Università degli Studi di Genova

PhD Program in Computer Science and Systems Engineering

ROBUST LEARNING UNDER COVARIATE SHIFT AND ANOMALIES

by

WATUSADISI MAVAKALA Arnaud

Thesis submitted for the degree of Doctor of Philosophy (XXXVIII cycle)

February 2026

Lorenzo Rosasco	Supervisor
Giorgio Delzanno	Head of the PhD Program
Giulia Denevi	Co-supervisor
Roberto Morelli	Co-supervisor

Department of Informatics, Bioengineering, Robotics and Systems Engineering

Acknowledgements

I am extremely grateful to Professor Lorenzo Rosasco, not only for having taught me everything I know about research, but also for his invaluable guidance beyond the academic sphere. I would like to extend special thanks to Andrea Della Vecchia, with whom I also had the honor of collaborating, as well as to all the people who spent time in the Machine Learning Genova Center laboratories for their insightful discussions, shared meals, and for helping create a welcoming and supportive community. I also wish to thank Professor Joseph Moto Kosarade of the Catholic University of Congo for his guidance and motivating support. Finally, I would like to note that the research conducted by myself (Arnaud M. Watusadisi) and Professor Lorenzo Rosasco was funded by a grant from the Liguria Region in collaboration with Leonardo SpA, under the Trustworthy AI project. I am particularly grateful to my co-supervisors, Giulia Denevi and Roberto Morelli, for their warm welcome and the fruitful collaboration at the aerospace company Leonardo SpA.

In loving memory of my father

WATUSADISI LAVIA Albert

Abstract

This thesis investigates the role of robustness in machine learning for high-dimensional and safety-critical data, focusing on both covariate-shifted regression and anomaly detection. We study robustness to distributional changes in supervised learning, addressing covariate shift in nonparametric regression within reproducing kernel Hilbert spaces (RKHSs). By leveraging random projection techniques, including Nyström approximations, to restrict the hypothesis space, we achieve substantial computational savings while preserving predictive accuracy under changing input distributions. We also examine robustness in anomaly detection for satellite telemetry. We employ a Controlled Latent Space Model (CLSM), a semi-supervised autoencoder that learns normal operational patterns and enables the detection of anomalies through reconstruction errors. The robustness of the model is evaluated under environmental perturbations and adversarial attacks. The results provide both practical and theoretical insights for monitoring and fault diagnosis, as well as for the development of trustworthy AI systems in operational environments such as satellite telemetry.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Classical supervised learning assumes that training and test data are drawn from the same underlying distribution [Vapnik, 1998]. In practice, this assumption often does not hold due to changes in measurement devices, environmental conditions, or operational contexts [Cortes et al., 2010a; Shimodaira, 2000]. *Covariate shift* arises when the marginal distribution of inputs changes between training and testing, while the conditional distribution of outputs given inputs remains unchanged [Gizewski et al., 2020; Shimodaira, 2000]. Addressing covariate shift requires models that are robust to such distributional changes. A standard solution is importance weighting, which reweights the training loss to match the test distribution [Cortes et al., 2010a; Shimodaira, 2000]. However, when combined with kernel methods, importance weighting can become computationally expensive due to the need to solve large linear systems.

Satellite telemetry provides multivariate time-series measurements of spacecraft subsystems. Telemetry is typically temporally correlated and noisy, and the physical quantity measured by each channel is often unknown, making automated anomaly detection challenging [Kotowski et al., 2024; Ruszczak et al., 2024]. Traditional rule-based methods rely on expert thresholds and scale poorly

as telemetry complexity increases [Hawkins, 1980; Kotowski et al., 2024; Ruszczak et al., 2024]. Data-driven approaches, especially unsupervised methods, learn normal operational patterns from telemetry without requiring labeled anomalies [Chandola et al., 2009; Sakurada and Yairi, 2014]. The European Space Agency Anomaly Benchmark (ESA-ADB) provides real spacecraft telemetry with expert annotations, enabling systematic evaluation [Kotowski et al., 2024]. A key challenge in anomaly detection for telemetry is robustness, defined as the ability of a model to maintain reliable detection performance under noise, sensor errors, and adversarial perturbations [Finlayson et al., 2019; Song et al., 2018].

The two problems are usually studied separately, yet they share a deeper structural link beyond the generic notion of robustness. In both cases, the model is trained on a distribution (the training input distribution for covariate shift, the normal operating condition for telemetry) and then applied to test data that may not follow that same distribution. Under covariate shift, the input distribution changes, and an importance weighting function provides a correction. In anomaly detection, anomalies are points that do not follow the nominal distribution, and no such weighting function is available. Instead, the model computes a *reconstruction error* from an autoencoder trained on nominal data to quantify how much a test sample differs from the nominal training data. Thus, both settings require methods that are robust to distributional mismatch, but they differ in the information available. Covariate shift uses an importance weighting function, while anomaly detection uses reconstruction error.

1.2 Research Focus

This thesis pursues two main research directions. The first concerns efficient kernel methods under covariate shift, where we study the trade-off between statistical robustness and computational efficiency. The second concerns robust semi-supervised

anomaly detection in satellite telemetry, where we provide a systematic empirical analysis of autoencoder based methods under various perturbations.

- *Efficient kernel methods under covariate shift.*

  Importance weighted kernel ridge regression (KRR) is a standard approach for covariate shift, with well established statistical guarantees [Gogolashvili et al., 2023; Ma et al., 2023]. However, the exact solution requires solving a linear system of size n, where n is the training sample size, which requires $O(n^3)$ time and $O(n^2)$ memory. This becomes impractical for large n. Approximation methods such as Nyström subsampling reduce these costs [Rudi et al., 2015]. Prior work by Myleiko and Solodky [2024] analyzes regularized Nyström subsampling under covariate shift and obtains optimal learning rates, but their analysis applies only to bounded importance weights and relies on a specific iterated Tikhonov regularization scheme. The main novelty of this thesis relative to prior art is as follows. First, we prove that a Nyström projection based on leverage score sampling preserves the *full weighted KRR learning rate* when the importance weighting function is bounded. This provides a complementary analysis to Myleiko and Solodky [2024], using a different sampling strategy. Second, we extend the analysis to unbounded importance weighting functions. Under stronger moment conditions, the same Nyström projection still achieves the optimal rate. This setting is not covered by prior analyses (including Myleiko and Solodky [2024]), which explicitly assume bounded importance weights. Our result thus provides the first guarantee for scalable kernel approximations under covariate shift with unbounded importance weighting.

- *Robust semi-supervised anomaly detection.*

  Recent work has introduced the European Space Agency Anomaly Bench-

mark (ESA ADB) and OPS SAT telemetry for anomaly detection research [Kotowski et al., 2024; Ruszczak et al., 2024]. Autoencoder based methods, synthetic anomaly injection, per channel thresholding, and event wise evaluation are now established practices. In this thesis, we adopt a 1D convolutional autoencoder with a controlled bottleneck dimension referred to as the Controlled Latent Space Model (CLSM) as a representative architecture. Our contribution is a systematic empirical analysis of robustness on the ESA ADB benchmark.

1.3 Publications

The main contributions of this thesis have been published and presented as:

- Della Vecchia, A., Watusadisi, A. M., De Vito, E., & Rosasco, L. (2025). **Computational Efficiency under Covariate Shift in Kernel Ridge Regression**. Spotlight at NeurIPS 2025 [Della Vecchia et al., 2025],
- A., Watusadisi, Giulia Denevi, Rosasco, L., & Roberto Morelli, **Detecting and Localizing Anomalies in Satellite Telemetry Using Semi-Supervised Autoencoder Ensembles**, Poster at ICAIR 2025.

1.4 Thesis Organization

The remainder of this thesis is structured as follows:

- **Chapter 2 – Kernel-Based Learning and Approximation Methods:** This chapter provides a comprehensive review of supervised learning frameworks, with a focus on kernel methods and their theoretical properties. Key concepts such as reproducing kernel Hilbert spaces (RKHS), kernel ridge regression (KRR), and regularization techniques are introduced. The

chapter also surveys approximation strategies, including random projections, Nyström methods, and other dimensionality reduction techniques, which form the foundation for scalable kernel learning in high-dimensional settings.

- **Chapter 3 – Kernel Ridge Regression under Covariate Shift:** Building on Chapter 2, this chapter investigates KRR when the training and test distributions differ. We discuss importance-weighted risk minimization, the impact of covariate shift on statistical guarantees, and computational challenges. We prove that ALS-based Nyström projections retain the full weighted-KRR learning rate for bounded importance weights, and extend the result to unbounded weights under stronger moment assumptions. Simulations and real-data experiments illustrate these theoretical findings and characterize the trade-off between accuracy, computational cost, and sensitivity to weight misspecification.

- **Chapter 4 – Robust Anomaly Detection in Satellite Telemetry:** This chapter focuses on the design and evaluation of semi-supervised models for multivariate time series anomaly detection. We present the Controlled Latent Space Model, a 1D convolutional autoencoder with latent compression, and describe methods for segmenting telemetry windows, modeling normal operational patterns, and injecting synthetic anomalies. Extensive experiments quantify robustness under environmental perturbations and adversarial attacks, while channel-wise detection and thresholding strategies enhance interpretability and reliability in operational settings.

- **Chapter 5 – Conclusions:** This chapter synthesizes the main contributions of the thesis, highlighting the role of robustness in both covariate-shifted regression and anomaly detection, and the importance of computational efficiency in kernel-based methods. It discusses limitations and outlines

future research directions, including analysis of source-target misalignment and adaptive kernel approximations for KRR, as well as advanced adversarial defenses and FDIR integration for anomaly detection in satellite telemetry.

Part I

KRR under Covariate Shift: Algorithms & Theory

Chapter 2

Learning with Kernels

Kernel methods map data into a high-dimensional feature space where linear methods can be applied. A positive definite kernel function computes inner products in that space without ever forming the feature vectors explicitly. This makes it possible to handle non-linear relationships while keeping the computational benefits of linear algebra. However, kernel methods become expensive when the sample size is large. Approximation techniques such as Nyström subsampling reduce these costs while preserving statistical performance. In this chapter, we first recall the standard supervised learning framework, then introduce kernel ridge regression, and finally discuss kernel approximation strategies that form the basis for the scalable algorithms studied later.

2.1 Supervised Learning Framework

In supervised learning, the training set consists of n examples, each represented as a pair

$$(x_1, y_1), \ldots, (x_n, y_n),$$

where x_i denotes the input and y_i the corresponding output (or label in classification problems). The inputs belong to a space X, typically a subset of $\mathbb{R}^d$, while the outputs take values in a space Y. The dimension d of the input space

corresponds to the number of features describing each example. Depending on the nature of the output space, we distinguish several classical settings: if $Y \subseteq \mathbb{R}$, the problem is a regression task; if $Y = \{-1, 1\}$, it is a binary classification problem; and if $Y = \{1, 2, \ldots, K\}$, it corresponds to a multiclass classification problem. The data space is denoted by $Z = X \times Y$. The goal of learning is to understand the relationship between inputs x and outputs y. From the training examples, we learn an approximate function, denoted by $\hat{f}$. To evaluate the quality of $\hat{f}$, we consider its generalization error: given a new, previously unseen input-output pair

$$(x_{\text{new}}, y_{\text{new}}) \in X \times Y,$$

we measure how far the prediction $\hat{f}(x_{\text{new}})$ is from the true value y_{new}, i.e., the error on this new example. Measuring the error only on the training set would favor models that reproduce the training data exactly, including the unavoidable noise. Such models suffer from overfitting, failing to capture the true target function f^*. In contrast, models that generalize well achieve better performance in approximating the real function f on new data, which become available only after training.

Formally, supervised learning can be studied from the perspective of statistical learning theory. This framework introduces a data sampling mechanism, loss functions to evaluate the quality of approximation, and the notion of a hypothesis space, within which the optimal function $\hat{f}$ is identified. Before defining the distributions underlying the learning process, we first formalize the probabilistic setting in which observations and outputs are generated.

2.1.1 Data Distribution

We consider the space $X \times Y$ as a probability space with a distribution ρ. The marginal distribution of ρ on X is denoted by ρ_X, and the conditional distribution

of Y given an observation $x \in X$ is denoted by $\rho(\cdot \mid x)$. For a pair of random variables $(X, Y) \in X \times Y$, the probability that the output takes the value y given the input x is:

$$\rho(y \mid x) = \mathbb{P}(Y = y \mid X = x). \tag{2.1}$$

The distribution ρ generating the input-output examples is unknown and cannot be observed directly. We only have access to a training set, assumed to consist of examples drawn independently and identically distributed (i.i.d.) according to ρ. In some contexts, however, the examples may have dependencies, as in time series data. Whether the data are independent or dependent, we still need a principled way to measure how well a model performs. This is the role of loss functions and the empirical risk.

2.1.2 Loss Functions and Empirical Risk

To learn from data, we need to evaluate the quality of predictions made by a model $f : \mathcal{X} \to \mathbb{R}$ on data points $(x, y) \in \mathcal{X} \times \mathcal{Y}$. This evaluation is performed using a loss function $l : \mathcal{Y} \times \mathbb{R} \to [0, \infty)$, which quantifies the discrepancy between the predicted value $f(x)$ and the true target y. The loss function serves two main purposes: it adapts to the type of learning task (e.g., regression or classification) and specifies how prediction errors are penalized. For example, in regression, the squared loss increases quadratically with the size of the error, so large deviations are penalized more severely than small ones. This choice of loss is therefore fundamental, as it shapes the behavior of the learning algorithm and determines what types of errors are considered more costly.

- For continuous-value regression ($\mathcal{Y} \subset \mathbb{R}$), the most commonly used loss is the squared loss:

$$l(y, a) = (y - a)^2, \quad y, a \in \mathbb{R}, \tag{2.1}$$

which is differentiable and strongly convex, making it suitable for gradient-based optimization. Another alternative is the absolute loss:

$$l(y, a) = |y - a|, \tag{2.2}$$

which is less sensitive to outliers and promotes sparse solutions in linear models [Tibshirani, 1996].

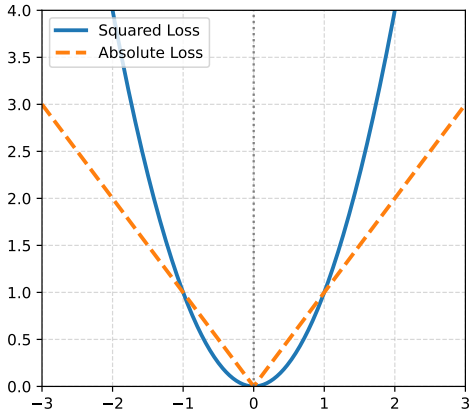


Figure 2.1: The regression losses as presented in Equation (2.1) and Equation (2.2)

- For binary classification ($\mathcal{Y} = \{-1, 1\}$), the natural choice is the 0-1 loss:

$$l(y, a) = \begin{cases} 1, & \text{if } y \neq \operatorname{sgn}(a), \\ 0, & \text{otherwise,} \end{cases} \qquad a \in \mathbb{R}. \tag{2.3}$$

  While this loss directly measures misclassification, it is discontinuous and non-convex, which makes gradient-based optimization infeasible [Feldman et al., 2012]. Therefore, convex surrogates are commonly used, such as:

  – Logistic loss:

$$l(y, a) = \log(1 + e^{-ya}), \tag{2.4}$$

    which corresponds to a probabilistic interpretation of classification.

– Hinge loss:

$$l(y, a) = |1 - ya|_+, \tag{2.5}$$

which is widely used in support vector machines and encourages a margin between classes.

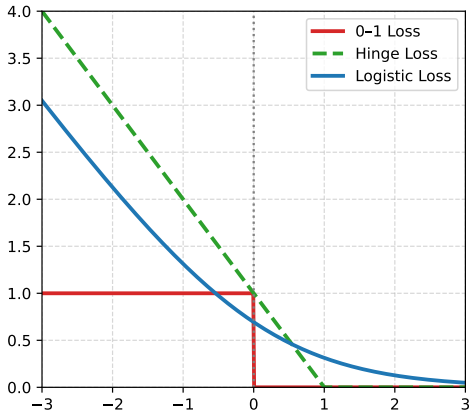


Figure 2.2: The classification losses as presented in Equation (2.3), Equation (2.4) and Equation (2.5)

- For multi-class classification, where $\mathcal{Y}$ consists of T discrete classes, each class is encoded as a one-hot vector in $\mathbb{R}^T$. Continuous losses such as cross-entropy are then used:

$$l(y, p) = -\sum_{i=1}^{T} y_i \log p_i, \tag{2.6}$$

where y is the one-hot encoding of the true label, and p is the predicted probability distribution over the classes. This formulation penalizes predictions that assign low probability to the correct class and is particularly suitable for neural networks and probabilistic models.

Given a probability distribution ρ over the data and a loss function, the ideal metric for evaluating a function f is the expected risk:

$$\mathbb{E}(f) = \int_{\mathcal{X}\times\mathcal{Y}} l(y, f(x))\, d\rho(x, y). \tag{2.7}$$

Minimizing the expected risk over all measurable functions corresponds to finding the optimal prediction function, which cannot be computed exactly because the true distribution ρ is unknown. In practice, we observe a finite training set $Z = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, which induces the empirical risk:

$$\hat{\mathbb{E}}(f) = \frac{1}{n} \sum_{i=1}^{n} l(y_i, f(x_i)). \tag{2.8}$$

The Empirical Risk Minimization (ERM) principle consists of finding

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \hat{\mathbb{E}}(f), \tag{2.9}$$

where $\mathcal{F}$ is the chosen hypothesis space. The size and complexity of $\mathcal{F}$ govern the trade-off between underfitting and overfitting:

- If $\mathcal{F}$ is too large, $\hat{f}$ may perfectly fit the training data, including noise, but generalize poorly to unseen samples.
- If $\mathcal{F}$ is too small, $\hat{f}$ may generalize well, but the best function within $\mathcal{F}$ could be far from the true target function f^*, resulting in systematic bias.

Balancing these effects is central to learning theory and motivates the use of regularization, structural constraints, and careful design of the hypothesis space. This balance between underfitting and overfitting requires a quantitative criterion to compare different candidate functions. That criterion is the risk, and its empirical evaluation provides a basis for choosing the hypothesis space. Regularization offers a practical way to control this tradeoff directly.

2.1.3 Risk Evaluation and Regularization

For regression, the excess risk of an estimator f measures how much worse f performs compared to the optimal predictor given by the regression function:

$$f_\rho(x) = \mathbb{E}[Y \mid X = x]. \tag{2.10}$$

Formally, the excess risk is defined as

$$R_\rho(f) := \mathbb{E}(f) - \mathbb{E}(f_\rho) = \|f - f_\rho\|_{\rho_X}^2, \tag{2.11}$$

where the second equality holds for the squared loss [Cucker and Smale, 2002]. Intuitively, the regression function f_ρ represents the conditional mean of Y given X, and minimizing the expected squared loss corresponds to approximating this conditional expectation as closely as possible. Therefore, the excess risk quantifies the deviation of an estimator from the theoretically optimal predictor. In the context of binary classification, the classifier that minimizes the misclassification probability is known as the Bayes classifier:

$$c^*(x) = \operatorname{sgn}(f_\rho(x)) = \arg\max_{y \in \mathcal{Y}} \rho(y \mid x),$$

which assigns each input x to the class with the highest conditional probability. The Bayes classifier is the theoretical gold standard, achieving the lowest possible classification error given the true data distribution. In practice, however, we only have a finite training set, and unrestricted optimization over a large hypothesis space can lead to overfitting, where the model fits the training data closely but generalizes poorly to unseen samples. Two common strategies are used to mitigate this:

1. Restrict the hypothesis space $\mathcal{F}$, for example by considering only linear or low-degree polynomial functions. This reduces model flexibility and prevents it from fitting noise in the training data.

2. Regularization, which introduces a penalty term on the complexity of the function:

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \hat{\mathbb{E}}(f) + \lambda \|f\|^2, \tag{2.12}$$

where $\lambda \geq 0$ is a hyperparameter controlling the trade-off between fitting

the data and keeping the function smooth or simple. A larger λ discourages overly complex models, reducing overfitting, while a smaller λ allows more flexibility to fit the data closely.

Regularization therefore enables the use of a larger hypothesis space (e.g., all smooth functions) while controlling the complexity of the solution. It provides a principled way to balance bias and variance, improving generalization performance on unseen data. Together with empirical risk minimization, regularization is one of the foundational tools in statistical learning theory. A natural mathematical framework for analyzing such regularized estimators is provided by reproducing kernel Hilbert spaces. These spaces allow us to define function complexity in terms of a norm and to apply the representer theorem, which connects infinite dimensional optimization to finite dimensional computation.

2.2 Reproducing Kernel Hilbert Spaces (RKHS)

In this thesis, we assume H to be a Hilbert space [Bach, 2024; Mohri et al., 2018], which provides a structured mathematical framework for studying functions with a well-defined notion of inner product and norm. One of the simplest examples of a Hilbert space is the space of linear functions:

$$H = \Big\{ f : X \to \mathbb{R} \;\Big|\; f(x) = \langle w, x \rangle,\ w \in \mathbb{R}^d \Big\},$$

associated with the standard Euclidean inner product $\langle f, g \rangle_H = \langle w_f, w_g \rangle_{\mathbb{R}^d}$ when $f(x) = \langle w_f, x \rangle$ and $g(x) = \langle w_g, x \rangle$. This space allows linear modeling of relationships between inputs and outputs. However, its expressive power is limited to linear relationships, which is often insufficient for capturing the complexity of real-world phenomena. For instance, predicting human height based on age requires modeling a non-linear growth pattern: linear functions fail to capture the initial rapid growth and subsequent stabilization occurring in later years.

To address this limitation and allow for learning more flexible, non-linear functions, we introduce Reproducing Kernel Hilbert Spaces (RKHS). RKHS provide a rigorous mathematical framework to extend linear models to infinite-dimensional spaces while preserving the geometric structure provided by the inner product. This allows us to exploit linear techniques in a transformed feature space that captures non-linear relationships in the original input space.

Definition 1 (Reproducing Kernel Hilbert Space (RKHS))**.** *A Hilbert space H of functions $f : X \to \mathbb{R}$ is called a Reproducing Kernel Hilbert Space if:*

1. *H is a Hilbert space with an inner product $\langle \cdot, \cdot \rangle_H$ and associated norm $\| \cdot \|_H$.*
2. *For every $x \in X$, there exists a function $k_x \in H$ such that the reproducing property holds:*
$$f(x) = \langle f, k_x \rangle_H, \quad \forall f \in H.$$

The function $k : X \times X \to \mathbb{R}$ defined as

$$k(x, x') = \langle k_x, k_{x'} \rangle_H$$

is called the *reproducing kernel* of the RKHS. It is symmetric and positive semi-definite, meaning that for any finite set of points $\{x_1, \ldots, x_n\} \subset X$ and coefficients $a_1, \ldots, a_n \in \mathbb{R}$,

$$\sum_{i,j=1}^{n} a_i a_j k(x_i, x_j) = \Big\| \sum_{i=1}^{n} a_i k_{x_i} \Big\|_H^2 \geq 0.$$

The Moore-Aronszajn theorem ensures the reciprocal statement: for every symmetric positive semi-definite function k, there exists a unique RKHS H in which k is the reproducing kernel. This establishes a one-to-one correspondence between kernels and RKHSs, forming the foundation for kernel-based learning methods that leverage inner products in high-dimensional feature spaces without explicitly computing them.

2.2.1 Feature Maps and the Kernel Trick

A Reproducing Kernel Hilbert Space (RKHS) can equivalently be characterized via a *feature map* $\phi : X \to W$ into a (possibly infinite-dimensional) Hilbert space W, such that the reproducing kernel k can be expressed as

$$k(x, x') = \langle \phi(x), \phi(x') \rangle_W, \quad \forall x, x' \in X.$$

Intuitively, the feature map ϕ *embeds input points into a higher-dimensional space*, where linear relationships in W correspond to potentially highly non-linear relationships in the original input space X. This lifting allows classical linear methods, such as linear regression or classification, to model complex non-linear dependencies without explicitly constructing complicated non-linear transformations in the input space.

Illustrative Example: Consider approximating the function $y = 0.5x^2 - 1$ from data:

- Using standard linear functions $f(x) = \langle w, x \rangle$ in $\mathbb{R}$ is insufficient and leads to severe underfitting.
- Introducing a feature map $\phi(x) = [x^2, x, 1]^\top$ transforms the problem into a linear one in $\mathbb{R}^3$, allowing $f(x) = \langle w, \phi(x) \rangle$ to perfectly represent the quadratic relationship.

The kernel trick provides a computationally efficient mechanism to work with these high- or infinite-dimensional feature spaces. Instead of explicitly computing $\phi(x)$, one can directly evaluate the inner product $k(x, x') = \langle \phi(x), \phi(x') \rangle_W$. This is particularly crucial when W is infinite-dimensional, as it avoids explicitly handling infinitely many coordinates while still exploiting the expressive capacity of the space. Common kernels for $X \subseteq \mathbb{R}^d$ include:

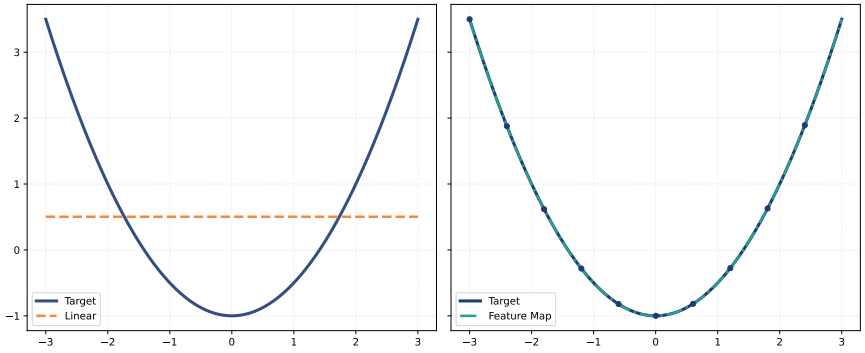


Figure 2.3: Illustrative Example.

- Linear kernel:

$$k(x, x') = x^\top x',$$

  which corresponds to the original linear function space.

- Polynomial kernel:

$$k(x, x') = (1 + x^\top x')^b, \quad b \in \mathbb{N},$$

  allowing the learning of polynomial relationships of degree b.

- Gaussian (RBF) kernel:

$$k(x, x') = \exp\big(-\frac{\gamma}{2}\|x - x'\|^2\big), \quad \gamma > 0,$$

  which induces an infinite-dimensional feature space with smooth, localized functions.

- Laplacian kernel:

$$k(x, x') = \exp\big(-\gamma\|x - x'\|\big), \quad \gamma > 0,$$

  which is also shift-invariant and emphasizes sparse interactions in the feature space.

By selecting an appropriate kernel, one can integrate prior knowledge about the data, such as smoothness, periodicity, or invariance properties, into the learning algorithm. The combination of feature maps and the kernel trick forms the cornerstone of many modern machine learning methods, enabling efficient and flexible learning in complex, high-dimensional settings. A concrete and widely used instance of this kernel based framework is Kernel Ridge Regression.

2.3 Kernel Ridge Regression (KRR)

Kernel Ridge Regression (KRR) is a fundamental supervised learning method that leverages the theory of Reproducing Kernel Hilbert Spaces (RKHS) to learn non-linear functions while remaining computationally manageable. Specifically, KRR combines three key components:

1. A hypothesis space H given by a RKHS with reproducing kernel k.
2. The *squared loss* function, measuring the discrepancy between predicted and observed outputs.
3. *Tikhonov regularization* to control model complexity and prevent overfitting.

The resulting optimization problem is

$$\min_{f \in H} \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \|f\|_H^2, \tag{2.13}$$

where $\lambda > 0$ is the regularization parameter controlling the trade-off between data fidelity and smoothness of the solution.

2.3.1 Representer Theorem

A crucial principle that enables practical computation in potentially infinite-dimensional RKHS is the Representer Theorem [Kimeldorf and Wahba, 1971;

Schölkopf et al., 2001], which states:

Theorem 1 (Representer Theorem)**.** *For any strictly monotone increasing regularizer $\Omega(\|f\|_H)$ and arbitrary loss function $L(f(x_1), \ldots, f(x_n))$, the minimizer $f^* \in H$ of the regularized risk can be expressed as a finite linear combination of kernel evaluations on the training points:*

$$f^*(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x), \quad \alpha_i \in \mathbb{R}.$$

This theorem reduces the infinite-dimensional optimization over H to a finite-dimensional problem over the coefficients $\alpha = (\alpha_1, \ldots, \alpha_n)^\top \in \mathbb{R}^n$. The RKHS norm can be expressed as

$$\|f\|_H^2 = \sum_{i,j=1}^{n} \alpha_i \alpha_j k(x_i, x_j) = \alpha^\top K \alpha,$$

where $K \in \mathbb{R}^{n \times n}$ is the kernel matrix with entries $K_{ij} = k(x_i, x_j)$. Similarly, the predictions on the training set are

$$f(x_i) = (K\alpha)_i.$$

Substituting the Representer Theorem into (2.13), the KRR problem reduces to a standard convex quadratic problem:

$$\hat{\alpha}_\lambda = \arg\min_{\alpha \in \mathbb{R}^n} \frac{1}{n} \|K\alpha - y\|^2 + \lambda \alpha^\top K \alpha, \tag{2.14}$$

with the closed-form solution:

$$\hat{\alpha}_\lambda = (K + n\lambda I)^{-1} y. \tag{2.15}$$

Thus, KRR allows learning a function in an infinite-dimensional RKHS without ever explicitly computing the feature map $\phi(x)$; all computations are expressed in terms of kernel evaluations $k(x_i, x_j)$. This is the essence of the *kernel trick*, which makes KRR both flexible and computationally efficient.

2.3.2 Statistical Properties of KRR

The statistical performance of Kernel Ridge Regression can be rigorously analyzed under a set of standard assumptions:

Assumption 1 (Bounded Kernel)**.** *There exists a constant $\kappa > 0$ such that*

$$k(x, x) \leq \kappa^2, \quad \forall x \in X.$$

Assumption 2 (Existence of Optimal Function in RKHS)**.** *There exists $f_H \in H$ such that it minimizes the expected risk*

$$f_H = \arg\min_{f \in H} \mathbb{E}[(f(X) - Y)^2].$$

Assumption 3 (Output Regularity)**.** *The outputs y are either bounded or satisfy a sub-Gaussian condition, ensuring that higher moments are controlled.*

Under these assumptions [Caponnetto and De Vito, 2007], the excess risk of the KRR estimator $\hat{f}_\lambda$ can be bounded with high probability (at least $1 - \delta$) by

$$R_H(\hat{f}_\lambda) := \mathbb{E}[(\hat{f}_\lambda(X) - Y)^2] - \mathbb{E}[(f_H(X) - Y)^2] \lesssim \lambda + \frac{1}{n} \log \frac{1}{\delta}. \tag{2.16}$$

This inequality emphasizes the classical *bias-variance trade-off*:

- A larger regularization parameter λ increases bias but reduces variance.
- A smaller λ decreases bias but may amplify variance due to overfitting.

Selecting λ appropriately as a function of the sample size n and the desired *certainty level* δ ensures minimization of the excess risk.

2.3.3 Computational Aspects of KRR

From a computational perspective, KRR involves two main steps: computing the kernel matrix and solving the resulting linear system.

- **Kernel matrix computation:** For n data points in d dimensions, computing $K_{ij} = k(x_i, x_j)$ requires $O(dn^2)$ operations and $O(n^2)$ memory.

- **Linear system solution:** Solving $(K + n\lambda I)\alpha = y$ using direct methods (e.g., Cholesky or LU decomposition) requires $O(n^3)$ operations.
- **Iterative methods:** For large datasets, conjugate gradient or other iterative solvers provide efficient alternatives that avoid storing or inverting the full kernel matrix.

