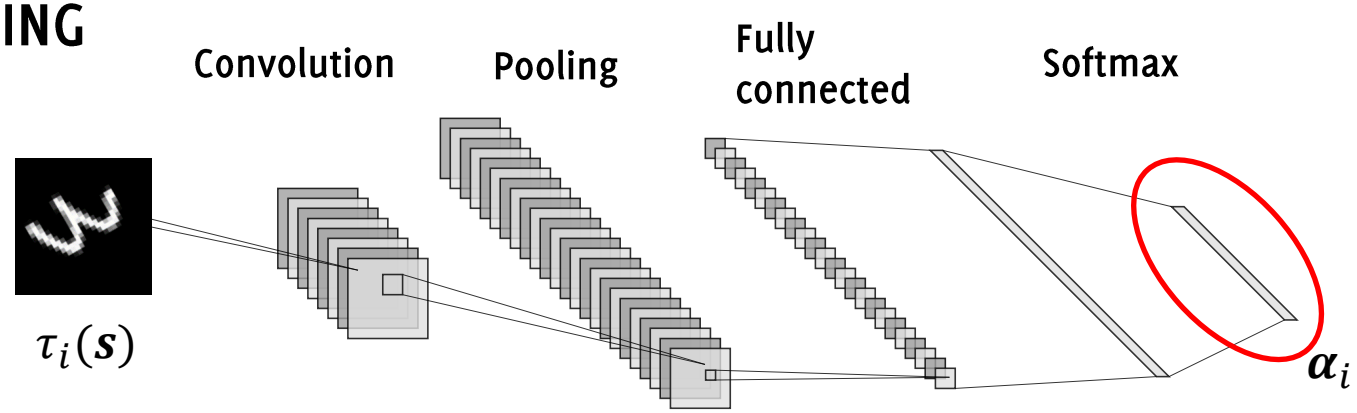
SELF-SUPERVISED LEARNING: TWO STEP APPROACH



We can use the second last layer of this classification network as a **feature vector** describing normal data and train an anomaly detector

Remark: we have to split our training set in two sets. The first set is used to train the classifier, the second one to train the anomaly detector

SELF-SUPERVISED LEARNING

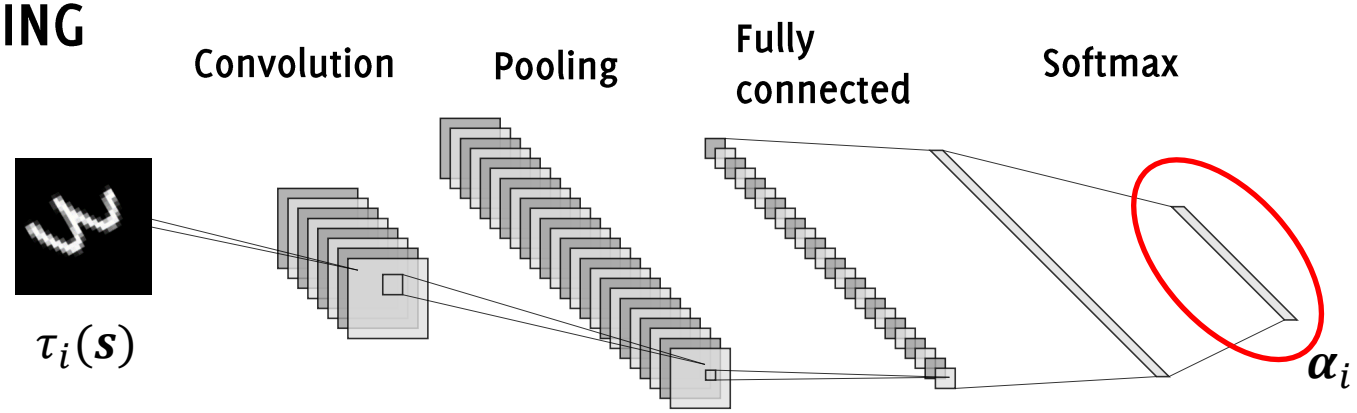


The novel approach is to feed the network, after training, with the all the trasformed versions $\tau_i(\boldsymbol{s})$ of the test sample $\boldsymbol{s}$ to obtain a set of vectors $\{\boldsymbol{\alpha}_i\}_{i=1}^{T}$

Here, $[\boldsymbol{\alpha}_i]_i$ is the posterior probability of label i given $\tau_i(\boldsymbol{s})$.

Golan, El-Yaniv, *"Deep Anomaly Detection Using Geometric Transformations"*, NeurIPS 2018

SELF-SUPERVISED LEARNING



The novel approach is to feed the network, after training, with the all the trasformed versions $\tau_i(\boldsymbol{s})$ of the test sample $\boldsymbol{s}$ to obtain a set of vectors $\{\boldsymbol{\alpha}_i\}_{i=1}^{T}$

Here, $[\boldsymbol{\alpha}_i]_i$ is the posterior probability of label i given $\tau_i(\boldsymbol{s})$.

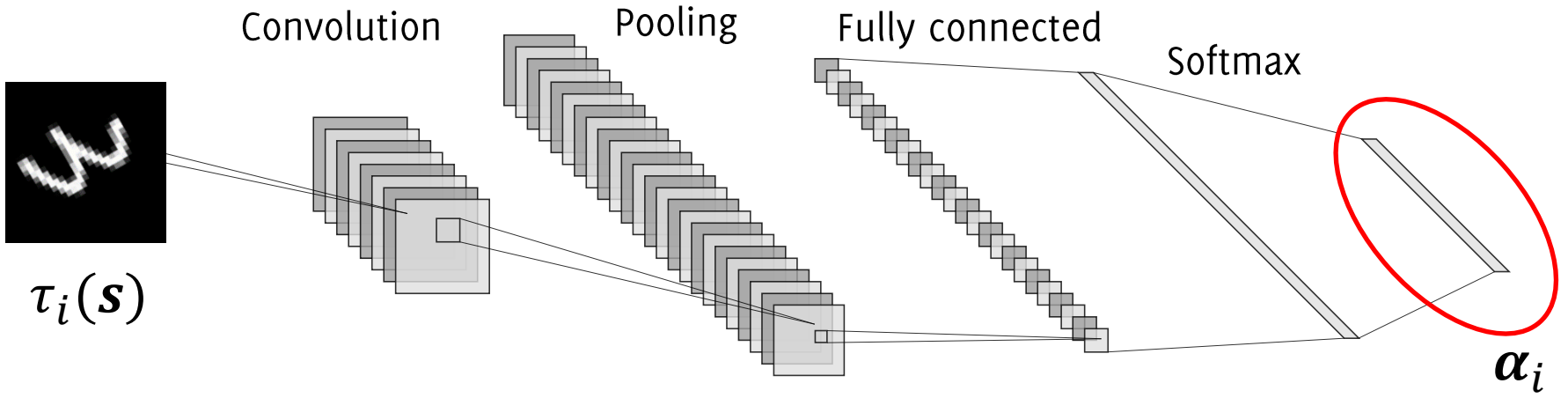
A simple anomaly score is:

$$score(\boldsymbol{s}) = 1 - \frac{1}{T}\sum_{i=1}^{T}[\boldsymbol{\alpha}_i]_i$$

Which is zero when all the samples are classified with posterior equal 1 to a class, thus when the classifier is very confident, thus the sample is normal

Golan, El-Yaniv, *"Deep Anomaly Detection Using Geometric Transformations"*, NeurIPS 2018

SELF-SUPERVISED LEARNING



Another anomaly score can be defined by modeling the distribution of the last layer

$$\boldsymbol{\alpha}_i = \psi(\boldsymbol{\tau}_i(\boldsymbol{s})) \in [0,1]^T$$

Estimate the conditional distributions $P(\boldsymbol{\alpha}_i|\tau_i)$ for each τ_i using the Dirichlet distribution (parametric model)

$$score(\boldsymbol{s}) = -\sum_{\boldsymbol{i}} \log P(\boldsymbol{\alpha}_i|\tau_i)$$

Golan, El-Yaniv, *"Deep Anomaly Detection Using Geometric Transformations"*, NeurIPS 2018

SELF-SUPERVISED LEARNING

The set of transformation has to be properly chosen:

- if during training the trained **classifier cannot discriminate** the transformed samples, it **does not extract meaningful feature** for anomaly detection
- **Non-geometric transformations** (Gaussian blur, gamma correction, sharpening) might eliminate important feature and **are less performing** than geometric ones

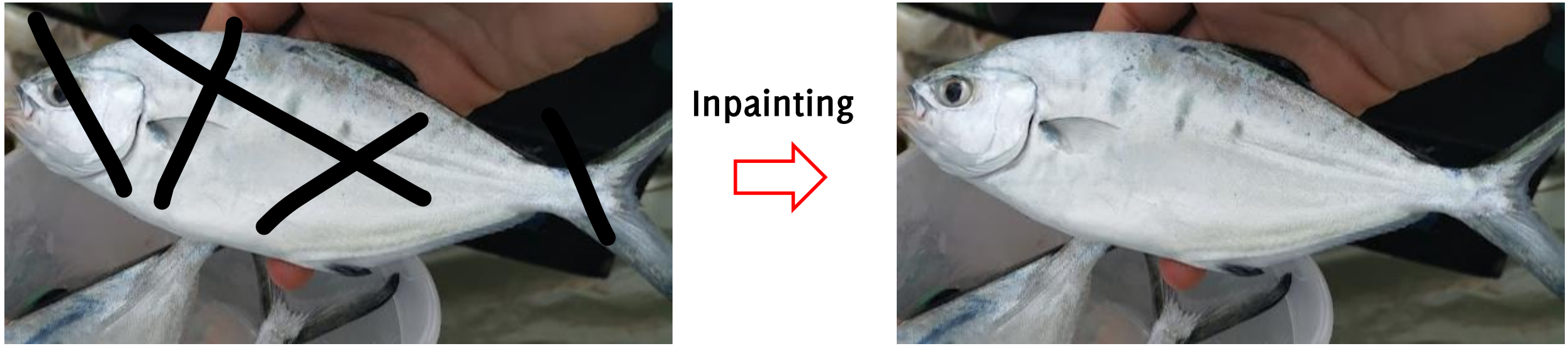
Golan, El-Yaniv, *"Deep Anomaly Detection Using Geometric Transformations"*, NeurIPS 2018

INPAINTING AS A FORM OF SELF SUPERVISION

Image inpainting is the problem of reconstructing missing or damaged areas of digital photographs and videos

A very challenging inverse problem where **Deep Autoencoders excel**

Training an autoencoder performing inpainting **does not require supervision**



Here, inpainting is used as a form of self-supervision

ANOMALY DETECTION BY IMAGE INPAINTING

Anomaly Detection is cast to a reconstruction by inpainting problem.

The rationale:

A deep AE trained to perform inpainting over normal images will reconstruct:

- Perfectly normal regions
- Poorly anomalous ones.

Rmk: Due to a high generalization capacity of (pre-trained) AE that are fine-tuned to reconstruct the entire normal input (without inpaining) , the anomalies are often reconstructed with a high fidelity.

MvTec AD dataset

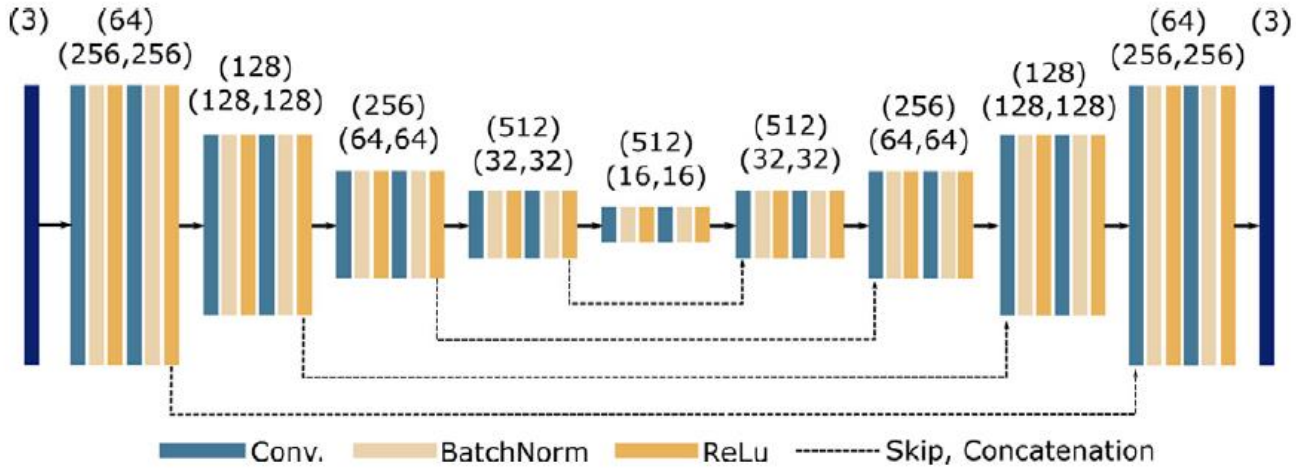
Zavrtanik, Vitjan, Matej Kristan, and Danijel Skočaj. "Reconstruction by inpainting for visual anomaly detection." *Pattern Recognition* 112 (2021): 107706.

ANOMALY DETECTION BY IMAGE INPAINTING

The AE network $\mathcal{D}(\mathcal{E}(\cdot))$ is trained to perform image inpainting over normal images

- Inpainted regions are set over a **tile of squares**
- In contrast with standard AE for anomaly detection, **it is possible to adopt skip connections**, since only inpainted regions are reconstructed, thus the trivial solution (the identity mapping) cannot be learned
- Training loss is based on **structural similarity indexes**, GSM and SSIM

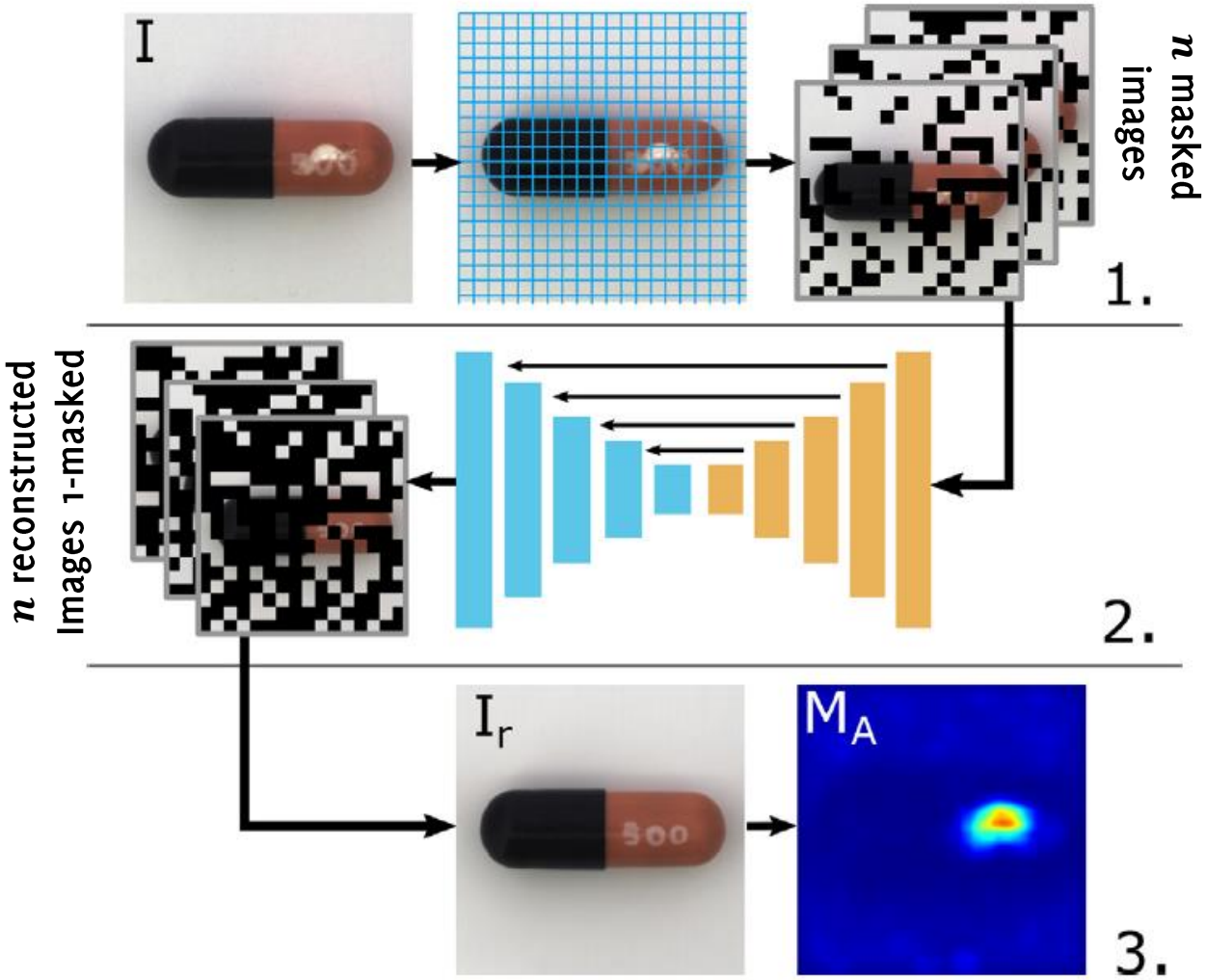
$$\mathcal{L}(\boldsymbol{s}, \hat{\boldsymbol{s}}) = \ell^2(\boldsymbol{s} - \hat{\boldsymbol{s}}) + \lambda_S SSIM(\boldsymbol{s}, \hat{\boldsymbol{s}}) + \lambda_G GSM(\boldsymbol{s}, \hat{\boldsymbol{s}})$$



Zavrtanik, Vitjan, Matej Kristan, and Danijel Skočaj. "Reconstruction by inpainting for visual anomaly detection." *Pattern Recognition* 112 (2021): 107706.

ANOMALY DETECTION BY IMAGE INPAINTING

- At test time, ***n*** **inpainted images are randomly generated over a grid**, reconstructed by the AE and then inverse masked.
- **Reconstruction quality** on inpainted regions is assessed as **an anomaly score** (measured by the loss functions used for training)
- The entire process is repeated multiple times **to consider different tile size to detect anomalies at different scales**



Zavrtanik, Vitjan, Matej Kristan, and Danijel Skočaj. "Reconstruction by inpainting for visual anomaly detection." *Pattern Recognition* 112 (2021): 107706.

SEMI-SUPERVISED APPROACHES

CNN as data-driven feature extractor

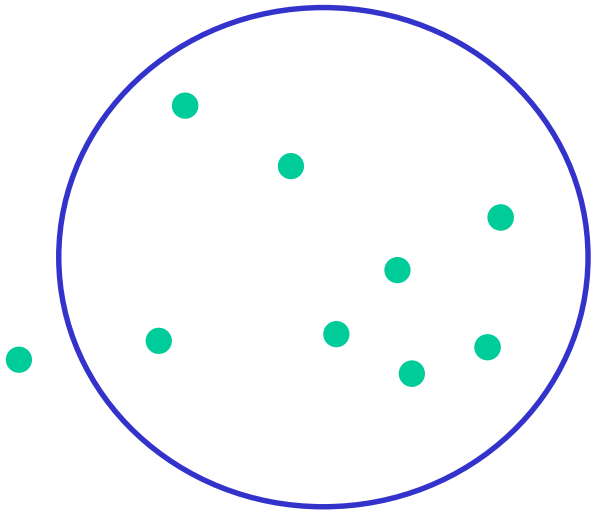
- Transfer learning
- Autoencoders
- Self-supervised learning
- Deep One-Class Classification

Generative models

- Variational Auto Encoders (VAEs)
- Generative Adversarial Networks (GANs)

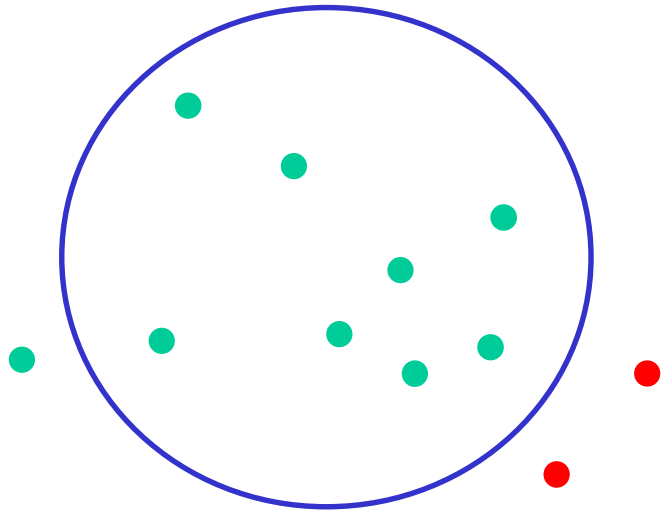
SUPPORT VECTOR DATA DESCRIPTION (SVDD) REVISITED

We want to find an **hypersphere** that, in the feature space, **encloses most of the normal** data



SUPPORT VECTOR DATA DESCRIPTION (SVDD) REVISITED

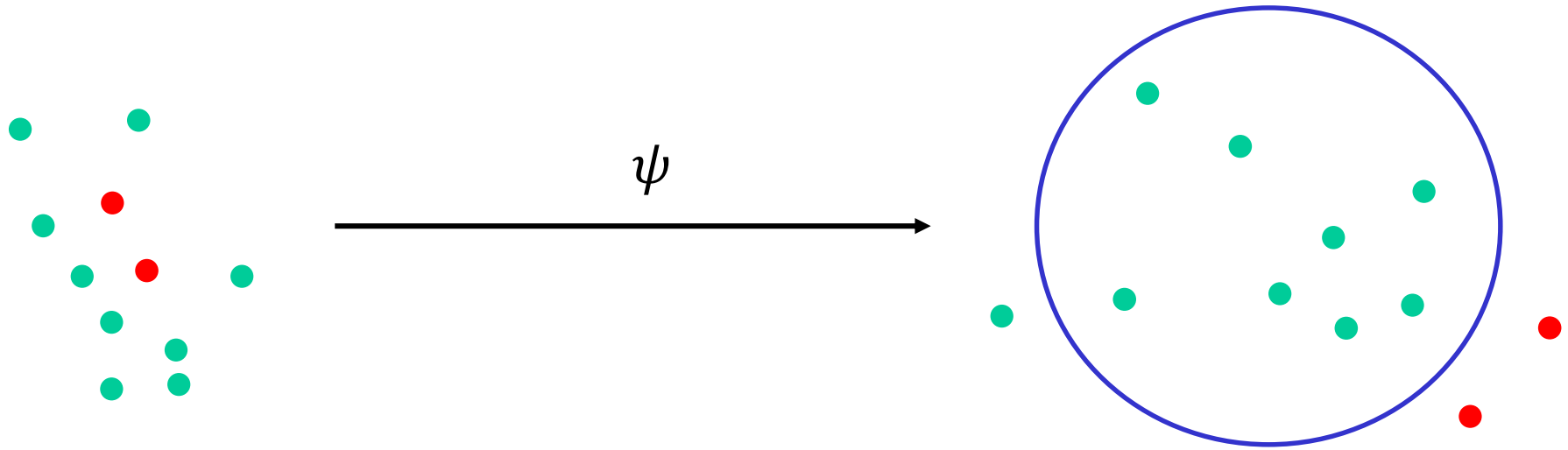
We want to find an **hypersphere** that, in the feature space, **encloses most of the normal** data



We expect that anomalous data lie outside the sphere

SUPPORT VECTOR DATA DESCRIPTION (SVDD) REVISITED

Typically the sphere is computed in a **high** (possibly infinite) **dimensional** feature **space**

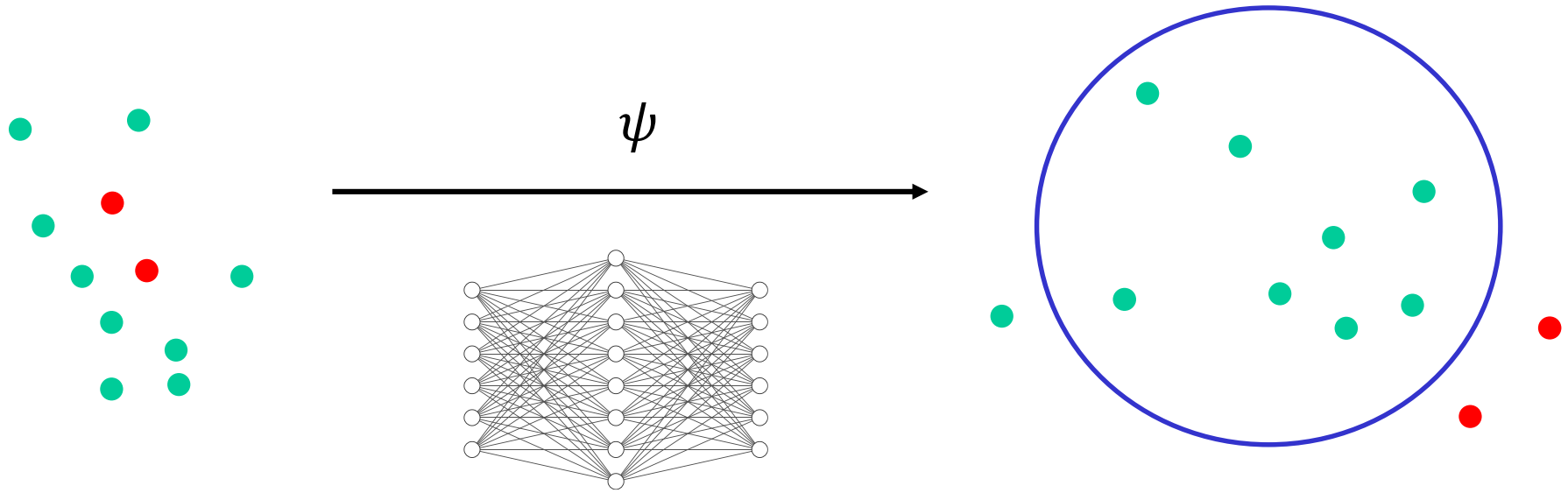


Feature are defined using **kernels**

- Polynomial kernel
- Gaussian kernel

SUPPORT VECTOR DATA DESCRIPTION (SVDD) REVISITED

Idea: can we learn the feature from normal data using a neural network?



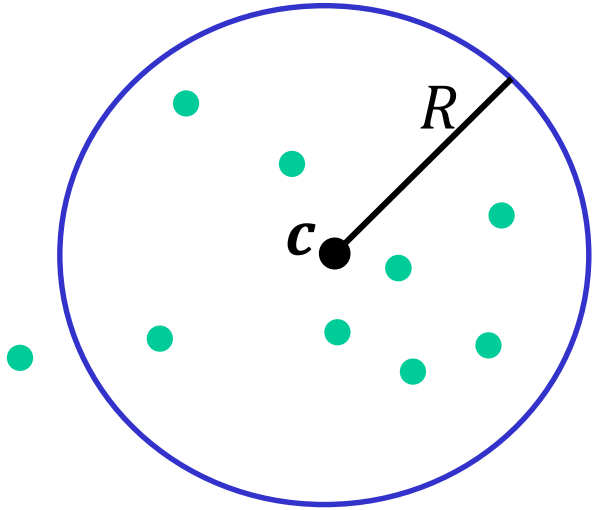
Ruff et al, *"Deep One-Class Classification", ICML 2018*

SOFT-BOUNDARY DEEP SVDD

Minimize the loss:

$$\min_{R,\boldsymbol{\theta}} R^2 + \frac{1}{\nu N}\sum_{n=1}^{N} \max\{0, \left|\left|\psi_{\boldsymbol{\theta}}(\boldsymbol{s}_n) - \boldsymbol{c}\right|\right|^2 - R^2\} + \lambda\left|\left|\boldsymbol{\theta}\right|\right|^2$$

- The samples $\boldsymbol{s}_n$ such that $\psi_{\boldsymbol{\theta}}(\boldsymbol{s}_n)$ is inside the sphere do not contribute to the loss
- ν provides a bound on the False Positive Rate
- $\lambda\left|\left|\boldsymbol{\theta}\right|\right|^2$ is a regularization term



A test sample $\boldsymbol{s}$ is anomalous if $\psi_{\boldsymbol{\theta}}(\boldsymbol{s}) - \boldsymbol{c} > R$

Ruff et al, *"Deep One-Class Classification", ICML 2018*

SOFT-BOUNDARY DEEP SVDD

Minimize the loss:

$$\min_{R,\boldsymbol{\theta}} R^2 + \frac{1}{\nu N}\sum_{n=1}^{N} \max\{0, \left|\left|\psi_{\boldsymbol{\theta}}(\boldsymbol{s}_n) - \boldsymbol{c}\right|\right|^2 - R^2\} + \lambda\left|\left|\boldsymbol{\theta}\right|\right|^2$$

Remarks:

- Some contraints must be imposed on the network $\psi_{\boldsymbol{\theta}}$ to avoid trivial solutions:
  - **No bias terms**
  - **Unbounded** activations
- $\boldsymbol{c}$ is not optimized but has to be precomputed from data
  - $\boldsymbol{c}$ must be different from $\boldsymbol{c}_0 = \psi_0(\boldsymbol{s})$

Ruff et al, *"Deep One-Class Classification", ICML 2018*

A SIMPLER FORMULATION: DEEP SVDD

$$\min_{\boldsymbol{\theta}} + \frac{1}{N}\sum_{n=1}^{N}\left|\left|\psi_{\boldsymbol{\theta}}(\boldsymbol{s}_n) - \boldsymbol{c}\right|\right|^2 + \lambda\left|\left|\boldsymbol{\theta}\right|\right|^2$$

Cons:

- No bound on the FPR provided by ν
- A threshold has to be chosen for the anomaly score:

$$\left|\left|\psi_{\boldsymbol{\theta}}(\boldsymbol{s}) - \boldsymbol{c}\right|\right|^2$$

Ruff et al, *"Deep One-Class Classification", ICML 2018*

SEMI-SUPERVISED APPROACHES

CNN as data-driven feature extractor

- Transfer learning
- Autoencoders
- Self-supervised learning
- Deep One-Class Classification

Generative models

- Variational Auto Encoders (VAEs)
- Generative Adversarial Networks (GANs)

Wait for Luigi Malagò ' Lecture

SEMI-SUPERVISED APPROACHES

CNN as data-driven feature extractor

- Transfer learning
- Autoencoders
- Self-supervised learning
- Deep One-Class Classification

Generative models

- Variational Auto Encoders (VAEs)
- Generative Adversarial Networks (GANs)

GENERATIVE MODELS

Given a training set of images S, these models can generate other images that are similar to those in S

WHAT FOR GENERATIVE MODELS?

- Generative models can be used for data augmentation, simulation and planning
- Realistic samples for artwork, super-resolution, colorization, etc.
- Getting close to the “holy grail” of modeling the distribution of natural images

Radford, et al “Unsupervised representation learning with deep convolutional generative adversarial networks. ICLR 2016

GENERATIVE ADVERSARIAL NETWORKS (GAN)

The GAN approach:

Do not look for an **explicit density model** ϕ_S describing the manifold of natural images.