Regularization and Stability

Tikhonov regularization not only controls overfitting but also improves numerical stability by reducing the condition number of the system:

$$\kappa(K + n\lambda I) \leq \kappa(K),$$

where $\kappa(\cdot)$ denotes the ratio of the largest to smallest eigenvalue. From a *spectral perspective*, the regularization term effectively *filters out* the influence of small eigenvalues, which correspond to directions in the feature space that are poorly constrained by the data. This enhances robustness to noise and ensures that the solution is well-conditioned even when the kernel matrix is nearly singular. Despite these theoretical advantages, classical Kernel Ridge Regression becomes computationally impractical for large datasets. This limitation motivates the development of approximation methods.

2.4 Approximation Methods for KRR

Classical KRR becomes computationally impractical for large datasets because it requires inverting an $n \times n$ matrix (n being the number of data points), which has a time complexity of $\mathcal{O}(n^3)$ and memory complexity of $\mathcal{O}(n^2)$. To address these challenges, several approximation approaches have been developed to reduce the dimensionality of the problem or approximate the kernel matrix while maintaining good predictive accuracy. The three most commonly used methods are Principal

Component Analysis (PCA) [Sterge et al., 2019], Random Fourier Features (RFF) [Rudi and Rosasco, 2017], and the Nyström method.

2.4.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a classical method for reducing the dimensionality of the data or kernel matrix by identifying the directions of maximum variance. In the context of kernel ridge regression, PCA can be applied to the data matrix $X_b \in \mathbb{R}^{n\times d}$ (for a linear kernel) or to the kernel matrix $K \in \mathbb{R}^{n\times n}$ (for a general kernel) to approximate the key subspace of the RKHS in which the solution lies [Sterge et al., 2019]. The core idea is to perform a singular value decomposition (SVD) of the data or kernel matrix:

$$K = U\Sigma^2 U^\top, \quad X_b = U\Sigma V^\top,$$

where $U \in \mathbb{R}^{n\times n}$ and $V \in \mathbb{R}^{d\times d}$ are orthogonal matrices containing the left and right singular vectors, and $\Sigma \in \mathbb{R}^{n\times d}$ contains the singular values. The directions associated with the largest singular values capture the most significant variance in the data. To reduce dimensionality, only the top $m \ll \min(n, d)$ singular values and corresponding vectors are kept. The data are then projected into this lower-dimensional space:

$$x \in \mathbb{R}^d \mapsto \tilde{x} = S^\top x \in \mathbb{R}^m, \quad S = V_m = X_b^\top U_m \Sigma_m^{-1},$$

where U_m and Σ_m contain the first m columns of U and the largest m singular values, respectively. This projection preserves the directions with the most variance and discards components contributing less to the overall data variability. Once the data are projected, kernel ridge regression reduces to a standard linear ridge regression problem in the m-dimensional space:

$$c_m = (X_{b,m}^\top X_{b,m} + \lambda n I)^{-1} X_{b,m}^\top Y_b, \quad f_{\lambda,m}(x) = c_m^\top \tilde{x}, \quad X_{b,m} = X_b S.$$

This approach provides several benefits. By focusing on the principal directions, PCA can reduce the computational cost from $\mathcal{O}(n^3)$ for the full KRR solution to $\mathcal{O}(nm^2)$, and the memory requirement from $\mathcal{O}(n^2)$ to $\mathcal{O}(nm)$. It also ensures that the main structure in the data is captured accurately within the reduced subspace.

However, there are limitations. Computing the full SVD of X_b or K remains expensive for very large datasets, making PCA less practical when n or d is very large. Additionally, PCA is inherently linear: it can only capture linear relationships between features. Nonlinear dependencies, which could be modeled by a nonlinear kernel in KRR, are not represented in the PCA subspace. As a result, PCA provides a good approximation primarily when the data exhibit mostly linear structure or when a linear kernel can capture the key patterns.

2.4.2 Random Fourier Features (RFF)

Random Fourier Features (RFF) provide a practical way to approximate kernels that depend only on the difference between inputs, such as Gaussian or Laplacian kernels [Rudi and Rosasco, 2017]. Recall that for a reproducing kernel Hilbert space (RKHS) with kernel K, there exists a feature map $\phi : \mathcal{X} \to \mathcal{H}$ such that

$$k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}.$$

For shift-invariant kernels, this inner product can be approximated using a randomized finite-dimensional feature map $z : \mathcal{X} \to \mathbb{R}^R$:

$$\langle \phi(x), \phi(x') \rangle_{\mathcal{H}} \approx z(x)^\top z(x').$$

Specifically, using Bochner's theorem, we define

$$z(x) = \frac{1}{\sqrt{R}} \begin{bmatrix} \sqrt{2}\cos(\omega_1^\top x + b_1) \\ \vdots \\ \sqrt{2}\cos(\omega_R^\top x + b_R) \end{bmatrix},$$

where ω_i are sampled according to the Fourier transform of the kernel and $b_i \sim \mathcal{U}[0, 2\pi]$. Using this approximation, the KRR solution, which in the RKHS is

$$f(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x) = \sum_{i=1}^{n} \alpha_i \langle \phi(x_i), \phi(x) \rangle_{\mathcal{H}},$$

can now be expressed as a finite-dimensional linear model:

$$f(x) \approx \sum_{i=1}^{n} \alpha_i z(x_i)^\top z(x) = \beta^\top z(x), \quad \text{with } \beta = \sum_{i=1}^{n} \alpha_i z(x_i).$$

The KRR problem then reduces to standard ridge regression in $\mathbb{R}^R$:

$$\beta = (Z^\top Z + n\lambda I)^{-1} Z^\top \hat{y}, \quad Z = \begin{bmatrix} z(x_1)^\top \\ \vdots \\ z(x_n)^\top \end{bmatrix}.$$

This approximation dramatically reduces computational cost when $R \ll n$, replacing the inversion of an $n \times n$ matrix with an $R \times R$ matrix. The quality of the approximation depends on the number of random features R: small R may reduce accuracy, while large R reduces computational advantage. Additionally, since the random features are chosen independently of the data, RFF may not capture certain dataset-specific structures. Nonetheless, it allows kernel methods to scale efficiently to large datasets while preserving the approximate RKHS structure.

2.4.3 Nyström Method

The Nyström method is a data-dependent approximation technique designed to reduce the computational cost of kernel ridge regression (KRR) Rudi et al. [2015]. By the representer theorem, the empirical risk minimization (ERM) solution of KRR can be written as

$$f(x) = \sum_{i=1}^{n} c_i K(x_i, x),$$

where the coefficients $c \in \mathbb{R}^n$ are obtained by solving the linear system

$$(K + \lambda n I)c = y.$$

This solution belongs to the finite-dimensional subspace

$$\mathcal{H}_n = \text{span}\{\phi(x_1), \ldots, \phi(x_n)\} \subset \mathcal{H},$$

and requires the full kernel matrix $K \in \mathbb{R}^{n \times n}$, leading to a computational cost of $\mathcal{O}(n^3)$ and a memory requirement of $\mathcal{O}(n^2)$. The Nyström method reduces this cost by restricting the hypothesis space to a smaller subspace generated by $m \ll n$ selected points, called Nyström centers or inducing points,

$$\{\tilde{x}_j\}_{j=1}^{m} \subset \{x_1, \ldots, x_n\}.$$

The corresponding approximation space is

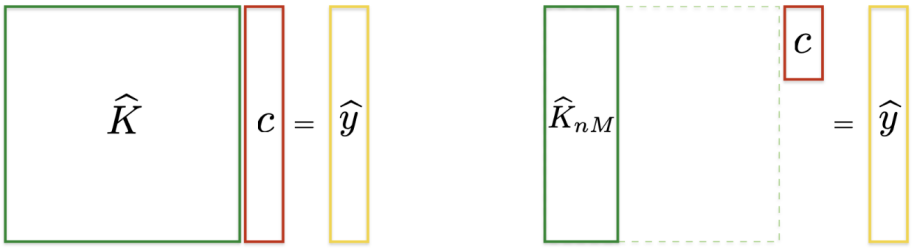
$$\mathcal{H}_m = \text{span}\{\phi(\tilde{x}_1), \ldots, \phi(\tilde{x}_m)\} \subset \mathcal{H}_n.$$

Restricting the solution to $\mathcal{H}_m$ gives an approximate predictor of the form

$$f(x) = \sum_{j=1}^{m} c_j K(\tilde{x}_j, x),$$

which involves only m kernel evaluations instead of n. From an RKHS perspective, the approximation error comes from projecting the exact KRR solution in $\mathcal{H}_n$

onto the smaller subspace $\mathcal{H}_m$.



The Nyström-regularized KRR problem is defined as

$$f_{\lambda,m} = \arg\min_{f \in \mathcal{H}_m} \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}}^2.$$

Writing $f_{\lambda,m}$ in terms of the Nyström centers gives

$$f_{\lambda,m}(x) = \sum_{j=1}^{m} c_j K(\tilde{x}_j, x), \quad \|f_{\lambda,m}\|_{\mathcal{H}}^2 = c^\top K_{mm} c,$$

where $K_{mm} \in \mathbb{R}^{m \times m}$ is the kernel matrix among the centers, and $K_{nm} \in \mathbb{R}^{n \times m}$ is the cross-kernel matrix between training points and centers. The optimization problem reduces to

$$c = \arg\min_{c \in \mathbb{R}^m} \frac{1}{n} \|K_{nm} c - y\|_2^2 + \lambda c^\top K_{mm} c,$$

with the explicit solution

$$c = (K_{nm}^\top K_{nm} + \lambda n K_{mm})^\dagger K_{nm}^\top y.$$

Another way to view the Nyström method is as an approximation of the full kernel matrix by a low-rank factorization

$$K \approx K_{nm} K_{mm}^{-1} K_{nm}^\top.$$

This replaces the inversion of an $n \times n$ matrix with operations on matrices of size at most $m \times m$, giving a computational cost of $\mathcal{O}(nm^2 + m^3)$ and a memory

cost of $\mathcal{O}(nm)$. The quality of the Nyström approximation depends on both the number of centers m and how they are chosen. While uniform random selection is commonly used and often effective, more advanced strategies such as leverage score sampling or determinantal point process sampling can better capture the structure of the RKHS. Importantly, the Nyström method preserves the RKHS structure of KRR: the resulting predictor is still a linear combination of kernel evaluations, ensuring that regularization properties and theoretical guarantees remain valid.

Chapter 3

KRR under Covariate Shift

3.1 Introduction

Classical supervised learning assumes that the training and test data distributions are identical [Vapnik, 1999]. However, in practice, it is often the case that there is a significant mismatch between the two [Quiñonero-Candela et al., 2022]. This discrepancy can arise from various factors, such as inconsistencies in measurement equipment, differences in data collection domains, or variations in subject populations [Koh et al., 2021]. Among these scenarios, one particularly interesting and common case is known as covariate shift [Sugiyama and Kawanabe, 2012]. Covariate shift occurs when the marginal distributions of the input covariates differ between the training and test data, while the conditional distribution of the output label given the input covariates remains unchanged [Cortes et al., 2008; Shimodaira, 2000]. This phenomenon is observed in a variety of well-studied learning problems, including in domain adaptation settings [Ben-David et al., 2006; Cortes and Mohri, 2014; Mansour et al., 2009; Zhang et al., 2012], active learning [Kanamori and Shimodaira, 2003; Sugiyama and Ridgeway, 2006; Wiens, 2000], natural language processing [Jiang and Zhai, 2007], and medical image analysis [Guan and Liu, 2021].

Despite its practical significance, covariate shift remains relatively under-

explored in theoretical frameworks compared to the classical setting where no distribution mismatch is assumed. Recently, several studies have attempted to bridge this gap. A widely adopted approach to addressing covariate shift involves correcting the learning objective by reweighting the loss function using the so-called importance weighting (IW) function. This function corresponds to the Radon-Nikodym derivative of the test marginal distribution with respect to the training marginal distribution [Fang et al., 2020; Huang et al., 2006; Sugiyama et al., 2012]. Shimodaira [2000] was the first to demonstrate the consistency of the importance-weighted maximum likelihood estimator, while in Cortes et al. [2010a] the authors derived suboptimal finite-sample bounds for restricted function classes with finite pseudodimension. In the context of nonparametric regression in reproducing kernel Hilbert spaces (RKHSs) and under the assumption that the regression function belongs to the RKHS (well-specified case), Gizewski et al. [2022] provided optimal excess risk convergence results for the minimizer of the reweighted empirical risk when the weight function is known and uniformly bounded. In Ma et al. [2023], authors recently showed that the standard unweighted kernel ridge regression estimator is still minimax optimal under an appropriate choice of the regularization parameter, provided the IW function is uniformly bounded or its second moment is bounded. Similar results for nonparametric classification were obtained in Kpotufe and Martinet [2021]. We also mention Pathak et al. [2022]; Schmidt-Hieber and Zamolodtchikov [2024] in the context of nonparametric regression for classes of functions different from RKHSs, and Lei et al. [2021]; Wen et al. [2014]; Yamazaki et al. [2007] for parametric models. Building on Ma et al. [2023], in Gogolashvili et al. [2023] the authors extended the analysis to (simplified) misspecified case, where the regression function is not assumed to lie in the RKHS itself, but its projection is. In this setting, the authors showed that, under covariate shift, the unweighted classic KRR predictor is not a consistent

estimator of the projection of the regression function. In such cases, IW correction is necessary.

Kernel methods provide a robust framework for nonparametric learning, but their scalability is limited by high computational and memory costs—challenges that clearly do not disappear under covariate shift. To address this, researchers have developed more efficient strategies, ranging from improved optimization [Johnson and Zhang, 2013; Schmidt et al., 2017] to randomized linear algebra techniques [Calandriello et al., 2017; Drineas and Mahoney, 2005b; Mahoney, 2011; Woodruff, 2014]. These aim to reduce costs, raising a key question: do such shortcuts compromise statistical accuracy? Recent work suggests they often do not [Bach, 2017; Bottou and Bousquet, 2008; Della Vecchia et al., 2021; Rudi and Rosasco, 2017; Rudi et al., 2015; Sun et al., 2018]. A promising approach is to restrict the hypothesis space to a lower-dimensional (random) subspace, as in sketching [Kpotufe and Sriperumbudur, 2019] and random projection methods [Woodruff, 2014], including Nyström methods for kernels [Smola and Schökopf, 2000; Williams and Seeger, 2000]. Recent results confirm that these methods can achieve both computational efficiency and statistical accuracy, for both smooth and general convex losses [Bach, 2013; Della Vecchia et al., 2024; Marteau-Ferey et al., 2019b; Rudi et al., 2015].

Although random projection techniques, such as the Nyström method have been widely studied in standard learning settings, their use under covariate shift remains largely unexamined. A recent step in this direction was taken by Myleiko and Solodky [2024], building on the framework of Gizewski et al. [2022]. The authors consider Nyström subsampling in a setting where weights are known and uniformly bounded, and derive optimal risk bounds that, in the case of Tikhonov regularization, can be recovered as a special case of our results.

In this chapter, we study importance-weighted kernel ridge regression (KRR)

under the covariate shift setting. Previous work in this area typically focuses either on optimal statistical guarantees without addressing computational complexity [Gogolashvili et al., 2023], or on Nyström-type approximations in the i.i.d. setting without distribution shift [Rudi et al., 2015]. In contrast, we combine covariate shift, and Nyström-based low-rank approximation to obtain scalable algorithms [Della Vecchia et al., 2025]. Our main contribution is to show that an ALS-based Nyström projection preserves the optimal learning rates of importance-weighted KRR in the regime of bounded importance weights. This result is significant because it establishes that optimal statistical guarantees are maintained despite both (i) distribution mismatch between training and test data and (ii) the approximation error induced by the Nyström projection. In particular, it clarifies what is new relative to prior work: while existing results on covariate shift in RKHSs derive optimal rates for exact importance-weighted KRR, and classical Nyström or random subspace methods achieve efficient kernel approximation only in the standard i.i.d. setting, we show that these two lines of work can be unified without loss of statistical optimality. More precisely, the novelty lies in proving that ALS-based Nyström approximation preserves the full weighted-KRR learning rate under bounded importance weights, something not established in previous work combining kernel approximation and covariate shift. We further extend the analysis to unbounded importance weights under stronger moment assumptions, showing that comparable learning rates can still be achieved in this more general setting. From a computational perspective, our method achieves state-of-the-art efficiency by significantly reducing the cost of solving the kernel system while retaining optimal statistical performance. The theoretical analysis requires new arguments to handle the interaction between covariate shift, importance weighting, and Nyström approximation, going beyond standard random projection techniques. Full technical details are provided in Appendix A.2. We validate our theoretical

findings through experiments on synthetic and real-world datasets.

As remarked in Section 3.4.1, Assumption 6 requires boundedness of all moments of the weight function. While stronger than the second-moment assumption in Ma et al. [2023], it is needed here to control the ALS-based Nyström approximation in the unbounded regime. Assumption 3.6 ensures that the regression function projected onto $\mathcal{H}$ is well-defined and lies in $\mathcal{H}$, rather than only on its boundary, which is slightly more general than the standard well-specified setting but conceptually aligned. We do not treat the fully misspecified case, which would require tools such as interpolation spaces (see Steinwart and Christmann [2008]). Finally, as discussed in Section 3.7, a more refined analysis of constants in the learning bounds could help explain why distributional misalignment is sometimes benign and sometimes highly detrimental in practice, depending on its interaction with the source condition. These directions are left for future work.

3.2 Background

We start defining some key quantities we will need in the rest of this part, see e.g. Caponnetto and De Vito [2007]; Smale and Zhou [2007]. Given a measurable space $\mathcal{X}$, a probability distribution μ on $\mathcal{X}$, a space of squared-integrable functions L^2_μ with respect to measure μ and an RKHS $\mathcal{H}$ of (bounded) kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, with $K_x(\cdot) = K(x, \cdot) \in \mathcal{H}$, define

$$S_\mu : \mathcal{H} \to L^2_\mu, \quad (S_\mu f)(x) = \langle f, K_x \rangle_{\mathcal{H}} = f(x) \quad \mu\text{−}a.s. \quad \text{and} \quad S^*_\mu g = \int K_x g(x) d\mu.$$

for $f \in \mathcal{H}$, $g \in L^2_\mu$. Covariance operator $\Sigma_\mu : \mathcal{H} \to \mathcal{H}$ is defined as $\Sigma_\mu := S^*_\mu S_\mu = \mathbb{E}_\mu[K_x \otimes K_x]$.

Define the sampling operator $\widehat{S} : \mathcal{H} \to \mathbb{R}^n$ associated with set $\{x_1, \ldots, x_n\} \in \mathcal{X}^n$,

for $f \in \mathcal{H}$, as

$$(\widehat{S}f)_i := f(x_i) = \langle f, K_{x_i} \rangle_{\mathcal{H}}, \quad i \in [n], \quad \text{and} \quad \widehat{S}^*(\widehat{y}) := \frac{1}{n} \sum_{i=1}^{n} y_i K_{x_i}, \quad \widehat{y} \in \mathbb{R}^n.$$

3.3 Learning under Covariate Shift

The covariate shift setting introduces an additional complexity: training and test distributions may differ, but only through their marginals, while sharing the same conditional distribution:

$$\rho^{te}(x, y) = \rho(y|x)\rho_X^{te}(x), \qquad \rho^{tr}(x, y) = \rho(y|x)\rho_X^{tr}(x).$$

Since the regression function g^* only depends on the conditional distribution, which is identical for both ρ^{te} and ρ^{tr}, it is unique. As in the standard setting, we are provided with n input-output pairs $(x_i, y_i)_{i=1}^n \in (\mathcal{X} \times \mathbb{R})$ sampled i.i.d. from ρ^{tr}, i.e. $(x_1, y_1), \ldots, (x_n, y_n) \sim \rho^{tr}(x, y)$.

The challenge consists in the fact that we train our model using samples from the ρ^{tr}, but we aim to evaluate its performance on new data drawn from ρ^{te}. Then, we want to upper bound the excess risk

$$\mathcal{E}(\widehat{f}_\lambda) = \|\widehat{f}_\lambda - g^*\|^2_{\rho_X^{te}}. \tag{3.1}$$

Note that the empirical risk computed from ρ^{tr} samples is a biased estimate of the expected risk under ρ^{te}. As a result, minimizing it may not yield a predictor that performs well on the test distribution.

3.3.1 Importance-Weighted Risk Minimization

The goal of importance weighting correction is to construct an unbiased estimator of the risk with respect to the test distribution ρ^{te} while using data sampled from the training distribution ρ^{tr}. The idea is to reweight empirical risk minimi-

sation samples based on their relevance to the test distribution, ensuring good performance under ρ^{te}.

Under the assumption that $\rho_X^{te} \ll \rho_X^{tr}$, the test distribution is absolutely continuous with respect to the training distribution. In other words, any region of the input space $\mathcal{X}$ that has never been observed in the training set will also not appear in the test set. This assumption ensures that the importance weighting function w, which assigns a weight to each point $x \in \mathcal{X}$, is well defined and given by the Radon-Nikodym derivative of ρ_X^{te} with respect to ρ_X^{tr}:

$$w(x) := \frac{d\rho_X^{te}}{d\rho_X^{tr}}(x). \tag{3.2}$$

Points that are likely to be encountered during testing (ρ_X^{te} is large) but are rare at training time (ρ_X^{tr} is small) receive higher weights. This reweighting corrects for the discrepancy between the two distributions and reflects the relative importance of samples at test time.

In practice, this assumption is reasonable when the test distribution is similar to the training distribution and when all regions likely to be present in the test data are well represented in the training data. However, it does not cover situations where certain parts of $\mathcal{X}$ are absent from the training data or where the differences between ρ_X^{te} and ρ_X^{tr} are significant. In such cases, importance weights can become highly extreme, leading to unstable estimators and degraded predictive performance. Therefore, the assumption $\rho_X^{te} \ll \rho_X^{tr}$ explicitly excludes scenarios in which the test distribution introduces new regions or behaviours that are not represented in the training distribution.

In the rest of this chapter, we will focus on applying importance weighting, based on the Radon–Nikodym derivative, in the context of kernel methods.

Assumption 4. *$\mathcal{H}$ is an RKHS with scalar product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and associated kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.*

We define the regularized importance-weighted empirical risk, for all $f \in \mathcal{H}$, as

$$\widehat{\mathcal{R}}^w_\lambda(f) := \frac{1}{n}\sum_{i=1}^n w(x_i)(y_i - \langle K_{x_i}, f\rangle)^2 + \lambda\|f\|^2_{\mathcal{H}}, \tag{3.3}$$

where $\mathcal{H} \ni K_x(\cdot) = K(x, \cdot)$. For the square loss, the minimizer $\widehat{f}^w_\lambda$ of eq. (3.3) is given by:

$$\widehat{f}^w_\lambda(x) = (\widehat{S}^*\widehat{M}_w\widehat{S} + \lambda \mathrm{I})^{-1}\widehat{S}^*\widehat{M}_w\widehat{y}, \tag{3.4}$$

where $\widehat{M}_w$ is the diagonal matrix with i-th entry $w(x_i)$. In case weights are all positive, we have

$$\widehat{f}^w_\lambda(x) = \sum_{i=1}^n c^w_i K_{x_i}(x) \quad \in \mathrm{span}\{K_{x_1}, \dots, K_{x_n}\}, \quad c^w = (\widehat{K} + n\lambda\widehat{M}_{1/w})^{-1}\widehat{y} \quad \in \mathbb{R}^n, \tag{3.5}$$

where $\widehat{K}$ is the kernel Gram matrix, and $\widehat{M}_{1/w}$ is the diagonal matrix with i-th entry $1/w(x_i)$.

Since the regression function g^* may not generally belong to $\mathcal{H}$ (i.e., the model may be misspecified), we introduce the best approximation $f_{\mathcal{H}} \in \mathcal{H}$ of g^* with respect to the $L^2_{\rho^{te}_X}$ distance.

Assumption 5. *There exists an $f_{\mathcal{H}} \in \mathcal{H}$ such that*

$$\mathcal{E}(f_{\mathcal{H}}) = \min_{f\in\mathcal{H}} \mathcal{E}(f) = \min_{f\in\mathcal{H}} \|f - g^*\|^2_{\rho^{te}_X}. \tag{3.6}$$

Note that, while the minimizer might not be unique, we select $f_{\mathcal{H}}$ as the unique minimizer with minimal norm [De Vito et al., 2021]. In the following, we will evaluate the performance of our estimator relative to the best estimator in $\mathcal{H}$, i.e. $f_{\mathcal{H}}$.

Computations A significant limitation of the procedure outlined above is the computational cost associated with the $n \times n$ matrix inversion required to compute

the estimator, see (3.5). This operation has a complexity of $\mathcal{O}(n^3)$ in time and $\mathcal{O}(n^2)$ in memory, making it impractical when $n > 10^5$.

3.4 ERM on Random Subspaces and the Nyström Method

We consider an efficient approximation of the above procedure based on considering a subspace $\mathcal{B} \subset \mathcal{H}$ and solving the corresponding importance-weighted regularized ERM problem

$$\min_{\beta \in \mathcal{B}} \widehat{\mathcal{R}}^w_\lambda(\beta), \tag{3.7}$$

with $\widehat{\beta}^w_\lambda$ the unique minimizer. As clear from (3.5), choosing $\mathcal{B} = \mathcal{H}_n = \operatorname{span}\{K_{x_1}, \dots, K_{x_n}\}$ is equivalent to considering the full space $\mathcal{H}$ and yields the same solution as in (3.4). However, a natural alternative is to consider a smaller subspace:

$$\mathcal{B} = \mathcal{H}_m = \operatorname{span}\{K_{\widetilde{x}_1}, \dots, K_{\widetilde{x}_m}\}, \tag{3.8}$$

where $\{\widetilde{x}_1, \dots, \widetilde{x}_m\} \subset \{x_1, \dots, x_n\}$ is a random subset of the input points and $m \leq n$. This is equivalent to Nyström approximation [Williams and Seeger, 2000]. A basic approach is to select these points uniformly at random from the training dataset. Alternatively, we can employ more refined sampling techniques, such as using leverage scores [Drineas et al., 2012]

$$l_i(\alpha) = (\widehat{K}(\widehat{K} + \alpha n \mathrm{I})^{-1})_{ii}, \qquad i = 1, \dots, n. \tag{3.9}$$

Since in practice computing leverage scores directly can be computationally expensive, approximations $(\hat{l}_i(\alpha))_{i=1}^n$ have been considered [Alaoui and Mahoney, 2015; Cohen et al., 2015; Drineas et al., 2012]. In particular, we consider the following one.

Definition 2 (T-approximate leverage scores)**.** *Let $(l_i(\alpha))_{i=1}^n$ be the leverage scores associated to the training set for a given α. Let $\delta > 0$, $t_0 > 0$ and $T \geqslant 1$. We*

say that $(\hat{l}_i(\alpha))_{i=1}^n$ *are* T*-approximate leverage scores with confidence* δ*, when with probability at least* $1-\delta$*,*

$$\frac{1}{T} l_i(\alpha) \leqslant \hat{l}_i(\alpha) \leqslant T l_i(\alpha), \qquad \forall i \in \{1, \dots, n\}, \;\; \alpha \geqslant t_0. \tag{3.10}$$

Given the T-approximate leverage scores for $\alpha \geqslant t_0$, ALS sampling proceeds by independently drawing samples $\tilde{x}_1, \dots, \tilde{x}_m$ from the training set with replacement, where each point x_i is selected with probability $Q_\alpha(i) = \hat{l}_i(\alpha) / \sum_j \hat{l}_j(\alpha)$. All the results in the next sections are obtained under ALS sampling.

We can now define the Nyström W-KRR problem as follows:

$$\widehat{f}^w_{\lambda,m} := \arg\min_{\beta \in \mathcal{H}_m} \frac{1}{n} \left\| \widehat{M}_w^{1/2} (\widehat{y} - \widehat{S}\beta) \right\|_2^2 + \lambda \|\beta\|_{\mathcal{H}}^2 \tag{3.11}$$

$$= \arg\min_{f \in \mathcal{H}} \frac{1}{n} \left\| \widehat{M}_w^{1/2} (\widehat{y} - \widehat{S} P_m f) \right\|_2^2 + \lambda \|f\|_{\mathcal{H}}^2 , \tag{3.12}$$

where P_m is the orthogonal projection operator onto $\mathcal{H}_m$, given by $P_m = VV^*$, see Appendix A.1. Taking the derivative and using the first order condition (see Appendix A.1), the Nyström estimator can be expressed as:

$$\widehat{f}^w_{\lambda,m} = V (V^* \widehat{S}^* \widehat{M}_w \widehat{S} V + \lambda \mathrm{I})^{-1} V^* \widehat{S}^* \widehat{M}_w \widehat{y}. \tag{3.13}$$

Alternatively, using some linear algebra, it can also be written as:

$$\widehat{f}^w_\lambda(x) = \sum_{i=1}^m \widetilde{c}^w_i K_{\widetilde{x}_i}(x), \qquad \widetilde{c}^w = (\widehat{K}_{nm}^T \widehat{M}_w \widehat{K}_{nm} + n\lambda \widehat{K}_{mm})^{-1} \widehat{K}_{nm}^T \widehat{M}_w \widehat{y}, \tag{3.14}$$

where $\widehat{f}^w_\lambda(x) \in \mathrm{span}\{K_{\widetilde{x}_1}, \dots, K_{\widetilde{x}_m}\}$, $\widetilde{c}^w \in \mathbb{R}^m$, $\widehat{K}_{nm} \in \mathbb{R}^{n \times m}$, $(\widehat{K}_{nm})_{ij} = K(x_i, \widetilde{x}_j)$ and $\widehat{K}_{mm} \in \mathbb{R}^{m \times m}$, $(\widehat{K}_{mm})_{ij} = K(\widetilde{x}_i, \widetilde{x}_j)$ (see derivation in Appendix A.1).

Computations From eq. (3.14), it is clear that the Nyström method can offer significant computational benefits. Unlike in eq. (3.13), computing our projected estimator only requires $\mathcal{O}(m^3 + m^2 n)$ time and $\mathcal{O}(mn)$ memory, compared to the previous $\mathcal{O}(n^3)$ and $\mathcal{O}(n^2)$. When $m \ll n$ the difference between the two can be

big and efficient implementations such as in Meanti et al. [2020]; Rudi et al. [2017] can drastically reduce computational requirements.

3.4.1 Further Assumptions

In this section, we aim to derive excess risk bounds for the Nyström estimator presented in eq. (3.13). We begin by introducing the technical assumptions required for the subsequent analysis.

The following assumption, inspired by Gogolashvili et al. [2023], ensures the boundedness of the importance-weighting (IW) function or of its moments.

Assumption 6. *Let $w = d\rho_X^{te}/d\rho_X^{tr}$ be the IW function. There exist constants $q \in [0,1], W > 0$ and $\sigma > 0$ such that $\forall p \in \mathbb{N},\ \ p \geqslant 2$*

$$\left(\int_{\mathcal{X}} w(x)^{\frac{p-1}{q}} d\rho_X^{te}\right)^q \leqslant \frac{1}{2} p! W^{p-2} \sigma^2, \tag{3.15}$$

where the left-hand side for $q = 0$ is defined as $\|w^{p-1}\|_{\infty,\rho_X^{te}}$, the ess sup *with respect to ρ_X^{te}.*

Considering the uniformly bounded case $\|w\|_\infty < \infty$, Assumption 6 is satisfied for $q = 0$. When w is not uniformly bounded, Assumption 6 can still be satisfied for $q \in (0,1]$ if the moments of w are bounded. For example, it is satisfied for $q \in (0,1]$ if $W \geqslant 1, \sigma^2 \geqslant 1$ and

$$2\rho_X^{te}\left(\left\{x \in X : \frac{d\rho_X^{te}}{d\rho_X^{tr}}(x) \geqslant t\right\}\right) \leqslant \sigma^2 \exp\left(-W^{-1} t^{1/q}\right) \qquad \text{for all } t > 0$$

(see Appendix A in Gogolashvili et al. [2023] for the detailed result). Equivalently, Assumption 6 can be stated as a condition on the Rényi divergence between ρ_X^{te} and ρ_X^{tr} [Cortes et al., 2010a; Mansour et al., 2009]. It follows that this assumption can be interpreted as a requirement for the test distribution ρ_X^{te} not to deviate too strongly from the training distribution ρ_X^{tr}, with the parameter $q \in [0,1]$ quantifying the severity of the deviation. Smaller q allows heavier tails (more

extreme weights), while larger q imposes faster moment decay. Introducing q is useful for presenting results such as Theorem 2 in a unified form, which can then be specialized to the two cases considered in Corollary 1: uniformly bounded weights ($q = 0$) and possibly unbounded weights ($q \neq 0$). Note that in Ma et al. [2023], a weaker assumption requiring only the second moment to be bounded is considered. The stronger condition in eq. (3.15) is crucial in our case to apply the Nyström method in the unbounded setting (see Appendix A.2 and how the proof differs from Rudi et al. [2015]).