Just find out a **model** able to **generate samples** that looks like training samples $S \subset \mathbb{R}^n$

Instead of sampling from ϕ_S, just use:

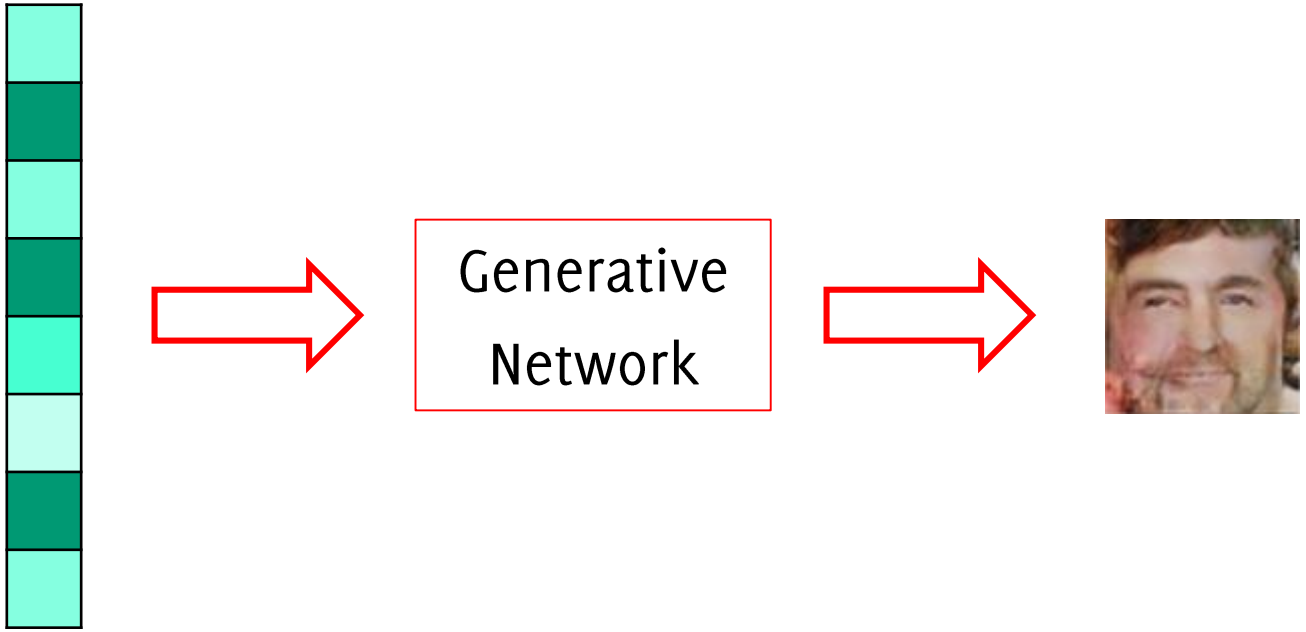
- Sample a seed from a known distribution ϕ_z
- Feed this seed to a learned transformation that generates realistic samples, as if they were drawn from ϕ_S

Use a **neural network to learn this transformation**

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, ... & Bengio, Y. Generative adversarial nets NIPS 2014

GENERATIVE ADVERSARIAL NETWORKS (GAN)

The GAN approach:



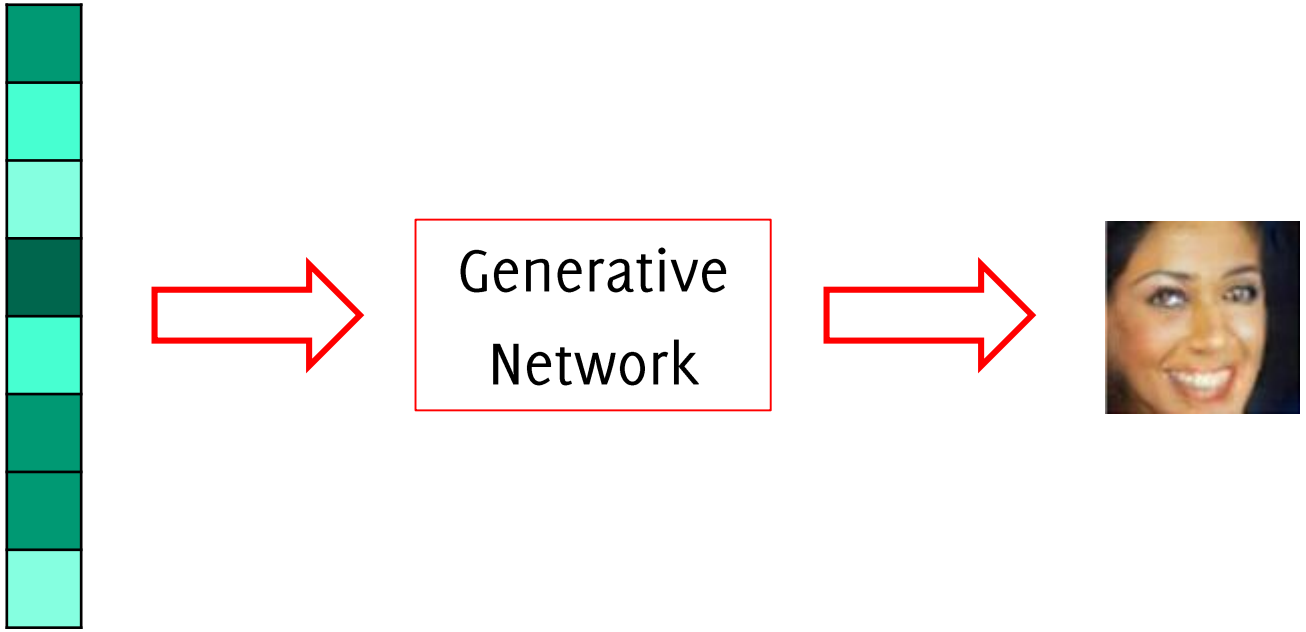
$z \sim \phi_z$

Draw a sample from
the noise distribution

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, ... & Bengio, Y. Generative adversarial nets NIPS 2014

GENERATIVE ADVERSARIAL NETWORKS (GAN)

The GAN approach:



$z \sim \phi_z$

Draw a sample from
the noise distribution

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, ... & Bengio, Y. Generative adversarial nets NIPS 2014

GENERATIVE ADVERSARIAL NETWORKS (GAN)

The GAN solution: Train a pair of neural networks with different tasks that compete in a sort of **two player game**.

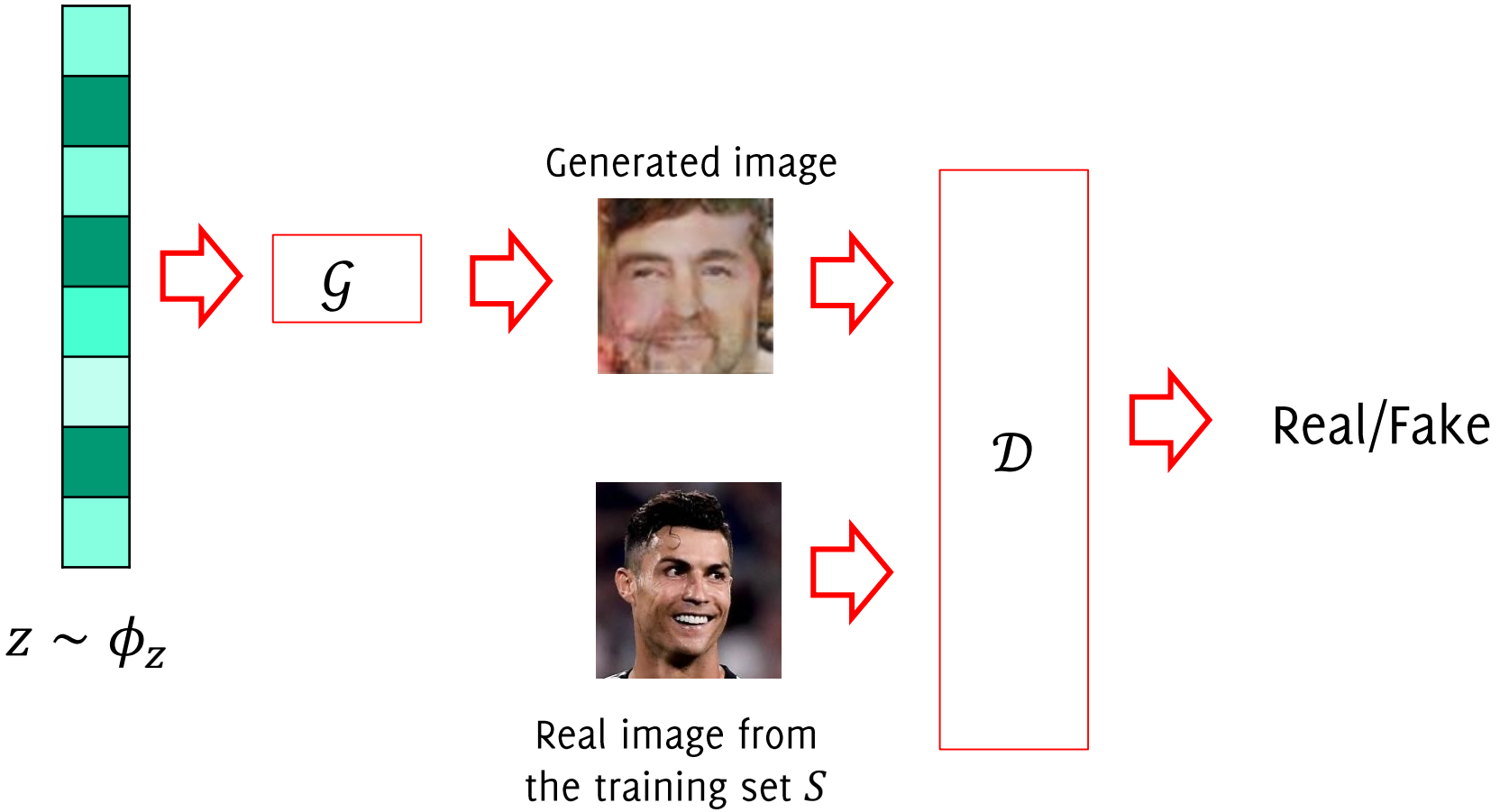
These models are:

- Generator $\mathcal{G}$ that produces realistic samples e.g. taking as input some random noise. $\mathcal{G}$ tries to fool the discriminator
- Discriminator $\mathcal{D}$ that takes as input an image and assess whether it is real or generated by $\mathcal{G}$

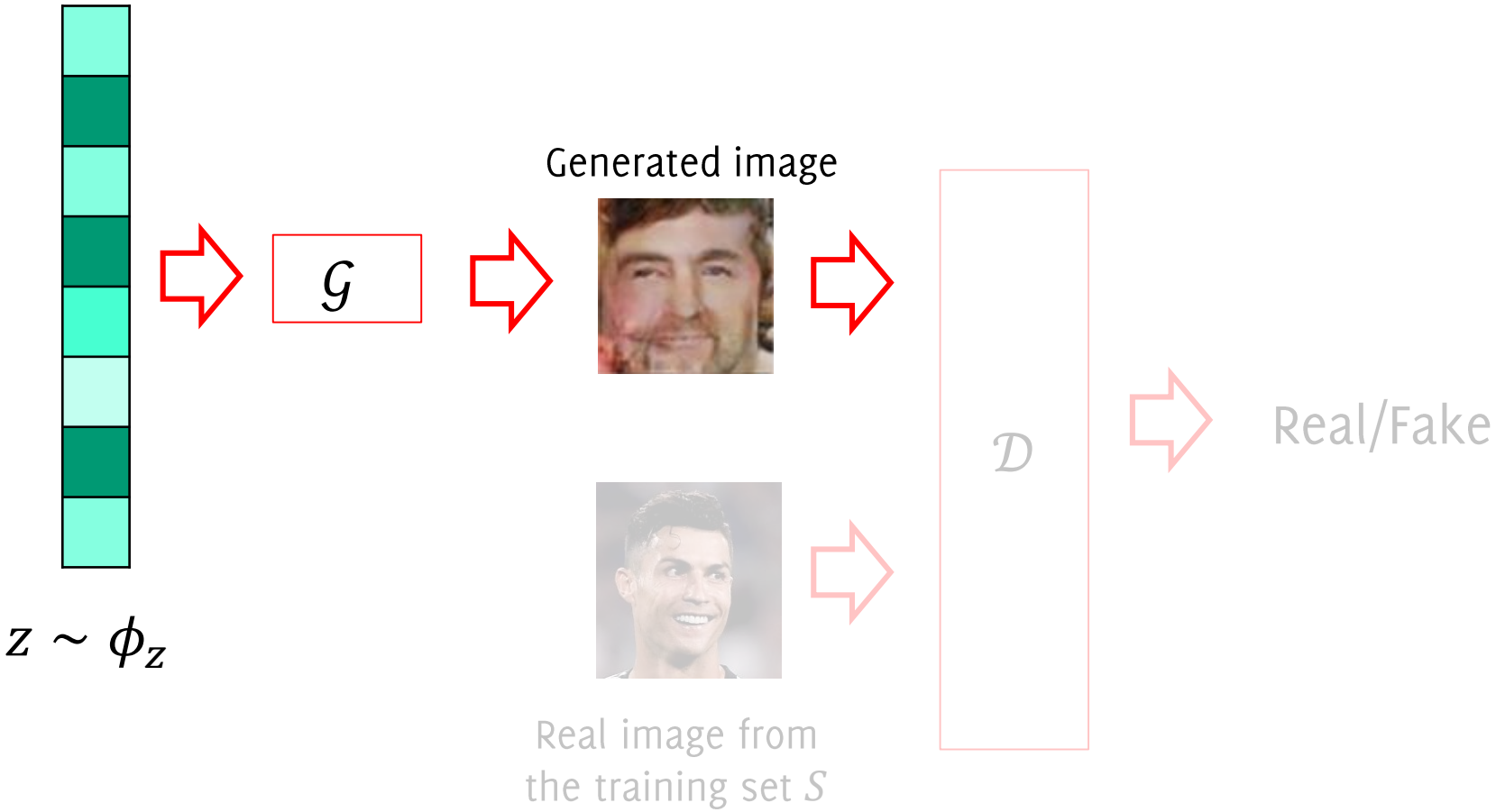
Train the two and at the end, keep only $\mathcal{G}$

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, ... & Bengio, Y. Generative adversarial nets NIPS 2014

GAN ARCHITECTURE

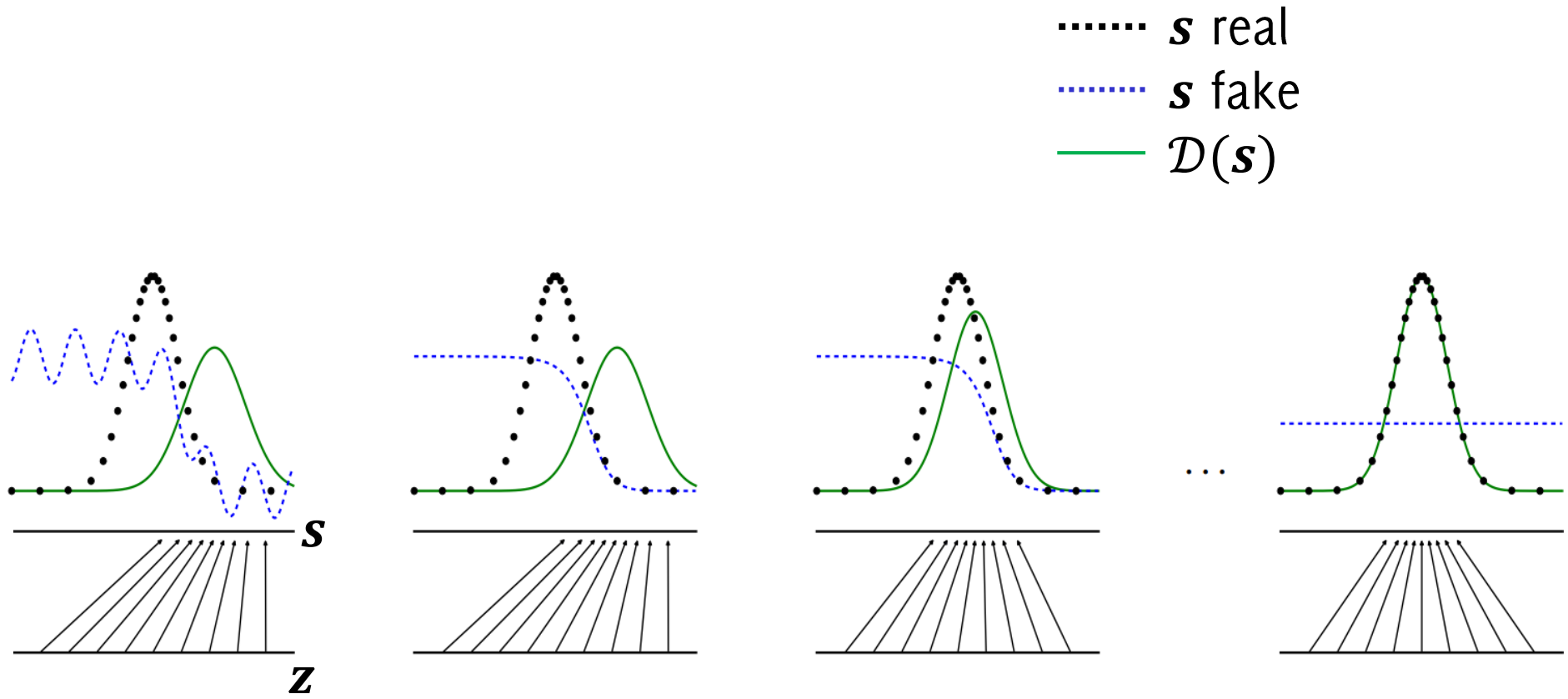


GENERATING IMAGES USING THE GAN



Discriminator $\mathcal{D}$ is completely useless and as such dropped. After a successful GAN training, $\mathcal{D}$ is not able to distinguish the real/fake

ILLUSTRATION OF WHAT HAPPENS AT THE END OF TRAINING



[Goodfellow 2014]

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, ... & Bengio, Y. Generative adversarial nets NIPS 2014

MNIST

Generated samples

closest training sample to the second-last column

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, ... & Bengio, Y. Generative adversarial nets NIPS 2014

INTERPOLATION IN THE LATENT SPACE

We can interpolate between two points in the latent space and obtain smooth transitions from a digit to another one

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, ... & Bengio, Y. Generative adversarial nets NIPS 2014

AT THE END OF THE DAY…

The discriminator $\mathcal{D}$ is discarded

The generator $\mathcal{G}$ and ϕ_Z are preserved as generative model

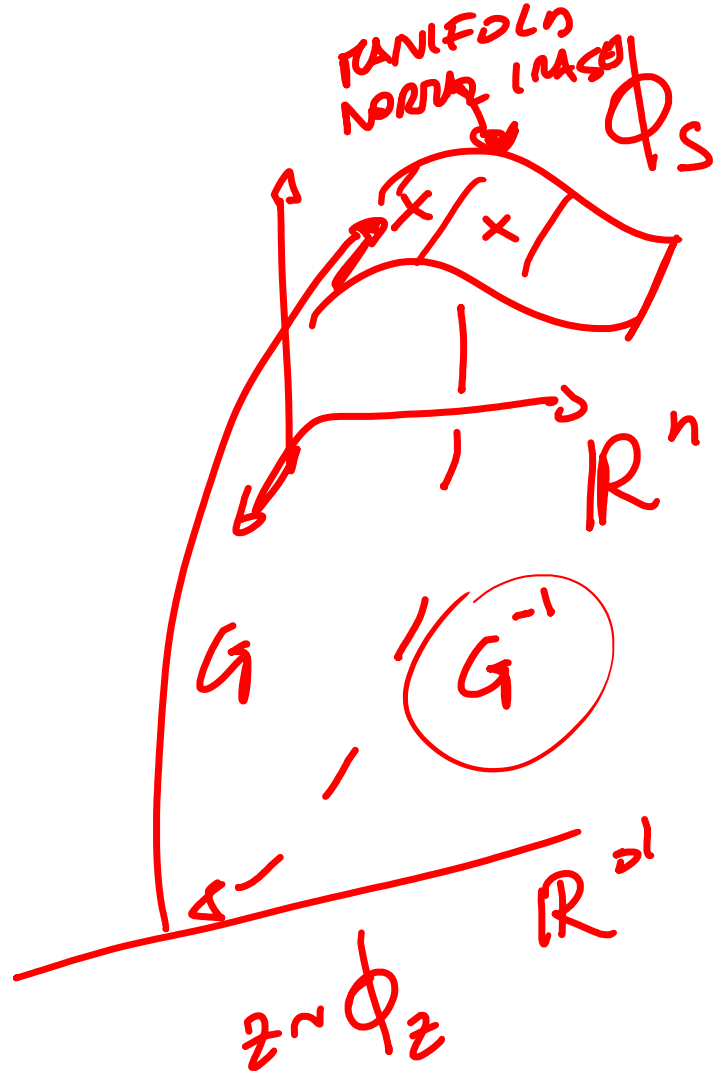
Remarks:

- The training is rather unstable, need to carefully synchronize the two steps (many later works in this direction, e.g. Wasserstein GAN)
- Training by standard tools: backpropagation and dropout
- Theoretical results provided
- Generator does not use S directly during training
- Generator performance is difficult to assess quantitatively
- **There is no explicit expression for the generator**, it is provided in an implicit form -> you cannot compute the likelihood of a sample w.r.t. the learned GAN

GAN FOR ANOMALY DETECTION

THANKS TO STEFANO PECCHIA, FORMER AN2DL STUDENT!


i
x
G
G⁻¹
G⁻¹
φ_z


MANIFOLD
NORMAL IMAGE
φ_S
x
x
R^n
G
G⁻¹
R^{n-1}
z ~ φ_z

GANS AND ANOMALY DETECTION

A Generator $\mathcal{G}$ trained exclusively on normal images in TR, already **provides a mapping**

- From the space of random vectors $z \sim \phi_z$
- To the manifold where images live $s \sim \phi_s$

Thus, **if we could invert the GAN**, we would have already an AD model

An anomaly score for a test image s would be

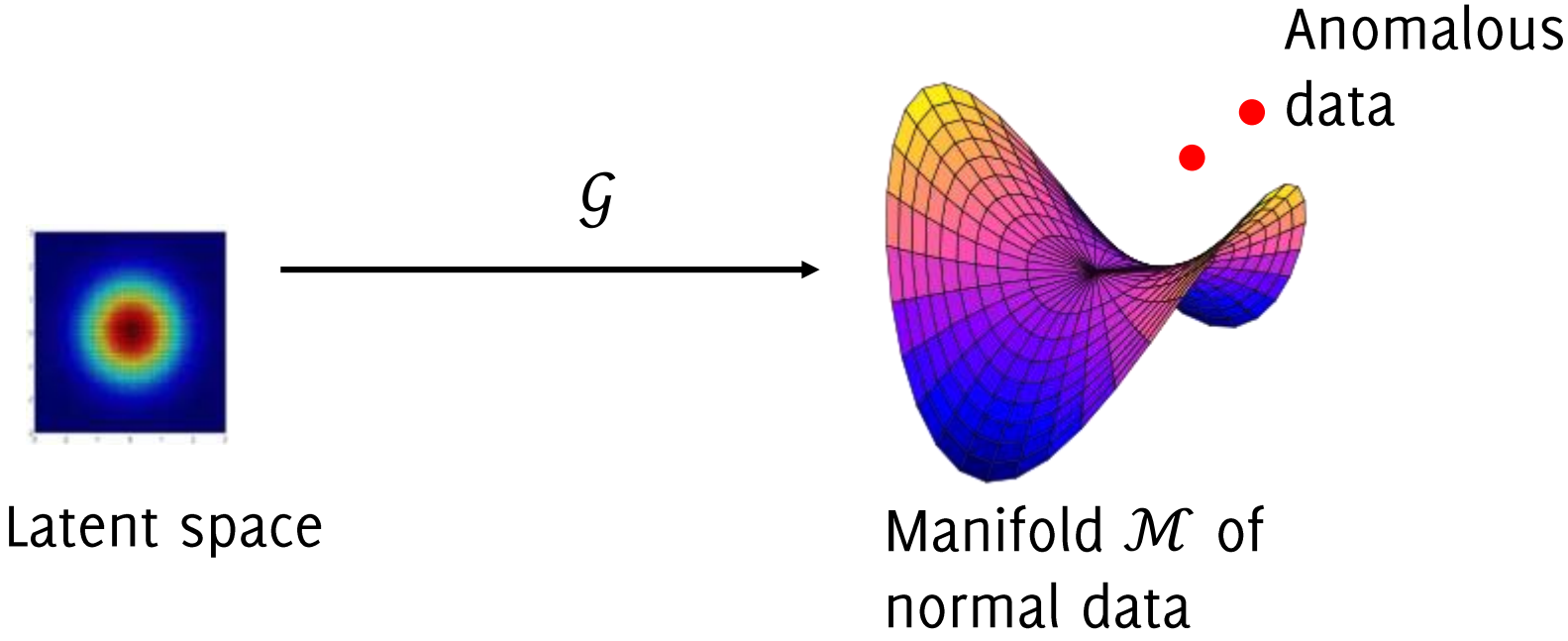
$$s \rightarrow \mathcal{G}^{-1}(s) \rightarrow \phi_z\left(\mathcal{G}^{-1}(s)\right)$$

Unfortunately, it is not possible to invert $\mathcal{G}$…

GAN FOR ANOMALY DETECTION

Idea: let us train a GAN on normal data. We expect that the generator $\mathcal{G}$ cannot generate any anomalous sample $\boldsymbol{s}$.

Problem: Given a test sample $\boldsymbol{s}$ how can we determine if it could be generated by $\mathcal{G}$?

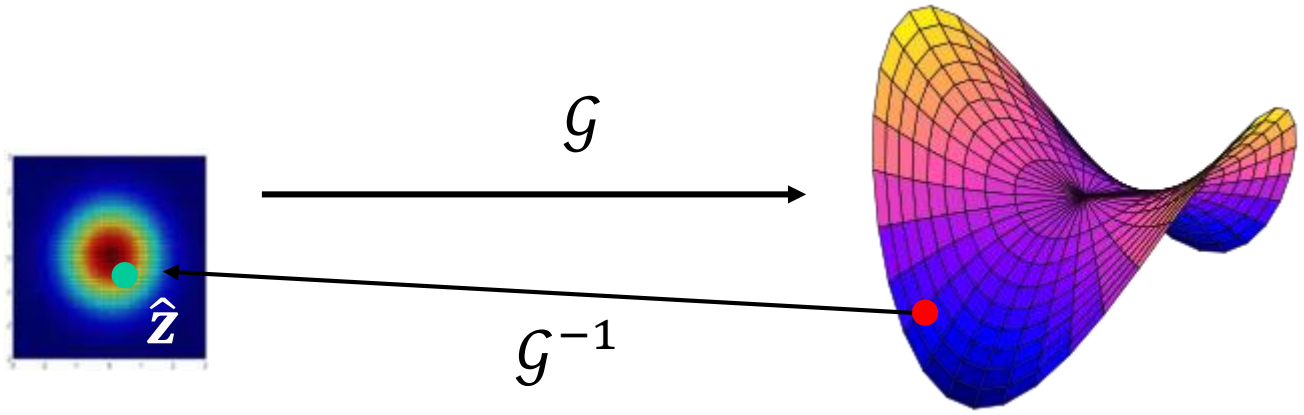


ANOGAN

Project the test sample $\boldsymbol{s}$ living on the manifold $\mathcal{M}$ to the random vector world by **solving the optimization problem:**

$$\hat{\boldsymbol{z}} = \min_{\boldsymbol{z}} \left|\left|\mathcal{G}(\boldsymbol{z}) - \boldsymbol{s}\right|\right| + \lambda \log(1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big))$$

- $\left|\left|\mathcal{G}(\boldsymbol{z}) - \boldsymbol{s}\right|\right|$ ensures that $\boldsymbol{s}$ is well approximated by the generator
- $\log(1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big))$ ensures that the projection $\mathcal{G}(\hat{\boldsymbol{z}})$ is similar to a real (normal) sample (as $\mathcal{D}$ would consider it normal)



Schlegl et al, "*Unsupervised anomaly detection with generative adversarial networks to guide marker discovery*", IPMI 2017

ANOGAN

Project the test sample $\boldsymbol{s}$ living on the manifold $\mathcal{M}$ to the random vector world by **solving the optimization problem:**

$$\hat{\boldsymbol{z}} = \min_{\boldsymbol{z}} \left|\left|\mathcal{G}(\boldsymbol{z}) - \boldsymbol{s}\right|\right| + \lambda \log(1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big))$$

- $\left|\left|\mathcal{G}(\boldsymbol{z}) - \boldsymbol{s}\right|\right|$ ensures that $\boldsymbol{s}$ is well approximated by the generator
- $\log(1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big))$ ensures that the projection $\mathcal{G}(\hat{\boldsymbol{z}})$ is similar to a real (normal) sample (as $\mathcal{D}$ would consider it normal)

Anomaly score:

$$score(\boldsymbol{s}) = \left|\left|\mathcal{G}(\hat{\boldsymbol{z}}) - \boldsymbol{s}\right|\right| + \lambda \log(1 - \mathcal{D}\big(\mathcal{G}(\hat{\boldsymbol{z}})\big))$$

Schlegl et al, "*Unsupervised anomaly detection with generative adversarial networks to guide marker discovery*", IPMI 2017

ANOGAN

Project the test sample $\boldsymbol{s}$ living on the manifold $\mathcal{M}$ to the random vector world by **solving the optimization problem:**

$$\hat{\boldsymbol{z}} = \min_{\boldsymbol{z}} \left|\left|\mathcal{G}(\boldsymbol{z}) - \boldsymbol{s}\right|\right| + \lambda \log(1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big))$$

- $\left|\left|\mathcal{G}(\boldsymbol{z}) - \boldsymbol{s}\right|\right|$ ensures that $\boldsymbol{s}$ is well approximated by the generator
- $\log(1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big))$ ensures that the projection $\mathcal{G}(\hat{\boldsymbol{z}})$ is similar to a real (normal) sample (as $\mathcal{D}$ would consider it normal)

Anomaly score:

$$score(\boldsymbol{s}) = \left|\left|\mathcal{G}(\hat{\boldsymbol{z}}) - \boldsymbol{s}\right|\right| + \lambda \log(1 - \mathcal{D}\big(\mathcal{G}(\hat{\boldsymbol{z}})\big))$$

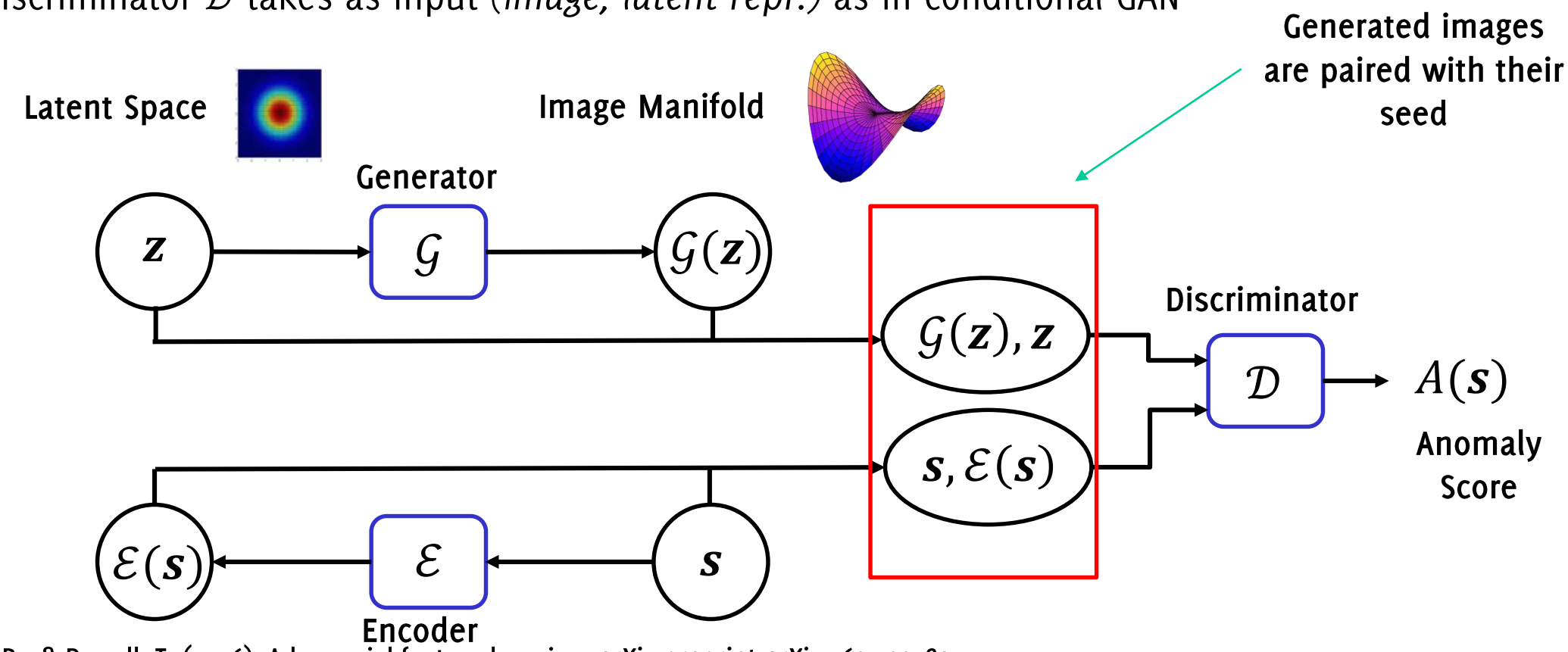
We need to solve an optimization problem for each test sample!
Can we train a Deep Neural Network to Solve this problem?