This assumption provides a formal way to control the effect of very large importance weights in the theoretical analysis. It allows risk bounds and Nyström approximations to hold even when weights are unbounded, by ensuring that extreme values do not dominate the empirical processes. The parameter q offers a unified framework to handle both bounded and unbounded cases, and the connection to Rényi divergence gives a quantitative measure of the deviation between training and test distributions that is relevant for the analysis.

We further assume the output variable to be bounded.

Assumption 7. *The range of the output $Y \in \mathbb{R}$ is upper bounded, i.e. $Y \in [-B, B]$, $B < \infty$.*

Furthermore, we make the following regularity assumption, usually known as *source condition*.

Assumption 8 (Source condition)**.** *There exist $1/2 \leqslant r \leqslant 1$ and $g \in L^2(X, \rho_X^{te})$ with $\|g\|_{\rho^{te}} \leqslant R$ for some $R > 0$ such that $f_{\mathcal{H}} = L^r g$, where $L := S_{\rho^{te}} S_{\rho^{te}}^*$ is the integral operator.*

Assumption 8 and its equivalent formulations (e.g., Assumption 4 in Rudi et al. [2015]) are common in literature [Caponnetto and De Vito, 2007; Smale and Zhou, 2007]. r quantifies the smoothness of the target function $f_{\mathcal{H}}$ and the extent to which it can be well approximated by functions in $\mathcal{H}$. For $r = 1/2$, the assumption is always satisfied. Intuitively, a larger r implies $f_{\mathcal{H}}$ being smoother.

Finally, we impose an assumption on the capacity of our RKHS, which roughly measures the number of eigenvalues of Σ greater than λ [Caponnetto and De Vito, 2007; Zhang, 2005].

Definition 3 (Effective dimension)**.** *For $\lambda > 0$, define the random variable $\mathcal{N}_x(\lambda) = \langle K_x, (\Sigma + \lambda \mathrm{I})^{-1} K_x \rangle_{\mathcal{H}}$ with $x \in \mathcal{X}$ distributed according to ρ, then $\mathcal{N}_\rho(\lambda) = \mathbb{E}_\rho \mathcal{N}_x(\lambda)$ is called effective dimension.*

Assumption 9 (Capacity condition)**.** *$\forall \lambda > 0$, there exists $0 \leqslant \gamma \leqslant 1$, $Q > 0$ s.t. $\mathcal{N}_{\rho_X^{te}}(\lambda) < Q\lambda^{-\gamma}$.*

It is known that the condition in Assumption 9 is ensured if the eigenvalues $(\eta_i)_i$ of the covariance operator Σ satisfy a polynomial decaying condition $\eta_i \sim i^{-1/\gamma}$ (see Appendix A.3).

3.4.2 Excess Risk Bounds

In this section, we present our main theoretical results. We will derive excess risk bounds for the Nyström predictor defined in eq. (3.13) and we will show that Nyström approximation does not affect the state-of-art rates of convergence while instead reducing both time and memory requirements. We present now the main result .

Theorem 2. *Under assumptions 4,5,6,7,8,9, for ALS sampling, let $\delta > 0$, $\left(\frac{256(W+\sigma^2)\log^2(4/\delta)}{n}\right)^{\frac{1}{\gamma(1-q)+1+q}} \leqslant \lambda \leqslant \|\Sigma\|_{op}$, and $m \geqslant 144 T^2 Q \lambda^{-\gamma} \log \frac{8n}{\delta}$, with probability greater or equal than $1-\delta$*

$$\left|\mathcal{R}(\widehat{f}^{w}_{\lambda,m}) - \mathcal{R}(f_{\mathcal{H}})\right|^{1/2} \leqslant 64B \left(\frac{W}{n\sqrt{\lambda}} + \sqrt{\frac{\sigma^2}{n\lambda^{\gamma(1-q)+q}}}\right) \log\left(\frac{8}{\delta}\right) + 43R\lambda^r.$$

proof. We need two propositions in Appendix A.2. We split the excess risk as

$$\begin{aligned}\left|\mathcal{R}(\widehat{f}^{w}_{\lambda,m}) - \mathcal{R}(f_{\mathcal{H}})\right|^{1/2} &= \left\|\widehat{f}^{w}_{\lambda,m} - f_{\mathcal{H}}\right\|_{\rho^{te}_X} = \left\|\Sigma^{1/2}(\widehat{f}^{w}_{\lambda,m} - f_{\mathcal{H}})\right\|_{\mathcal{H}} \\ &= \left\|\Sigma^{1/2}(V(V^*\widehat{\Sigma}_w V + \lambda \mathrm{I})^{-1}V^*\widehat{S}^* M_w y - f_{\mathcal{H}})\right\|_{\mathcal{H}} \\ &\leqslant \underbrace{\left\|\Sigma^{1/2}V(V^*\widehat{\Sigma}_w V + \lambda \mathrm{I})^{-1}V^*\widehat{S}^* M_w (y - \widehat{S} f_{\mathcal{H}})\right\|_{\mathcal{H}}}_{\mathbb{A}} + \\ &\quad + \underbrace{\left\|\Sigma^{1/2}(\mathrm{I} - V(V^*\widehat{\Sigma}_w V + \lambda \mathrm{I})^{-1}V^*\widehat{\Sigma}_w) f_{\mathcal{H}}\right\|_{\mathcal{H}}}_{\mathbb{B}}\end{aligned}$$

Term $\mathbb{A}$

$$\mathbb{A} \leqslant \underbrace{\left\|\Sigma^{1/2}\widehat{\Sigma}^{-1/2}_{w\lambda}\right\|}_{\mathbb{A}_1} \underbrace{\left\|\widehat{\Sigma}^{1/2}_{w\lambda}V(V^*\widehat{\Sigma}_{w\lambda}V)^{-1}V^*\widehat{\Sigma}^{1/2}_{w\lambda}\right\|}_{\mathbb{A}_2} \underbrace{\left\|\widehat{\Sigma}^{-1/2}_{w\lambda}\Sigma^{1/2}_{\lambda}\right\|}_{\mathbb{A}_3 = \beta} \underbrace{\left\|\Sigma^{-1/2}_{\lambda}\widehat{S}^* M_w (y - \widehat{S} f_{\mathcal{H}})\right\|_{\mathcal{H}}}_{\mathbb{A}_4}$$

- $\mathbb{A}_1$:

$$\mathbb{A}_1 \leqslant \underbrace{\left\|\Sigma^{1/2}\Sigma^{-1/2}_{\lambda}\right\|}_{\leqslant 1} \underbrace{\left\|\Sigma^{1/2}_{\lambda}\widehat{\Sigma}^{-1/2}_{w\lambda}\right\|}_{\beta} \leqslant \beta$$

- $\mathbb{A}_2$: using lemma 8 in Rudi et al. [2015] it's easy to show that $\left\|\widehat{\Sigma}^{1/2}_{w\lambda}V(V^*\widehat{\Sigma}_{w\lambda}V)^{-1}V^*\widehat{\Sigma}^{1/2}_{w\lambda}\right\|^2 = \left\|\widehat{\Sigma}^{1/2}_{w\lambda}V(V^*\widehat{\Sigma}_{w\lambda}V)^{-1}V^*\widehat{\Sigma}^{1/2}_{w\lambda}\right\|$, and therefore the only possible values for $\mathbb{A}_2$ are 0 and 1. Then

$$\mathbb{A}_2 \leqslant 1$$

- $\mathbb{A}_3 = \beta$: using proposition 7 in Rudi et al. [2015] we have

$$\beta \leqslant \frac{1}{1-b}$$

if $b = \lambda_{\max}\left[\Sigma^{-1/2}_{\lambda}(\Sigma - \widehat{\Sigma}_w)\Sigma^{-1/2}_{\lambda}\right] < 1$.

Applying Lemma 18 in Gogolashvili et al. [2023] we get, with probability

greater than $1-\delta$

$$\lambda_{\max}\left[\Sigma_\lambda^{-1/2}(\Sigma-\widehat{\Sigma}_w)\Sigma_\lambda^{-1/2}\right] \leqslant \left\|\Sigma_\lambda^{-\frac{1}{2}}\left(\Sigma-\widehat{\Sigma}_w\right)\Sigma_\lambda^{-\frac{1}{2}}\right\|_{\mathrm{HS}} \leqslant 4\left(\frac{W}{\lambda n}+\sqrt{\frac{\sigma^2\mathcal{N}_{\rho_X^{te}}(\lambda)^{1-q}}{\lambda^{1+q}n}}\right)\log\left(\frac{2}{\delta}\right) \tag{3.16}$$

and for $n\lambda^{1+q} \geqslant 64\,(W+\sigma^2)\,\mathcal{N}_{\rho_X^{te}}(\lambda)^{1-q}\log^2\left(\frac{2}{\delta}\right)$, with probability greater than $1-\delta$ the above quantity is less or equal than $3/4$.

- $\mathbb{A}_4$:

  To control this term we use lemma 19 in Gogolashvili et al. [2023], where $\xi_i = \Sigma_\lambda^{-1/2}w(x_i)K_{x_i}y_i$. We obtain that, with probability greater or equal than $1-\delta$

$$\left\|\frac{1}{n}\sum^n\xi_i-\mathbb{E}[\xi]\right\|_{\mathcal{H}} = \left\|\Sigma_\lambda^{-1/2}\widehat{S}^*M_w(y-\widehat{S}f_{\mathcal{H}})\right\|_{\mathcal{H}} \leqslant 4B\left(\frac{W}{n\sqrt{\lambda}}+\sqrt{\frac{\sigma^2\mathcal{N}_{\rho_X^{te}}(\lambda)^{1-q}}{n\lambda^q}}\right)\log\left(\frac{2}{\delta}\right).$$

Term $\mathbb{B}$

As regards $\mathbb{B}$, following Rudi et al. [2015] we proceed as follows. Noting that $V(V^*\widehat{\Sigma}_{w\lambda}V)^{-1}V^*\widehat{\Sigma}_{w\lambda}VV^* = VV^*$, we have

$$\begin{aligned}
I-V(V^*\widehat{\Sigma}_{w\lambda}V)^{-1}V^*\widehat{\Sigma}_w &= I-V(V^*\widehat{\Sigma}_{w\lambda}V)^{-1}V^*\widehat{\Sigma}_{w\lambda}+\lambda V(V^*\widehat{\Sigma}_{w\lambda}V)^{-1}V^*\\
&= I-V(V^*\widehat{\Sigma}_{w\lambda}V)^{-1}V^*\widehat{\Sigma}_{w\lambda}VV^*-V(V^*\widehat{\Sigma}_{w\lambda}V)^{-1}V^*\widehat{\Sigma}_{w\lambda}\left(I-VV^*\right)+\lambda V(V^*\widehat{\Sigma}_{w\lambda}V)^{-1}V^*\\
&= \left(I-VV^*\right)+\lambda V(V^*\widehat{\Sigma}_{w\lambda}V)^{-1}V^*-V(V^*\widehat{\Sigma}_{w\lambda}V)^{-1}V^*\widehat{\Sigma}_{w\lambda}\left(I-VV^*\right).
\end{aligned}$$

By assumption 8, we have $\left\|\Sigma_{w\lambda}^{-(r-1/2)}f_{\mathcal{H}}\right\|_{\mathcal{H}} \leqslant \left\|\Sigma_{w\lambda}^{-r}f_{\mathcal{H}}\right\|_{\mathcal{H}} \leqslant \left\|\Sigma_w^{-r}f_{\mathcal{H}}\right\|_{\mathcal{H}} \leqslant R$. Define $r' := r-1/2$ to simplify the notation. Using the above decomposition, we can rewrite term $\mathbb{B}$ as

$$\begin{aligned}
\mathbb{B} &\leqslant \left\|\Sigma^{1/2}\left(I-V(V^*\widehat{\Sigma}_{w\lambda}V)^{-1}V^*\widehat{\Sigma}_w\right)\Sigma_{w\lambda}^{r'}\right\|\left\|\Sigma_{w\lambda}^{-r'}f_{\mathcal{H}}\right\|_{\mathcal{H}}\\
&\leqslant R\left\|\Sigma^{1/2}\Sigma_\lambda^{-1/2}\right\|\left\|\Sigma_\lambda^{1/2}\left(I-VV^*\right)\Sigma_\lambda^{r'}\right\| + R\lambda\left\|\Sigma^{1/2}\widehat{\Sigma}_{w\lambda}^{-1/2}\right\|\left\|\widehat{\Sigma}_{w\lambda}^{1/2}V(V^*\widehat{\Sigma}_{w\lambda}V)^{-1}V^*\Sigma_\lambda^{r'}\right\|\\
&\quad + R\left\|\Sigma^{1/2}\widehat{\Sigma}_{w\lambda}^{-1/2}\right\|\left\|\widehat{\Sigma}_{w\lambda}^{1/2}V(V^*\widehat{\Sigma}_{w\lambda}V)^{-1}V^*\widehat{\Sigma}_{w\lambda}^{1/2}\right\|\left\|\widehat{\Sigma}_{w\lambda}^{1/2}\Sigma_\lambda^{-1/2}\right\|\left\|\Sigma_\lambda^{1/2}\left(I-VV^*\right)\Sigma_\lambda^{r'}\right\|\\
&\leqslant R(1+\beta\theta)\underbrace{\left\|\Sigma_\lambda^{1/2}\left(I-VV^*\right)\Sigma_\lambda^{r'}\right\|}_{\mathbb{B}.1} + R\beta\,\underbrace{\lambda\left\|\widehat{\Sigma}_{w\lambda}^{1/2}V(V^*\widehat{\Sigma}_{w\lambda}V)^{-1}V^*\Sigma_\lambda^{r'}\right\|}_{\mathbb{B}.2},
\end{aligned}$$

with $\theta = \left\| \widehat{\Sigma}_{w\lambda}^{1/2} \Sigma_\lambda^{-1/2} \right\|$.

- $\mathbb{B}.1$:

$$\mathbb{B}.1 = \left\| \Sigma_\lambda^{1/2} \left(I - VV^*\right)^2 \Sigma_\lambda^{r'} \right\| \leqslant \left\| \Sigma_\lambda^{1/2} \left(I - VV^*\right) \right\| \left\| \left(I - VV^*\right) \Sigma_\lambda^{r'} \right\|.$$

  Since VV^* is a projection operator, we have that $(I - VV^*) = (\mathrm{I} - VV^*)^s$, for any $s > 0$, therefore, by applying Cordes inequality (see proposition 6 in Appendix A.3) to $\left\| \left(I - VV^*\right) \Sigma_\lambda^{r'} \right\|$, we have

$$\left\| \left(\mathrm{I} - VV^*\right) \Sigma_\lambda^{r'} \right\| = \left\| \left(\mathrm{I} - VV^*\right)^{2r} \Sigma_\lambda^{\frac{1}{2}2r'} \right\| \leqslant \left\| \left(\mathrm{I} - VV^*\right) \Sigma_\lambda^{1/2} \right\|^{2r'}.$$

  This term can be controlled using Proposition 2 in Appendix A.2.

- $\mathbb{B}.2$:

$$\begin{aligned}
\mathbb{B}.2 &\leqslant \lambda \left\| \widehat{\Sigma}_{w\lambda}^{1/2} V (V^* \widehat{\Sigma}_{w\lambda} V)^{-1} V^* \widehat{\Sigma}_{w\lambda}^{r'} \right\| \left\| \widehat{\Sigma}_{w\lambda}^{-r'} \Sigma_\lambda^{r'} \right\| \\
&\leqslant \lambda \left\| \widehat{\Sigma}_{w\lambda}^{1/2} V (V^* \widehat{\Sigma}_{w\lambda} V)^{-1} V^* \widehat{\Sigma}_{w\lambda}^{r'} \right\| \left\| \widehat{\Sigma}_{w\lambda}^{-1/2} \Sigma_\lambda^{1/2} \right\|^{2r'} \\
&\leqslant \beta^{2r'} \lambda \left\| \left(V^* \widehat{\Sigma}_{w\lambda} V\right)^{1/2} \left(V^* \widehat{\Sigma}_{w\lambda} V\right)^{-1} \left(V^* \widehat{\Sigma}_{w\lambda} V\right)^{r'} \right\| \\
&= \beta^{2r'} \lambda \left\| \left(V^* \widehat{\Sigma}_w V + \lambda I\right)^{-(1/2 - r')} \right\| \leqslant \beta \lambda^{1/2 + r'} = \beta \lambda^r,
\end{aligned}$$

  where the first step is obtained multiplying and dividing by $\widehat{\Sigma}_{w\lambda}^{r'}$, the second step by applying Cordes inequality, the third step by Prop. 6 in Rudi et al. [2015].

Putting all together, for $\left(\frac{256(W+\sigma^2)\log^2(4/\delta)}{n}\right)^{\frac{1}{\gamma(1-q)+1+q}} \leqslant \lambda \leqslant \|\Sigma\|$, and $m \geqslant 144 T^2 Q \lambda^{-\gamma} \log \frac{8n}{\delta}$, with probability greater or equal than $1 - \delta$

$$\left| \mathcal{R}(\widehat{f}_{\lambda,m}^w) - \mathcal{R}(f_{\mathcal{H}}) \right|^{1/2} \leqslant 4B\beta^2 \left(\frac{W}{n\sqrt{\lambda}} + \sqrt{\frac{\sigma^2}{n\lambda^{\gamma(1-q)+q}}} \right) \log\left(\frac{8}{\delta}\right) + 3R(1 + \beta\theta)\lambda^r + R\beta^2\lambda^r.$$

Under the above conditions on λ we have that $\beta \leqslant 4$ and $\theta \leqslant 2$ and then for $\left(\frac{256(W+\sigma^2)\log^2(4/\delta)}{n}\right)^{\frac{1}{\gamma(1-q)+1+q}} \leqslant \lambda \leqslant \|\Sigma\|$, and $m \geqslant 144 T^2 Q \lambda^{-\gamma} \log \frac{8n}{\delta}$, with probability greater or equal than $1 - \delta$

$$\left| \mathcal{R}(\widehat{f}_{\lambda,m}^w) - \mathcal{R}(f_{\mathcal{H}}) \right|^{1/2} \leqslant 64B \left(\frac{W}{n\sqrt{\lambda}} + \sqrt{\frac{\sigma^2}{n\lambda^{\gamma(1-q)+q}}} \right) \log\left(\frac{8}{\delta}\right) + 43R\lambda^r.$$

□

The constraint on the regularization parameter involves quantities that are unknown in practice. While this is standard in theoretical analyses, it should not be interpreted as a practical tuning rule. Instead, it characterizes a regime in which the analysis of Theorem 2 holds and specifies the scaling of λ with respect to the sample size n. In practice, λ is selected empirically (e.g. by cross-validation), while the theoretical condition ensures that the assumptions required in Theorem 2 are satisfied. The analysis of Theorem 2 is based on a decomposition of the excess risk into bias and variance terms. The bias is controlled by the source condition and leads to a term of order λ^r. The variance arises from the estimation error under importance weighting. Assumption 6 controls the effect of large importance weights and determines the dependence on W, σ^2, and the exponent $\gamma(1-q)+q$. The Nyström approximation with approximate leverage score sampling ensures that the kernel operator is well approximated when $m \gtrsim \lambda^{-\gamma}$. This allows the same type of bounds as in the full kernel setting. Finally, the choice of λ balances the bias and variance terms within the bound, and the parameter q determines how importance weighting affects the variance term.

Corollary 1. *Under the assumptions and conditions as in Theorem 2,*

(a) with Assumption 6 satisfied by $q = 0$ and choosing $\lambda \asymp (\|w\|_\infty/n)^{\frac{1}{2r+\gamma}}$, with $m \gtrsim (n/\|w\|_\infty)^{\frac{\gamma}{2r+\gamma}} \log n$, with high probability

$$\mathcal{E}(f^w_{\lambda,m}) = \|\widehat{f}^w_{\lambda,m} - f_{\mathcal{H}}\|^2_{\rho^{te}_X} \lesssim \left(\frac{\|w\|_\infty}{n}\right)^{\frac{2r}{2r+\gamma}}, \tag{3.17}$$

(b) with Assumption 6 satisfied by $q = 1$ and choosing $\lambda \asymp (n/(W+\sigma^2))^{-\frac{1}{2r+1}}$, with $m \gtrsim (n/(W+\sigma^2))^{\frac{\gamma}{2r+1}} \log n$, with high probability

$$\mathcal{E}(f^w_{\lambda,m}) = \|\widehat{f}^w_{\lambda,m} - f_{\mathcal{H}}\|^2_{\rho^{te}_X} \lesssim \left(\frac{1}{n}\right)^{\frac{2r}{2r+1}}. \tag{3.18}$$

The rate in eq. (3.17) matches the optimal convergence rate of standard kernel ridge regression (KRR) established in Caponnetto and De Vito [2007]. However, it explicitly depends on $\|w\|_\infty$, which can become arbitrarily large as the training and test distributions diverge. This result also recovers Theorem 2.1 from Myleiko and Solodky [2024] for Tikhonov regularization in the special case $\gamma = 1$. Compared to their work, we consider ALS sampling, which enables fast rates under a suitable capacity condition. Furthermore, we extend the analysis to the case of unbounded importance weights. Specifically, eq. (3.18) shows that when the weighting function is unbounded (i.e., $q = 1$ in Assumption 6), the convergence rate deteriorates to $\mathcal{O}(n^{-\frac{2r}{2r+1}})$. This slower rate, which does not depend on the capacity assumption 9), is always worse than the rate in eq. (3.17). These findings are consistent with the results reported in Gogolashvili et al. [2023] for the full (non-projected) model.

Note that the rate in eq. (3.18) can possibly be improved, even under our Nyström approximation, by clipping the unbounded IW function w at a threshold that depends on n (see Section 5.2.2 in Gogolashvili et al. [2023]). This idea follows Corollary 2 of Ma et al. [2023], although the result is not directly comparable due to differing assumptions.

Example 1. *To illustrate the benefits of the Nyström approach, consider for example the common setting $q = 0$, $r = 1/2$ and $\gamma = 1$. From eq. (3.17), we achieve the optimal rate of $\mathcal{O}(n^{-1/2})$ [Caponnetto and De Vito, 2007], with $m = \mathcal{O}(\sqrt{n}\log(n))$. This results in computational costs of $\mathcal{O}(m^3 + m^2 n) = \mathcal{O}(n\sqrt{n} + n^2)$ in time and $\mathcal{O}(mn) = \mathcal{O}(n\sqrt{n})$ in memory. These are significantly lower than the costs of the non-approximated method, respectively $\mathcal{O}(n^3)$ and $\mathcal{O}(n^2)$.*

3.5 Unknown Weights

Clearly, when dealing with real data, assuming knowledge of the exact weighting function w is unrealistic. Instead, if we have access to some samples from both distributions, we can attempt to estimate an approximate weighting function

$v \approx w$ and control the error resulting from this approximation. Let us choose a function v such that there exists a probability distribution ρ_X^v with

$$v(x) = \frac{d\rho_X^v(x)}{d\rho_X^{tr}(x)}, \tag{3.19}$$

i.e. $v(x)$ is the Radon-Nikodym derivative of ρ_X^v with respect to ρ_X^{tr}. Note that, in general, $\rho_X^v \neq \rho_X^{te}$. Similarly to eq. (3.6) in Assumption 5, we assume that

$$f_{\mathcal{H}}^v = \arg\min_{f \in \mathcal{H}} \mathcal{E}^v(f) = \|f - g^*\|_{\rho_X^v}^2 \tag{3.20}$$

exists (again we consider the unique with minimal norm in case of multiple minimizers). Furthermore, we assume that v satisfies Assumptions 6, 8, 9 with constants V, η, r, γ, Q, respectively. Since v is chosen, weights $v(x)$ are known for all $x \in \mathcal{X}$, allowing us to compute the estimator

$$\widehat{f}_{\lambda,m}^v = V(V^*\widehat{S}^*\widehat{M}_v\widehat{S}V + \lambda \mathrm{I})^{-1}V^*\widehat{S}^*\widehat{M}_v\widehat{y}, \tag{3.21}$$

where $\widehat{M}_v$ is the diagonal matrix with diagonal entries $v(x_1), \ldots, v(x_n)$. Still, we are interested in the excess risk of this estimator with respect to the original ρ^{te} distribution.

Denoting $\Sigma = \Sigma_{\rho_X^{te}}$ and $\Sigma_v = \Sigma_{\rho_X^v}$, we decompose the excess risk of $\widehat{f}_{\lambda,m}^v$ as:

$$\mathcal{E}(f_{\lambda,m}^v) = \|\widehat{f}_{\lambda,m}^v - f_{\mathcal{H}}\|_{\rho_X^{te}}^2 \leqslant \|\Sigma\Sigma_{\lambda v}^{-1}\|_{op}\|\widehat{f}_{\lambda,m}^v - f_{\mathcal{H}}^v\|_{\rho_X^v}^2 + \|f_{\mathcal{H}}^v - f_{\mathcal{H}}\|_{\rho_X^{te}}^2. \tag{3.22}$$

Well Specified Case If $g^* \in \mathcal{H}$, then clearly $g^* = f_{\mathcal{H}} = f_{\mathcal{H}}^v$ and eq. (3.22) simplifies to:

$$\|\widehat{f}_{\lambda,m}^v - f_{\mathcal{H}}\|_{\rho_X^{te}}^2 \leqslant \|\Sigma\Sigma_{\lambda v}^{-1}\|_{op}\|\widehat{f}_{\lambda,m}^v - f_{\mathcal{H}}\|_{\rho_X^v}^2. \tag{3.23}$$

The second term corresponds to what we have already analyzed in Theorem 2, with the distribution ρ_X^{te} and its associated weighting function w replaced by ρ_X^v and v. The first term quantifies the additional cost incurred due to the mismatch

between ρ_X^{te} and ρ_X^v. Using Proposition 5 from Appendix A.3, we can show that if we set $v(x) \equiv 1$, i.e. $\rho_X^v \equiv \rho_X^{tr}$, then $\|\Sigma\Sigma_{\lambda v}^{-1}\|_{op} \leqslant \|w\|_\infty$. This indicates that, when w is bounded, we have a finite control on the appearing term involving the two covariance operators. By setting $\lambda \asymp (\|w\|_\infty n)^{\frac{1}{2r+\gamma}}$, and ensuring $m \gtrsim n^{\frac{\gamma}{2r+\gamma}} \log n$, then, with high probability

$$\|\widehat{f}_{\lambda,m}^{v\equiv 1} - f_{\mathcal{H}}\|_{\rho_X^{te}}^2 \lesssim \|w\|_\infty \left(\frac{1}{n}\right)^{\frac{2r}{2r+\gamma}}, \tag{3.24}$$

where we used the fact that $\|v\|_\infty = 1$. It is important to note that, when compared with the bound in eq. (3.17), which assumes knowledge of the true (typically unknown) weights, we achieve the same rate in n. This means that, when the model is well-specified, the classical Nyström-ERM algorithm gives the same rate as the importance-weighted variant, despite the covariate shift between train and test distributions. This result agrees with findings in Gogolashvili et al. [2023]; Ma et al. [2023], where anyway random projection approximations are not involved. More than that, we emphasize that although the rate remains the same, the dependence on $\|w\|_\infty$, which is assumed finite but can be arbitrarily large, is worse than in eq. (3.17), where the true w is employed.

Misspecified Case In the misspecified case, the situation is more complex since in general $g^* \neq f_{\mathcal{H}} \neq f_{\mathcal{H}}^v$. In particular, the term $\|f_{\mathcal{H}}^v - f_{\mathcal{H}}\|_{\rho_X^{te}}^2$ in eq. (3.22) is not zero, and its magnitude can become arbitrarily large, depending on the severity of the mismatch between ρ_X^{te} and ρ_X^v.

In Theorem 2, when the exact importance weights w are replaced by an approximate weighting function v, the guarantees of the theorem remain robust in the well-specified case, provided v satisfies the same assumptions as w. In this setting, the replacement of w by v only affects the analysis through the best approximation in $\mathcal{H}$, which changes from $f_{\mathcal{H}}$ to $f_{\mathcal{H}}^v$. This induces an additional term in the excess

risk decomposition of the form $\|f^v_{\mathcal{H}} - f_{\mathcal{H}}\|^2_{\rho^{te}_X}$. Consequently, the convergence rate in n of the excess risk $\|\hat{f}^v_{\lambda,m} - f_{\mathcal{H}}\|^2_{\rho^{te}_X}$ is preserved and remains effectively the same as in Theorem 2 for any such v (even if $v \neq w$). In contrast, in the misspecified case, the discrepancy between best approximations can be of non-negligible order and does not decay with n, thereby degrading the excess risk bound. Therefore, the robustness of the guarantees depends on both the choice of v and the model being well-specified.

Robustness of Theorem 2 under approximate weights. If the approximate weighting function v satisfies the same assumptions as w and is treated as a fixed function, then the bound of Theorem 2 holds with w replaced by v, up to an extra term $\|f^v_{\mathcal{H}} - f_{\mathcal{H}}\|^2_{\rho^{te}_X}$. In the well-specified case ($g^* \in \mathcal{H}$), this extra term is zero, so the guarantee is exactly the same as with the exact w, regardless of how different v is from w. In the misspecified case, the extra term does not decay with n and may degrade the bound. Hence, the guarantees are fully robust to the choice of v in the well-specified setting, but not in the misspecified setting. When v is estimated from data (rather than fixed), the analysis does not apply; see the limitation below.

The analysis above assumes that the approximate weighting function v is known without error. In practice, v is estimated from samples of the training and test distributions using a density ratio estimation method. This creates a fundamental gap. The theoretical guarantees do not cover the additional randomness and error introduced by weight estimation. While one might expect that if the estimation error is small in a suitable norm then the rates would be preserved, a rigorous analysis that combines weight estimation error with Nyström approximation under covariate shift remains open, especially in the misspecified case.

3.6 Simulations and real data experiments

As emphasized in the introduction, the main goal of this study is to show that the Nyström method can deliver significant computational savings under covariate shift without compromising accuracy.

3.6.1 Simulations

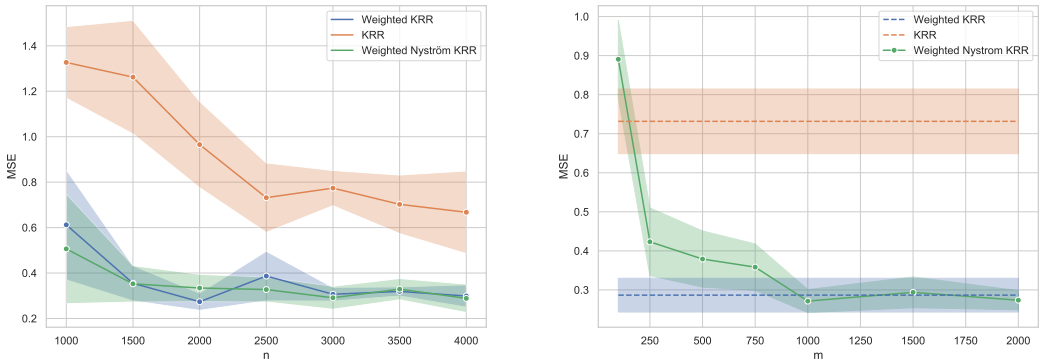


Figure 3.1: On the left: MSE for the different models varying the number of train samples n. The result for Nyström W-KRR model is obtained for optimal m. On the right: $n = 3000$, optimal m is selected as the smallest for which Nyström W-KRR matches the full W-KRR model ($m = 1000$ here).

We start by reproducing the experimental setting in Gogolashvili et al. [2023]. We want to solve a regression problem using KRR with RBF kernel in the context of distribution shift, assuming $\rho_X^{tr} \sim \mathcal{N}(\mu_{tr}, \Sigma_{tr})$, $\rho_X^{te} \sim \mathcal{N}(\mu_{te}, \Sigma_{te})$ and $\mu_{tr} \neq \mu_{te}$, $\Sigma_{tr} \neq \Sigma_{te}$. The regression function is

$$g^*(x) := c_1 e^{-\frac{c_2}{\|x\|_2^{2k}}}, \quad k \in \mathbb{N},\ c \in \mathbb{N},$$

where the parameter k controls the level of misspecification. Note that, in fact, when k increases, the regression function becomes essentially piece-wise constant, and neither constant nor discontinuous functions belong to the RKHS of the

Gaussian kernel. Data samples are generated following

$$y_i^{tr} = g^*(x_i^{tr}) + \xi_i, \quad y_i^{te} = g^*(x_i^{te}),$$

with $x_i^{tr} \sim \rho_X^{tr}$, $x_i^{te} \sim \rho_X^{te}$ and $\xi_i \sim \mathcal{N}(0, \varepsilon^2)$. Figure 3.1 shows the results in this setting for $k = 50$. In the plot on the left, we observe that the two weighted models, namely KRR with IW correction (W-KRR) and its Nyström-approximated version (Nyström W-KRR), perform similarly. As expected, the simple (unweighted) KRR model shows a performance gap. On the right, despite reaching the same error of the weighted KRR model, Nyström approximation can lead to important computational savings allowing for choosing a number of Nyström centers $m \ll n$.

3.6.2 Experiments on Benchmark Datasets

As regards real-world applications, we conduct experiments on commonly used benchmark datasets in the domain adaptation field [Dinu et al., 2023; He et al., 2023; Wang and Sun, 2024; Wilson et al., 2020]. The size of the original datasets is reduced in case of memory issues with KRR and W-KRR when building the full Gram matrix $\hat{K}$ (see memory bottlenecks in Section 3.4). In these experiments, weights are estimated using RuLSIF method [Liu et al., 2013; Yamada et al., 2013]. We consider 4 real-world datasets: HHAR [Stisen et al., 2015], WISDM [Kwapisz et al., 2011], HAR70+ [Ustad et al., 2023] and HARChildren [Tørring et al., 2024]. These datasets consist of data collected from multiple users using wearable sensors, such as accelerometers and gyroscopes. To simulate covariate shift, we train each model on data collected from one user and evaluate it on data from a different user. This cross-user evaluation protocol is commonly regarded as an approximate covariate shift setting, where different users induce different feature distributions while sharing the same activity label space. In practice, this relies on the assumption that the conditional distribution remains approximately

unchanged across users, although this assumption is not strictly verified and may not hold exactly due to user-specific variability in sensor signals and behavior. We employ an RBF kernel with length-scale parameter γ and regularization parameter λ, both selected via cross-validation. The results reported in Table 3.1 are obtained using ALS sampling (see Definition 2), specifically through the BLESS fast implementation described in Rudi et al. [2018]. For comparison, Table A.1 in Appendix A.4 presents analogous results obtained with uniform sampling of the Nyström centers. Additional details on the datasets and experimental setup are provided in Appendix A.4.

Table 3.1: Performances of the various methods, both in terms of MSE and training/prediction time.

	HAR70+ ($n = 20000$)			HARChildren ($n = 15000$)			HHAR ($n = 15500$)			WISDM ($n = 25000$)		
	MSE	t train (s)	t pred (s)	MSE	t train (s)	t pred (s)	MSE	t train (s)	t pred (s)	MSE	t train (s)	t pred (s)
KRR	10 ± 1	1694 ± 2	15.0 ± 0.5	26.6 ± 0.9	762 ± 12	10.2 ± 0.4	3.7 ± 0.3	876 ± 6	10.5 ± 0.9	7.8 ± 0.1	3280 ± 48	38 ± 5
W-KRR	5.0 ± 0.2	1785 ± 2	15.1 ± 0.3	13.5 ± 0.8	809 ± 26	9.0 ± 0.1	1.8 ± 0.3	1034 ± 93	9.9 ± 0.1	4.8 ± 0.2	3364 ± 30	33 ± 2
Ny W-KRR	5.1 ± 0.2	89 ± 19	1.0 ± 0.1	13.2 ± 0.6	8.0 ± 0.2	1.6 ± 0.1	1.8 ± 0.1	6.5 ± 0.4	1.1 ± 0.1	4.7 ± 0.3	9.9 ± 0.4	1.4 ± 0.1

The above Table 3.1 shows that the two methods using IW correction achieve the best and essentially equal performance. However, in terms of computational efficiency, our Nyström W-KRR method, offers significant time and memory savings. The number of Nyström points m required by Nyström W-KRR is 1100, 1800, 1400, and 1550, for HAR70+, HARChildren, HHAR, and WISDM, respectively.

3.7 Conclusions

In this part, we showed that even under covariate shift, random projection techniques — particularly the Nyström method — can significantly enhance

computational efficiency without any loss in learning performance. We provide new statistical bounds for our compressed Nyström algorithm, showing that it matches the optimal statistical guarantees of the full W-KRR model. Leveraging results from random projection theory, we developed novel technical proofs to account for the mismatch between training and test distributions and the potential unboundedness of the IW function. We evaluated the effectiveness of our approach through simulations and experiments on real-world datasets.

However, several questions remain open for future investigation. Although optimal rates are achieved in the well-specified case, the misalignment between the training and test distributions relative to the target function (see the *source condition* in Assumption 8) appears to play a critical role empirically, as it can make covariate shift either benign or severely adversarial. This effect helps explain why some shifts are easier than others. The impact of a distribution shift depends on how the two data sets are related in the space used by the model. When this relation is strong, learning remains stable (benign case). When it is weak, learning becomes more difficult and less stable (adversarial case). Recent works in domain adaptation and transfer learning try to make this idea more precise by quantifying how difficult a covariate shift is. This is done by defining measures that compare the training and test distributions, for example through divergence-based distances, or by studying how well they align in the representation space of the model. In particular, some approaches use operator-based or spectral quantities to measure this alignment, which allows one to distinguish between easier and harder shifts in a more formal way. In this perspective, the interaction between source and target covariance operators provides a natural RKHS-based way to formalize such difficulty notions. A deeper understanding of this phenomenon may come from a more detailed analysis of the constants in the learning bounds and their influence on the overall rate (see eq. (3.22) and the interaction between

the covariance operators of source and target distributions).

Part II

Robustness under Anomaly Detection

Chapter 4

Robust Anomaly Detection

4.1 Introduction

Satellite telemetry provides multivariate time-series measurements of spacecraft subsystems, capturing both operational states and environmental conditions. These data are essential for mission monitoring, fault diagnosis, and autonomous decision-making, as early anomaly detection can prevent failures and extend mission duration. Telemetry is typically multivariate, temporally correlated, and noisy, and the physical quantity measured by each channel is often unknown, which makes automated and reliable anomaly detection particularly challenging [Kotowski et al., 2024; Ruszczak et al., 2024]. Traditional rule-based methods rely on expert-defined thresholds to identify unusual system behavior, but they scale poorly as telemetry complexity increases and cannot capture complex inter-parameter dependencies [Hawkins, 1980; Kotowski et al., 2024; Ruszczak et al., 2024]. To address these limitations, data-driven approaches, particularly unsupervised machine learning methods, have been increasingly adopted. These techniques learn normal operational patterns directly from telemetry, allowing the detection of atypical or unexpected behavior without requiring labeled anomalies [Chandola et al., 2009; Sakurada and Yairi, 2014]. Evaluation of anomaly detection methods has historically been limited by the lack of standardized benchmarks.

The European Space Agency Anomaly Benchmark (ESA-ADB) provides real spacecraft telemetry along with expert annotations. This allows researchers to systematically evaluate anomaly detection methods, reflecting the complexity and variability of real spacecraft operations [Kotowski et al., 2024].

A key challenge in anomaly detection for telemetry is robustness, defined as the ability of a model to maintain reliable detection performance in the presence of noise, perturbations, or variability in the input data. Telemetry signals are often affected by sensor noise, communication errors, or environmental effects. In addition, neural networks, including autoencoders, can be vulnerable to adversarial attacks, where small changes in the input are designed to mislead the model, potentially causing an anomaly to be misclassified as normal or a normal sample to be detected as an anomaly [Finlayson et al., 2019; Goodfellow et al., 2015; Song et al., 2018]. Studying adversarial attacks is important because they represent a worst-case scenario that could be exploited to hide an anomaly or cause false alarms. Ensuring robustness to such attacks is therefore essential for operational reliability.

In this chapter, we adopt a 1D convolutional autoencoder with a controlled bottleneck dimension referred to as the Controlled Latent Space Model (CLSM) as a representative architecture. The CLSM builds on well-known autoencoder designs. Our contribution is not a new model. Instead, we provide a comprehensive empirical characterization of robustness on the ESA-ADB benchmark. Specifically, we first apply different anomaly injection strategies to the input [Enttsel et al., 2024], and then subject the perturbed data to adversarial attacks [Goodge et al., 2021]. Compared to earlier spacecraft anomaly detection baselines (e.g., threshold-based methods or simple autoencoders without controlled bottleneck), our work provides practical guidelines for deploying autoencoder-based detectors under realistic operational conditions where both environmental perturbations and adversarial

attacks are present. The results offer clear insights into the factors that influence autoencoder performance and highlight trade-offs between detection accuracy and robustness.

4.1.1 Data Description

The ESA-ADB [Kotowski et al., 2024] dataset consists of 14 years of real sensor measurements collected from a large spacecraft operated by the European Space Agency. These data were made publicly available as part of the ESA Spacecraft Anomaly Detection Challenge on Kaggle [Kaggle, 2026], providing a standardized benchmark for developing and evaluating anomaly detection algorithms on real-world spacecraft telemetry. The dataset captures a long, continuous multivariate time series, including sensor readings, subsystem states, health indicators, and operator telecommands, reflecting the operational behavior of the spacecraft over extended periods. Formally, the telemetry data can be represented as a multivariate time series. At each time step t, the spacecraft state is described by a vector of measurements from a subset of channels:

$$\mathbf{x}_t = \left[x_t^{(1)}, x_t^{(2)}, \ldots, x_t^{(d)}\right]^\top \in \mathbb{R}^d, \quad d \subset C,$$

where C denotes the full set of 76 available channels, d is the subset of channels selected at time t, and each $x_t^{(i)} \in \mathbb{R}$ is the real-valued measurement of the i-th channel. Over a temporal horizon of length T, the complete telemetry sequence can be written in matrix form as

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_T^\top \end{bmatrix} \in \mathbb{R}^{T \times d},$$

which preserves both the temporal evolution of each channel and the dependencies between channels. The training set contains measurements, among which some points are labeled as anomalies. These numbers highlight the rare and anomalous events, a characteristic that closely mirrors real operational environments. The

Description	Count
Total measurements in the training dataset	14,728,321
Labeled anomalies in training dataset	1,544,104
Test set	521,280

Table 4.1: Overview of the ESA-ADB in Kaggle [Kaggle, 2026].

dataset is highly imbalanced, with normal operational data dominating, while rare and potentially critical events appear infrequently. This imbalance creates a challenging scenario for anomaly detection, where models must effectively distinguish true anomalous behavior from normal variability while minimizing false alarms. The test set in Table 4.1 is provided through the ESA ADB Kaggle competition and does not include reference labels. In Kaggle-style challenges, the test data are intentionally released without labels so that participants cannot tune or overfit their models to the specific test responses; instead, model predictions are submitted to the platform and evaluated against hidden labels to determine the leaderboard scores [Kaggle, 2026]. In the training, we have a single *is_anomaly* label, where a time point is marked as anomalous if at least one monitored channel is affected. As suggested in Kaggle [2026], a subset of six channels (and so $d = 6$) was chosen to focus on the most relevant variables for the mission. This selection reduces computational complexity while maintaining a representative set for developing and evaluating anomaly detection algorithms. The correlation heatmap (Fig-4.1) illustrates the relationships among all channels. The observed correlation structure shows that, unlike other channels, the selected subset of six channels (41–46) exhibits little correlation among themselves. This justifies

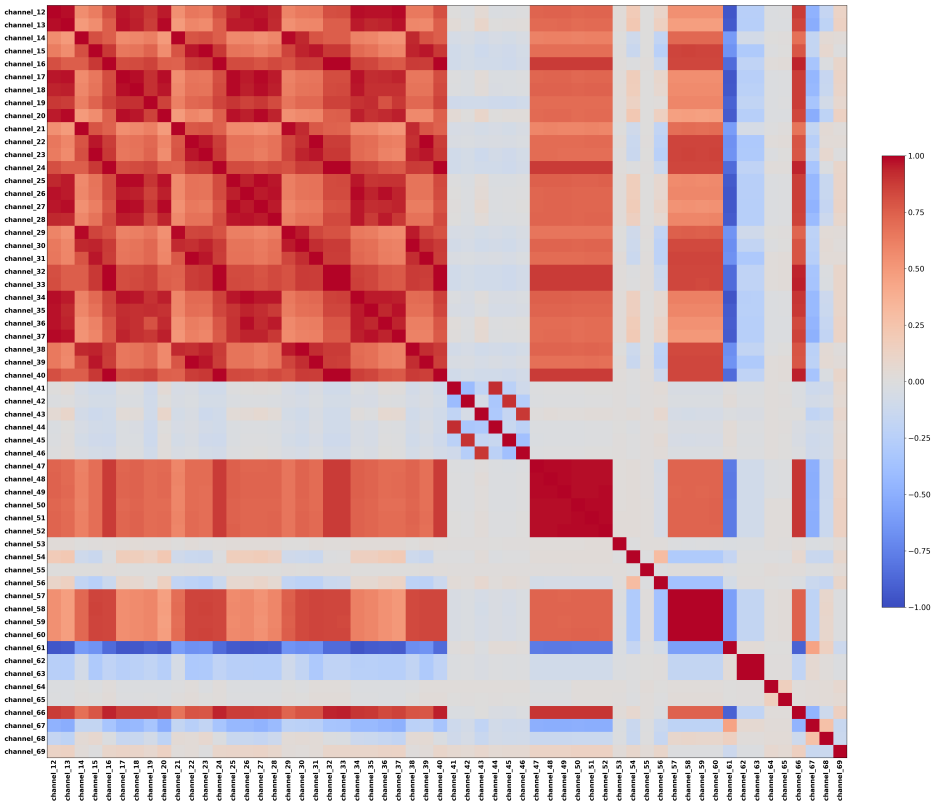


Figure 4.1: Correlation heatmap of a selected subset of telemetry channels.

focusing on this subset as a representative subset for anomaly detection.

4.1.2 Data Preprocessing

Given the high temporal resolution of spacecraft telemetry, several preprocessing steps are applied to prepare the data for anomaly detection models. These steps aim to reduce high-frequency fluctuations, preserve temporal dependencies, and structure the data in a form suitable for learning temporal patterns while preventing information leakage between training and validation sets. In this context, high-frequency fluctuations refer to rapid variations in the signal relative to the nominal sampling frequency of each telemetry channel, which ranges from approximately 0.03 Hz to 0.07 Hz across the ESA missions considered [Kaggle,

2026]. Therefore, fluctuations are considered high-frequency if they occur over time intervals shorter than the typical sampling period, i.e., on the order of tens of seconds. The following data preprocessing steps were applied in this chapter:

- **Temporal Segmentation :** To ensure a realistic evaluation and prevent data leakage, we used the labeled training data from the Kaggle ESA dataset to perform temporal segmentation. The training dataset was divided into $S = 12$ chronological segments, with 10 segments randomly assigned for training, 1 for validation, and 1 for testing (Fig-4.2). This segmentation preserves the chronological order within each segment, ensuring that validation windows always follow the corresponding training windows. Using only the labeled training data guarantees that the random segment allocation is meaningful, as these are the only sequences for which ground-truth labels are available.

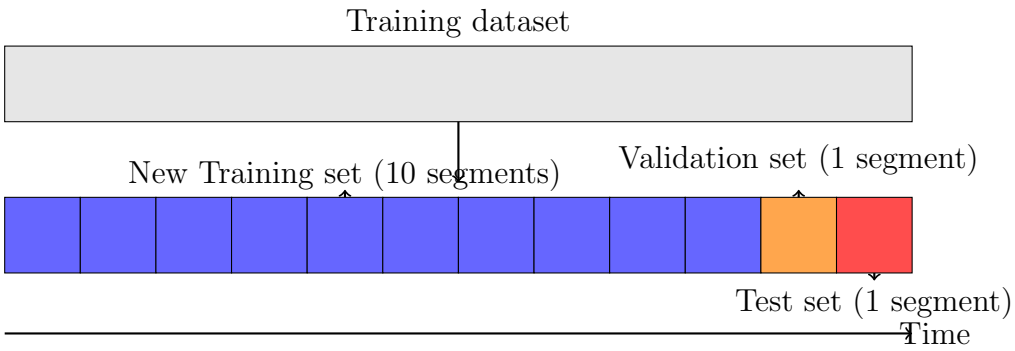


Figure 4.2: Temporal segmentation of the training dataset into $S = 12$ chronological segments, with 10 segments randomly assigned to the new training set (blue), 1 to the validation set (orange), and 1 to the test set (red).

- **Temporal Smoothing :** To reduce high-frequency fluctuations representing irrelevant or non-informative dynamics, temporal smoothing will be applied independently to each telemetry channel. Formally, for each channel $x_t^{(j)}$,

the smoothed value at time t is:

$$\tilde{x}_t^{(j)} = \frac{1}{w} \sum_{k=0}^{w-1} x_{t-k}^{(j)},$$

where w is the window size. This operation attenuates noise while preserving underlying trends and dynamic behaviors that are critical for anomaly detection. Temporal smoothing is particularly important in spacecraft telemetry, where minor oscillations frequently occur during nominal operation.

- **Temporal Windowing :** Each segment is divided into fixed-length temporal windows to enable learning over local temporal contexts. Let $\mathbf{X}_i \in \mathbb{R}^{w \times d}$ denote the i-th temporal window within a segment (Fig. 4.3), defined as

$$\mathbf{X}_i = \begin{bmatrix} \mathbf{x}_{(i-1)s+1}^\top \\ \mathbf{x}_{(i-1)s+2}^\top \\ \vdots \\ \mathbf{x}_{(i-1)s+w}^\top \end{bmatrix},$$

where s is the step size between consecutive windows, and $i = 1, 2, \ldots, N$ indexes the windows within the segment. The first sample of window i is at position $(i-1)s+1$, which means each window starts s samples after the previous one. If $s < w$, consecutive windows overlap, allowing the model to capture detailed temporal transitions, whereas if $s = w$, windows are non-overlapping. This approach allows the model to learn local temporal patterns while maintaining the chronological order of the data within each segment. At the same time, the step size s determines how many windows are generated from each segment, effectively controlling the total number of windows.

- **Data Normalization :** The time-series data are divided into fixed-length segments and are processed using a robust preprocessing pipeline to reduce

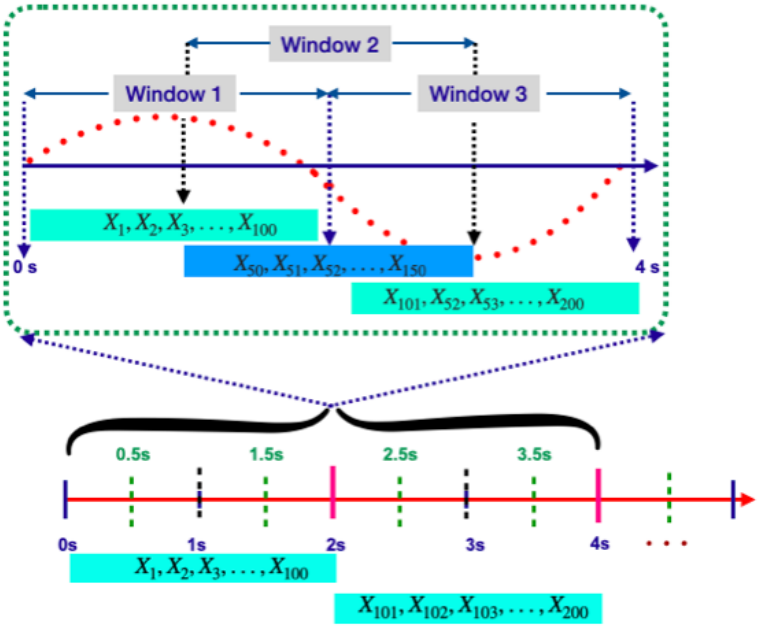


Figure 4.3: Graphical illustration of sliding window data partitioning.

noise and standardize the input features. First, each channel is normalized using a robust scaling method to limit the influence of outliers and extreme values. Second, high-frequency noise is reduced through a smoothing procedure, such as a Savitzky–Golay filter or a uniform moving average, allowing the underlying temporal patterns to become more apparent. Finally, a second normalization step is applied to the smoothed data to ensure consistent scaling across all channels. This two-step normalization process ensures that the model focuses on meaningful temporal dynamics rather than noise or scaling inconsistencies.

This preprocessing pipeline transforms the original telemetry measurements into a structured, temporally consistent representation suitable for multivariate anomaly detection. By combining temporal smoothing, overlapping time windows, and strict temporal splitting, the data is prepared to support model learning while accurately reflecting the operational constraints of real spacecraft telemetry.

4.1.3 Anomaly Injection Strategy

In this work, we use the ESA-ADB [Kotowski et al., 2024] dataset, which contains telemetry data with originally annotated real anomalies. We also generate synthetic anomalies following the methodology described in Enttsel et al. [2024]. Experiments are conducted separately on the ESA data with its real anomalies and on synthetic datasets containing only synthetic anomalies, allowing a systematic evaluation of model performance on both types of anomalies. For both datasets, time series were processed using separate procedures for normal and anomalous sequences before creating the final datasets. Normal sequences were first segmented into temporal windows (see Section 4.1.2), and any windows containing anomalies were removed so that the model would learn exclusively normal behavior. Each window was then smoothed using a moving average filter and standardized with a smoothed scaler computed solely on normal training windows, ensuring stable and comparable windows suitable for training. Anomalous sequences were processed differently. Raw windows were standardized using a scaler fitted only on normal training data, after which synthetic anomalies were injected at controlled locations and amplitudes in the synthetic dataset, allowing precise control over anomaly properties for systematic evaluation. The windows were then denormalized to restore their original value range, smoothed to reduce artificial noise, and standardized again with the smoothed scaler to ensure consistent scaling between normal and anomalous windows. Finally, normal windows and synthetically altered windows were concatenated to form the final synthetic dataset, with each window assigned a binary label indicating the absence (0) or presence (1) of an anomaly. This strategy, in which the model is trained exclusively on normal data and evaluated on synthetic datasets containing controlled injected anomalies, provides a principled and reproducible framework for assessing unsupervised anomaly detection models.

Types of perturbations (Fig-4.4). Let $\mathbf{X}_{\text{nom}}^{(i)} \in \mathbb{R}^{d \times w}$ denote the i-th normal input window. A perturbation applied to specific points within this window is represented as $\boldsymbol{\eta}^{(i)} \in \mathbb{R}^{d \times w}$. The resulting anomalous window is defined by

$$\tilde{\mathbf{X}}^{(i)} = \mathbf{X}_{\text{nom}}^{(i)} + \boldsymbol{\eta}^{(i)} \odot \mathbf{M}^{(i)},$$

where $\mathbf{M}^{(i)} \in \{0,1\}^{d \times w}$ is a binary mask indicating the locations of anomalous points, and $\odot$ denotes the Hadamard product (element-wise multiplication). Each type of anomaly is characterized by a base amplitude a, derived from a control parameter δ that regulates anomaly intensity, and a temporal pattern specifying which points are affected.

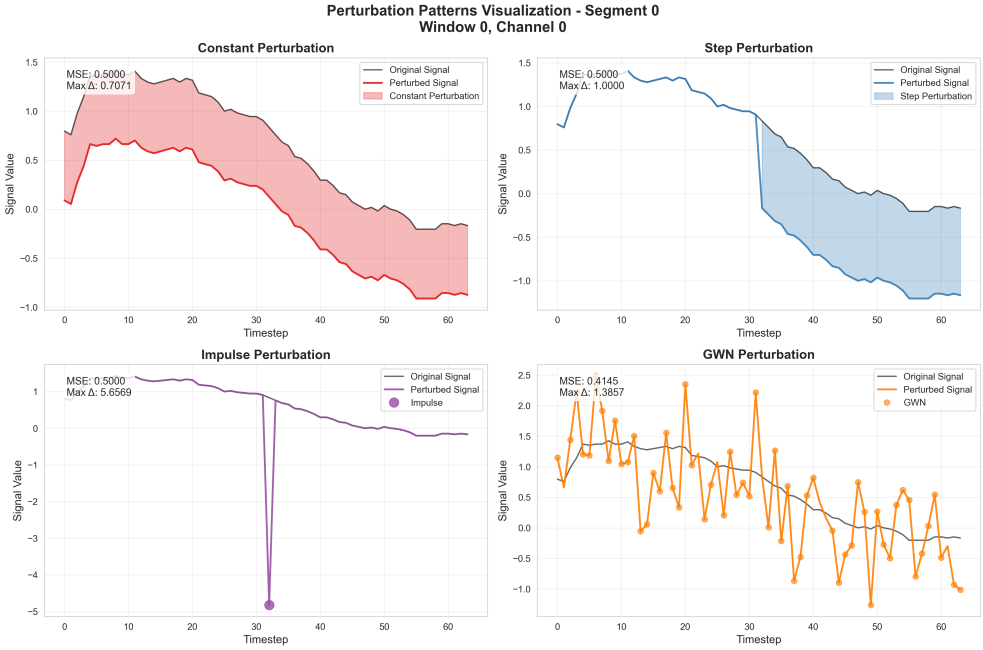


Figure 4.4: Types of perturbations applied to a temporal window.

$$M_{c,t}^{(i)} = \begin{cases} 1, & \text{if the point } (c,t) \text{ in window } i \text{ is anomalous,} \\ 0, & \text{otherwise.} \end{cases}$$

The mask allows perturbations to be applied only at anomalous locations, preserving nominal dynamics elsewhere. The amplitude of injected anomalies is controlled by the parameter $\delta > 0$, and $\epsilon_{c,t} \sim \mathcal{N}(0,1)$ represents standard Gaussian noise for stochastic perturbations.

1. **Constant Anomaly:** a uniform shift applied to all anomalous points in window i:

$$\tilde{\mathbf{X}}^{(i)}_{c,t} = \mathbf{X}^{(i)}_{\text{nom},c,t} + a \cdot s \cdot M^{(i)}_{c,t}, \quad a = \sqrt{\delta}, \quad s \in \{-1, +1\}.$$

   where $\mathbf{X}^{(i)}_{\text{nom}} \in \mathbb{R}^{d \times w}$ denotes the i-th normal input window, and its elements are written as $\mathbf{X}^{(i)}_{\text{nom},c,t}$, representing the value of channel c at time step t in window i. This perturbation applies the same offset to every anomalous point while leaving nominal points unchanged due to the mask. It simulates persistent drifts or sensor biases and allows the autoencoder to be evaluated on its ability to detect sustained deviations.

2. **Step Anomaly:** a sudden change starting at the midpoint of window i:

$$\tilde{\mathbf{X}}^{(i)}_{c,t} = \begin{cases} \mathbf{X}^{(i)}_{\text{nom},c,t}, & t < w/2, \\ \mathbf{X}^{(i)}_{\text{nom},c,t} + a \cdot s \cdot M^{(i)}_{c,t}, & t \geq w/2, \end{cases} \quad a = \alpha\sqrt{2\delta}, \quad 0 < \alpha \leq 1, \quad s \in \{-1, +1\}.$$

   This perturbation applies a constant offset to anomalous points only in the second half of the window, leaving nominal points unchanged due to the mask. It evaluates the autoencoder's ability to detect abrupt regime shifts and sudden changes in signal trends.

3. **Impulse Anomaly:** a single-point spike at the center of window i:

$$\tilde{\mathbf{X}}^{(i)}_{c,t} = \begin{cases} \mathbf{X}^{(i)}_{\text{nom},c,t} + a \cdot M^{(i)}_{c,t}, & t = w/2, \\ \mathbf{X}^{(i)}_{\text{nom},c,t}, & \text{otherwise}, \end{cases} \quad a = \alpha\sqrt{\frac{\delta}{w}}.$$

This perturbation applies an offset to a single anomalous point at the center of the sequence, leaving all other points unchanged due to the mask. It evaluates the autoencoder's sensitivity to short-lived transient events.

4. **Gaussian White Noise (GWN):** continuous stochastic perturbation in window i:

$$\tilde{\mathbf{X}}_{c,t}^{(i)} = \mathbf{X}_{\text{nom},c,t}^{(i)} + a \cdot \epsilon_{c,t} \cdot M_{c,t}^{(i)}, \quad \epsilon_{c,t} \sim \mathcal{N}(0,1), \quad a = \sqrt{\delta}, \quad t = 1, \ldots, w.$$

This perturbation applies stochastic noise to anomalous points only, as indicated by the mask $M^{(i)}$, affecting the sequence while preserving the dynamics of nominal points. It evaluates the autoencoder's robustness to random fluctuations.

Using these perturbation patterns, we can systematically analyze reconstruction errors and sensitivity of autoencoder models across different anomaly types, durations, and amplitudes. Each anomaly type can be applied independently or simultaneously across multiple channels. The injection process consists of the following steps:

1. **Target channel selection:** One or more channels are selected for perturbation, either randomly to simulate unexpected failures or deterministically to study specific sensor or subsystem vulnerabilities.

2. **Anomaly localization and duration:** For localized anomalies, such as step changes or impulse spikes, both the position within the temporal window and the duration of the anomaly are explicitly specified, allowing precise control over its temporal extent.

3. **Anomaly amplitude scaling:** The perturbation magnitude δ is selected based on domain knowledge or experimental design. Multiple amplitude

levels can be considered to evaluate detection sensitivity under varying anomaly severities.

4. **Preprocessing consistency enforcement:** To ensure compatibility with the model's input distribution, the anomalous window is first denormalized, optionally smoothed, and then re-normalized using the same two-scaler preprocessing pipeline applied during training.

5. **Label generation:** Binary labels are assigned at the time-step level, with injected samples marked as anomalous and all remaining points labeled as normal, enabling detailed evaluation of detection performance.

This injection strategy ensures controlled, realistic, and reproducible anomaly generation.

4.2 Semi-Supervised Approaches

Detecting anomalies in multivariate time series is challenging because anomalous events are rare, diverse, and context-dependent. Semi-supervised approaches address this by modeling normal behavior, so that any observation deviating from the learned patterns is identified as potentially anomalous, even in the absence of labeled anomalous data. This approach ensures that the model captures the essential characteristics of the system under normal operation, enabling reliable detection of anomalous events when labeled anomalies are limited or unavailable.

4.2.1 Learning from normal behavior

Let $\mathbf{x}_t \in \mathbb{R}^d$ denote the multivariate observation at time t, where d is the number of channels and $t = 1, \ldots, T$ indexes the full set of observations in the dataset.

Our goal is to estimate the distribution of normal system behavior,

$$\rho_{\text{normal}}(\mathbf{x}),$$

using only the subset of observations assumed to be nominal, i.e., corresponding to normal operation:

$$\{\mathbf{x}_t\}_{t=1}^{T_{\text{train}}} \sim \rho_{\text{normal}}(\mathbf{x}),$$

where $T_{\text{train}} \leq T$ denotes the number of time steps in this training set of normal observations. An anomaly score $s(\mathbf{x})$ is then defined such that:

$$s(\mathbf{x}) = \begin{cases} \text{low}, & \text{if } \mathbf{x} \sim \rho_{\text{normal}} \\ \text{high}, & \text{if } \mathbf{x} \not\sim \rho_{\text{normal}} \end{cases}.$$

In reconstruction-based approaches, such as autoencoders, $s(\mathbf{x})$ is commonly defined as the reconstruction error:

$$s(\mathbf{x}) = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2,$$

where $\hat{\mathbf{x}}$ is the model's reconstruction of $\mathbf{x}$.