Schlegl et al, "*Unsupervised anomaly detection with generative adversarial networks to guide marker discovery*", IPMI 2017

BIDIRECTIONALGANS (BIGANS)

The BiGAN **adds an encoder $\mathcal{E}$ to the adversarial game which**

- Brings an image back to the space of "noise vectors" $\mathcal{E}(\boldsymbol{s})$
- Can be used to reconstruct an input image $\boldsymbol{s}$ (as in autoencoders) $\mathcal{G}(\mathcal{E}(s))$
- The discriminator $\mathcal{D}$ takes as input *(image, latent repr.)* as in conditional GAN



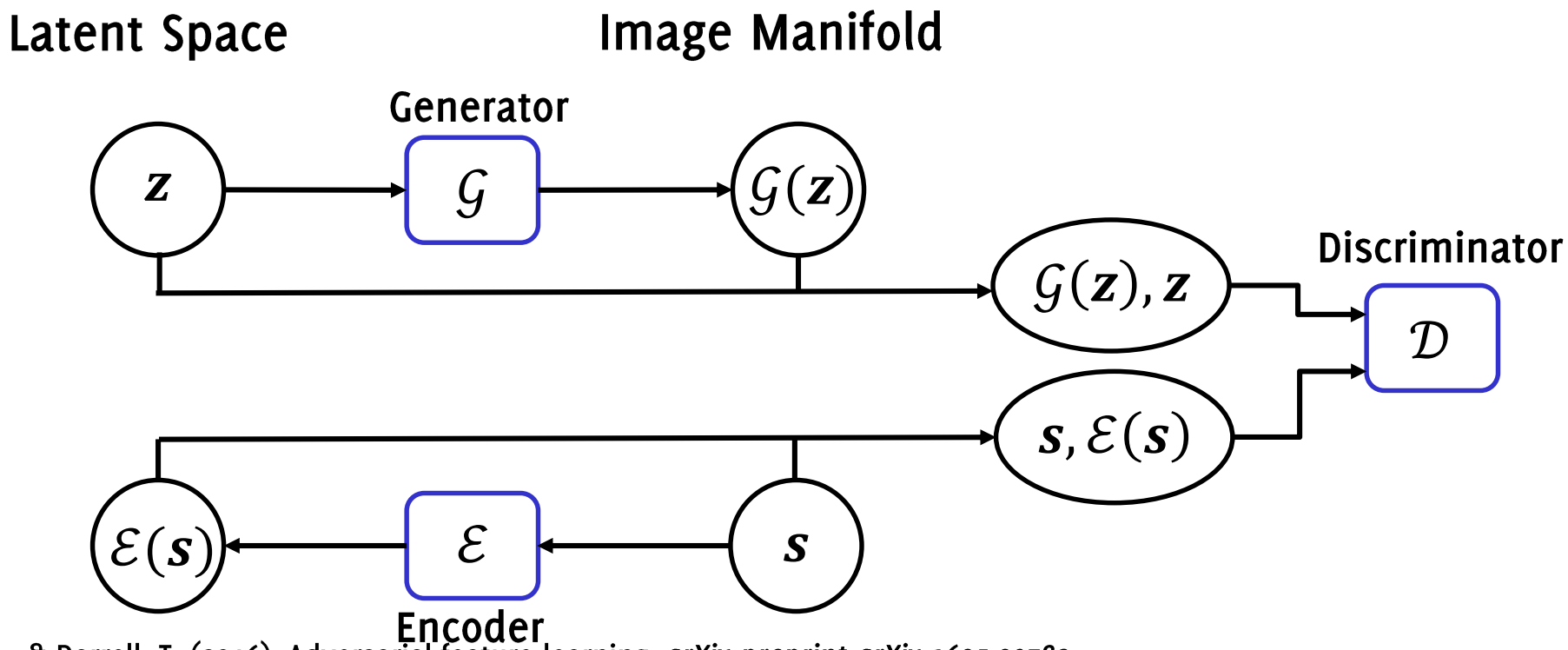
Donahue, J., Krähenbühl, P., & Darrell, T. (2016). Adversarial feature learning. *arXiv preprint arXiv:1605.09782.*

BIGAN TRAINING

The BIGANs training is also an **adversarial game**

$$\min_{G,E} \max_{\mathcal{D}} V(\mathcal{D}, E, G)$$

$$V(\mathcal{D}, E, G) = \mathrm{E}_{s\sim\phi_S}[\log \mathcal{D}(s, E(s))] + \mathrm{E}_{z\sim\phi_z}[1 - \log \mathcal{D}(G(z), z))]$$



Donahue, J., Krähenbühl, P., & Darrell, T. (2016). Adversarial feature learning. *arXiv preprint arXiv:1605.09782.*

STRAIGHTFORWARD ANOMALY DETECTION USING $\mathcal{E}$

In principle, the encoder $\mathcal{E}(\cdot)$ can be used for anomaly detection by computing the likelihood of $\phi_z\big(\mathcal{E}(\boldsymbol{s})\big)$ and consider as anomalous all the images $\boldsymbol{s}$ corresponding to a low **likelihood** (provided that ϕ_z was not a uniform distribution)

$$\phi_z\big(\mathcal{E}(\boldsymbol{s})\big)$$

Another option is to use the posterior of the discriminator as anomaly score

$$\mathcal{D}\big(\boldsymbol{s}, \mathcal{E}(\boldsymbol{s})\big) \text{ or } \mathcal{D}(\boldsymbol{s}) \text{ in standard GANs}$$

since the discriminator will consider the anomalous sample as fake.

Donahue, J., Krähenbühl, P., & Darrell, T. (2016). Adversarial feature learning. *arXiv preprint arXiv:1605.09782*.

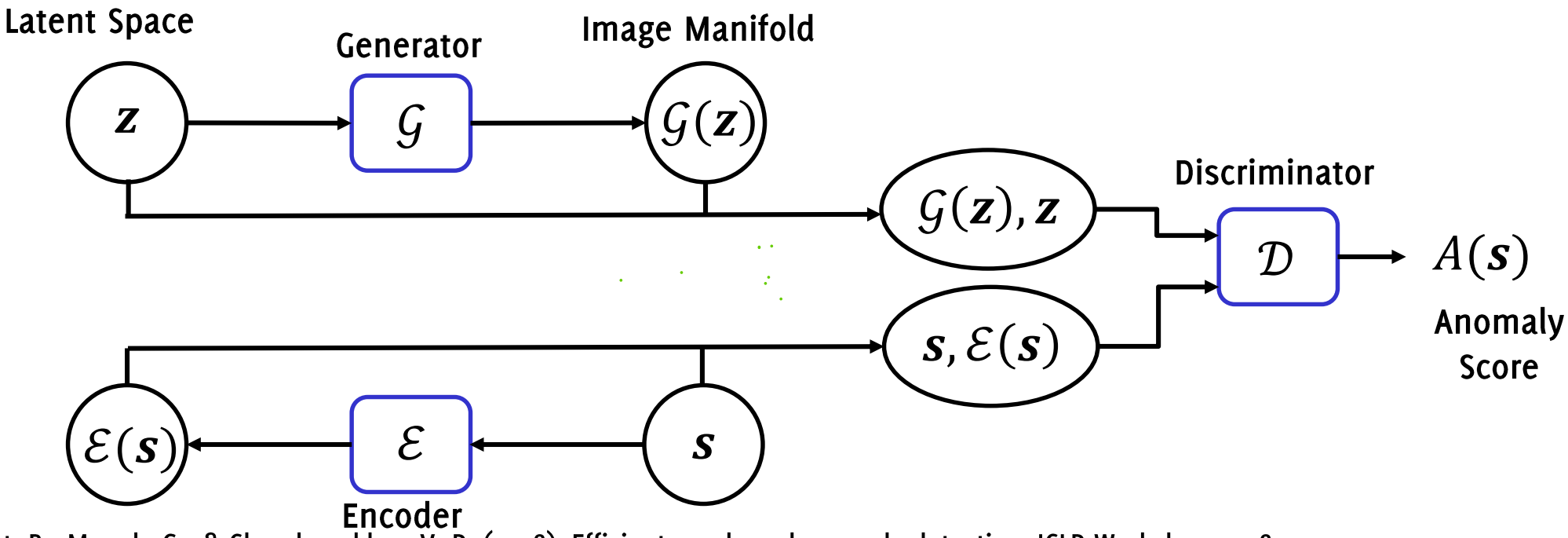
ANOMALY DETECTION WITH BIDIRECTIONALGANS (BIGANS)

However, there are more effective anomaly scores

$$A(s) = \alpha \left|\left| f\left(\mathcal{D}(s, \mathcal{E}(s))\right) - f\left(\mathcal{D}\left(\mathcal{G}(\mathcal{E}(s)), \mathcal{E}(s)\right)\right) \right|\right|_2 + (1-\alpha) \left|\left| \mathcal{G}(\mathcal{E}(s)) - s \right|\right|_2$$

Distance among latent representations of $\mathcal{D}$.
f is the operation of extracting a latent representation

Reconstruction Loss



Zenati, H., Foo, C. S., Lecouat, B., Manek, G., & Chandrasekhar, V. R. (2018). Efficient gan-based anomaly detection. ICLR Workshop 2018

ANOGAN IMPROVED

We can use BiGAN to efficiently invert the generator in AnoGAN:

$$\hat{\boldsymbol{z}} = \min_{\boldsymbol{z}} \left|\left|\mathcal{G}(\boldsymbol{z}) - \boldsymbol{s}\right|\right| + \lambda \log(1 - \mathcal{D}(\mathcal{G}(\boldsymbol{z})))$$

$$\hat{\boldsymbol{z}} = \mathcal{E}(\boldsymbol{s})$$

Zenati et al, "*Efficient GAN-based anomaly detection*", Workshop ICLR 2018

ANOGAN IMPROVED

We can use BiGAN to efficiently invert the generator in AnoGAN:

$$\hat{\boldsymbol{z}} = \min_{\boldsymbol{z}} \left|\left|\mathcal{G}(\boldsymbol{z}) - \boldsymbol{s}\right|\right| + \lambda \log(1 - \mathcal{D}(\mathcal{G}(\boldsymbol{z})))$$

$$\hat{\boldsymbol{z}} = \mathcal{E}(\boldsymbol{s})$$

$$score(\boldsymbol{s}) = \left|\left|\mathcal{G}(\hat{\boldsymbol{z}}) - \boldsymbol{s}\right|\right| + \lambda \log\left(1 - \mathcal{D}\left(\mathcal{G}(\hat{\boldsymbol{z}})\right)\right) =$$

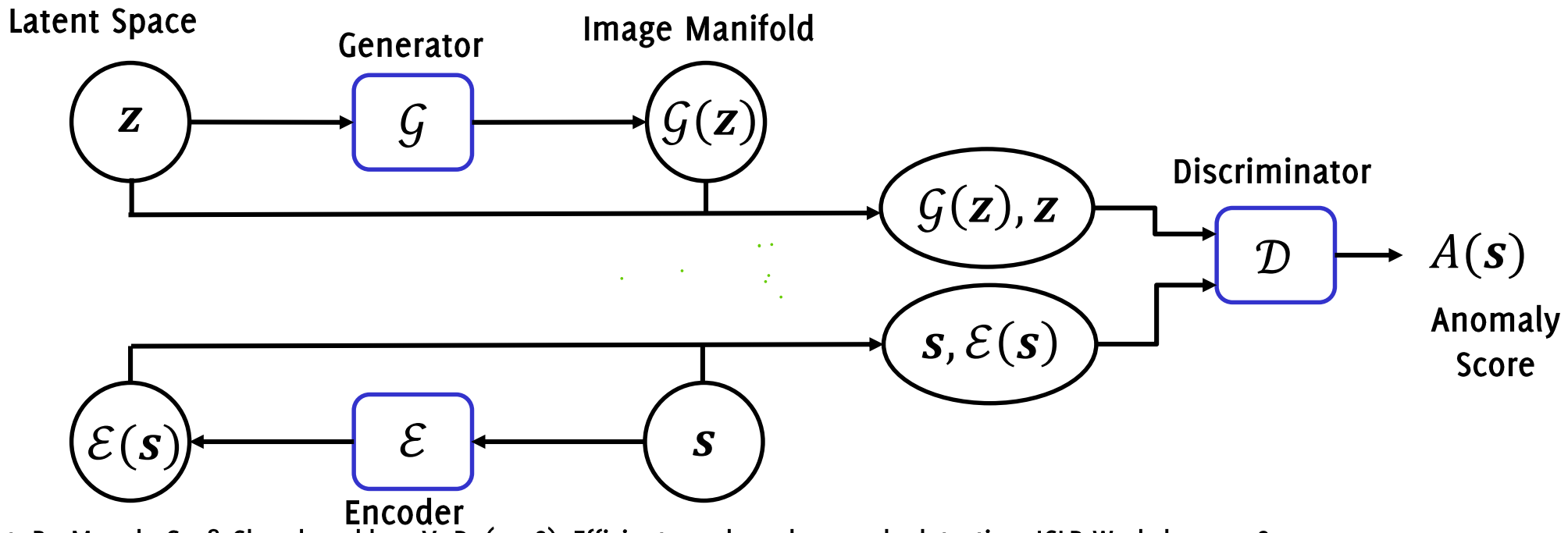
$$= \left|\left|\mathcal{G}(\mathcal{E}(\boldsymbol{s})) - \boldsymbol{s}\right|\right| + \lambda \log\left(1 - \mathcal{D}\left(\mathcal{G}(\mathcal{E}(\boldsymbol{s}))\right)\right)$$

Zenati et al, "*Efficient GAN-based anomaly detection*", Workshop ICLR 2018

ANOMALY DETECTION WITH BIDIRECTIONALGANS (BIGANS)

Limitations

- Image-wise / patch-wise training and testing. It is not simple to make $\mathcal{D}$ fully convolutional, as it takes two inputs.
- Little stability during training
- No way to promote better quality of reconstructed images

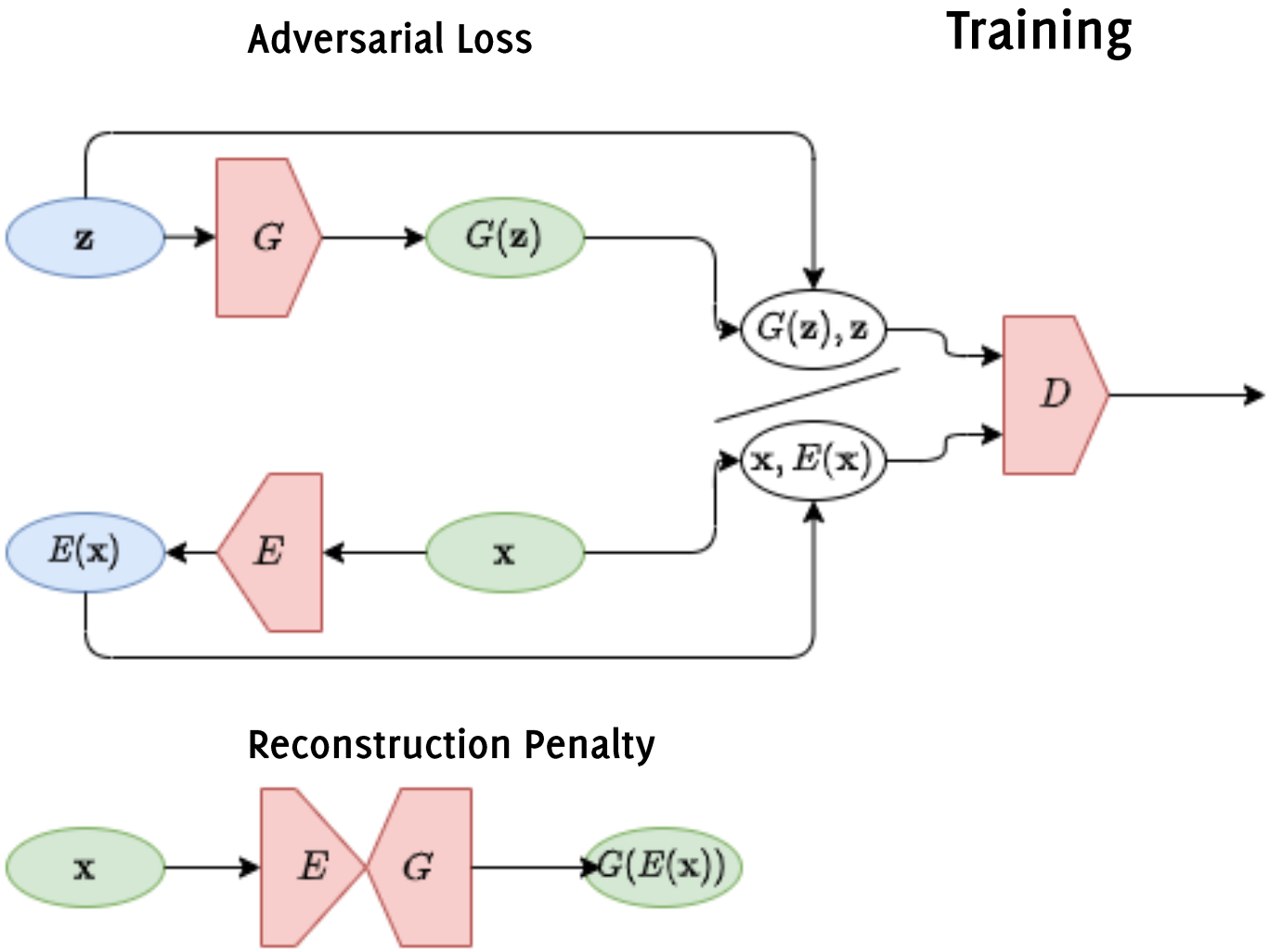


Zenati, H., Foo, C. S., Lecouat, B., Manek, G., & Chandrasekhar, V. R. (2018). Efficient gan-based anomaly detection. ICLR Workshop 2018

FULLY CONVOLUTIONAL ANOMALY DETECTION BY GANS

Training

- All the layers are made fully convolutional (much more efficient processing)
- LS-losses used to train the model (this improves stability)



Credits Stefano Pecchia

FULLY CONVOLUTIONAL ANOMALY DETECTION BY GANS

Inference

Anomaly score: combines

I. image reconstruction error

$$||\mathcal{G}(\mathcal{E}(\boldsymbol{s})) - \boldsymbol{s}|\Big|_{\mathbf{2}}$$

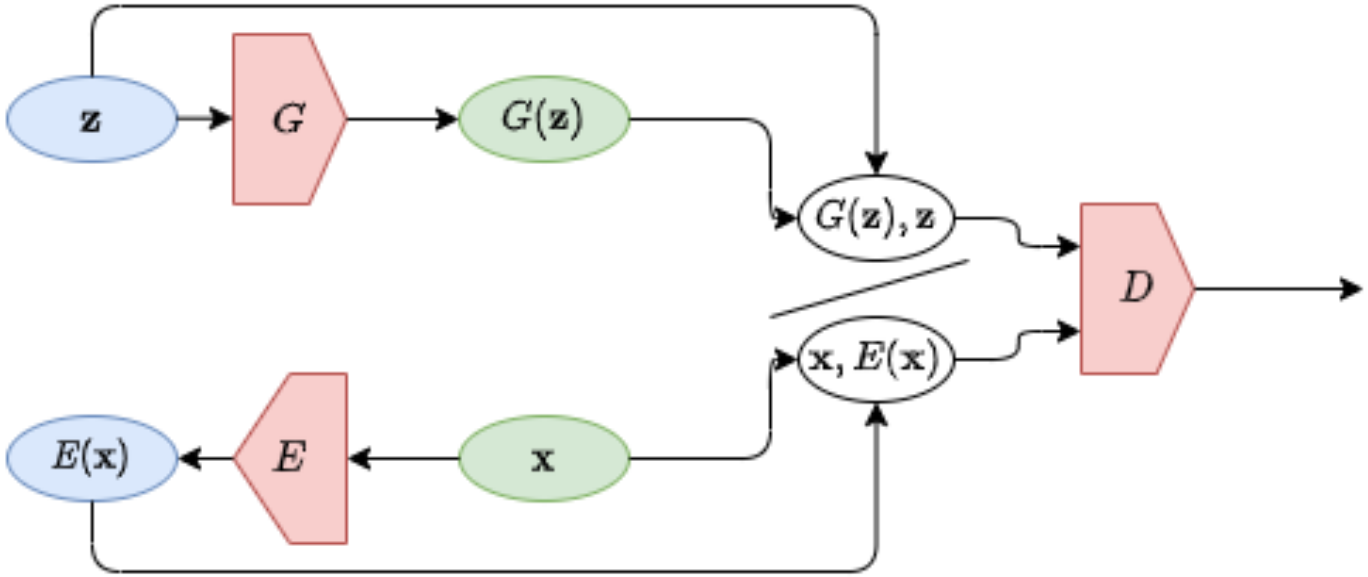
II. discriminator loss

$$\left(\mathcal{D}(s, \mathcal{E}(s)) - 1\right)^2$$

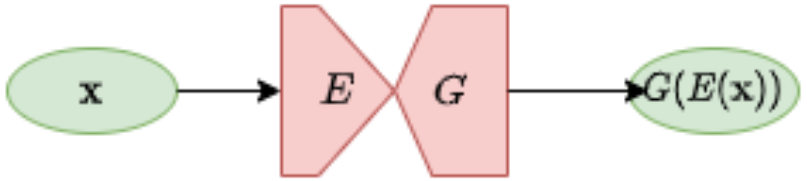
(since normal images are assumed to return 1)

Possibly also likelihood w.r.t estimated distribution of $\mathcal{E}(s)$

Adversarial Loss

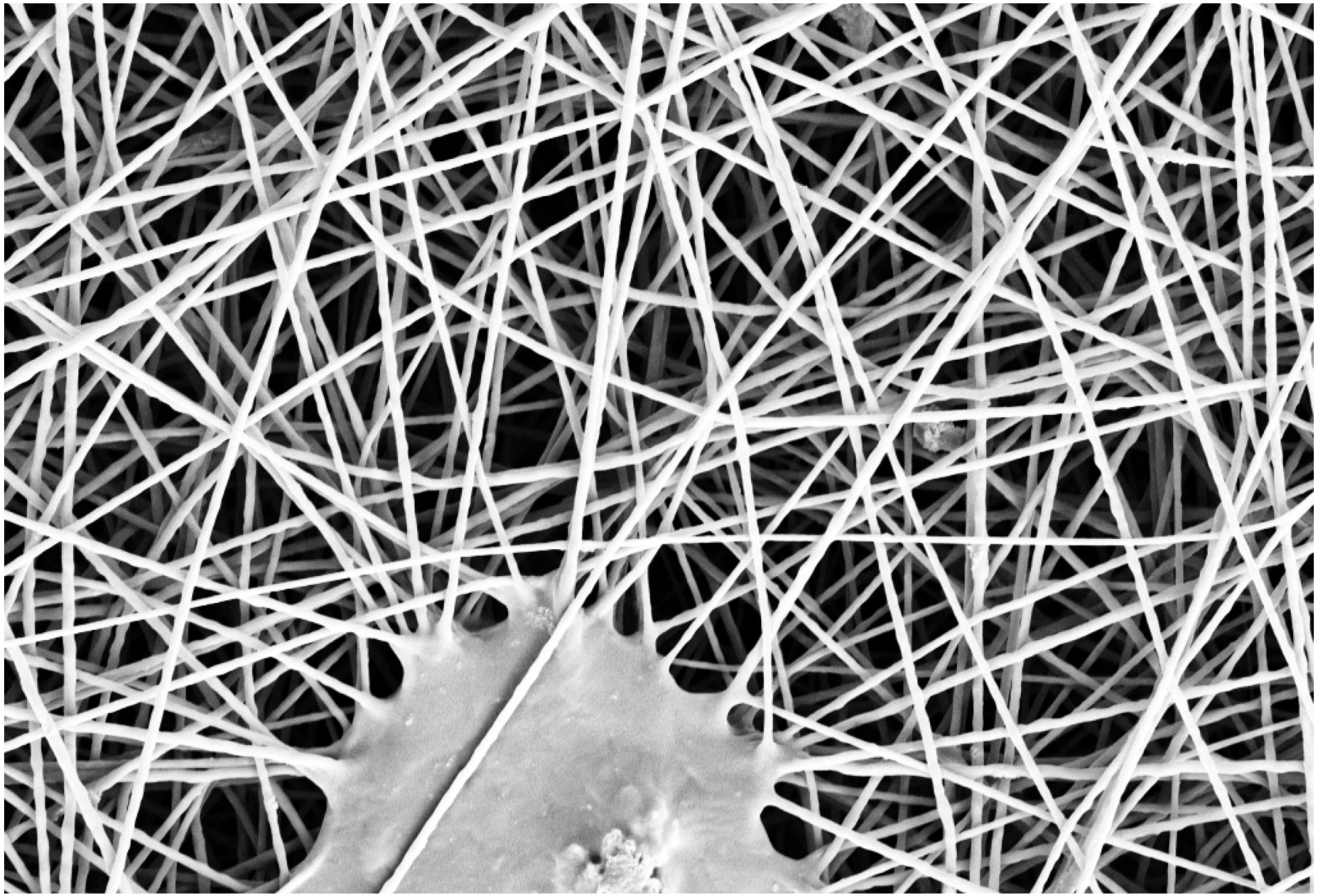


Reconstruction Penalty



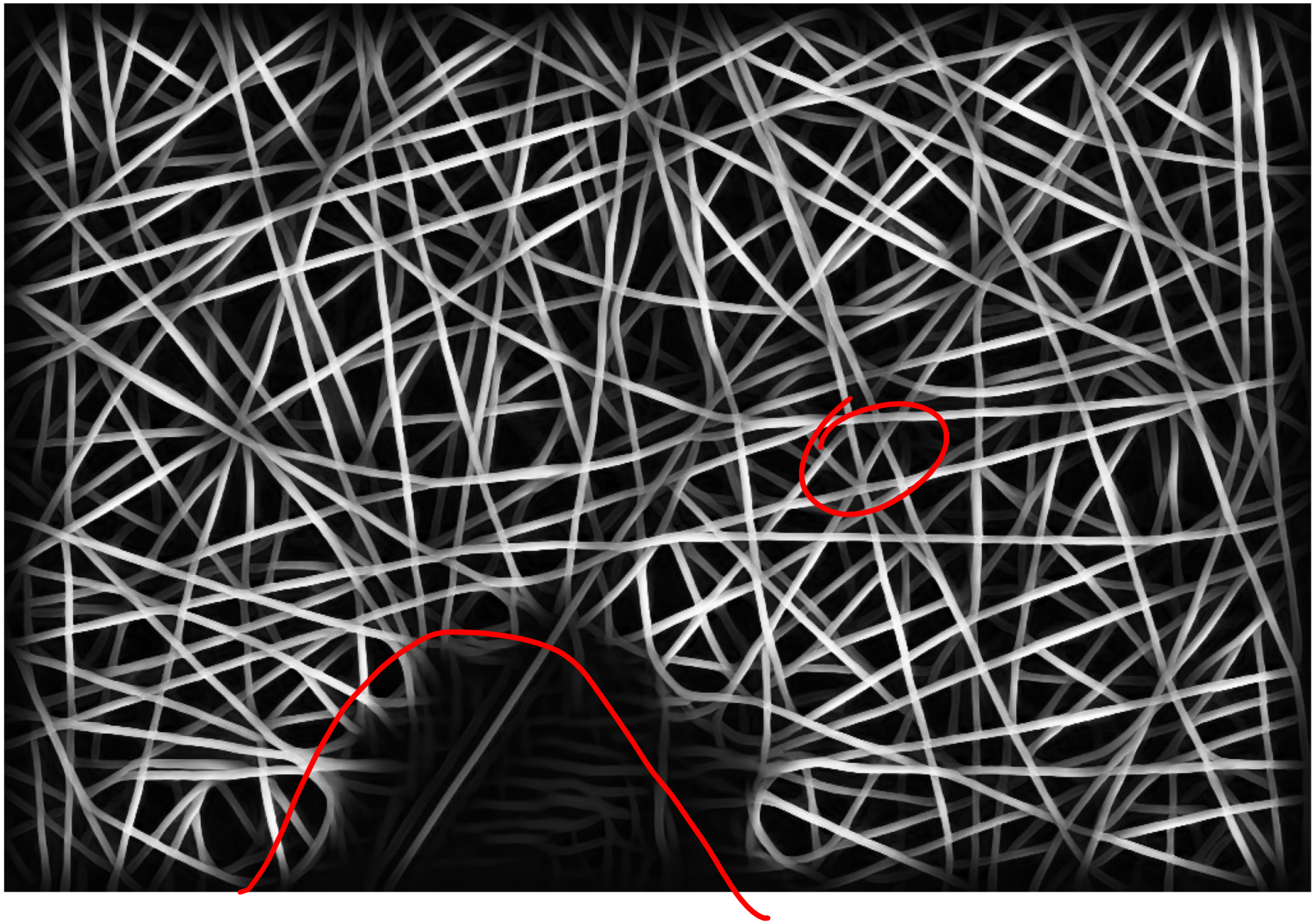
Credits Stefano Pecchia

INPUT IMAGE



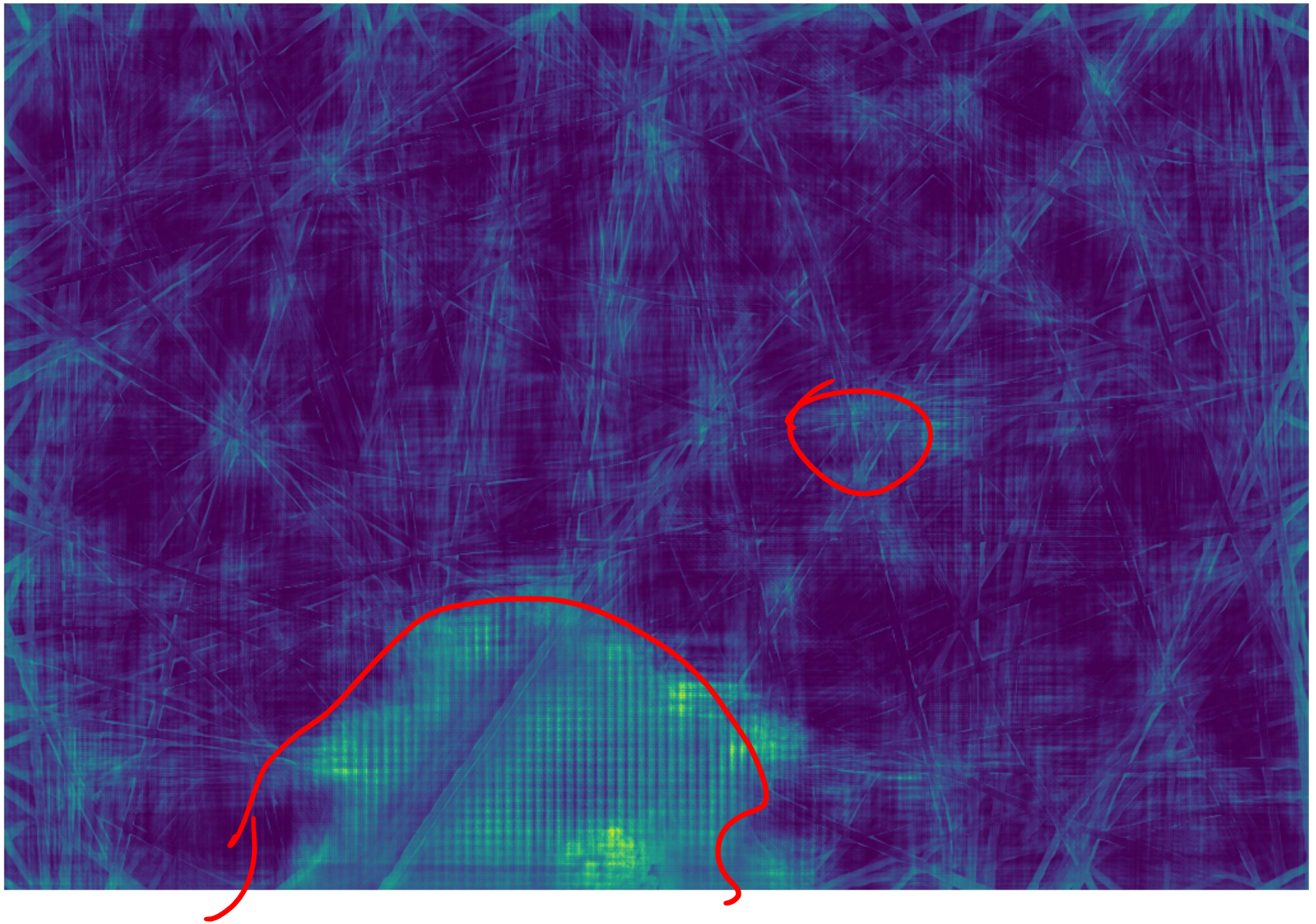
Credits Stefano Pecchia

RECONSTRUCTION $\mathcal{G}(\mathcal{E}(\boldsymbol{s}))$



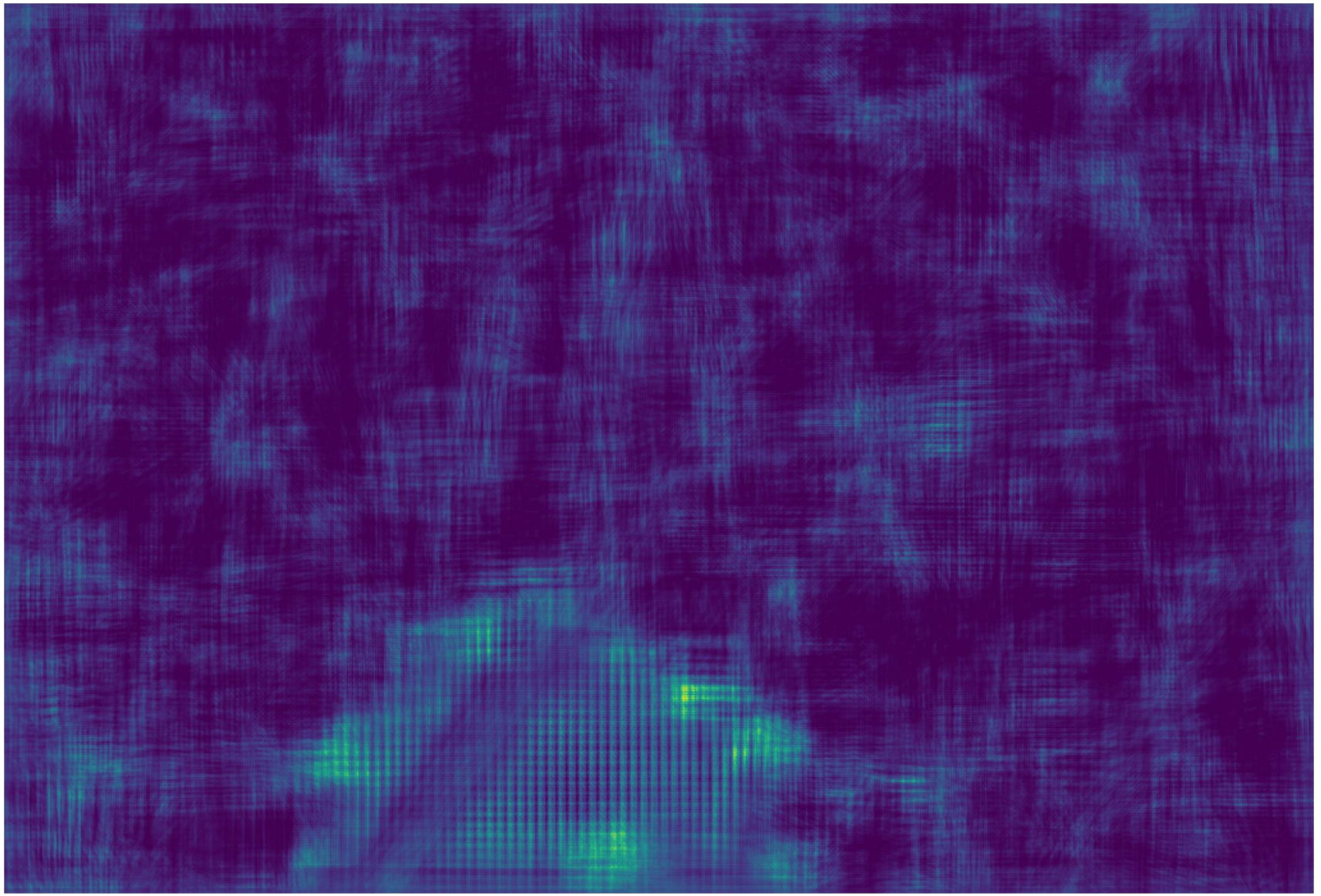
Credits Stefano Pecchia

RECONSTRUCTION LOSS $||\mathcal{G}(\mathcal{E}(\boldsymbol{s})) - \boldsymbol{s}||_2$