4.3 Model Architectures

This section presents the architectures of one-dimensional convolutional autoencoders (Fig-4.5) used in this study. We introduce the fundamental components of 1D CAEs, including convolutional layers, activation functions, and encoding–decoding mechanisms. We then describe the internal hierarchical structure of the models, highlighting how features are progressively extracted and compressed. Finally, we focus on a CAE with controlled latent compression, explaining how the latent dimensionality is regulated to retain essential information while removing unnecessary information.

4.3.1 1D Convolutional Autoencoders

One-dimensional convolutional autoencoders (1D CAEs) are designed to learn compact and informative representations of multivariate temporal signals while preserving the underlying temporal dependencies and inter-channel correlations. Let $\mathbf{X} \in \mathbb{R}^{d \times w}$ denote an input signal with d channels and temporal length w. A 1D convolutional autoencoder (CAE) consists of an encoder $f_\theta : \mathbf{X} \in \mathbb{R}^{d \times w} \mapsto \mathbf{z} \in \mathbb{R}^{d_z}$ and a decoder $g_\phi : \mathbf{z} \in \mathbb{R}^{d_z} \mapsto \hat{\mathbf{X}} \in \mathbb{R}^{d \times w}$, such that

$$\hat{\mathbf{X}} = g_\phi(f_\theta(\mathbf{X})), \tag{4.1}$$

where $\mathbf{z}$ is the latent representation and θ and ϕ denote the trainable parameters of the encoder and decoder, respectively. The 1D CAEs achieve a hierarchical abstraction of temporal signals, condensing information into a latent space while maintaining the ability to accurately reconstruct complex multivariate sequences.

Encoder. The encoder f_θ transforms the raw input signal into progressively abstract feature maps. Each layer applies a set of learned operations:

$$\mathbf{H}_{\text{enc}}^{(\ell)} = \sigma\Big(\mathbf{W}_{\text{enc}}^{(\ell)} * \mathbf{H}_{\text{enc}}^{(\ell-1)} + \mathbf{b}_{\text{enc}}^{(\ell)}\Big), \quad \ell = 1, \ldots, L, \tag{4.2}$$

where $\mathbf{H}_{\text{enc}}^{(0)} = \mathbf{X}$, $*$ denotes 1D convolution along the temporal dimension, and $\sigma(\cdot)$ is a non-linear activation function. In the first layer, $\ell = 1$, $\mathbf{W}_{\text{enc}}^{(1)}$ contains learned weight vectors that slide along the input signal to detect patterns over time and across channels. Each weight vector learns which combinations of input values are important for representing the signal. The bias term $\mathbf{b}_{\text{enc}}^{(1)}$ is added to the output of each weight vector before applying the non-linear activation function. Optionally, a pooling operation $P(\cdot)$ reduces the temporal resolution:

$$\mathbf{H}_{\text{pooled}}^{(\ell)} = P\big(\mathbf{H}_{\text{enc}}^{(\ell)}\big), \tag{4.3}$$

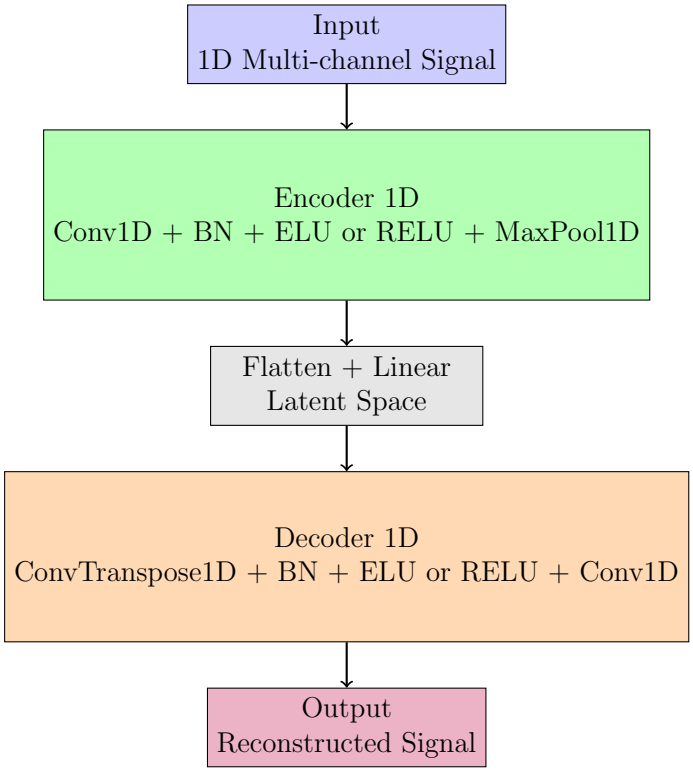


Figure 4.5: 1D Convolutional Autoencoder.

which introduces a structural bottleneck and makes the representation more robust to small shifts in time. Each layer of the encoder progressively abstracts the input:

- **Lower layers ($\ell \ll L$):** Capture fine-grained temporal structures such as peaks, rapid oscillations, and local inter-channel correlations.
- **Intermediate layers:** Integrate information over larger temporal windows, capturing medium-range dependencies and emergent patterns across channels.
- **Higher layers ($\ell \approx L$):** Encode global temporal structures, long-term trends, and high-level correlations, effectively condensing information for the latent space.

Latent Space. The latent space $\mathbf{z} \in \mathbb{R}^{d_z}$ provides a compact representation of the input signal:

$$\mathbf{z} = \mathbf{W}_{\text{enc}}^{(L)} \operatorname{vec}(\mathbf{H}_{\text{enc}}^{(L)}) + \mathbf{b}_{\text{enc}}^{(L)}, \tag{4.4}$$

where $\operatorname{vec}(\cdot)$ flattens the encoder's feature maps into a one-dimensional vector. The latent vector $\mathbf{z}$ consolidates the hierarchical features extracted by the encoder into a single vector, retaining the most relevant information for reconstruction. This compact representation facilitates both efficient signal reconstruction and learning of the essential structure of the input.

Decoder. The decoder g_ϕ reconstructs the input signal from the latent vector $\mathbf{z} \in \mathbb{R}^{d_z}$. To begin, the latent vector is transformed and reshaped into a tensor suitable for the decoder's convolutional layers:

$$\mathbf{H}_{\text{dec}}^{(L+1)} = \operatorname{reshape}(\mathbf{W}_{\text{dec}}^{(L+1)}\mathbf{z} + \mathbf{b}_{\text{dec}}^{(L+1)}), \tag{4.5}$$

where $\mathbf{W}_{\text{dec}}^{(L+1)} \in \mathbb{R}^{d_{L+1} \times d_z}$ and $\mathbf{b}_{\text{dec}}^{(L+1)} \in \mathbb{R}^{d_{L+1}}$ are the weights and biases of the first decoder layer. The dimension d_{L+1} is chosen so that the reshaped tensor matches the input size expected by the first convolutional layer of the decoder. The intermediate decoder layers, indexed from $\ell = L + 2$ to $\ell = L + D$, apply 1D convolutions along the temporal dimension to progressively reconstruct the signal:

$$\mathbf{H}_{\text{dec}}^{(\ell)} = \sigma\Big(\mathbf{W}_{\text{dec}}^{(\ell)} * \mathbf{H}_{\text{dec}}^{(\ell-1)} + \mathbf{b}_{\text{dec}}^{(\ell)}\Big), \tag{4.6}$$

where $\mathbf{W}_{\text{dec}}^{(\ell)}$ are the learned weight vectors and $\mathbf{b}_{\text{dec}}^{(\ell)}$ are the corresponding bias vectors. Finally, the output layer, indexed as $L + D + 1$, maps the last decoder feature map to the reconstructed signal:

$$\hat{\mathbf{X}} = \mathbf{W}_{\text{dec}}^{(L+D+1)} * \mathbf{H}_{\text{dec}}^{(L+D)} + \mathbf{b}_{\text{dec}}^{(L+D+1)}, \tag{4.7}$$

where $\mathbf{W}_{\text{dec}}^{(L+D+1)}$ and $\mathbf{b}_{\text{dec}}^{(L+D+1)}$ are the weights and biases that produce the final reconstruction with the same dimensions as the original input $\mathbf{X}$. This indexing ensures that the decoder layers directly follow the encoder layers, with $L+1$ corresponding to the initial reshape, $L+2$ to $L+D$ to the intermediate convolutional layers, and $L+D+1$ to the output layer. Each weight vector and bias in the decoder is learned during training to progressively expand the latent representation while preserving the essential features captured by the encoder.

1D Convolutional Autoencoder with Controlled Latent Compression. As introduced in Section 4.3.1, the encoder and decoder define the basic 1D convolutional autoencoder architecture. In this section, we focus exclusively on the controlled latent compression. The latent dimension is defined through a compression factor $\alpha \geqslant 1$:

$$d_z = \frac{d \cdot w}{\alpha}.$$

When $\alpha = 1$, the latent space has the same dimension as the input. For $\alpha > 1$, a bottleneck is created, ensuring that the model retains only the most important features and discards redundancy. This controlled latent compression ensures that the autoencoder focuses on the most relevant features, enhancing generalization while still allowing for efficient reconstruction.

Baseline Model without Latent Space. As a baseline, we consider a 1D convolutional autoencoder without a latent space. In this setup, the encoder outputs a representation that has the same size as the input, without any dimensionality reduction or bottleneck. All features, including redundant information, are preserved and passed directly to the decoder. This baseline provides a reference point to evaluate how introducing a compressed latent space affects feature selection and reconstruction quality.

Hyperparameters. The performance and hierarchical feature extraction capabilities of 1D CAEs are influenced by several hyperparameters such as :

- **Window size** w: Determines the temporal context captured by each convolution.
- **Number of layers** L **and** D: Defines the depth of the encoder and decoder, controlling the degree of abstraction.
- **Kernel size** (k): Influences the receptive field of each convolution. Larger kernels capture longer temporal patterns at the expense of local resolution, while smaller kernels emphasize fine-grained structures.
- **Pooling strategy and stride:** Pooling (e.g., max-pooling) reduces temporal resolution and enforces a structural bottleneck. The pooling size determines how many elements are summarized at each step, while the stride controls the step of the operation, affecting downsampling rate and the overlap between consecutive windows.
- **Latent dimension** d_z **and compression factor** α: Govern the trade-off between information retention and model generalization.
- **Activation functions and normalization**: Influence stability, gradient flow, and the ability to model non-linear temporal dependencies.

4.4 Adversarial attacks of Anomalous Points

In this section, we study adversarial perturbations that specifically target anomalous points within a signal segment with the intention of leaving nominal points unchanged [Goodge et al., 2021]. This allows controlled experimentation on how anomalies influence the model's predictions. Each temporal window extracted

from the multivariate time series is represented as a matrix $\mathbf{X}^{(i)} \in \mathbb{R}^{d \times w}$. The index i denotes the position of the window in the sequence. When a window contains only nominal (non-anomalous) observations, it is denoted $\mathbf{X}^{(i)}_{\text{nom}}$, where the subscript nom indicates that the window follows the distribution of normal system behavior. Thus, $\mathbf{X}^{(i)}$ is the general notation for any window, while $\mathbf{X}^{(i)}_{\text{nom}}$ specifically refers to windows assumed to be normal. To indicate the temporal location of anomalies within a window, we define a binary vector $\mathbf{y}^{(i)} \in \{0,1\}^w$, where $y^{(i)}_t = 1$ if time step t in window i is anomalous, and $y^{(i)}_t = 0$ if it is nominal. Let

$$\mathcal{A}^{(i)} = \{\, t \in \{1, \ldots, w\} \mid y^{(i)}_t = 1 \,\}$$

denote the set of anomalous time indices within window i. For each anomalous window $\mathbf{X}^{(i)}$ with $i \in \mathcal{W}_a$ (the set of anomalous window indices), we define a binary mask $\mathbf{M}^{(i)} \in \{0,1\}^{d \times w}$ specifying the locations of points to perturb:

$$\mathbf{M}^{(i)}_{c,t} = \begin{cases} 1, & \text{if } t \in \mathcal{A}^{(i)} \text{ (anomalous)}, \\ 0, & \text{otherwise}, \end{cases} \qquad \forall c = 1, \ldots, d.$$

Let $\boldsymbol{\eta}^{(i)} \in \mathbb{R}^{d \times w}$ denote the perturbation tensor. The effective perturbation is obtained via element-wise multiplication with the mask:

$$\tilde{\boldsymbol{\eta}}^{(i)} = \mathbf{M}^{(i)} \odot \boldsymbol{\eta}^{(i)}, \quad \tilde{\eta}^{(i)}_{c,t} = M^{(i)}_{c,t}\, \eta^{(i)}_{c,t}.$$

The perturbed window is then

$$\tilde{\mathbf{X}}^{(i)} = \mathbf{X}^{(i)} + \tilde{\boldsymbol{\eta}}^{(i)},$$

so that only the anomalous points in the window are modified, while the nominal points remain unchanged.

1. **Random Attack:** Small stochastic perturbations are applied exclusively

to the anomalous points within each window. For each anomalous window $\mathbf{X}^{(i)}$ with $i \in \mathcal{W}_a$, a random tensor $\boldsymbol{\eta}^{(i)} \in \mathbb{R}^{d \times w}$ is drawn from a zero-mean Gaussian distribution:

$$\boldsymbol{\eta}^{(i)} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}),$$

where σ controls the perturbation magnitude and $\mathbf{I}$ is the identity matrix. The effective perturbation is then masked to affect only anomalous points:

$$\tilde{\boldsymbol{\eta}}^{(i)} = \mathbf{M}^{(i)} \odot \boldsymbol{\eta}^{(i)},$$

and the perturbed window is obtained as

$$\tilde{\mathbf{X}}^{(i)} = \mathbf{X}^{(i)} + \tilde{\boldsymbol{\eta}}^{(i)}.$$

2. **Gradient-Based ℓ_2 Attack:** This attack uses the gradient of the autoencoder's reconstruction loss to iteratively perturb only the anomalous points. Let f_θ denote the autoencoder parameterized by θ, and $\mathcal{W}_a$ the set of indices of anomalous windows. For each anomalous window $\mathbf{X}^{(i)}$ with $i \in \mathcal{W}_a$, the objective function is

$$\mathcal{L}(\theta) = \sum_{i \in \mathcal{W}_a} \left\| f_\theta(\mathbf{X}^{(i)}) - \mathbf{X}^{(i)} \right\|_2^2,$$

where $\mathcal{L}(\theta)$ explicitly depends on the autoencoder parameters θ. The gradient with respect to the input is masked to affect only anomalous points:

$$\nabla \mathbf{X}^{(i)} \leftarrow \mathbf{M}^{(i)} \odot \nabla \mathbf{X}^{(i)}, \quad i \in \mathcal{W}_a.$$

The adversarial update rule is then

$$\tilde{\mathbf{X}}^{(i)} \leftarrow \mathbf{X}^{(i)} - \alpha \nabla \mathbf{X}^{(i)}, \quad i \in \mathcal{W}_a,$$

where α is the step size. A relative perturbation budget β can be enforced for each window:

$$\frac{\sum |\tilde{\mathbf{X}}^{(i)} - \mathbf{X}^{(i)}|}{\sum |\mathbf{X}^{(i)}|} \leqslant \beta, \quad i \in \mathcal{W}_a.$$

This framework allows a systematic study of how controlled perturbations on anomalous points affect the autoencoder's robustness.

4.5 Evaluations

The evaluation of semi-supervised anomaly detection models is critical for assessing their reliability and sensitivity in industrial time series. Due to the rarity and heterogeneity of anomalies, a comprehensive evaluation strategy must carefully consider reconstruction errors and the model's ability to detect deviations from normal patterns.

4.5.1 Error Computation

During training and validation, the network is optimized using a reconstruction loss computed over each input window. Let $\mathbf{X}^{(i)} \in \mathbb{R}^{d \times w}$ denote the i-th window extracted from the time series, and $\hat{\mathbf{X}}^{(i)}$ its reconstruction. Here, $i = 1, \ldots, N$ indexes the temporal windows, $c = 1, \ldots, d$ indexes the channels, and $t = 1, \ldots, w$ indexes the points within each window. The reconstruction loss for a window is computed as

$$\frac{1}{d \cdot w} \sum_{c=1}^{d} \sum_{t=1}^{w} \ell\big(X_{c,t}^{(i)}, \hat{X}_{c,t}^{(i)}\big),$$

where $\ell(x, \hat{x})$ is the squared error during training and the absolute error during validation. This provides a consistent reconstruction error metric across all windows and phases. Because overlapping windows are used during inference, each time point in the original series may belong to multiple windows. Reconstruction errors are aggregated across all windows that contain a given time point to produce

a single anomaly score per point in the time series.

4.5.2 Metrics for Quantitative Evaluation

To rigorously assess the performance of anomaly detection models, a set of complementary metrics is computed both per anomaly type and across channels. These metrics quantify the model's ability to correctly identify anomalous time points while minimizing false alarms, which is critical in industrial applications where undetected failures and false alerts can have significant operational consequences. Let TP, FP, and FN denote the number of true positives, false positives, and false negatives, respectively, computed at the sample level.

- **Precision (P):** Measures the proportion of correctly detected anomalies among all predicted anomalies.

$$P = \frac{TP}{TP + FP}.$$

- **Recall (R):** Measures the proportion of true anomalies that are correctly detected.

$$R = \frac{TP}{TP + FN}.$$

- F_1**-score:** The harmonic mean of Precision and Recall, providing a balanced measure of detection performance.

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R}.$$

- $F_{0.5}$**-score:** A weighted harmonic mean of precision (P) and recall (R), giving more weight to precision. In time-series anomaly detection, it is usually computed per timestamp, so longer anomalies contribute more since

each point is counted individually:

$$F_{0.5} = (1 + 0.5^2) \cdot \frac{P \cdot R}{0.5^2 \cdot P + R}.$$

This is the standard $F_{0.5}$ definition without event-based adjustments.

- **Area Under the ROC Curve (AUC-ROC):** The AUC-ROC is the area under the Receiver Operating Characteristic (ROC) curve, which plots the true positive rate (TPR) against the false positive rate (FPR) as the decision threshold τ varies. Formally, it can be expressed as the probability that a randomly chosen anomalous point x^+ has a higher anomaly score $s(x^+)$ than a randomly chosen normal point x^-:

$$\text{AUC-ROC} = \Pr\big(s(x^+) > s(x^-)\big), \quad x^+ \sim \text{anomalous}, \ x^- \sim \text{normal}.$$

This metric is threshold-independent and evaluates the model's ability to separate normal and anomalous points across all possible thresholds.

These metrics together provide a comprehensive evaluation framework, enabling analysis of both overall detection capability and the trade-offs between sensitivity and specificity. They are particularly useful for comparing model performance across different anomaly types, channels, and severity levels.

4.5.3 Channel-wise Detection and Thresholding

Anomaly detection is performed independently for each channel. Let $s_t^{(c)} \in \mathbb{R}$ denote the anomaly score at time t for channel c. For each channel, a percentile-based threshold τ_c is computed from the distribution of training scores:

$$\tau_c = \text{Percentile}_p\left(\{s_t^{(c)} \mid t \in \text{training set}\}\right),$$

where p is selected to maximize the F1-score on the validation set. A time point t is classified as anomalous for channel c if:

$$\hat{y}_t^{(c)} = \begin{cases} 1, & s_t^{(c)} > \tau_c, \\ 0, & \text{otherwise.} \end{cases}$$

The final prediction at time t is obtained by aggregating channel-wise decisions using a logical OR operation:

$$\hat{y}_t = \bigvee_{c=1}^{d} \hat{y}_t^{(c)}.$$

Thus, an anomaly detected in any individual channel is sufficient to trigger an alert at the corresponding time step.

4.5.4 Evaluation Using Synthetic and Real Anomalies

Model evaluation is conducted by treating synthetic anomalies and real anomalies as two distinct and independent cases, each serving a specific purpose within the assessment process.

1. **Synthetic anomalies:** Controlled perturbations, including constant shifts, step changes, impulse spikes, and stochastic noise are injected into the validation and test windows (see Section 4.1.3). These artificially generated anomalies enable a systematic and detailed analysis of model behavior, allowing threshold calibration, evaluation of sensitivity to different anomaly types and amplitudes, and channel-wise performance assessment under fully controlled conditions.

2. **Real anomalies:** The original anomalous events in the ESA-ADB dataset are kept intact, without any synthetic anomalies, and used exclusively in the validation and test sets. They represent actual anomalous behaviors observed in real industrial operations. Evaluating on these original anomalies ensures

that performance improvements observed under controlled experiments generalize to complex and unpredictable real-world conditions.

For each anomaly type, performance metrics are computed independently per channel and then aggregated to provide an overall assessment.

4.6 Experimental Results

As described in Section 4.1, the experiments conducted in this thesis are based on real multivariate temporal data from the European Space Agency (ESA). These data originate from critical space systems, with heterogeneous sensor dynamics and rare but significant anomalies. The detailed dataset description, including anomaly proportions and temporal structure, is provided in Section 4.1.1. Two configurations of one-dimensional convolutional autoencoders (1D CAEs) are evaluated:

- A baseline model without a dedicated latent space.
- A model with a controlled latent space, where the dimensionality d_z is explicitly constrained using a compression factor α.

All experiments are performed on an Intel Xeon Gold 6248R CPU with NVIDIA Tesla V100 GPU and 128 GB RAM. Random seeds are fixed to ensure reproducibility, and model weights are initialized identically across runs.

4.6.1 The baseline model

For the baseline model without latent space, each temporal window contains $w = 64$ time steps and $d = 6$ input channels (features). Consecutive windows have a 50% overlap, and each sequence is divided into $S = 12$ segments, of which 10 are randomly assigned to the training set, 1 to the validation set, and 1 to the test

set. The model is trained in batches of size $B = 64$ over MAX_EPOCHS $= 10$ epochs with a learning rate $\eta = 3 \times 10^{-4}$. To smooth reconstruction errors, a smoothing window of 5 is applied. The encoder consists of $L = 3$ convolutional layers with kernel size $k = 3$, each followed by ReLU activations and max-pooling. The decoder mirrors this structure with $D = 3$ transposed convolutional layers and a final convolution to reconstruct the input.

Table 4.2: Summary of temporal window creation, segment selection, and filtering of anomalous windows for the new training set, derived from the ESA Kaggle training data (Table 4.1). The naturally occurring anomalies are retained in the validation and test sets, and no synthetic anomalies are injected.

Description	**Count**
Temporal windows created from Kaggle training data	
Windows shape	(460,260, 6, 64)
Total windows	**460.260**
Segment selection	
Training windows	**383.550**
Validation set	38.355
Test set	38.355
Filtering anomalous windows from training only	
Total training windows	**383.550**
Normal windows for training	345.026
Anomalous windows excluded from training	38.524

Channel-Wise Threshold Optimization. A key feature of the proposed approach is the use of channel-wise detection thresholds (Sections 4.5.3). Each sensor channel has an independently optimized threshold τ to account for channel-specific variations in noise, sensitivity, and signal dynamics. The threshold optimization was performed on an independent validation segment by maximizing the F0.5-score, which emphasizes precision over recall. This is particularly relevant for ESA systems, where false positives can have significant operational consequences. The

optimal average threshold was found to be $\tau = 6.0$, providing a robust balance between anomaly detection sensitivity and false alarm limitation. The channel-wise approach is advantageous in multivariate settings, as certain sensors may have higher noise levels or different dynamic ranges, making a global threshold suboptimal.

Evaluation on a Test Segment Derived from Kaggle Training Data. The model was evaluated on a test segment obtained by splitting the ESA Kaggle training data (Table 4.2), which provides labeled samples for performance assessment. A simplified preprocessing pipeline was applied, including temporal smoothing, robust normalization, and sliding-window segmentation. Channel-wise performance metrics on this split test segment are summarized in Table 4.3. The confusion matrix for this evaluation is shown in Fig-4.6, providing a visual

F0.5-score	F1-score	Precision	Recall	AUC ROC
0.7544	0.5516	0.9993	0.3810	0.6886

Table 4.3: Performance of the model on the test segment derived from the ESA Kaggle training data.

overview of model predictions on the labeled segment. These results in Table 4.3 indicate that the model achieves extremely high precision, meaning that nearly all detected anomalies correspond to true abnormal events, a crucial property for ESA operational systems. The recall is moderate, suggesting that some subtle anomalies may remain undetected. This reflects the model's conservative detection strategy optimized for the F0.5-score. Overall, the AUC ROC demonstrates that the model effectively differentiates between normal and anomalous behavior across multiple sensor channels.

Generalization to Kaggle Test Data. The autoencoder model was applied to the Kaggle ESA test set, which contains 521.280 observations and is completely

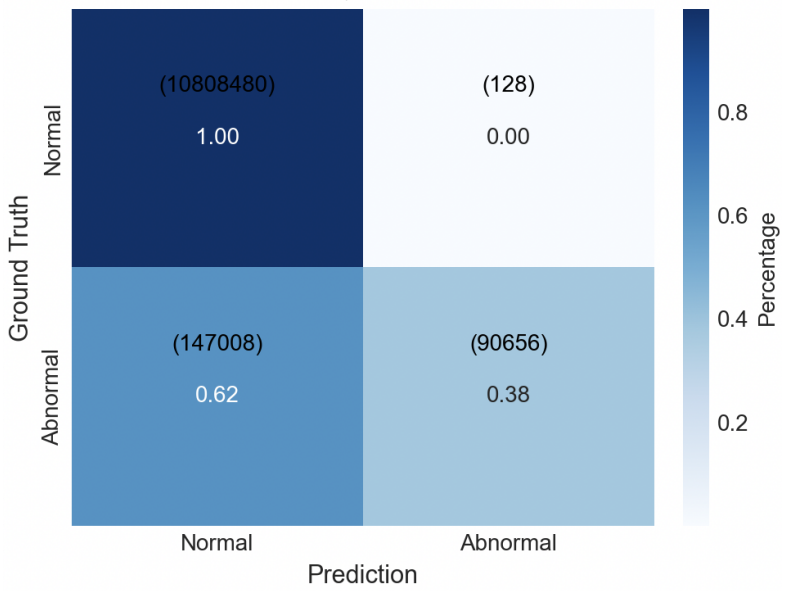


Figure 4.6: Confusion matrix showing the model's performance on the test segment derived from the ESA Kaggle training data.

unlabeled, meaning that the true anomaly locations are not provided (Table 4.1). Of these observations, 6.528 points ($\approx 1.25\%$) were detected as anomalous by the model. The resulting submission achieved a public leaderboard score of 83% using the *Corrected event-wise F0.5-score*, the official evaluation metric. The Corrected event-wise F0.5-score, introduced by Sehili & Zhang [El Amine Sehili and Zhang, 2023] and adopted as a primary metric in the ESA-ADB benchmark [Kotowski et al., 2024], is a harmonic mean of corrected event-wise precision and event-wise recall, giving precision twice the weight of recall. Event-wise precision and recall treat each continuous anomalous segment as a single event, regardless of its size. The result demonstrates strong performance on the publicly evaluated portion of unseen data. On the private leaderboard, which evaluates the model on a separate, non-public portion of the test set and provides a more reliable estimate of generalization, the model obtained a score of 67.5%. This evaluation highlights both the strengths and limitations of the proposed approach. While the high

public score confirms the model's ability to capture dominant anomaly patterns, the lower private score underscores the challenge of generalizing to fully unseen signal distributions.

Evaluation Results on Injected Anomalies. We evaluated the model's performance on the validation and test sets obtained by splitting the ESA Kaggle training data, applying synthetic anomalies to both sets. The validation and test segments were chosen to be relatively close in distribution to each other, avoiding large discrepancies that could artificially degrade performance. The types of injected anomalies include Constant, Step, Impulse, and Gaussian White Noise (GWN), as described in Section 4.1.3. Each anomaly is applied to a randomly selected channel within the input window, enabling controlled and localized perturbations. Using the strategies described in Section 4.1.3, Table 4.5 and Table 4.6 summarize the performance metrics on the validation and test sets (Table 4.4), respectively. The metrics include $F_{0.5}$, F_1, Precision, Recall, and AUC.

Table 4.5: Performance on the validation set obtained from the ESA Kaggle training data for different anomaly injection strategies.

Perturbation	$F_{0.5}$	F_1	Precision	Recall	AUC
Constant	0.7128	0.7971	0.6659	0.9926	0.9221
Step	0.3882	0.4356	0.3619	0.5470	0.8638
Impulse	0.0147	0.0229	0.0119	0.3081	0.8197
GWN	0.7144	0.7992	0.6672	0.9964	0.9477

The results on the validation set (Table 4.5) show that the model performs best on Constant and GWN anomalies, achieving the highest F_1 and $F_{0.5}$ scores. Detection behavior differs between these two types. Constant anomalies are identified with very high precision but lower recall, indicating a conservative strategy that flags fewer points with high confidence. GWN anomalies, in contrast, exhibit extremely high recall but lower precision, meaning most anomalous points

Table 4.4: Summary of temporal window creation, segment selection, and normal windows after filtering anomalies for the new training set, as well as windows after anomaly injection for the validation and test sets derived from the ESA Kaggle training data.

Description	**Count**
Temporal windows created	
Windows shape	(460,260, 6, 64)
Total windows	**460.260**
Segment selection	
Training windows	**383.550**
Validation set	38.355
Test set	38.355
Normal windows after filtering anomalies	
Training set	345.026
Validation set	33.336
Test set	33.336
Windows after anomaly injection	
Validation set	66,672
Test set	66,672

are detected at the cost of more false positives. Step anomalies show moderate recall and relatively low precision, leading to modest overall scores. Impulse anomalies remain the most challenging, with very low F_1 and $F_{0.5}$ scores despite moderate recall, reflecting the difficulty of detecting brief, transient events where false positives dominate.

Table 4.6: Performance on the test set obtained from the ESA Kaggle training data for different anomaly injection strategies.

Perturbation	$F_{0.5}$	F_1	Precision	Recall	AUC
Constant	0.7133	0.7977	0.6662	0.9939	0.9385
Step	0.3884	0.4402	0.3602	0.5660	0.8892
Impulse	0.0145	0.0226	0.0117	0.3230	0.8501
GWN	0.7144	0.7992	0.6672	0.9963	0.9577

The test set results (Table 4.6) largely confirm the validation trends. Performance remains strong for Constant and GWN anomalies, maintaining the complementary precision and recall profiles observed in validation. Step anomalies continue to show moderate detectability, while Impulse anomalies remain difficult to detect, producing very low precision and overall scores despite non-negligible recall. The similarity between validation and test outcomes suggests stable model behavior across datasets. These results provide a detailed understanding of the model's behavior on injected anomalies, emphasizing its strengths and limitations across different anomaly types. Building on this analysis, we now explore a controlled latent space approach, which aims to further enhance anomaly detection by structuring the representation space for more robust and interpretable detection.

4.6.2 Controlled Latent Space Model

For the enhanced model with latent compression, each temporal window $\mathbf{X} \in \mathbb{R}^{d \times w}$ is projected into a latent vector $\mathbf{z} \in \mathbb{R}^{d_z}$, enforcing a bottleneck that retains only the most informative features. The architecture extends the baseline 1D convolutional autoencoder with a bottleneck between the encoder and decoder. The encoder compresses the input through L convolutional layers (kernel size $k = 5$) with ELU activations, batch normalization, and max-pooling (stride equal to pooling size). Each sequence is split into $S = 12$ segments. Using the labeled Kaggle training data, 10 segments are randomly assigned to training, 1 to validation, and 1 to testing. Consecutive windows have a 50% overlap. Synthetic anomalies are injected on a randomly selected channel within each window, allowing controlled, localized perturbations. Training uses batch size $B = 64$ over MAX_EPOCHS $= 10$ with learning rate $\eta = 3 \times 10^{-4}$, and a Savitzky-Golay filter (smoothing window is 3, polyorder 2) is applied to smooth reconstruction errors. We systematically vary the compression factor $\alpha \in \{1, 2, 4, 16, 32, 64, 128, 192, 384\}$

as well as the number of layers in the encoder and decoder, $L, D \in \{2, 3\}$, to study the trade-off between reconstruction fidelity and compactness of the latent representation. The model is trained fully unsupervised using only nominal windows. Over 10 epochs, the 3-layer CONV_AE1D shows a rapid decrease in training loss for low compression factors (1–4), reaching 0.006–0.008. Medium compression (8–32) increases the final loss to 0.0096–0.0145, indicating that the bottleneck begins to limit reconstruction. High compression (64–384) further increases the loss (0.016–0.090) and slows convergence. The 2-layer model exhibits a similar trend, with slightly lower loss at low compression (1 = 0.004875, 2 = 0.003914, 4 = 0.006174). Medium compression results in 0.00918–0.01456, and high compression reaches 0.016–0.09957, comparable to the 3-layer model. Overall, final loss increases with compression, particularly beyond factor 16. Factors 1–4 provide the best reconstruction, while factor 8 offers a good compromise between compression and reconstruction quality. Loss decreases quickly in early epochs before stabilizing. These reconstruction errors are then used for anomaly detection on a per-sample and per-channel basis (Sections 4.5.1 and 4.5.3).

Table 4.7: Validation AUC for different compression factors and injection types using the 2-layer 1D convolutional autoencoder.

Compression Factor	Constant	GWN	Step	Impulse
1	0.9710	0.9462	0.8886	0.8196
2	0.9259	0.9439	0.8651	0.8197
4	0.9547	0.9448	0.8788	0.8197
8	0.9597	0.9427	0.8820	0.8197
16	0.9457	0.9349	0.8738	0.8195
32	0.9420	0.9341	0.8730	0.8195
64	0.9399	0.9324	0.8727	0.8193
128	0.9386	0.9282	0.8705	0.8187
192	0.9338	0.9255	0.8683	0.8187
384	0.9221	0.9191	0.8621	0.8177

Table 4.8: Test AUC for different compression factors and injection types using the 2-layer 1D convolutional autoencoder.

Compression Factor	Constant	GWN	Step	Impulse
1	0.9764	0.9571	0.9088	0.8500
2	0.9414	0.9558	0.8903	0.8502
4	0.9632	0.9561	0.9010	0.8501
8	0.9674	0.9544	0.9036	0.8502
16	0.9575	0.9489	0.8976	0.8501
32	0.9546	0.9482	0.8969	0.8501
64	0.9524	0.9467	0.8965	0.8499
128	0.9510	0.9434	0.8945	0.8493
192	0.9463	0.9408	0.8923	0.8493
384	0.9390	0.9361	0.8879	0.8482

Tables 4.7 and 4.8 show AUC scores for the 2-layer CONV_AE1D model. Constant and GWN achieve high AUC values (validation: Constant 0.971, GWN 0.946; test: Constant 0.976, GWN 0.957), while Step and Impulse are lower, especially at extreme compressions (validation: Step 0.889–0.871, Impulse 0.820–0.818; test: Step 0.909–0.895, Impulse 0.850–0.849). A compression factor of 8 balances performance, maintaining high AUC for Constant (validation 0.960, test 0.967) and GWN (validation 0.943, test 0.954), while improving Step (validation 0.882, test 0.904) and Impulse (validation 0.820, test 0.850), supporting accurate reconstruction and anomaly detection.

Tables 4.9 and 4.10 show that a compression factor of 2 provides the best overall balance across injection types. For Constant and GWN signals, AUC is highest (Validation: 0.9678 and 0.9465; Test: 0.9743 and 0.9576). Step and Impulse also improve compared with lower factors. Higher compression factors reduce AUC for Constant and GWN without significant gains for Step or Impulse, confirming that factor 2 is the optimal choice for 3-layer models.

Table 4.9: Validation AUC for different compression factors and injection types using the 3-layer 1D convolutional autoencoder.

Compression Factor	Constant	GWN	Step	Impulse
1	0.9459	0.9441	0.8745	0.8197
2	0.9678	0.9465	0.8863	0.8197
4	0.9632	0.9432	0.8834	0.8198
8	0.9476	0.9405	0.8751	0.8197
16	0.9439	0.9351	0.8735	0.8196
32	0.9432	0.9344	0.8735	0.8195
64	0.9390	0.9326	0.8725	0.8193
128	0.9389	0.9283	0.8706	0.8187
192	0.9343	0.9254	0.8683	0.8185
384	0.9197	0.9162	0.8598	0.8153

Table 4.10: Test AUC for different compression factors and injection types using the 3-layer 1D convolutional autoencoder.

Compression Factor	Constant	GWN	Step	Impulse
1	0.9568	0.9555	0.8976	0.8501
2	0.9743	0.9576	0.9073	0.8501
4	0.9712	0.9550	0.9050	0.8502
8	0.9580	0.9527	0.8982	0.8502
16	0.9559	0.9488	0.8973	0.8501
32	0.9554	0.9484	0.8973	0.8501
64	0.9517	0.9468	0.8963	0.8499
128	0.9511	0.9435	0.8945	0.8493
192	0.9471	0.9410	0.8926	0.8493
384	0.9367	0.9342	0.8862	0.8468

4.6.3 Robustness evaluation.

We evaluated the robustness of the best-performing 2-layer and 3-layer 1D convolutional autoencoders against adversarial attacks on the injections. Tables 4.11 and 4.12 report the main performance metrics on the validation and test sets, both derived from the Kaggle ESA training data.

For the 2-layer autoencoder with a compression factor of 8 (Tables 4.11

Table 4.11: AUC for Validation set obtained from the ESA Kaggle training data with a 2-layer autoencoder (compression factor 8) under different anomaly injections and adversarial attack settings.

Injection	Injection only	Random Attack	L2 Attack ($\beta = 20\%$)
Constant	0.9597	0.9594	0.9097
Step	0.8820	0.8817	0.8578
Impulse	0.8197	0.8197	0.8193
GWN	0.9427	0.9424	0.8941

Table 4.12: AUC for Test set obtained from the ESA Kaggle training data with a 2-layer autoencoder (compression factor 8) under different anomaly injections and adversarial attack settings.

Injection	Injection only	Random Attack	L2 Attack ($\beta = 20\%$)
Constant	0.9674	0.9672	0.9285
Step	0.9036	0.9034	0.8844
Impulse	0.8502	0.8502	0.8498
GWN	0.9544	0.9541	0.9167

and 4.12), the model achieves high AUC across both validation and test sets, demonstrating strong robustness in detecting injected anomalies. Constant and GWN injections achieve the highest AUC (0.959–0.967), indicating that sustained or stochastic anomalies are reliably detected, while Step and Impulse injections have lower AUC (0.819–0.904), showing that abrupt or localized anomalies are more challenging. Adversarial attacks moderately reduce performance. Random attacks result in minimal drops across all injection types, whereas L2 attacks reduce AUC more noticeably for Constant, Step, and GWN injections, while Impulse injections remain largely unaffected. Despite these reductions, the autoencoder continues to detect the majority of anomalies correctly, confirming its robustness under both normal anomaly injections and adversarially perturbed injections.

The 3-layer CONV_AE1D model with a compression factor of 2 (Tables 4.13

Table 4.13: AUC for Validation set obtained from the ESA Kaggle training data for the CONV_AE1D model with 3 layers and compression factor 2 under different anomaly injections and adversarial attack settings

Injection	Injection only	Random Attack	L2 Attack ($\beta = 20\%$)
Constant	0.9678	0.9674	0.9017
Step	0.8863	0.8860	0.8540
Impulse	0.8197	0.8197	0.8191
GWN	0.9465	0.9459	0.8775

Table 4.14: AUC for Test set obtained from the ESA Kaggle training data for the CONV_AE1D model with 3 layers and compression factor 2 under different anomaly injections and adversarial attack settings

Injection	Injection only	Random Attack	L2 Attack ($\beta = 20\%$)
Constant	0.9743	0.9740	0.9221
Step	0.9073	0.9070	0.8812
Impulse	0.8501	0.8501	0.8496
GWN	0.9576	0.9572	0.9035

and 4.14) achieves high AUC across both validation and test sets, indicating effective detection of injected anomalies. For the "Injection only" scenario, AUC is highest for Constant and GWN injections (0.946–0.974) and lower for Step and Impulse injections (0.819–0.907), suggesting that sustained or stochastic anomalies are reliably detected, while abrupt or localized anomalies are more challenging. Random adversarial attacks have a negligible impact on performance, with only minimal drops in AUC across all injection types. L2 attacks, at $\beta = 20\%$, reduce AUC notably for Constant, Step, and GWN injections, while Impulse injections remain largely unaffected. Despite these reductions, the autoencoder still detects the majority of anomalies correctly, demonstrating robustness to adversarial perturbations applied to injected anomalies. Comparing the 2-layer and 3-layer CONV_AE1D models reveals a trade-off between reconstruction quality

and robustness. The 3-layer model with compression factor 2 achieves higher AUC under Injection only conditions (0.946–0.974) than the 2-layer model with higher compression (0.819–0.967), indicating better reconstruction of normal patterns and stronger detection of injected anomalies. However, when adversarial attacks are applied to the injected anomalies, the 2-layer model shows slightly smaller decreases in AUC, especially under strong L2 attacks ($\beta = 20\%$), suggesting it is more stable under perturbations. The 3-layer model, while achieving higher baseline AUC, experiences larger drops for abrupt or localized anomalies (Step and Impulse) under adversarial perturbations. Overall, the 3-layer model is more accurate on clean injected anomalies, whereas the 2-layer model is relatively more robust to adversarial and injection-type perturbations, highlighting a trade-off between detection performance and robustness

Summary The experiments in this chapter cover three distinct evaluation settings. Table 4.15 separates the conclusions supported by each setting and indicates which rely on real ESA anomalies and which are based only on synthetic injections.

All experiments reported above use a six channel subset extracted from one of the three ESA ADB subsystems, specifically the subsystem that contains the highest number of labeled anomalies. As reported by Kotowski et al. [2024], performance on the full set of channels is substantially worse than on reduced channel subsets. The authors attribute this to the higher dimensionality, the presence of many subtle events, and an increased chance of false detections. Consequently, the quantitative results obtained on our six channel subset (e.g., AUC and optimal compression factors) do not necessarily transfer to the full benchmark. However, the qualitative conclusions, that autoencoders with a controlled latent space can detect Constant and GWN anomalies effectively, that Impulse anomalies

Table 4.15: Summary of conclusions per evaluation setting.

Evaluation setting	Supported conclusions
Real ESA anomalies	A 1D convolutional autoencoder with channel-wise thresholding can detect real spacecraft anomalies on a six-channel subsystem. The resulting conservative strategy achieves very high precision (0.999) and a public leaderboard score of 83% (67.5% on the private test set). This conclusion is supported by real annotated anomalies in ESA telemetry data.
Synthetic injected anomalies	Detection performance depends strongly on anomaly type. It is high for Constant and GWN (AUC 0.94–0.97), moderate for Step (0.88–0.91), and lower for Impulse (0.82–0.85). Introducing a controlled latent space (compression factor 2 or 8) improves performance for Constant and GWN compared to a model without bottleneck. These results are obtained from synthetic anomaly injections.
Adversarially perturbed injections	Adversarial perturbations moderately reduce AUC for Constant, Step, and GWN anomalies, while Impulse anomalies remain largely unaffected. The 2-layer autoencoder with compression factor 8 shows higher robustness than the 3-layer model with compression factor 2, indicating a trade-off between baseline accuracy and adversarial robustness. These conclusions are based on adversarially perturbed synthetic data.

remain difficult, and that adversarial attacks cause moderate degradation, are expected to hold in other settings as well, as they reflect fundamental properties of reconstruction based detectors. Future work should validate these findings directly on the full benchmark and across all three subsystems.

4.7 Conclusions

We evaluated 2-layer and 3-layer 1D convolutional autoencoders for anomaly detection on ESA sensor data, considering various anomaly injections and adversarial

attacks applied to these injections. The 3-layer model with a lower compression factor achieves the highest AUC for Constant and GWN injections, demonstrating strong reconstruction capabilities and reliable detection of sustained or stochastic anomalies. However, its performance decreases more under L2 adversarial attacks, particularly for Constant, Step, and GWN injections, indicating higher sensitivity to perturbations. The 2-layer model with a higher compression factor achieves slightly lower AUC on clean injected anomalies but maintains more consistent detection under L2 adversarial attacks, reflecting greater robustness. Both models reliably detect Constant and GWN injections, while Step and Impulse produce moderate reductions in detection performance. These results show that deeper models improve anomaly reconstruction and detection confidence but are more affected by adversarial perturbations, whereas shallower models are more stable under attacks but may miss subtle or localized anomalies. Future work will explore adversarial defenses, hybrid temporal or graph-based architectures, online adaptation strategies, and integration with FDIR techniques to further improve operational anomaly detection.

Chapter 5

Conclusions

This thesis investigated two complementary challenges related to robustness in machine learning. In the first part, we studied kernel ridge regression (KRR) when the training and test distributions differ, a situation commonly known as covariate shift. By leveraging random projection techniques, particularly the Nyström method, we showed that computational cost can be drastically reduced while maintaining predictive accuracy. New statistical bounds for the compressed Nyström estimator were derived, demonstrating that it preserves the optimal guarantees of the full weighted KRR solution even in the presence of distributional mismatches and potentially unbounded importance weights. Empirical evaluations on synthetic and real-world datasets confirmed that KRR can achieve both efficiency and robustness. In the second part, we focused on robust anomaly detection for multivariate time series in satellite telemetry. We used the *Controlled Latent Space Model* (CLSM), a semi-supervised 1D convolutional autoencoder that compresses temporal windows into a structured latent space. The CLSM learns normal operational patterns and detects anomalies without requiring labeled data. Its performance was evaluated on naturally occurring anomalies and under adversarial attacks applied on injected anomalies. The 3-layer CLSM with a compression factor of 2 achieves the highest AUC on injected anomalies, demon-

strating strong reconstruction and reliable detection of sustained or stochastic anomalies. Its performance decreases under adversarial attacks, particularly for abrupt or localized anomalies, indicating sensitivity to perturbations on injected anomalies. The 2-layer CLSM with a compression factor of 8 achieves slightly lower AUC on clean injected anomalies but maintains more consistent performance under adversarial attacks, reflecting moderate robustness. These results highlight a trade-off between model depth, compression factor, and robustness. Deeper models improve anomaly reconstruction and detection confidence but are more affected by perturbations, whereas shallower models are more stable under attacks on injected anomalies but may miss subtle or localized anomalies. The unifying theme of this work is robustness: in KRR, robustness ensures reliable predictions despite shifts in input distributions, while in anomaly detection, robustness guarantees stable detection under adversarial or challenging conditions. Future research directions include the integration of advanced adversarial defenses, hybrid temporal or graph-based models, online learning for evolving telemetry streams, and incorporation of FDIR (Fault Detection, Isolation, and Recovery) strategies to enhance operational resilience.

References

Alekh Agarwal, Sahand Negahban, and Martin J Wainwright. Stochastic optimization and sparse statistical recovery: Optimal algorithms for high dimensions. In Advances in Neural Information Processing Systems, pages 1538–1546, 2012.

Charu C. Aggarwal. Outlier Analysis. 2017.

Amr Akl and Hatem Elattar. Hybrid anomaly detection in spacecraft telemetry data using sparse feature-based methods and spatial-temporal generative adversarial networks. Journal of Physics: Conference Series, 3070:012021, 2025. doi: 10.1088/1742-6596/3070/1/012021.

Ahmed Alaoui and Michael W Mahoney. Fast randomized kernel ridge regression with statistical guarantees. In Advances in Neural Information Processing Systems, pages 775–783, 2015. 37

Jean-Yves Audibert and Alexandre B. Tsybakov. Fast learning rates for plug-in classifiers. The Annals of Statistics, 35(2):608–633, 2007.

Haim Avron, Vikas Sindhwani, and David Woodruff. Sketching structured matrices for faster nonlinear regression. In Advances in neural information processing systems, pages 2994–3002, 2013.

Haim Avron, Vikas Sindhwani, Jiyan Yang, and Michael W Mahoney. Quasi-monte

carlo feature maps for shift-invariant kernels. The Journal of Machine Learning Research, 17(1):4096–4133, 2016.

Francis Bach. Sharp analysis of low-rank kernel matrix approximations. In Conference on learning theory, pages 185–209. PMLR, 2013. 31

Francis Bach. On the equivalence between kernel quadrature rules and random feature expansions. The Journal of Machine Learning Research, 18(1):714–751, 2017. 31

Francis Bach. Learning theory from first principles. Draft of a book, version of Sept, 6:2021, 2021.

Francis Bach. Learning theory from first principles. MIT press, 2024. 15

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. Journal of Machine Learning Research, 3 (Nov):463–482, 2002.

Peter L Bartlett, Olivier Bousquet, Shahar Mendelson, et al. Local rademacher complexities. The Annals of Statistics, 33(4):1497–1537, 2005.

Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, classification, and risk bounds. Journal of the American Statistical Association, 101 (473):138–156, 2006.

Frank Bauer, Sergei Pereverzev, and Lorenzo Rosasco. On regularization algorithms in learning theory. Journal of complexity, 23(1):52–72, 2007.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. Advances in neural information processing systems, 19, 2006. 29

Shai Ben-David et al. A theory of learning from different domains. Machine Learning, 2010.

Adi Ben-Israel and Thomas NE Greville. Generalized inverses: theory and applications. Springer Science & Business Media, 2006. 127

John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boomboxes: Cross-domain sentiment classification. In ACL, 2007.

Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In Advances in neural information processing systems, pages 161–168, 2008. 31

Stéphane Boucheron, Olivier Bousquet, and Gábor Lugosi. Theory of classification: A survey of some recent advances. ESAIM: probability and statistics, 9:323–375, 2005.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. Concentration Inequalities: A Nonasymptotic Theory of Independence. Oxford University Press, Oxford, 2013. 128

Olivier Bousquet and André Elisseeff. Stability and generalization. Journal of machine learning research, 2(Mar):499–526, 2002.

Stephen Boyd and Lieven Vandenberghe. Convex Optimization. Cambridge University Press, 2004.

Federico Buccellato, Davide Nicolini, Eleonora Vacca, Corrado De Sio, and Luca Sterpone. Ai-powered anomaly detection for satellite telemetry. In CF '25: 22nd ACM International Conference on Computing Frontiers, pages 222–223, 2025. doi: 10.1145/3719276.3727953.

Jonathon Byrd and Zachary Lipton. What is the effect of importance weighting in deep learning? In International conference on machine learning, pages 872–881. PMLR, 2019.

Daniele Calandriello and Lorenzo Rosasco. Statistical and computational trade-offs in kernel k-means. In Advances in Neural Information Processing Systems, pages 9357–9367, 2018.

Daniele Calandriello, Alessandro Lazaric, and Michal Valko. Distributed adaptive sampling for kernel matrix approximation. In Artificial Intelligence and Statistics, pages 1421–1429. PMLR, 2017. 31

Andrea Caponnetto and Ernesto De Vito. Optimal rates for the regularized least-squares algorithm. Foundations of Computational Mathematics, 7:331–368, 2007. 21, 33, 40, 41, 46, 128, 133

Andrea Caponnetto and Yuan Yao. Adaptive rates for regularization operators in learning theory. Analysis and Applications, 8, 2010.

Andrea Caponnetto, Lorenzo Rosasco, Ernesto De Vito, and Alessandro Verri. Empirical effective dimension and optimal rates for regularized least squares algorithm. 2005.

Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. ACM Computing Surveys, 2019.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. ACM Computing Surveys, 41(3):15, 2009. doi: 10.1145/1541880.1541882. 2, 56

Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector

machines. ACM transactions on intelligent systems and technology (TIST), 2 (3):1–27, 2011.

Veronika Cheplygina, Marleen de Bruijne, and Josien Pluim. Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. Medical Image Analysis, 2019.

Michael B Cohen, Yin Tat Lee, Cameron Musco, Christopher Musco, Richard Peng, and Aaron Sidford. Uniform sampling for matrix approximation. In Proceedings of the 2015 conference on innovations in theoretical computer science, pages 181–190, 2015. 37

Corinna Cortes and Mehryar Mohri. Domain adaptation and sample bias correction theory and algorithm for regression. Theoretical Computer Science, 519:103–126, 2014. 29

Corinna Cortes, Mehryar Mohri, Michael Riley, and Afshin Rostamizadeh. Sample selection bias correction theory. In International conference on algorithmic learning theory, pages 38–53. Springer, 2008. 29

Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning bounds for importance weighting. Advances in neural information processing systems, 23, 2010a. 1, 30, 39

Corinna Cortes, Mehryar Mohri, and Ameet Talwalkar. On the impact of kernel approximation on learning accuracy. In Proceedings of the thirteenth international conference on artificial intelligence and statistics, pages 113–120. JMLR Workshop and Conference Proceedings, 2010b.

Felipe Cucker and Steve Smale. On the mathematical foundations of learning. Bulletin of the American mathematical society, 39(1):1–49, 2002. 14

Bo Dai, Bo Xie, Niao He, Yingyu Liang, Anant Raj, Maria-Florina F Balcan, and Le Song. Scalable kernel methods via doubly stochastic gradients. Advances in neural information processing systems, 27:3041–3049, 2014.

Ernesto De Vito, Andrea Caponnetto, and Lorenzo Rosasco. Model selection for regularized least-squares algorithm in learning theory. Foundations of Computational Mathematics, 5:59–85, 2005a.

Ernesto De Vito, Lorenzo Rosasco, Andrea Caponnetto, Umberto De Giovannini, Francesca Odone, and Peter Bartlett. Learning from examples as an inverse problem. Journal of Machine Learning Research, 6(5), 2005b.

Ernesto De Vito, Lorenzo Rosasco, and Andrea Caponnetto. Discretization error analysis for tikhonov regularization. Analysis and Applications, 4(01):81–99, 2006.

Ernesto De Vito, Lorenzo Rosasco, and Alessandro Rudi. Regularization: From inverse problems to large-scale machine learning. Harmonic and Applied Analysis: From Radon Transforms to Machine Learning, pages 245–296, 2021. 36

Andrea Della Vecchia, Jaouad Mourtada, Ernesto De Vito, and Lorenzo Rosasco. Regularized erm on random subspaces. In International Conference on Artificial Intelligence and Statistics, pages 4006–4014. PMLR, 2021. 31, 133

Andrea Della Vecchia, Ernesto De Vito, Jaouad Mourtada, and Lorenzo Rosasco. The nyström method for convex loss functions. Journal of Machine Learning Research, 25(360):1–60, 2024. 31

Andrea Della Vecchia, Arnaud Mavakala Watusadisi, Ernesto De Vito, and Lorenzo Rosasco. Computational efficiency under covariate shift in kernel ridge regression. arXiv preprint arXiv:2505.14083, 2025. 4, 32

Luc Devroye, László Györfi, and Gábor Lugosi. A probabilistic theory of pattern recognition, volume 31. Springer Science & Business Media, 2013.

Marius-Constantin Dinu, Markus Holzleitner, Maximilian Beck, Hoan Duc Nguyen, Andrea Huber, Hamid Eghbal-zadeh, Bernhard A. Moser, Sergei Pereverzyev, Sepp Hochreiter, and Werner Zellinger. Addressing parameter choice issues in unsupervised domain adaptation by aggregation. In The Eleventh International Conference on Learning Representations, 2023. URL https://openreview.net/forum?id=M95oDwJXayG. 51

Petros Drineas and Michael W Mahoney. Approximating a gram matrix for improved kernel-based learning. In International Conference on Computational Learning Theory, pages 323–337. Springer, 2005a.

Petros Drineas and Michael W Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. journal of machine learning research, 6(Dec):2153–2175, 2005b. 31

Petros Drineas, Malik Magdon-Ismail, Michael W Mahoney, and David P Woodruff. Fast approximation of matrix coherence and statistical leverage. The Journal of Machine Learning Research, 13(1):3475–3506, 2012. 37

Ahmed El Alaoui and Michael W Mahoney. Fast randomized kernel methods with statistical guarantees. stat, 1050:2, 2014.

Mohamed El Amine Sehili and Zonghua Zhang. Multivariate time series anomaly detection: Fancy algorithms and flawed evaluation methodology. In Technology Conference on Performance Evaluation and Benchmarking, pages 1–17. Springer, 2023. 84

Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. Regularization of inverse problems, volume 375. Springer Science & Business Media, 1996.

Andriy Enttsel, Silvia Onofri, Alex Marchioni, Mauro Mangia, Gianluca Setti, and Riccardo Rovatti. A general framework for the assessment of detectors of anomalies in time series. IEEE Transactions on Industrial Informatics, pages 1–11, 2024. doi: 10.1109/TII.2024.3413359. 57, 64

European Space Agency, Krzysztof Kotowski, Bogdan Ruszczak, et al. Esa anomaly detection benchmark (telemetry anomaly dataset), 2024. URL https://doi.org/10.5281/zenodo.12528695.

Tongtong Fang, Nan Lu, Gang Niu, and Masashi Sugiyama. Rethinking importance weighting for deep learning under distribution shift. Advances in neural information processing systems, 33:11996–12007, 2020. 30

Asma Fejjari, Alexis Delavault, Robert Camilleri, and Gianluca Valentino. A review of anomaly detection in spacecraft telemetry data. Applied Sciences, 15 (10):5653, 2025. doi: 10.3390/app15105653.

Vitaly Feldman, Venkatesan Guruswami, Prasad Raghavendra, and Yi Wu. Agnostic learning of monomials by halfspaces is hard. SIAM Journal on Computing, 41(6):1558–1590, 2012. 11

Samuel G. Finlayson et al. Adversarial attacks on medical machine learning. Science, 2019. 2, 57

Junichi Fujii, Masatoshi Fujii, Takayuki Furuta, and Ritsuo Nakamoto. Norm inequalities equivalent to heinz inequality. Proceedings of the American Mathematical Society, 118(3):827–830, 1993. 135

João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. ACM computing surveys (CSUR), 46(4):1–37, 2014.

Saurabh Garg, Yifan Wu, Sivaraman Balakrishnan, and Zachary Lipton. A unified view of label shift estimation. Advances in Neural Information Processing Systems, 33:3290–3300, 2020.

Jiawei Ge, Shange Tang, Jianqing Fan, Cong Ma, and Chi Jin. Maximum likelihood estimation is all you need for well-specified covariate shift. arXiv preprint arXiv:2311.15961, 2023.

Evarist Giné and Joel Zinn. Some limit theorems for empirical processes. The Annals of Probability, pages 929–989, 1984.

Alex Gittens and Michael W Mahoney. Revisiting the nyström method for improved large-scale machine learning. The Journal of Machine Learning Research, 17(1):3977–4041, 2016.

Elke R Gizewski, Lukas Mayer, Bernhard A Moser, Duc Hoan Nguyen, Sergiy Pereverzyev Jr, Sergei V Pereverzyev, Natalia Shepeleva, and Werner Zellinger. On a regularization of unsupervised domain adaptation in rkhs. Applied and Computational Harmonic Analysis, 57:201–227, 2022. 30, 31

Julian Gizewski et al. Kernel ridge regression under covariate shift. Journal of Machine Learning Research, 2020. 1

Davit Gogolashvili, Matteo Zecchin, Motonobu Kanagawa, Marios Kountouris, and Maurizio Filippone. When is importance weighting correction needed for covariate shift adaptation? arXiv preprint arXiv:2303.04020, 2023. 3, 30, 32, 39, 42, 43, 46, 48, 50, 127, 128, 132, 134

Gene H Golub and Charles F Van Loan. Matrix computations. JHU press, 2013.

Ian Goodfellow et al. Explaining and harnessing adversarial examples. ICLR, 2015. 57

Adam Goodge, Bryan Hooi, See Kiong Ng, and Wee Siong Ng. Robustness of autoencoders for anomaly detection under adversarial impact. In Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence, pages 1244–1250, 2021. 57, 74

Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. Covariate shift by kernel mean matching. 2008.

Arthur Gretton et al. A kernel two-sample test. Journal of Machine Learning Research, 2012.

Hao Guan and Mingxia Liu. Domain adaptation for medical image analysis: a survey. IEEE Transactions on Biomedical Engineering, 69(3):1173–1185, 2021. 29

Frank Hansen. An operator inequality. Mathematische Annalen, 246:249–250, 1979. URL http://eudml.org/doc/163343.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. The elements of statistical learning: data mining, inference, and prediction, volume 2. Springer, 2009.

Douglas M. Hawkins. Identification of outliers. Chapman and Hall, 1980. 2, 56

Huan He, Owen Queen, Teddy Koker, Consuelo Cuevas, Theodoros Tsiligkaridis, and Marinka Zitnik. Domain adaptation for time series under feature and label shifts. In International Conference on Machine Learning, pages 12746–12774. PMLR, 2023. 51

James J Heckman. Sample selection bias as a specification error. Econometrica: Journal of the econometric society, pages 153–161, 1979.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. arXiv preprint arXiv:1903.12261, 2019.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In Proceedings of the IEEE/CVF international conference on computer vision, pages 8340–8349, 2021.

Cho-Jui Hsieh, Si Si, and Inderjit S Dhillon. Fast prediction for large-scale kernel machines. In Advances in Neural Information Processing Systems, pages 3689–3697, 2014.

Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex Smola. Correcting sample selection bias by unlabeled data. Advances in neural information processing systems, 19, 2006. 30

Kyle Hundman et al. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In KDD, 2018.

Jing Jiang and ChengXiang Zhai. Instance weighting for domain adaptation in nlp. ACL, 2007. 29

Rong Jin, Tianbao Yang, Mehrdad Mahdavi, Yu-Feng Li, and Zhi-Hua Zhou. Improved bounds for the nyström method with application to kernel classification. IEEE Transactions on Information Theory, 59(10):6939–6949, 2013.

Thorsten Joachims. Making large-scale svm learning practical. Technical report, Technical Report, 1998.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In Advances in neural information processing systems, pages 315–323, 2013. 31

Cijo Jose, Prasoon Goyal, Parv Aggrwal, and Manik Varma. Local deep kernel learning for efficient non-linear svm prediction. In International conference on machine learning, pages 486–494, 2013.

Kaggle. Spacecraft anomaly detection challenge on esa dataset. https://www.kaggle.com/competitions/esa-adb-challenge/overview, 2026. Accessed: 2026-02-15. ix, 58, 59, 60

Sham M. Kakade, Karthik Sridharan, and Ambuj Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In Advances in Neural Information Processing Systems 21, pages 793–800, 2009.

Takafumi Kanamori and Hidetoshi Shimodaira. Active learning algorithm using the maximum weighted log-likelihood estimator. Journal of statistical planning and inference, 116(1):149–162, 2003. 29

Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. A least-squares approach to direct importance estimation. The Journal of Machine Learning Research, 10:1391–1445, 2009.

Takafumi Kanamori, Taiji Suzuki, and Masashi Sugiyama. Statistical analysis of kernel-based least-squares density-ratio estimation. Machine Learning, 86: 335–367, 2012.

George Kimeldorf and Grace Wahba. Some results on tchebycheffian spline functions. Journal of mathematical analysis and applications, 33(1):82–95, 1971. 19

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In International conference on machine learning, pages 5637–5664. PMLR, 2021. 29

Vladimir Koltchinskii. Oracle inequalities in empirical risk minimization and sparse recovery problems: École D'Été de Probabilités de Saint-Flour XXXVIII-2008, volume 2033 of École d'Été de Probabilités de Saint-Flour. Springer Science & Business Media, 2011.

Vladimir Koltchinskii and Karim Lounici. Concentration inequalities and moment bounds for sample covariance operators. arXiv preprint arXiv:1405.2468, 2014.

Vladimir Koltchinskii et al. Local rademacher complexities and oracle inequalities in risk minimization. The Annals of Statistics, 34(6):2593–2656, 2006.