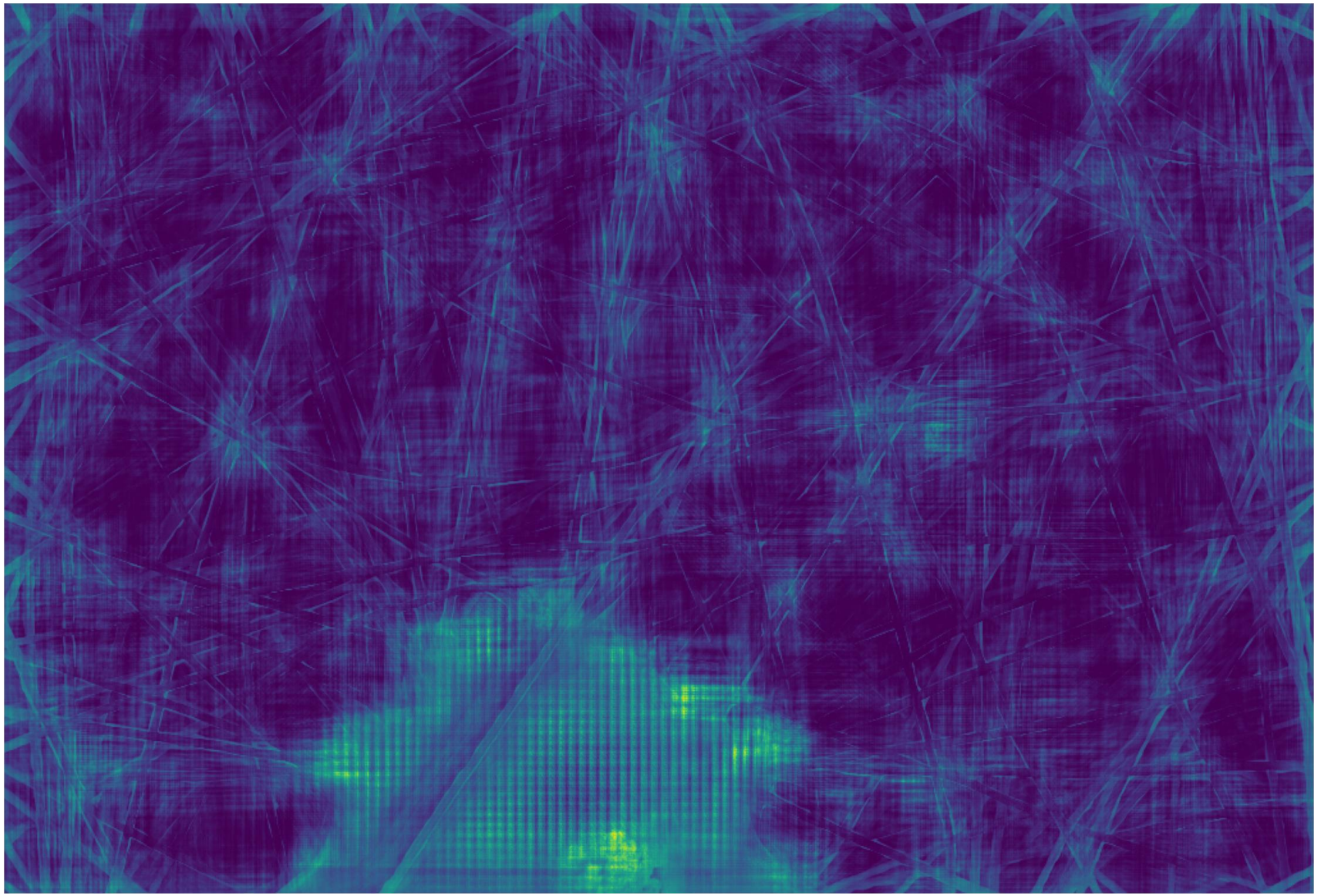
Credits Stefano Pecchia

DISCRIMINATOR SCORE $\left(\mathcal{D}\left(s, \mathcal{E}(s)\right) - 1\right)^2$



Credits Stefano Pecchia

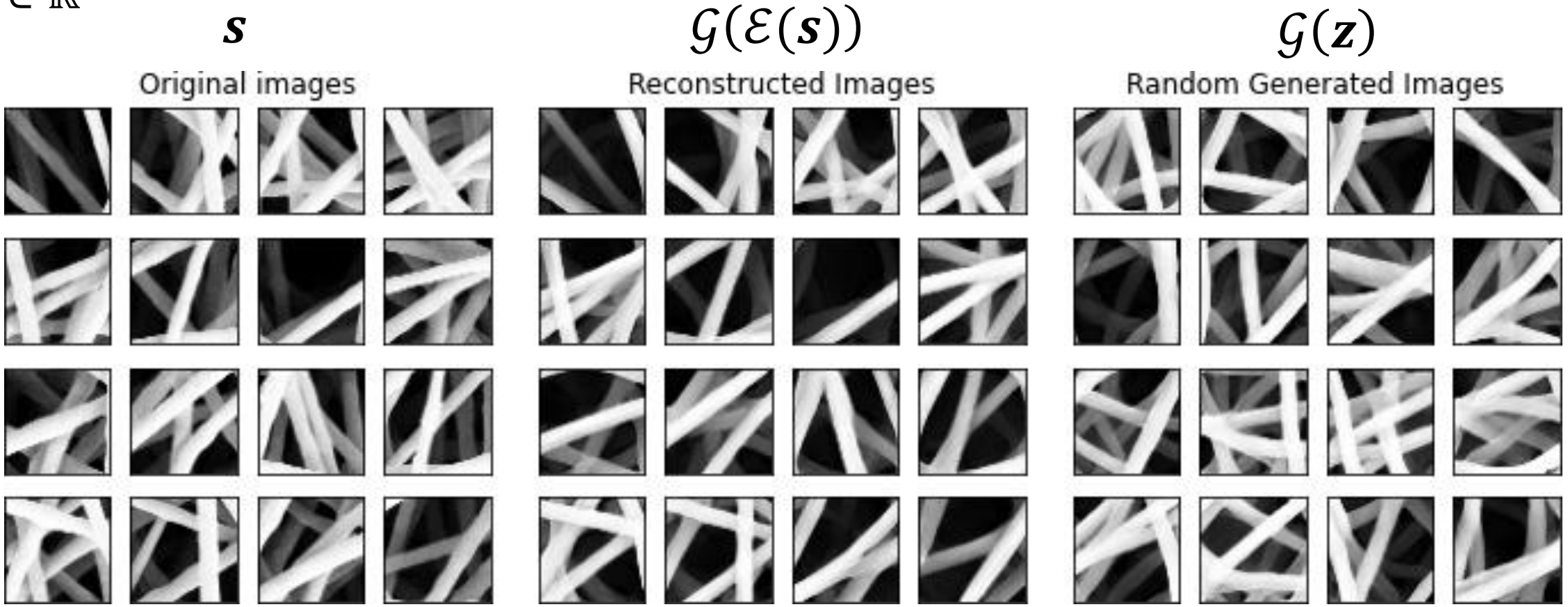
ANOMALY SCORE $\alpha\left(\mathcal{D}\left(s, \mathcal{E}(s)\right) - 1\right)^2 + (1-\alpha)||\mathcal{G}\left(\mathcal{E}(\boldsymbol{s})\right) - \boldsymbol{s}||_2$



Credits Stefano Pecchia

NORMAL IMAGE GENERATION BY OUR GAN

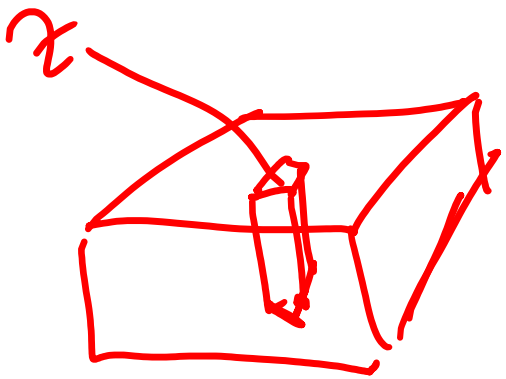
$z \in \mathbb{R}^{128}$

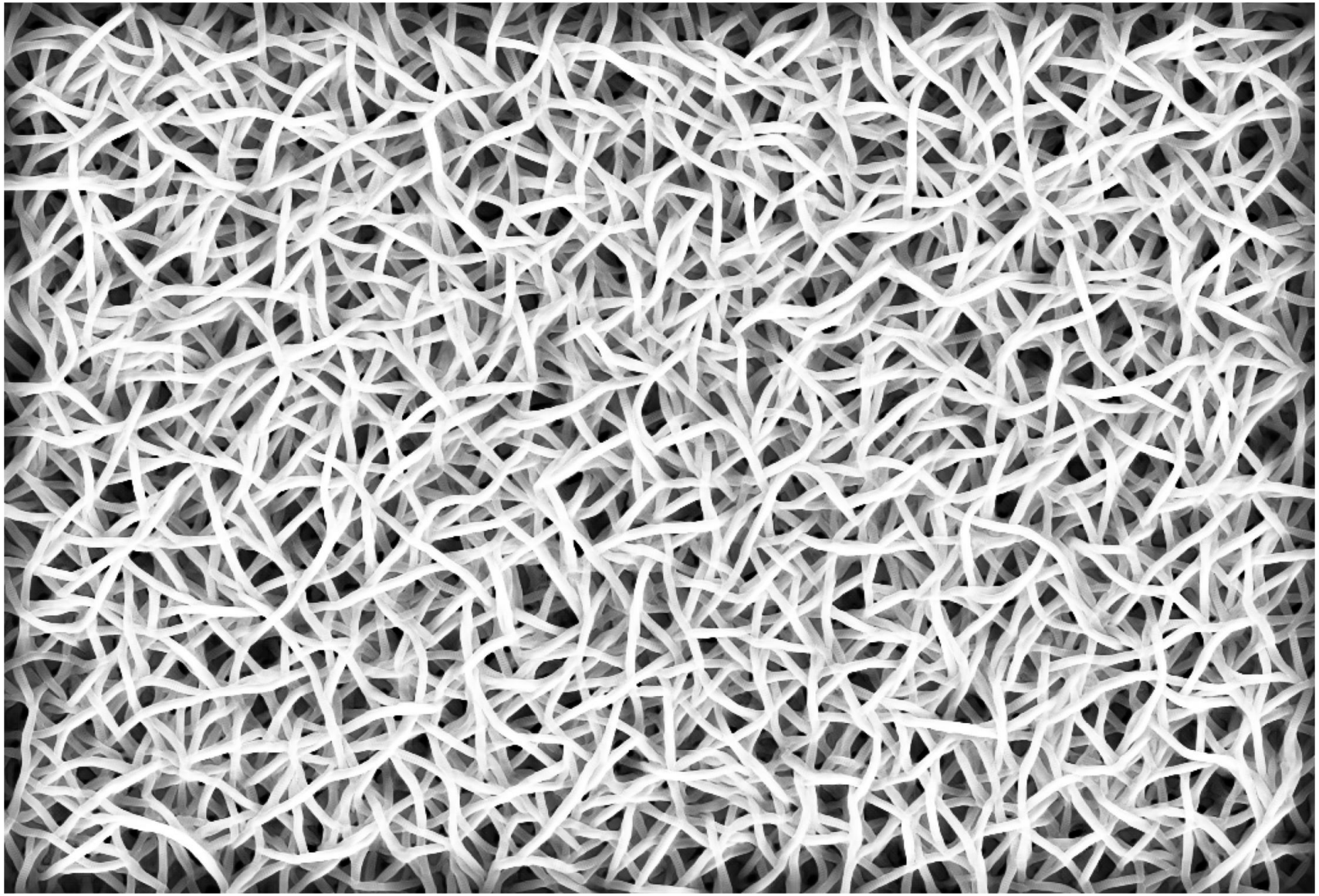


Credits Stefano Pecchia Thesis

IT IS A GENERTIVE MODEL!

$$z \in \mathbb{R}^{80 \times 120 \times 128}$$

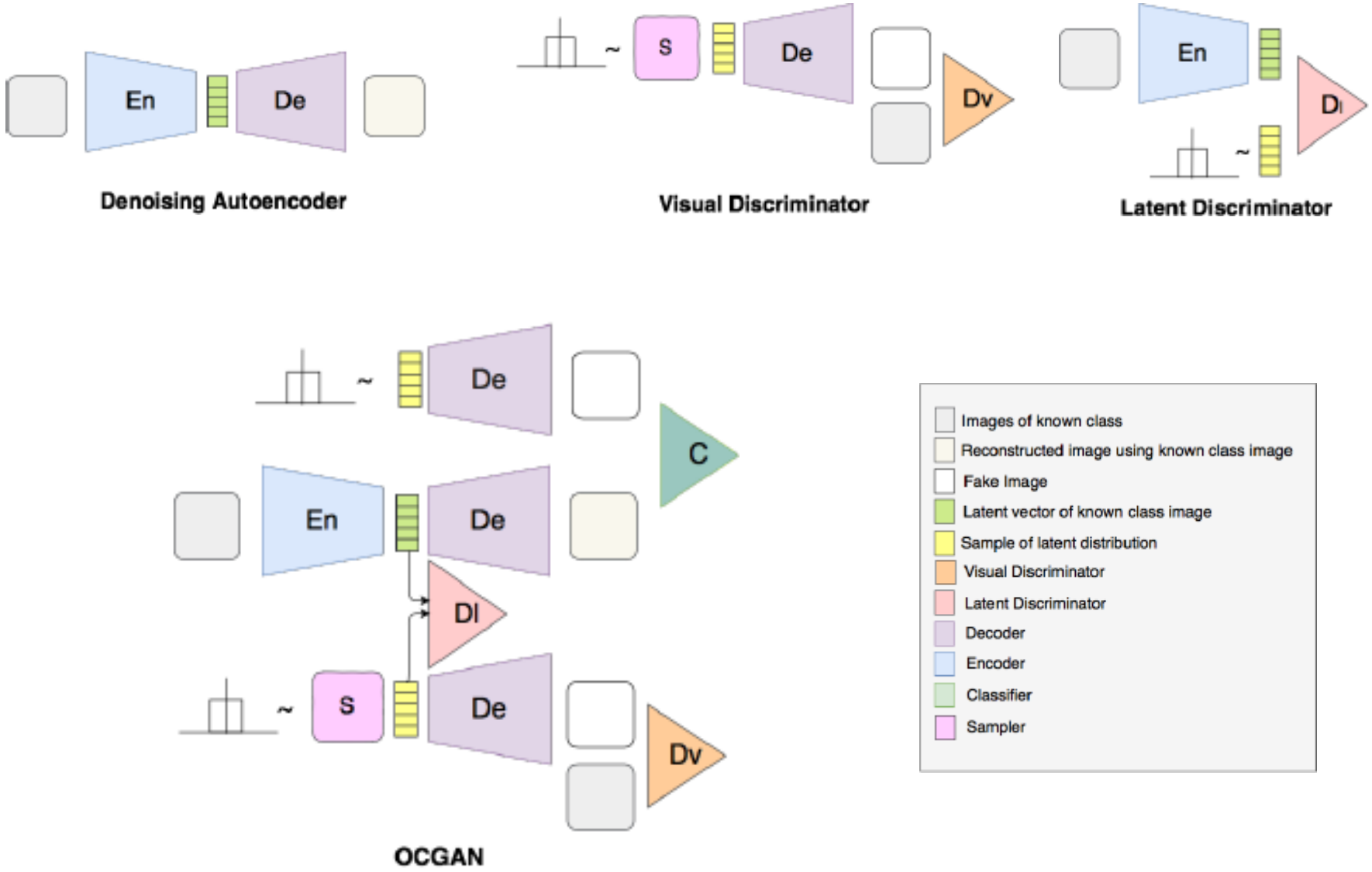




Local regions are well connected, but the GAN do not enforce a global image structure