Krzysztof Kotowski, Christoph Haskamp, Jacek Andrzejewski, Bogdan Ruszczak, Jakub Nalepa, Daniel Lakey, Peter Collins, Aybike Kolmas, Mauro Bartesaghi, Jose Martinez-Heras, et al. European space agency benchmark for anomaly detection in satellite telemetry. arXiv preprint arXiv:2406.17826, 2024. 1, 2, 4, 56, 57, 58, 64, 84, 93

Samory Kpotufe and Guillaume Martinet. Marginal singularity and the benefits of labels in covariate-shift. The Annals of Statistics, 49(6):3299–3323, 2021. 30

Samory Kpotufe and Bharath K Sriperumbudur. Kernel sketching yields kernel jl. arXiv preprint arXiv:1908.05818, 2019. 31

Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Ensemble nystrom method. Advances in Neural Information Processing Systems, 22:1060–1068, 2009.

Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Sampling methods for the nyström method. The Journal of Machine Learning Research, 13(1):981–1006, 2012.

Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. Activity recognition using cell phone accelerometers. ACM SigKDD Explorations Newsletter, 12(2): 74–82, 2011. 51, 137

Quoc Le, Tamás Sarlós, and Alexander Smola. Fastfood-computing hilbert space expansions in loglinear time. In International Conference on Machine Learning, volume 85, pages 244–252. PMLR, 2013.

Michel Ledoux and Michel Talagrand. Probability in Banach Spaces: isoperimetry and processes, volume 23. Springer Science & Business Media, 1991.

Qi Lei, Wei Hu, and Jason Lee. Near-optimal linear regression under distribution shift. In International Conference on Machine Learning, pages 6164–6174. PMLR, 2021. 30

Mu Li, James Tin-Yau Kwok, and Baoliang Lü. Making large-scale nyström approximation possible. In Proceedings of the 27th International Conference on Machine Learning, ICML 2010, page 631, 2010.

Zhe Li, Tianbao Yang, Lijun Zhang, and Rong Jin. Fast and accurate refined nyström-based kernel svm. In Thirtieth AAAI Conference on Artificial Intelligence, 2016.

Zhu Li, Jean-Francois Ton, Dino Oglic, and Dino Sejdinovic. Towards a unified analysis of random fourier features. In International Conference on Machine Learning, pages 3905–3914. PMLR, 2019.

Junhong Lin, Alessandro Rudi, Lorenzo Rosasco, and Volkan Cevher. Optimal rates for spectral algorithms with least-squares regression over hilbert spaces. Applied and Computational Harmonic Analysis, 48(3):868–890, 2020.

Qihang Lin, Zhaosong Lu, and Lin Xiao. An accelerated proximal coordinate gradient method. In Advances in Neural Information Processing Systems, pages 3059–3067, 2014.

Liang Liu, Ling Tian, Zhao Kang, and Tianqi Wan. Spacecraft anomaly detection with attention temporal convolution network. arXiv preprint arXiv:2303.06879, 2023.

Song Liu, Makoto Yamada, Nigel Collier, and Masashi Sugiyama. Change-point detection in time-series data by relative density-ratio estimation. Neural Networks, 43:72–83, 2013. 51, 135

Aleksej Logacjov, Kerstin Bach, Atle Kongsvold, Hilde Bremseth Bårdstu, and Paul Jarle Mork. Harth: a human activity recognition dataset for machine learning. Sensors, 21(23):7853, 2021.

Cong Ma, Reese Pathak, and Martin J Wainwright. Optimally tackling covariate shift in rkhs-based nonparametric regression. The Annals of Statistics, 51(2): 738–761, 2023. 3, 30, 33, 40, 46, 48

Michael W Mahoney. Randomized algorithms for matrices and data. Foundations and Trends® in Machine Learning, 3(2):123–224, 2011. 31

Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. Lstm-based encoder-decoder for multi-sensor anomaly detection. arXiv preprint arXiv:1607.00148, 2016. URL https://arxiv.org/abs/1607.00148.

Enno Mammen and Alexandre B. Tsybakov. Smooth discrimination analysis. The Annals of Statistics, 27(6):1808–1829, 1999. ISSN 00905364. URL http://www.jstor.org/stable/2673938.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. arXiv preprint arXiv:0902.3430, 2009. 29, 39

Ulysse Marteau-Ferey, Francis Bach, and Alessandro Rudi. Globally convergent newton methods for ill-conditioned generalized self-concordant losses. arXiv preprint arXiv:1907.01771, 2019a.

Ulysse Marteau-Ferey, Dmitrii Ostrovskii, Francis Bach, and Alessandro Rudi. Beyond least-squares: Fast rates for regularized empirical risk minimization through self-concordance. In Conference on Learning Theory, pages 2294–2340. PMLR, 2019b. 31

Pascal Massart, Élodie Nédélec, et al. Risk bounds for statistical learning. The Annals of Statistics, 34(5):2326–2366, 2006.

Giacomo Meanti, Luigi Carratino, Lorenzo Rosasco, and Alessandro Rudi. Kernel methods through the roof: handling billions of points efficiently. Advances in Neural Information Processing Systems, 33:14410–14422, 2020. 39

Ron Meir and Tong Zhang. Generalization error bounds for bayesian mixture algorithms. Journal of Machine Learning Research, 4(Oct):839–860, 2003.

Shahar Mendelson and Joseph Neeman. Regularization in kernel learning. 2010.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. Foundations of machine learning. MIT press, 2018. 15

Sofia Mosci, Lorenzo Rosasco, and Alessandro Verri. Dimensionality reduction and generalization. In Proceedings of the 24th international conference on Machine learning, pages 657–664, 2007.

Jaouad Mourtada and Lorenzo Rosasco. An elementary analysis of ridge regression with random design. Comptes Rendus. Mathématique, 360(G9):1055–1063, 2022.

Nicole Mücke, Gergely Neu, and Lorenzo Rosasco. Beating sgd saturation with tail-averaging and minibatching. In Advances in Neural Information Processing Systems, pages 12568–12577, 2019.

Cameron Musco and Christopher Musco. Recursive sampling for the nyström method. In Advances in Neural Information Processing Systems, 2017.

Hanna L Myleiko and Sergei G Solodky. Regularized nyström subsampling in covariate shift domain adaptation problems. Numerical Functional Analysis and Optimization, 45(3):165–188, 2024. 3, 31, 46

Yurii Nesterov. Lectures on convex optimization, volume 137. Springer, 2018.

Francesco Orabona. Simultaneous model selection and optimization through parameter-free stochastic learning. In Advances in Neural Information Processing Systems, pages 1116–1124, 2014.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 2010.

Guansong Pang et al. Robust anomaly detection: A survey. ACM Computing Surveys, 2022.

Reese Pathak, Cong Ma, and Martin Wainwright. A new similarity measure for covariate shift with applications to nonparametric regression. In International Conference on Machine Learning, pages 17517–17530. PMLR, 2022. 30

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. the Journal of machine Learning research, 12:2825–2830, 2011.

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. Dataset shift in machine learning. Mit Press, 2022. 29

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. Advances in neural information processing systems, 20, 2007.

Alan Ramponi and Barbara Plank. Neural unsupervised domain adaptation in nlp—a survey. arXiv preprint arXiv:2006.00632, 2020.

R Tyrrell Rockafellar. Convex analysis. Number 28. Princeton university press, 1970.

Lorenzo Rosasco, Ernesto De Vito, Andrea Caponnetto, Michele Piana, and Alessandro Verri. Are loss functions all the same? Neural computation, 16(5): 1063–1076, 2004.

Lorenzo Rosasco, Ernesto De Vito, and Alessandro Verri. Spectral methods for regularization in learning theory. DISI, Universita degli Studi di Genova, Italy, Technical Report DISI-TR-05-18, 2005.

Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. In Advances in Neural Information Processing Systems 30, pages 3215–3225, 2017. 23, 24, 31

Alessandro Rudi, Guillermo D Canas, and Lorenzo Rosasco. On the sample complexity of subspace learning. Advances in Neural Information Processing Systems, 26, 2013.

Alessandro Rudi, Raffaello Camoriano, and Lorenzo Rosasco. Less is more: Nyström computational regularization. arXiv preprint arXiv:1507.04717, 28, 2015. 3, 26, 31, 32, 40, 42, 43, 44, 127, 128, 130, 131

Alessandro Rudi, Luigi Carratino, and Lorenzo Rosasco. Falkon: An optimal large scale kernel method. In Advances in Neural Information Processing Systems, pages 3888–3898, 2017. 39

Alessandro Rudi, Daniele Calandriello, Luigi Carratino, and Lorenzo Rosasco. On fast leverage score sampling and optimal learning. In Advances in Neural Information Processing Systems, pages 5672–5682, 2018. 52

Bogdan Ruszczak, Krzysztof Kotowski, David Evans, and Jakub Nalepa. The ops-sat benchmark for detecting anomalies in satellite telemetry. arXiv preprint arXiv:2407.04730, 2024. 1, 2, 4, 56

Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis, pages 4–11, 2014. doi: 10.1145/2689746.2689747. 2, 56

Mark J Schervish and Morris H DeGroot. Probability and statistics, volume 563. Pearson Education London, UK:, 2014.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. Mathematical Programming, 162(1-2):83–112, 2017. 31

Johannes Schmidt-Hieber and Petr Zamolodtchikov. Local convergence rates of the nonparametric least squares estimator with applications to transfer learning. Bernoulli, 30(3):1845–1877, 2024. 30

Bernhard Schölkopf and Alexander J Smola. Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press, 2002.

Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. A generalized representer theorem. In International conference on computational learning theory, pages 416–426. Springer, 2001. 20

Shai Shalev-Shwartz and Shai Ben-David. Understanding machine learning: From theory to algorithms. Cambridge university press, 2014.

Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. Journal of Machine Learning Research, 14 (Feb):567–599, 2013.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. Journal of Machine Learning Research, 11(Oct):2635–2670, 2010.

Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal estimated sub-gradient solver for svm. Mathematical programming, 127 (1):3–30, 2011.

John Shawe-Taylor and Nello Cristianini. Kernel Methods for Pattern Analysis. Cambridge University Press, 2004.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. Journal of statistical planning and inference, 90(2):227–244, 2000. 1, 29, 30

Si Si, Cho-Jui Hsieh, and Inderjit S Dhillon. Memory efficient kernel approximation. Journal of Machine Learning Research, 18(20):1–32, 2017.

Steve Smale and Ding-Xuan Zhou. Learning theory estimates via integral operators and their approximations. Constructive approximation, 26(2):153–172, 2007. 33, 40

Alex J Smola and Bernhard Schökopf. Sparse greedy matrix approximation for machine learning. In Proceedings of the seventeenth international conference on machine learning, pages 911–918, 2000. 31

Yang Song et al. Adversarial attacks and defenses for anomaly detection. NeurIPS, 2018. 2, 57

Karthik Sridharan, Shai Shalev-Shwartz, and Nathan Srebro. Fast rates for regularized objectives. In Advances in Neural Information Processing Systems 21, pages 1545–1552, 2009.

Ingo Steinwart and Andreas Christmann. Support vector machines. Springer Science & Business Media, 2008. 33

Ingo Steinwart, Don R Hush, Clint Scovel, et al. Optimal rates for regularized least squares regression. In COLT, pages 79–93, 2009.

Nicholas Sterge, Bharath Sriperumbudur, Lorenzo Rosasco, and Alessandro Rudi. Gain with no pain: Efficient kernel-pca by nyström sampling. arXiv preprint arXiv:1907.05226, 2019. 23

Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. Smart devices are different: Assessing and mitigatingmobile sensing hetero-

geneities for activity recognition. In Proceedings of the 13th ACM conference on embedded networked sensor systems, pages 127–140, 2015. 51, 135

Masashi Sugiyama and Motoaki Kawanabe. Machine learning in non-stationary environments: Introduction to covariate shift adaptation. MIT press, 2012. 29

Masashi Sugiyama and Greg Ridgeway. Active learning in approximately linear regression based on conditional expectation of generalization error. Journal of Machine Learning Research, 7(1), 2006. 29

Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul Buenau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. Advances in neural information processing systems, 20, 2007.

Masashi Sugiyama, Taiji Suzuki, Shinichi Nakajima, Hisashi Kashima, Paul Von Bünau, and Motoaki Kawanabe. Direct importance estimation for covariate shift adaptation. Annals of the Institute of Statistical Mathematics, 60:699–746, 2008.

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. Density ratio estimation in machine learning. Cambridge University Press, 2012. 30

Yitong Sun, Anna Gilbert, and Ambuj Tewari. But how does it work in theory? linear svm with random features. In Advances in Neural Information Processing Systems, pages 3379–3388, 2018. 31

Robert Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society Series B: Statistical Methodology, 58(1):267–288, 1996. 11

Marte Fossflaten Tørring, Aleksej Logacjov, Siri Merete Brændvik, Astrid Ustad, Karin Roeleveld, and Ellen Marie Bardal. Validation of two novel human activity recognition models for typically developing children and children with cerebral palsy. Plos one, 19(9):e0308853, 2024. 51, 137

Nilesh Tripuraneni, Ben Adlam, and Jeffrey Pennington. Overparameterization improves robustness to covariate shift in high dimensions. Advances in Neural Information Processing Systems, 34:13883–13897, 2021.

Joel A Tropp. User-friendly tail bounds for sums of random matrices. Foundations of computational mathematics, 12(4):389–434, 2012.

Joel A. Tropp. An introduction to matrix concentration inequalities. Foundations and Trends® in Machine Learning, 8(1-2):1–230, 2015.

Yuta Tsuboi, Hisashi Kashima, Shohei Hido, Steffen Bickel, and Masashi Sugiyama. Direct density ratio estimation for large-scale covariate shift adaptation. Journal of Information Processing, 17:138–155, 2009.

Alexandre B. Tsybakov. Optimal aggregation of classifiers in statistical learning. The Annals of Statistics, 32(1):135–166, 2004.

Astrid Ustad, Aleksej Logacjov, Stine Øverengen Trollebø, Pernille Thingstad, Beatrix Vereijken, Kerstin Bach, and Nina Skjæret Maroni. Validation of an activity type recognition model classifying daily physical behavior in older adults: the har70+ model. Sensors, 23(5):2368, 2023. 51, 136

Vladimir Vapnik. Statistical Learning Theory. Wiley, 1998. 1

Vladimir Vapnik. The Nature of Statistical Learning Theory. Springer Science & Business Media, 1999. 29

Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. arXiv preprint arXiv:1011.3027, 2010.

Ernesto De Vito and Andrea Caponnetto. Risk bounds for regularized least-squares algorithm with operator-valued kernels. 2005.

Ernesto De Vito, Lorenzo Rosasco, Andrea Caponnetto, Umberto De Giovannini, and Francesca Odone. Learning from examples as an inverse problem. Journal of Machine Learning Research, 6(May):883–904, 2005.

Grace Wahba. Spline models for observational data, volume 59. Siam, 1990.

Kaizheng Wang. Pseudo-labeling for kernel ridge regression under covariate shift. arXiv preprint arXiv:2302.10160, 2023.

Ke Alexander Wang, Niladri S Chatterji, Saminul Haque, and Tatsunori Hashimoto. Is importance weighting incompatible with interpolating classifiers? arXiv preprint arXiv:2112.12986, 2021.

Shusen Wang and Zhihua Zhang. Improving cur matrix decomposition and the nyström approximation via adaptive sampling. The Journal of Machine Learning Research, 14(1):2729–2769, 2013.

Shusen Wang and Zhihua Zhang. Efficient algorithms and error analysis for the modified nystrom method. In Artificial intelligence and statistics, pages 996–1004. PMLR, 2014.

Xinli Wang and Rencheng Sun. Domain adaptation of time series classification. IEEE Access, 2024. 51

Junfeng Wen, Chun-Nam Yu, and Russell Greiner. Robust learning under uncertain test distributions: Relating covariate shift to model misspecification. In

International Conference on Machine Learning, pages 631–639. PMLR, 2014. 30

Halbert White. Consequences and detection of misspecified nonlinear regression models. Journal of the American Statistical Association, 76(374):419–433, 1981.

Douglas P Wiens. Robust weights and designs for biased regression models: Least squares and generalized m-estimation. Journal of Statistical Planning and Inference, 83(2):395–412, 2000. 29

Christopher Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. Advances in neural information processing systems, 13, 2000. 31, 37

Garrett Wilson, Janardhan Rao Doppa, and Diane J Cook. Multi-source deep domain adaptation with weak supervision for time-series sensor data. In Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pages 1768–1778, 2020. 51

David P Woodruff. Sketching as a tool for numerical linear algebra. arXiv preprint arXiv:1411.4357, 2014. 31

Da Xu, Yuting Ye, and Chuanwei Ruan. Understanding the role of importance weighting for deep learning. arXiv preprint arXiv:2103.15209, 2021.

Makoto Yamada, Taiji Suzuki, Takafumi Kanamori, Hirotaka Hachiya, and Masashi Sugiyama. Relative density-ratio estimation for robust distribution comparison. Neural computation, 25(5):1324–1370, 2013. 51, 135

Keisuke Yamazaki, Motoaki Kawanabe, Sumio Watanabe, Masashi Sugiyama, and Klaus-Robert Müller. Asymptotic bayesian generalization error when training

and test distributions are different. In Proceedings of the 24th international conference on Machine learning, pages 1079–1086, 2007. 30

Jiyan Yang, Vikas Sindhwani, Haim Avron, and Michael Mahoney. Quasi-monte carlo feature maps for shift-invariant kernels. In International Conference on Machine Learning, pages 485–493. PMLR, 2014.

Tianbao Yang, Yu-Feng Li, Mehrdad Mahdavi, Rong Jin, and Zhi-Hua Zhou. Nyström method vs random fourier features: A theoretical and empirical comparison. Advances in neural information processing systems, 25:476–484, 2012.

Yiming Ying and Massimiliano Pontil. Online gradient descent learning algorithms. Foundations of Computational Mathematics, 8:561–596, 2008.

Runtian Zhai, Chen Dan, Zico Kolter, and Pradeep Ravikumar. Understanding why generalized reweighting does not improve over erm. arXiv preprint arXiv:2201.12293, 2022.

Chao Zhang, Lei Zhang, and Jieping Ye. Generalization bounds for domain adaptation. Advances in neural information processing systems, 25, 2012. 29

Kai Zhang, Ivor W Tsang, and James T Kwok. Improved nyström low-rank approximation and error analysis. In Proceedings of the 25th international conference on Machine learning, pages 1232–1239, 2008.

Tong Zhang. Learning bounds for kernel regression using effective data dimensionality. Neural Computation, 17(9):2077–2098, 2005. 41

Y. Zhang et al. Stage training denoising autoencoder for anomaly detection in satellite telemetry. IEEE Access, 2019.

Yuchen Zhang, John Duchi, and Martin Wainwright. Divide and conquer kernel ridge regression. In Conference on learning theory, pages 592–617. PMLR, 2013.

Taichun Zhou, Guohang Guo, Zhigang Xiao, and Hu Li. Anomaly detection of satellite telemetry data based on latent space interpolation autoencoder. Chinese Journal of Space Science, 44(6):1155–1165, 2024. doi: 10.11728/cjss2024.06.2023-0147.

Appendix A

Appendix

A.1 Derivation of the estimators

We define some quantities we will need when deriving our estimators. Following Section 3.2, we define the sampling operator $\widehat{Z}_m : \mathcal{H} \to \mathbb{R}^m$ associated with the subset $\{\widetilde{x}_1, \ldots, \widetilde{x}_m\} \subset \{x_1, \ldots, x_n\}$ as

$$\widehat{Z}_m : \mathcal{H}_m \to \mathbb{R}^m, \quad (\widehat{Z}_m f) = \langle f, K_{\widetilde{x}_i} \rangle_{\mathcal{H}_m}, \tag{A.1}$$

with its adjoint

$$\widehat{Z}_m^* : \mathbb{R}^m \to \mathcal{H}_m, \quad \widehat{Z}_m^* \widetilde{c} = \sum_{i=1}^{m} \widetilde{c}_i K_{\widetilde{x}_i}, \ \forall \widetilde{c} \in \mathbb{R}^m. \tag{A.2}$$

Moreover, consider the singular value decomposition (SVD) of $\widehat{Z}_m$ and $\widehat{Z}_m^*$

$$\widehat{Z}_m = UDV^*, \qquad \widehat{Z}_m^* = VDU^*$$

with $U : \mathbb{R}^k \to \mathbb{R}^m$, $D : \mathbb{R}^k \to \mathbb{R}^k$ the diagonal matrix of singular values sorted in non-decreasing order $D = \text{diag}(\sigma_1, ..., \sigma_t)$ with $\sigma_1 \geqslant ... \geqslant \sigma_k > 0$, $V : \mathbb{R}^k \to \widetilde{\mathcal{H}}_m$, $k \leqslant m$ such that $U^*U = I_k$ and $V^*V = I_k$. The projection operator with range $\mathcal{H}_m$ is given by $P_m = VV^*$. We used this fact in Section 3.4. We derive an expression for the KRR minimizer when considering IW correction and Nyström

approximation.

Lemma 1 (Nyström W-KRR estimator)**.** *Given the minimization problem in (3.12), the unique minimizer can be written as*

$$\widehat{f}^{w}_{\lambda,m} = V(V^*\widehat{S}^*\widehat{M}_w\widehat{S}V + \lambda \mathrm{I})^{-1}V^*\widehat{S}^*\widehat{M}_w\widehat{y} \tag{A.3}$$

where $\lambda > 0$, $\widehat{y} = (y_1, ..., y_n)^T$ *and the matrix* $\widehat{M}_w = \mathrm{diag}(w(x_1), ..., w(x_n))$.

Proof. Eq. (3.12) can be rewritten as

$$\left\|\widehat{M}_w^{\frac{1}{2}}(\widehat{S}P_m f - \widehat{y})\right\|^2 = \left(\widehat{M}_w^{\frac{1}{2}}(\widehat{S}P_m f - \widehat{y})\right)^T \left(\widehat{M}_w^{\frac{1}{2}}(\widehat{S}P_m f - \widehat{y})\right). \tag{A.4}$$

Using first-order condition we have that

$$\frac{2}{n}P_m\widehat{S}^*\widehat{M}_w\widehat{S}P_m\widehat{f}^{w}_{\lambda,m} - \frac{2}{n}P_m\widehat{S}^*\widehat{M}_w\widehat{y} + 2\lambda\widehat{f}^{w}_{\lambda,m} = 0, \tag{A.5}$$

that is

$$(P_m\widehat{S}^*\widehat{M}_w\widehat{S}P_m + \lambda nI)\widehat{f}^{w}_{\lambda,m} = P_m\widehat{S}^*\widehat{M}_w\widehat{y}. \tag{A.6}$$

Replacing $P_m = VV^*$ we obtain

$$V(V^*\widehat{S}^*\widehat{M}_w\widehat{S}V + \lambda nI)V^*\widehat{f}^{w}_{\lambda,m} = VV^*\widehat{S}^*\widehat{M}_w\widehat{y}. \tag{A.7}$$

Multiplying both sides by V^* and using that $(V^*\widehat{S}^*\widehat{M}_w\widehat{S}V + \lambda nI)$ is invertible, we obtain

$$V^*\widehat{f}^{w}_{\lambda,m} = (V^*\widehat{S}^*\widehat{M}_w\widehat{S}V + \lambda nI)^{-1}V^*\widehat{S}^*\widehat{M}_w\widehat{y}. \tag{A.8}$$

The result is obtained multiplying both side by V and remembering that $\widehat{f}^{w}_{\lambda,m} \in \mathcal{H}_m$. □

We can express our estimator in an alternative form which will be useful in the actual implementation of the algorithm.

Lemma 2 (Nyström W-KRR estimator, representer theorem form)**.** *The above minimizer* $\widehat{f}^{w}_{\lambda,m}$ *can be also written as*

$$\widehat{f}^{w}_{\lambda,m}(x) = \sum_{i=1}^{m}\widetilde{c}^{w}_i K(\widetilde{x}_i, x), \quad \widetilde{c}^{w} = (\widehat{K}^T_{nm}\widehat{M}_w\widehat{K}_{nm} + n\lambda\widehat{K}_{mm})^{-1}\widehat{K}^T_{nm}\widehat{M}_w\widehat{y} \tag{A.9}$$

where $\widehat{K}_{nm} = \widehat{S}\widehat{Z}^*_m \in \mathbb{R}^{n\times m}$, $(\widehat{K}_{nm})_{ij} = K(x_i, \widetilde{x}_j)$ *and* $\widehat{K}_{mm} = \widehat{Z}_m\widehat{Z}^*_m \in \mathbb{R}^{m\times m}$, $(\widehat{K}_{mm})_{ij} = K(\widetilde{x}_i, \widetilde{x}_j)$.

Proof. Eq. (A.3) can be rewritten as

$$\begin{aligned}\widehat{f}^{w}_{\lambda,m} &= V(V^*\widehat{S}^*\widehat{M}_w\widehat{S}V + \lambda nI)^{-1}V^*\widehat{S}^*\widehat{M}_w\widehat{y} \\ &= VDU^*UD^{-1}(V^*\widehat{S}^*\widehat{M}_w\widehat{S}V + \lambda nI)^{-1}D^{-1}U^*UDV^*\widehat{S}^*\widehat{M}_w\widehat{y} \\ &= \widehat{Z}^*_m(\widehat{Z}_m\widehat{S}^*\widehat{M}_w\widehat{S}\widehat{Z}^*_m + \lambda\widehat{Z}_m\widehat{Z}^*_m)^\dagger\widehat{Z}_m\widehat{S}^*\widehat{M}_w\widehat{y} \end{aligned} \tag{A.10}$$

where we used $(FGH)^\dagger = H^\dagger G^{-1}F^\dagger$ (see the full-rank factorization of the pseudo-inverse [Ben-Israel and Greville, 2006]) with $F = UD$, $G = V^*\widehat{S}^*\widehat{M}_w\widehat{S}V + \lambda nI$ and $H = DU^T$.
Using the definitions of $\widehat{K}_{mm}$ and $\widehat{K}_{nm}$ we have

$$(\widehat{K}^T_{nm}\widehat{M}_w\widehat{K}_{nm} + \lambda n\widehat{K}_{mm})^\dagger\widehat{K}^T_{nm}\widehat{M}_w y = (\widehat{Z}_m\widehat{S}^*\widehat{M}_w\widehat{S}\widehat{Z}^*_m + \lambda\widehat{Z}_m\widehat{Z}^*_m)^\dagger\widehat{Z}_m\widehat{S}^*\widehat{M}_w\widehat{y} \tag{A.11}$$

and substituting this expression above we get the result. □

A.2 Propositions and Proofs

To prove Theorem 2, we will need the following two propositions.

Proposition 1 (Empirical Effective Dimension)**.** *Let* $\widehat{\mathcal{N}}_w(\lambda) = \operatorname{Tr}\widehat{\Sigma}_w\widehat{\Sigma}^{-1}_{\lambda,w}$. *Under assumption 6 for any* $\delta > 0$ *and* $\left(\frac{128(W+\sigma^2)\log^2(4/\delta)}{n}\right)^{\frac{1}{1+q}} \leqslant \lambda \leqslant \|\Sigma\|$, *then the following holds with probability* $1-\delta$,

$$\frac{|\widehat{\mathcal{N}}_w(\lambda) - \mathcal{N}_{\rho^{te}_X}(\lambda)|}{\mathcal{N}_{\rho^{te}_X}} \leqslant 2.$$

Proof. Let's call $\mathcal{N}_{\rho^{te}_X}(\lambda) = \mathcal{N}(\lambda)$ to simplify the notation. The proof partially follows the structure of Proposition 1 in Rudi et al. [2015], with some complications deriving from the presence of the (possibly unbounded) weights. Define $\widehat{B}^w = \Sigma^{-1/2}_\lambda(\Sigma - \widehat{\Sigma}_w)\Sigma^{-1/2}_\lambda$. Using Lemma 18 in Gogolashvili et al. [2023] we have that $\|\widehat{B}^w\|_{\text{HS}} \leqslant 3/4$, when $n\lambda^{1+q} \geqslant 64\,(W+\sigma^2)\,\mathcal{N}_{\rho^{te}_X}(\lambda)^{1-q}\log^2\left(\frac{2}{\delta}\right)$. We can rewrite

$$\left|\widehat{\mathcal{N}}_w(\lambda) - \mathcal{N}(\lambda)\right| = \left|\operatorname{Tr}\left(\widehat{\Sigma}^{-1}_{\lambda,w}\widehat{\Sigma}_w - \Sigma\Sigma^{-1}_\lambda\right)\right| = \left|\lambda\operatorname{Tr}\widehat{\Sigma}^{-1}_{\lambda,w}\left(\widehat{\Sigma}_w - \Sigma\right)\Sigma^{-1}_\lambda\right| \tag{A.12}$$

$$= \left|\lambda\operatorname{Tr}\Sigma^{-1/2}_\lambda\left(\mathrm{I} - \widehat{B}^w\right)^{-1}\Sigma^{-1/2}_\lambda\left(\widehat{\Sigma}_w - \Sigma\right)\Sigma^{-1/2}_\lambda\Sigma^{-1/2}_\lambda\right| \tag{A.13}$$

$$= \left|\lambda\operatorname{Tr}\Sigma^{-1/2}_\lambda\left(\mathrm{I} - \widehat{B}^w\right)^{-1}\widehat{B}^w\Sigma^{-1/2}_\lambda\right|. \tag{A.14}$$

Following Rudi et al. [2015] and using that, for any symmetric linear operator $X : \mathcal{H} \to \mathcal{H}$ the following identity holds

$$(\mathrm{I} - X)^{-1}X = X + X(\mathrm{I} - X)^{-1}X.$$

Applying the above identity with $X = \widehat{B}$

$$\lambda \left| \operatorname{Tr} \Sigma_\lambda^{-1/2} \left(\mathrm{I} - \widehat{B}^w\right)^{-1} \widehat{B}^w \Sigma_\lambda^{-1/2} \right| \leqslant \underbrace{\lambda \left| \operatorname{Tr} \Sigma_\lambda^{-1/2} \widehat{B}^w \Sigma_\lambda^{-1/2} \right|}_{\mathbb{A}} + \underbrace{\lambda \left| \operatorname{Tr} \Sigma_\lambda^{-1/2} \widehat{B}^w \left(\mathrm{I} - \widehat{B}^w\right)^{-1} \widehat{B}^w \Sigma_\lambda^{-1/2} \right|}_{\mathbb{B}}.$$

To find an upper bound for $\mathbb{A}$ notice that

$$\mathbb{A} = \left| \mu - \frac{1}{n} \sum_{i=1}^{n} \xi_i \right|$$

with $\xi_i = \left\langle K_{x_i}, \lambda w(x_i) \Sigma_\lambda^{-2} K_{x_i} \right\rangle \in \mathbb{R}$ i.i.d. random variables with $i \in [n]$ and $\mu = \mathbb{E}[\xi_i]$. Using Lemma 18 in Gogolashvili et al. [2023] and a general version of Bernstein inequality requiring, instead of boundedness, only an appropriate control of moments [Boucheron et al., 2013], we have with probability greater than $1 - \delta$

$$\mathbb{A} \leqslant 4 \left(\frac{W}{\lambda n} + \sigma \sqrt{\frac{\mathcal{N}_{\rho_X^{te}}(\lambda)^{1-q}}{\lambda^{1+q} n}} \right) \log\left(\frac{2}{\delta}\right).$$

As regards $\mathbb{B}$, write $\mathbb{B} = \|Q\|_{HS}^2$ where $Q = \lambda^{1/2} \Sigma_\lambda^{-1/2} \widehat{B}^w \left(\mathrm{I} - \widehat{B}^w\right)^{-1/2}$, moreover

$$\|Q\|_{HS}^2 \leqslant \left\| \lambda^{1/2} \Sigma_\lambda^{-1/2} \right\|^2 \left\| \widehat{B}^w \right\|_{HS}^2 \left\| \left(\mathrm{I} - \widehat{B}^w\right)^{-1/2} \right\|^2 \leqslant 4 \|\widehat{B}^w\|_{HS}^2,$$

since $\left\| \left(\mathrm{I} - \widehat{B}^w\right)^{-1/2} \right\|^2 \leqslant (1 - \|\widehat{B}^w\|)^{-1} \leqslant 4$, for $n\lambda^{1+q} \geqslant 64 \left(W + \sigma^2\right) \mathcal{N}_{\rho_X^{te}}(\lambda)^{1-q} \log^2\left(\frac{2}{\delta}\right)$. Using again Lemma 18 in Gogolashvili et al. [2023] and a version of the Bernstein inequality for Hilbert space-valued random variables (see for example Caponnetto and De Vito [2007]) we obtain with probability greater than $1 - \delta$

$$\mathbb{B} \leqslant 16 \left(\frac{W}{\lambda n} + \sigma \sqrt{\frac{\mathcal{N}_{\rho_X^{te}}(\lambda)^{1-q}}{\lambda^{1+q} n}} \right)^2 \log^2\left(\frac{2}{\delta}\right)$$