Credits Stefano Pecchia

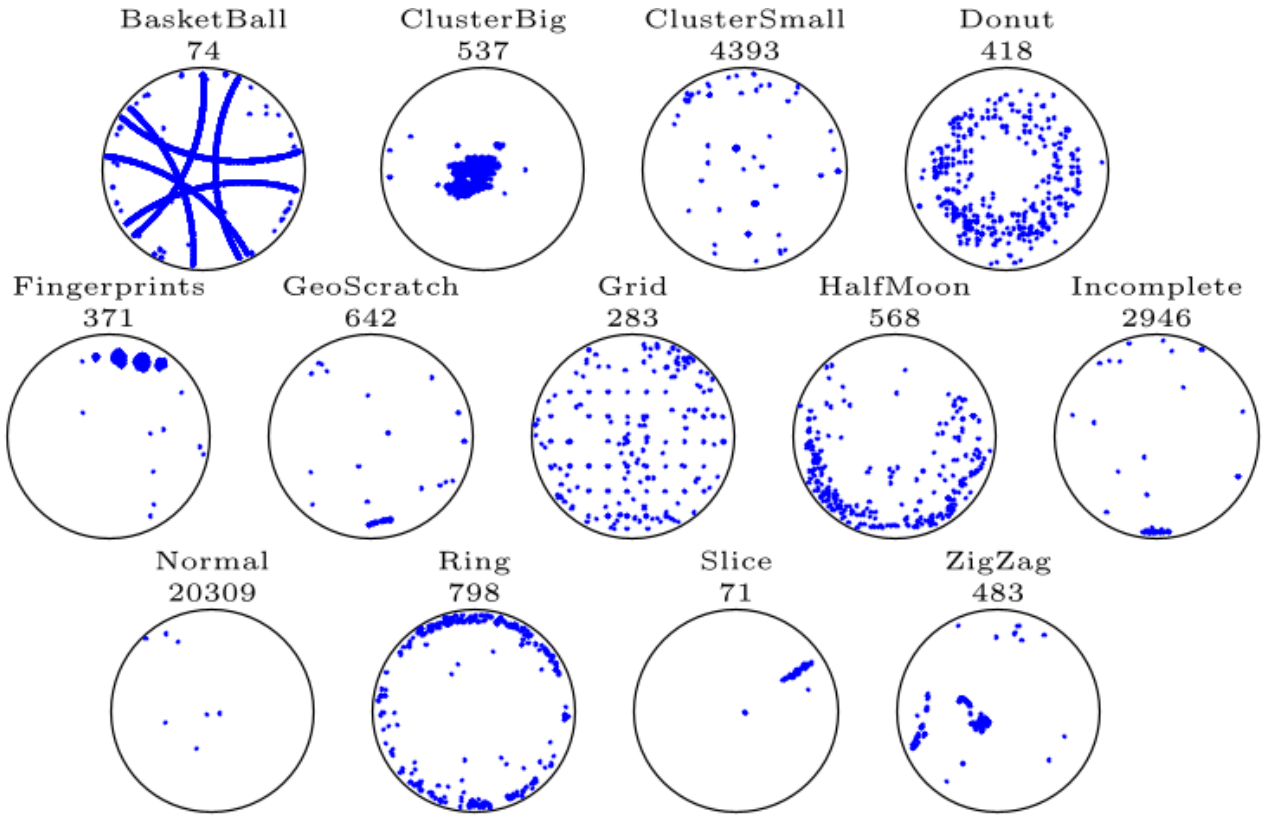
… MUCH MORE COMPLICATED ARCHITECTURES



Perera et al, "*OCGAN: One-class novelty detection using GANs with constrained latent representations*", CVPR 2019

A FEW PROJECT WORTH MENTIONING

OPEN SET RECOGNITION FOR WDM MONITORING



Goals

classify WDMs into 12 known classes of defect patterns + the normal class (no pattern)

detect WDMs from unknown classes, i.e. containing defect patterns that have not been observed yet

Challenge

WDMs are lists of defect coordinates on a 20,000x20,000 grid, thus cannot be directly processed by CNNs

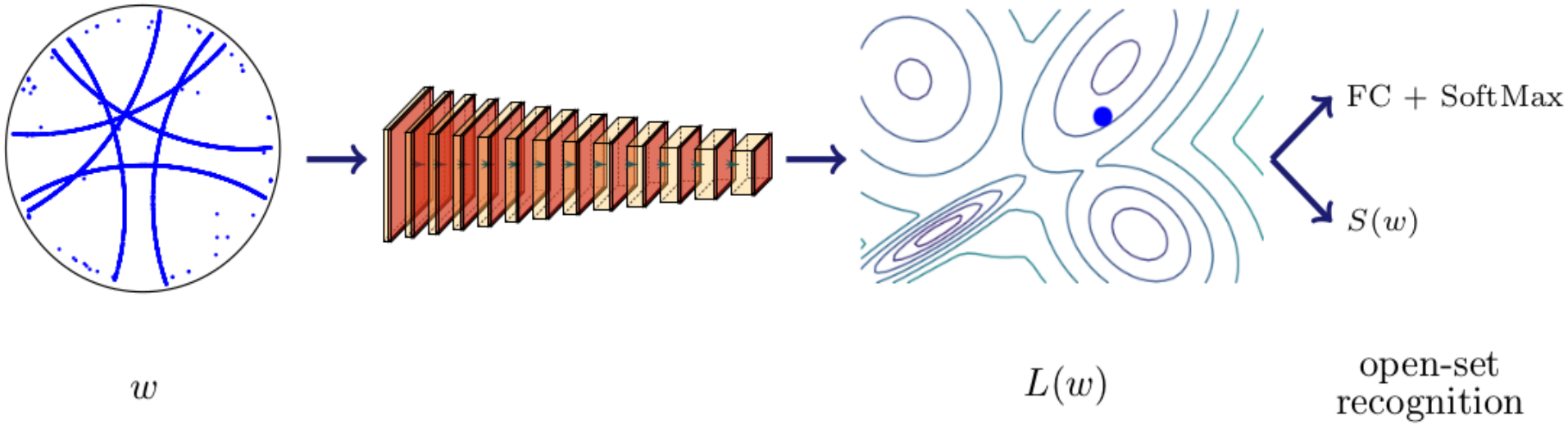
Di Bella, Carrera, Rossi, Fragneto, Boracchi *Wafer Defect Map Classification Using Sparse Convolutional Neural Networks* ICIAP 2009
Frittoli, Carrera, Rossi, Fragneto, Boracchi, *Deep Open-Set Recognition for Silicon Wafer Production Monitoring* Pattern Recognition 2021

OPEN SET RECOGNITION FOR WDM MONITORING

We propose a Submanifold Sparse Convolutional Network (SSCN) [1] trained on full-resolution WDMs fro known classes

We fit a GMM on the latent representation of known classes, and use the log-likelihood as anomaly score [2]

$$S(w) = -\log\left(\hat{\phi}\left(L(w)\right)\right)$$



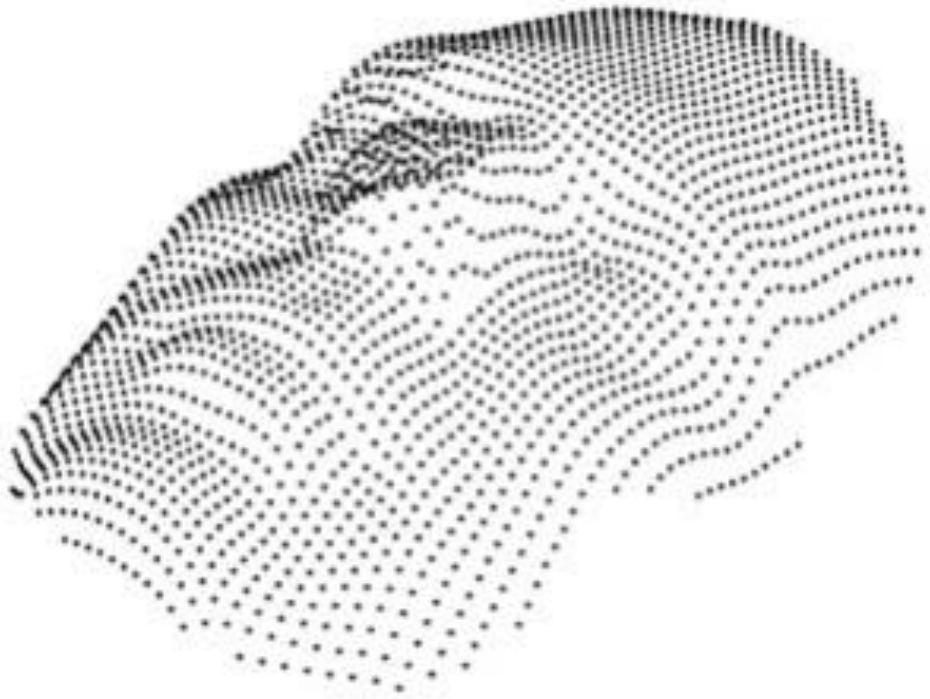
Di Bella, Carrera, Rossi, Fragneto, Boracchi *Wafer Defect Map Classification Using Sparse Convolutional Neural Networks* ICIAP 2009
Frittoli, Carrera, Rossi, Fragneto, Boracchi, *Deep Open-Set Recognition for Silicon Wafer Production Monitoring* Pattern Recognition 2021

ANOMALY DETECTION IN 3D POINT CLOUDS

3D point clouds are a compact and informative data format, which are becoming an increasingly popular object in the deep learning research

Deep learning on 3D point clouds addresses mostly supervised tasks such as classification and semantic segmentation

Unsupervised anomaly detection has not been widely investigated yet

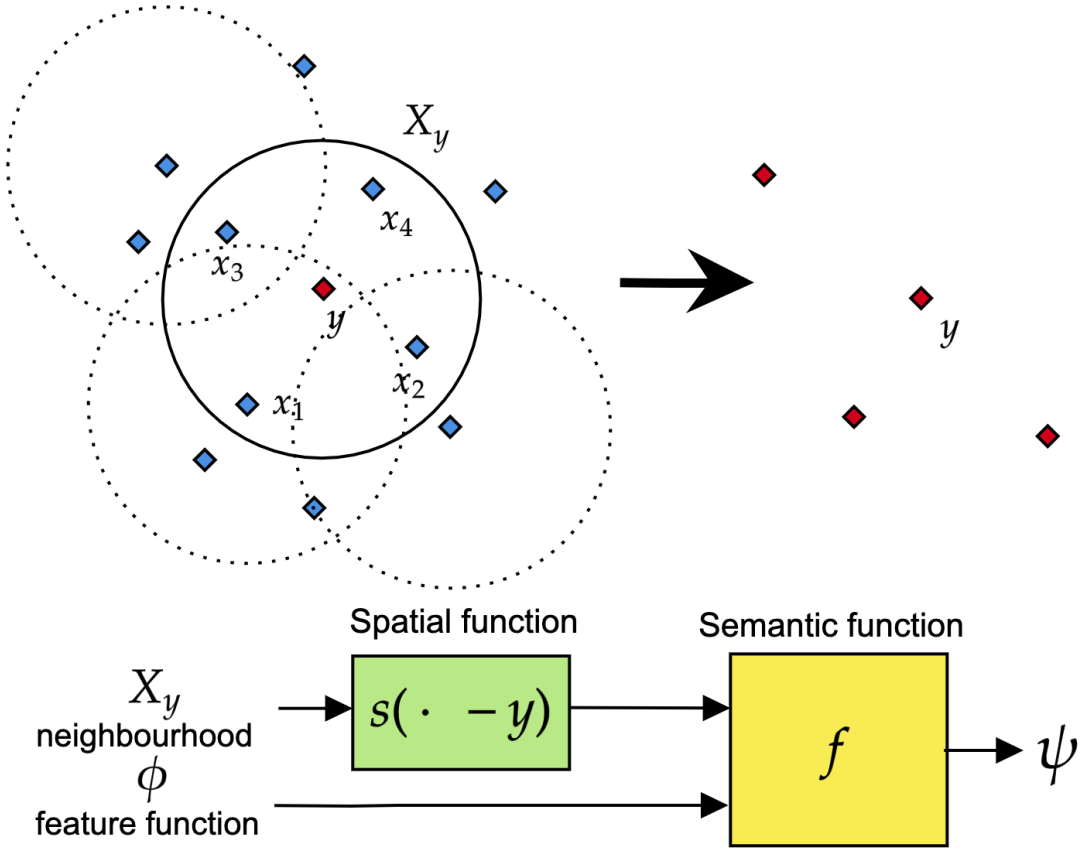


Guo et al. “Deep learning for 3D point clouds: a survey” TPAMI 2020

ANOMALY DETECTION IN 3D POINT CLOUDS

We propose the composite layer, a novel operator for deep NNs that separately processes spatial information and features in 3D point clouds

Our CompositeNets perform similarly to existing point-CNNs in classification, and we successfully apply them to anomaly detection following a self-supervised approach based on geometric transformations



Credits Alberto Floris
Golan and El-Yaniv “Deep anomaly detection using geometric transformations” NeurIPS 2018

CONCLUDING REMARKS

A FEW CONCLUDING REMARKS

Nowadays, anomaly detection problems are **ubiquitous** in **engineering and applied sciences**.

The presented general **framework encompasses most of algorithms** in the literature, which often **boil down to**

- **Feature extraction**
- **Definition of suitable statistics**
- **Applying decision rules** to a set of **random variables.**

A FEW CONCLUDING REMARKS

When **data** are characterized by **complex structures**, as in case of images and signals, the feature extraction phase is the most critical one.

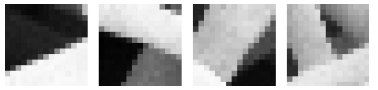
Data-driven models provide **meaningful representations** to images, that can be used to extract good feature for detection.

Nowadays the most powerful algorithms for feature extraction are based on deep learning, and in particular **Convolutional Neural Networks**

A FEW CONCLUDING REMARKS

CNNs can be used either:

- As **data-driven feature extractor** that are put on top of an anomaly detector designed for random vectors
  - The best performance are achieved where the CNN and the anomaly detector are **jointly learned**
- As a **generative model** that allows to sample from the distribution of normal images
  - This generator has to be somehow inverted for anomaly detection



A FEW CONCLUDING REMARKS

An increased availability of annotated datasets have boosted the research in Deep Learning models for Anomaly Detection.

MVTEC AD seems so for the reference standard

NanoTwice: http://web.mi.imati.cnr.it/ettore/NanoTWICE/

MVTEC AD https://www.mvtec.com/company/research/datasets/mvtec-ad

MVTEC 3D-AD https://www.mvtec.com/company/research/datasets/mvtec-3d-ad

STEEL https://www.kaggle.com/c/severstal-steel-defect-detection/data

SOME RESOURCES

Public software:

- Anomaly detection using sparse representations
  http://boracchi.faculty.polimi.it/boracchi/Projects/projects.html
- https://github.com/PramuPerera/OCGAN
- https://github.com/izikgo/AnomalyDetectionTransformations
- https://github.com/houssamzenati/Efficient-GAN-Anomaly-Detection.git
- https://github.com/lukasruff/Deep-SVDD