Putting all together, with probability $1-\delta$:

$$\left|\widehat{\mathcal{N}}_w(\lambda)-\mathcal{N}(\lambda)\right| \leqslant 4\left(\frac{W}{\lambda n}+\sigma\sqrt{\frac{\mathcal{N}_{\rho_X^{te}}(\lambda)^{1-q}}{\lambda^{1+q}n}}\right)\log\left(\frac{4}{\delta}\right)+16\left(\frac{W}{\lambda n}+\sigma\sqrt{\frac{\mathcal{N}_{\rho_X^{te}}(\lambda)^{1-q}}{\lambda^{1+q}n}}\right)^2\log^2\left(\frac{4}{\delta}\right) \tag{A.15}$$

$$\leqslant 4\left(\frac{W}{\lambda n}+\sigma\sqrt{\frac{1}{\lambda^{\gamma(1-q)+1+q}n}}\right)\log\left(\frac{4}{\delta}\right)+16\left(\frac{W}{\lambda n}+\sigma\sqrt{\frac{1}{\lambda^{\gamma(1-q)+1+q}n}}\right)^2\log^2\left(\frac{4}{\delta}\right) \tag{A.16}$$

Using that $\mathcal{N}(\lambda)\geqslant\|\Sigma\Sigma_\lambda^{-1}\|\geqslant 1/2$ if $\lambda\leqslant\|\Sigma\|$ we have

$$\left|\widehat{\mathcal{N}}_w(\lambda)-\mathcal{N}(\lambda)\right| \tag{A.17}$$

$$\leqslant\left(4\left(\frac{W}{\mathcal{N}_{\rho_X^{te}}(\lambda)\lambda n}+\sigma\sqrt{\frac{1}{\mathcal{N}_{\rho_X^{te}}(\lambda)^{1+q}\lambda^{1+q}n}}\right)\log\left(\frac{4}{\delta}\right)+\right. \tag{A.18}$$

$$\left.+16\left(\frac{W}{\mathcal{N}_{\rho_X^{te}}(\lambda)\lambda n}+\sigma\sqrt{\frac{1}{\mathcal{N}_{\rho_X^{te}}(\lambda)^{1+q}\lambda^{1+q}n}}\right)^2\log^2\left(\frac{4}{\delta}\right)\right)\mathcal{N}_{\rho_X^{te}}(\lambda) \tag{A.19}$$

$$\leqslant\left(4\left(\frac{2W}{\lambda n}+\sigma\sqrt{\frac{4}{\lambda^{1+q}n}}\right)\log\left(\frac{4}{\delta}\right)+16\left(\frac{2W}{\lambda n}+\sigma\sqrt{\frac{4}{\lambda^{1+q}n}}\right)^2\log^2\left(\frac{4}{\delta}\right)\right)\mathcal{N}_{\rho_X^{te}}(\lambda). \tag{A.20}$$

Then for $\left(\frac{256(W+\sigma^2)\log^2(4/\delta)}{n}\right)^{\frac{1}{1+q}}\leqslant\lambda\leqslant\|\Sigma\|$ with probability $1-\delta$

$$\left|\widehat{\mathcal{N}}_w(\lambda)-\mathcal{N}(\lambda)\right|\leqslant 2\mathcal{N}(\lambda)$$

□

Proposition 2 (Nyström approximation with ALS sampling)**.** *Let $\left(\hat{l}_i(t)\right)_{i=1}^n$ be the collection of approximate leverage scores. Let $\lambda>0$ and P_λ be defined as $P_\lambda(i)=\hat{l}_i(\lambda)/\sum_{j\in N}\hat{l}_j(\lambda)$ for any $i\in N$ with $N=\{1,\dots,n\}$. Let $\mathfrak{I}=(i_1,\dots,i_m)$ be a collection of indices independently sampled with replacement from N according to the probability distribution P_λ. Let P_m be the projection operator on the subspace $\mathcal{H}_m=\operatorname{span}\left\{K_{x_j}\mid j\in J\right\}$ and J be the subcollection of $\mathfrak{I}$ with all the duplicates removed. Under Assumption 6 and 9 , for any $\delta>0$ the following holds with probability $1-2\delta$*

$$\|(I-P_m)\Sigma_\lambda^{1/2}\|\leqslant 3\lambda$$

when the following conditions are satisfied:

- *there exists* $T \geqslant 1$ *and* $\lambda_0 > 0$ *such that* $\left(\hat{l}_i(t)\right)_{i=1}^n$ *are* T*-approximate leverage scores for any* $t \geqslant \lambda_0$ *(see Definition 2),*
- $\lambda_0 \vee \left(\frac{256(W+\sigma^2)\log^2(4/\delta)}{n}\right)^{\frac{1}{\gamma(1-q)+1+q}} \leqslant \lambda \leqslant \|\Sigma\|$,
- $m \geqslant 144T^2\mathcal{N}_{\rho_X^{te}}(\lambda)\log\frac{8n}{\delta}$.

Proof. Let's call $\Sigma = \Sigma_{\rho_X^{te}}$ to simplify the notation. Define $\tau = \delta/4$. Next, define the diagonal matrix $H \in \mathbb{R}^{n\times n}$ with $(H)_{ii} = 0$ when $P_\lambda(i) = 0$ and $(H)_{ii} = \frac{nq(i)}{mP_\lambda(i)}$ when $P_\lambda(i) > 0$, where $q(i)$ is the number of times the index i is present in the collection $\mathfrak{I}$. We have that

$$\widehat{S}_w^* H \widehat{S}_w = \frac{1}{m}\sum_{i=1}^n w(x_i)\frac{q(i)}{P_\lambda(i)} K_{x_i}\otimes K_{x_i} = \frac{1}{m}\sum_{j\in J} w(x_j)\frac{q(j)}{P_\lambda(j)} K_{x_j}\otimes K_{x_j}.$$

Now, considering that $\frac{q(j)}{P_\lambda(j)} > 0$ for any $j \in J$, thus $\operatorname{ran}\widehat{S}_w^* H \widehat{S}_w = \mathcal{H}_m$. Therefore, by using Prop. 3 and 7 in Rudi et al. [2015], we exploit the fact that the range of P_m is the same of $\widehat{S}_w^* H \widehat{S}_w$, to obtain

$$\left\|(I - P_m)\,\Sigma_\lambda^{1/2}\right\|^2 \leqslant \lambda \left\|\left(\widehat{S}_w^* H \widehat{S}_w + \lambda I\right)^{-1/2}\Sigma^{1/2}\right\|^2 \leqslant \frac{\lambda}{1-\beta(\lambda)},$$

with $\beta(\lambda) = \lambda_{\max}\left(\Sigma_\lambda^{-1/2}\left(\Sigma - \widehat{S}_w^* H \widehat{S}_w\right)\Sigma_\lambda^{-1/2}\right)$. Considering that the function $(1-x)^{-1}$ is increasing on $-\infty < x < 1$, in order to bound $\lambda/(1-\beta(\lambda))$ we need an upperbound for $\beta(\lambda)$. Here we split $\beta(\lambda)$ in the following way,

$$\beta(\lambda) \leqslant \underbrace{\lambda_{\max}\left(\Sigma_\lambda^{-1/2}\left(\Sigma - \widehat{\Sigma}_w\right)\Sigma_\lambda^{-1/2}\right)}_{\beta_1(\lambda)} + \underbrace{\lambda_{\max}\left(\Sigma_\lambda^{-1/2}\left(\widehat{\Sigma}_w - \widehat{S}_w^* H \widehat{S}_w\right)\Sigma_\lambda^{-1/2}\right)}_{\beta_2(\lambda)}.$$

β_1 can be bounded as in eq. (3.16).
As regards β_2:

$$\begin{aligned}\beta_2(\lambda) &\leqslant \left\|\Sigma_\lambda^{-1/2}\left(\widehat{\Sigma}_w - \widehat{S}_w^* H \widehat{S}_w\right)\Sigma_\lambda^{-1/2}\right\| \\ &\leqslant \left\|\Sigma_\lambda^{-1/2}\widehat{\Sigma}_{w\lambda}^{1/2}\right\|^2 \left\|\widehat{\Sigma}_{w\lambda}^{-1/2}\left(\widehat{\Sigma}_w - \widehat{S}_w^* H \widehat{S}_w\right)\widehat{\Sigma}_{w\lambda}^{-1/2}\right\|.\end{aligned}$$

Let

$$\beta_3(\lambda) = \left\| \widehat{\Sigma}_{w\lambda}^{-1/2} \left(\widehat{\Sigma}_w - \widehat{S}_w^* H \widehat{S}_w \right) \widehat{\Sigma}_{w\lambda}^{-1/2} \right\| = \left\| \widehat{\Sigma}_{w\lambda}^{-1/2} \widehat{S}_w^* (I - H) \widehat{S}_w \widehat{\Sigma}_{w\lambda}^{-1/2} \right\|.$$

Note that $\widehat{S}_w \widehat{\Sigma}_{w\lambda}^{-1} \widehat{S}_w^* = \widehat{S}_w (\widehat{S}_w^* \widehat{S}_w + \lambda \mathrm{I})^{-1} \widehat{S}_w^* = \left(\widehat{K}_w + \lambda n I \right)^{-1} \widehat{K}_w$ since $\widehat{K}_w = n \widehat{S}_w \widehat{S}_w^*$, with $(\widehat{K}_w)_{ij} = w(x_i)^{1/2} w(x_j)^{1/2} K(x_i, x_j)$.
Thus, if we let $UDU^\top$ be the eigendecomposition of $\widehat{K}_w$, we have that $\left(\widehat{K}_w + \lambda n I \right)^{-1} \widehat{K}_w = U(D + \lambda n I)^{-1} D U^\top$ and thus $\widehat{S}_w \widehat{\Sigma}_{w\lambda}^{-1} \widehat{S}_w^* = U(D + \lambda n I)^{-1} D U^\top$. In particular this implies that $\widehat{S}_w \widehat{\Sigma}_{w\lambda}^{-1} \widehat{S}_w^* = U \widehat{Q}^{1/2} \widehat{Q}^{1/2} U^\top$ with $\widehat{Q} = (D + \lambda n I)^{-1} D$. Therefore we have

$$\beta_3(\lambda) = \left\| \widehat{\Sigma}_{w\lambda}^{-1/2} \widehat{S}_w^* (I - H) \widehat{S}_w \widehat{\Sigma}_{w\lambda}^{-1/2} \right\| = \left\| \widehat{Q}^{1/2} U^\top (I - H) U \widehat{Q}^{1/2} \right\|.$$

Consider the matrix $A = \widehat{Q}^{1/2} U^\top$ and let a_i be the i-th column of A, and e_i be the i-th canonical basis vector for each $i \in N$. We prove that $\|a_i\|^2 = l_i(\lambda)$, the true leverage score, since

$$\|a_i\|^2 = \left\| \widehat{Q}^{1/2} U^\top e_i \right\|^2 = e_i^\top U \widehat{Q} U^\top e_i = \left(\left(\widehat{K}_w + \lambda n I \right)^{-1} \widehat{K}_w \right)_{ii} = l_i(\lambda)$$

Considering that $\sum_{k=1}^n \frac{q(k)}{P_\lambda(k)} a_k a_k^\top = \sum_{i=\mathfrak{I}} \frac{1}{P_\lambda(i)} a_i a_i^\top$, we have

$$\beta_3(\lambda) = \left\| A A^\top - \frac{1}{m} \sum_{i \in \mathfrak{I}} \frac{1}{P_\lambda(i)} a_i a_i^\top \right\|.$$

Moreover, by the T-approximation property of the approximate leverage scores (see Def. 1 in Rudi et al. [2015]), we have that for all $i \in \{1, \ldots, n\}$, when $\lambda \geqslant \lambda_0$, the following holds with probability $1 - \delta$

$$P_\lambda(i) = \frac{\hat{l}_i(\lambda)}{\sum_j \hat{l}_j(\lambda)} \geqslant T^{-2} \frac{l_i(\lambda)}{\sum_j l_j(\lambda)} = T^{-2} \frac{\|a_i\|^2}{\operatorname{Tr} A A^\top}.$$

Then, we can apply Prop. 9 in Rudi et al. [2015], so that, after a union bound, we obtain the following inequality with probability $1 - \delta - \tau$:

$$\beta_3(\lambda) \leqslant \frac{2\|A\|^2 \log \frac{2n}{\tau}}{3m} + \sqrt{\frac{2\|A\|^2 T^2 \operatorname{Tr} A A^\top \log \frac{2n}{\tau}}{m}} \leqslant \frac{2 \log \frac{2n}{\tau}}{3m} + \sqrt{\frac{2T^2 \widehat{\mathfrak{N}}_w(\lambda) \log \frac{2n}{\tau}}{m}},$$

where the last step follows from $\|A\|^2 = \left\| \left(\widehat{K}_w + \lambda n I\right)^{-1} \widehat{K}_w \right\| \leqslant 1$ and $\operatorname{Tr}\left(AA^\top\right) = \operatorname{Tr}\left(\widehat{\Sigma}_{w\lambda}^{-1}\widehat{\Sigma}_w\right) := \widehat{\mathcal{N}}_w(\lambda)$. Applying proposition 1 we have that $\widehat{\mathcal{N}}_w(\lambda) \leqslant 3\mathcal{N}_{\rho_X^{te}}(\lambda)$ with probability $1-\tau$, when $\left(\frac{128(W+\sigma^2)\log^2(4/\delta)}{n}\right)^{\frac{1}{1+q}} \leqslant \lambda \leqslant \|\Sigma\|$. Thus, by taking a union bound again, we have

$$\beta_3(\lambda) \leqslant \frac{2\log\frac{2n}{\tau}}{3m} + \sqrt{\frac{16T^2\mathcal{N}_{\rho_X^{te}}(\lambda)\log\frac{2n}{\tau}}{m}}$$

with probability $1-2\tau-\delta$. The last step is to bound $\left\|\Sigma_\lambda^{-1/2}\widehat{\Sigma}_{w\lambda}^{1/2}\right\|^2$, as follows

$$\left\|\Sigma_\lambda^{-1/2}\widehat{\Sigma}_{\lambda w}^{1/2}\right\|^2 = \left\|\Sigma_\lambda^{-1/2}\widehat{\Sigma}_{\lambda w}\Sigma_\lambda^{-1/2}\right\| = \left\|I + \Sigma_\lambda^{-1/2}\left(\widehat{\Sigma}_w - \Sigma\right)\Sigma_\lambda^{-1/2}\right\| \leqslant 1+\eta,$$

with $\eta = \left\|\Sigma_\lambda^{-1/2}\left(\widehat{\Sigma}_w - \Sigma\right)\Sigma_\lambda^{-1/2}\right\|$. We can bound η using Lemma 18 in Gogolashvili et al. [2023] (see eq. (3.16)). Finally, by collecting the above results and taking a union bound we have

$$\beta(\lambda) \leqslant 4\left(\frac{W}{\lambda n} + \sqrt{\frac{\sigma^2\mathcal{N}_{\rho_X^{te}}(\lambda)^{1-q}}{\lambda^{1+q}n}}\right)\log\left(\frac{2}{\tau}\right) + (1+\eta)\left(\frac{2\log\frac{2n}{\tau}}{3m} + \sqrt{\frac{16T^2\mathcal{N}_{\rho_X^{te}}(\lambda)\log\frac{2n}{\tau}}{m}}\right),$$

with probability $1-4\tau-\delta = 1-2\delta$ when $\left(\frac{256(W+\sigma^2)\log^2(4/\delta)}{n}\right)^{\frac{1}{1+q}} \leqslant \lambda \leqslant \|\Sigma\|$. Note that, if we select $\left(\frac{256(W+\sigma^2)\log^2(4/\delta)}{n}\right)^{\frac{1}{\gamma(1-q)+1+q}} \leqslant \lambda \leqslant \|\Sigma\|$, and $m \geqslant 144T^2\mathcal{N}_{\rho_X^{te}}(\lambda)\log\frac{8n}{\delta}$, we have $\beta(\lambda) \leqslant 5/6$, so that

$$\left\|(I-P_m)\Sigma^{1/2}\right\|^2 \leqslant 6\lambda$$

with probability $1-2\delta$. □

A.3 Known results

In this section, we derive tight bounds for the effective dimension $\mathcal{N}_\rho(\lambda)$ defined in Definition 3 when assuming polynomial decay of the eigenvalues $\sigma_j(\Sigma)$ of the covariance operator Σ.

Proposition 3 (Polynomial eigenvalues decay, Proposition 3 in Caponnetto and De Vito [2007]). *If for some* $\gamma \in \mathbb{R}^+$ *and* $1 < \beta < +\infty$

$$\sigma_i \leqslant \gamma i^{-\beta}$$

then

$$\mathcal{N}_\rho(\lambda) \leqslant \gamma \frac{\beta}{\beta - 1} \lambda^{-1/\beta} \tag{A.21}$$

Proof. Since the function $\sigma/(\sigma+\lambda)$ is increasing in σ and using the spectral theorem $\Sigma = UDU^*$ combined with the fact that $\mathrm{Tr}(UDU^*) = \mathrm{Tr}(U(U^*D)) = \mathrm{Tr}D$

$$\mathcal{N}_\rho(\lambda) = \mathrm{Tr}(\Sigma(\Sigma + \lambda I)^{-1}) = \sum_{i=1}^{\infty} \frac{\sigma_i}{\sigma_i + \lambda} \leqslant \sum_{i=1}^{\infty} \frac{\gamma}{\gamma + i^\beta \lambda} \tag{A.22}$$

The function $\gamma/(\gamma + x^\beta \lambda)$ is positive and decreasing, so

$$\begin{aligned} \mathcal{N}_\rho(\lambda) &\leqslant \int_0^\infty \frac{\gamma}{\gamma + x^\beta \lambda} dx \\ &= \alpha^{-1/\beta} \int_0^\infty \frac{\gamma}{\gamma + \tau^\beta} d\tau \\ &\leqslant \gamma \frac{\beta}{\beta - 1} \lambda^{-1/\beta} \end{aligned} \tag{A.23}$$

since $\int_0^\infty (\gamma + \tau^\beta)^{-1} \leqslant \beta/(\beta - 1)$. □

Further improvements can be found assuming exponential decay of the eigenvalues $\sigma_j(\Sigma)$ of Σ.

Proposition 4 (Exponential eigenvalues decay, Proposition 5 in Della Vecchia et al. [2021]). *If for some* $\gamma, \beta \in \mathbb{R}^+ \sigma_i \leqslant \gamma e^{-\beta i}$ *then*

$$\mathcal{N}_\rho(\lambda) \leqslant \frac{\log(1 + \gamma/\lambda)}{\beta}. \tag{A.24}$$

Proof.

$$\mathcal{N}_\rho(\lambda) = \sum_{i=1}^{\infty} \frac{\sigma_i}{\sigma_i + \lambda} = \sum_{i=1}^{\infty} \frac{1}{1 + \lambda/\sigma_i} \leqslant \sum_{i=1}^{\infty} \frac{1}{1 + \lambda' e^{\beta i}} \leqslant \int_0^{+\infty} \frac{1}{1 + \lambda' e^{\beta x}} dx, \tag{A.25}$$

where $\lambda' = \lambda/\gamma$. Using the change of variables $t = e^{\beta x}$ we get

$$(A.25) = \frac{1}{\beta}\int_1^{+\infty} \frac{1}{1+\lambda' t}\,\frac{1}{t}dt = \frac{1}{\beta}\int_1^{+\infty}\Big[\frac{1}{t} - \frac{\lambda'}{1+\lambda' t}\Big]dt = \frac{1}{\beta}\Big[\log t - \log(1+\lambda' t)\Big]_1^{+\infty}$$
$$= \frac{1}{\beta}\Big[\log\Big(\frac{t}{1+\lambda' t}\Big)\Big]_1^{+\infty} = \frac{1}{\beta}\Big[\log(1/\lambda') + \log(1+\lambda')\Big]. \tag{A.26}$$

So we finally obtain

$$\mathcal{N}_\rho(\lambda) \leqslant \frac{1}{\beta}\Big[\log(\gamma/\lambda) + \log(1+\lambda/\gamma)\Big] = \frac{\log(1+\gamma/\lambda)}{\beta}. \tag{A.27}$$

□

Next proposition gives some relations between weights and covariance operators.

Proposition 5. *Suppose that ρ_X^{te} is absolutely continuous with respect to ρ_X^v, and the Radon-Nikodym derivative $d\rho_X^{te}/d\rho_X^v$ is bounded by $G \in \mathbb{R}$. Then*

$$\left\|\Sigma\Sigma_{v\lambda}^{-1}\right\| \leqslant G,$$

see Gogolashvili et al. [2023].
Moreover, if $w(x) < \infty$ and $v(x) > 0$ for all $x \in \mathcal{X}$, with w and v defined as in eq. (??) and (3.19) respectively, then

$$\left\|\Sigma\Sigma_{v\lambda}^{-1}\right\| \leqslant \left\|\frac{w}{v}\right\|_\infty.$$

Proof. For operators A and B on $\mathcal{H}$, denote by $A \geqslant B$ that $A - B$ is a non-negative operator. Let $G := \|d\rho_X^{te}/d\rho_X^v\|_\infty$. For all $f \in \mathcal{H}$, we have

$$\langle f, \Sigma f\rangle_{\mathcal{H}} = \left\langle f, \int K_x f(x) d\rho_X^{te}(x)\right\rangle_{\mathcal{H}} = \int \langle f, K_x\rangle_{\mathcal{H}} f(x) d\rho_X^{te}(x) = \int f^2(x) d\rho_X^{te}(x)$$
$$= \int f^2(x)\frac{d\rho_X^{te}}{d\rho_X^v}(x) d\rho_X^v(x) \leqslant G\int f^2(x) d\rho_X^v(x) = G\,\langle f, \Sigma_v f\rangle_{\mathcal{H}}.$$

Therefore, we have

$$\Sigma \preccurlyeq G\Sigma_v \preccurlyeq G\,(\Sigma_v + \lambda \mathrm{I}) \quad \Longrightarrow \quad \Sigma\,(\Sigma_v + \lambda \mathrm{I})^{-1} \preccurlyeq G\mathbb{I},$$

where $\mathbb{I} : \mathcal{H} \mapsto \mathcal{H}$ is the identity operator. This implies $\left\|\Sigma\,(\Sigma_v + \lambda)^{-1}\right\| \leqslant G$,

which proves the first assertion. The second part comes simply from the fact that

$$\langle f, \Sigma f \rangle_{\mathcal{H}} = \int f^2(x) \frac{d\rho_X^{te}}{d\rho_X^{tr}}(x) \frac{d\rho_X^{tr}}{d\rho_X^{v}}(x) d\rho_X^{v}(x) = \int f^2(x) w(x) \frac{1}{v(x)} d\rho_X^{v}(x) \leqslant \left\| \frac{w}{v} \right\|_{\infty} \langle f, \Sigma_v f \rangle_{\mathcal{H}} .$$

□

Proposition 6 (Cordes Inequality Fujii et al. [1993]). *Let A, B two positive semidefinite bounded linear operators on a separable Hilbert space. Then*

$$\|A^s B^s\| \leqslant \|AB\|^s \quad \text{when } 0 \leqslant s \leqslant 1.$$

A.4 Experimental Details and Datasets

This section includes some additional details on the experimental setting and a detailed summary of all the datasets presented.

The four datasets consist of data collected from multiple users using wearable sensors, such as accelerometers and gyroscopes. We allocate 1% of the test dataset for weight estimation, reserving the remaining 99% for predictions. Weights are estimated using Relative Unconstrained Least-Squares Importance Fitting (RuLSIF) method, which minimizes the relative density-ratio divergence between distributions [Liu et al., 2013; Yamada et al., 2013].The experiments were conducted in Python on a 2018 MacBook Pro with a 2.3 GHz Intel Core i5 Quad-Core processor, 16GB of RAM, and no GPU.

We give here a brief description of the four datasets.

- The HHAR dataset [Stisen et al., 2015], accessible on DR-NTU (Data), dataset DOI: https://doi.org/10.21979/N9/OWDFXO, version used: 3.0 (May 27, 2022), license: CC BY-NC 4.0, authors of the dataset version: Mohamed Ragab and Emadeldeen Eldele. It comprises 13,062,475 samples with 10 features for human activity recognition using smartphone and smartwatch sensors. It explores the effect of sensor heterogeneity on activity recognition algorithms, aiming to classify physical activities. Data were collected from

accelerometer and gyroscope sensors during activities like Biking, Sitting, Standing, Walking, Stair Up, and Stair Down. Nine users participated, using 4 smartwatches (2 LG, 2 Samsung Galaxy) and 8 smartphones (2 each of Samsung Galaxy S3 mini, Samsung Galaxy S3, LG Nexus 4, and Samsung Galaxy S+). Features include accelerometer readings (x, y, z), user ID, device, model, and activity labels. Non-essential columns—index, arrival time, and creation time—were removed. The categorical columns (device and model) were converted to numerical values. User A has 1,218,871 samples designated as training data, from which 15,500 samples were randomly selected, while user H provides the test data.

- The HAR70+ dataset [Ustad et al., 2023], available from the UCI Machine Learning Repository, dataset DOI: https://doi.org/10.24432/C5CW3D, version used: Latest version available via the Norwegian Centre for Research Data (NSD), license: CC BY 4.0 (Creative Commons Attribution 4.0 International), dataset authors: Aleksej Logacjov, Astrid Ustad. It contains 2,259,597 samples with 6 features for activity classification. Data were recorded from 18 older adults (ages 70–95) wearing two Axivity AX3 accelerometers at 50 Hz for 40 minutes in a semi-structured, free-living setting. Five participants used walking aids. Sensors were placed on the right thigh and lower back. Activities such as walking, shuffling, ascending/descending stairs, standing, sitting, and lying were annotated frame-by-frame using chest-mounted camera video. For training and testing, we used data from two files: 518.csv, containing 141,714 samples with 6 features, and 516.csv, containing 138,278 samples with 6 features. From 518.csv, 20,000 samples were randomly selected for training. To introduce covariate shift and study feature selection, the timestamp and thigh accelerations in the x and z directions were excluded.

- The HARChildren dataset [Tørring et al., 2024], available on DR-NTU (Data), dataset DOI: https://doi.org/10.18710/EPCXCC, version used: published on August 30, 2024, license: CC0 1.0, dataset authors: Marte Fossflaten Tørring et al. It contains over 5 million samples with 8 features for activity classification. It includes data from 63 typically developing (TD) children and 16 children with Cerebral Palsy (CP), classified at levels I and II of the Gross Motor Function Classification System (GMFCS). These children wore two accelerometers, on the lower back and thigh. The features initially included a timestamp and accelerometer readings (x, y, z), but the timestamp was removed during preprocessing. The recorded activities include walking, running, shuffling, ascending and descending stairs, standing, sitting, lying, bending, cycling seated, cycling standing, and jumping. For training and testing, two files were used: 004.csv, which contains 354,139 samples, and 010.csv, which contains 238,574 samples. From 010.csv, 15,000 samples were randomly selected for training, while 004.csv was used for testing.

- The WISDM dataset [Kwapisz et al., 2011], available on DR-NTU (Data), dataset DOI: https://doi.org/10.21979/N9/KJWE5B, version used: 1.0 (May 27, 2022), license: CC BY-NC 4.0, dataset authors: Mohamed Ragab and Emadeldeen Eldele. It contains 1,098,209 samples with 5 features from accelerometer and gyroscope data collected at 20 Hz using smartphones and smartwatches. Data were recorded from 36 users performing 18 activities for 3 minutes each. Features include a user ID, activity code (label), and sensor readings (x, y, z). User 12 has 32,641 samples designated as training data, from which 25,000 samples were randomly selected, while user 19 provides the test data.

A common way to specify the grid of possible values of λ is to consider a

geometric series.

Remark 1 (Geometric grid)**.** *Let λ_{min} and λ_{max} the smallest and largest values of the regularization parameter we wish to consider and let*

$$b = \left(\frac{\lambda_{max}}{\lambda_{min}}\right)^{1/(Q-1)}.$$

We generate a geometric grid of Q values of the regularization if for $q = 1, \dots, Q$, we let

$$\lambda_q = b^{q-1}\lambda_{min},$$

so that $\lambda_1 = \lambda_{min}$ and $\lambda_Q = \lambda_{max}$.

We used $\lambda_1 = 10^{-4}$, $\lambda_Q = 1$ and $Q = 10$ to generate the geometric grid. We also propose a method for generating a geometric sequence of γ values, derived from distinct formulas for even and odd indices.

Remark 2. *The following values represent the parameter γ for the Gaussian kernel:*

$$\gamma_k = \begin{cases} 10^{-3+\frac{k-1}{2}}, & k \text{ odd}, \\ 5 \cdot 10^{-3+\frac{k-2}{2}}, & k \text{ even}, \end{cases} \qquad k = 1, 2, \dots, 6.$$

We optimize the hyperparameters using hold-out cross-validation, partitioning the dataset into 70% for training and 30% for validation. For each combination of hyperparameters λ and γ, we train the model on X_{train} and y_{train} and validate it on X_{val} and y_{val}.

As regards the simulations, we report here the parameters used to generate the results in Figure 3.1: $k = 50$ (that we can consider misspecified), $\mu_{tr} = (0.7, 0.7)$, $\sigma^2_{tr} = \text{diag}(0.7, 0.7)$, $\mu_{te} = (1.8, 1.8)$, $\sigma^2_{te} = \text{diag}(0.5, 0.5)$, $\epsilon^2 = 0.2$, $c_1 = c_2 = 10$.

Table A.1 shows that even when using uniform sampling, the two methods with importance weighting (IW) correction achieve the best and essentially equal performance. However, the number of Nyström points m required by Nyström W-KRR is 5500, 6500, 1750, and 4500 for HAR70+, HARChildren, HHAR, and WISDM respectively, which represents an increase compared to the ALS method.

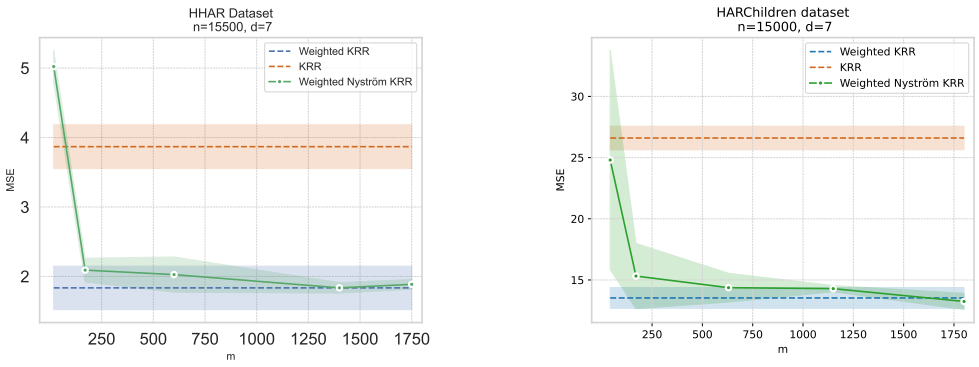


Figure A.1: These plots illustrate the results in Table 3.1 for HHAR and HARChildren datasets.

Table A.1: Performances of the various methods, both in terms of MSE and training/prediction time using uniform sampling.

	HAR70+ ($n = 20000$)			HARChildren ($n = 15000$)			HHAR ($n = 15500$)			WISDM ($n = 25000$)		
	MSE	t train (s)	t pred (s)	MSE	t train (s)	t pred (s)	MSE	t train (s)	t pred (s)	MSE	t train (s)	t pred (s)
KRR	10 ± 1	1694 ± 2	15.0 ± 0.5	26.6 ± 0.9	762 ± 12	10.2 ± 0.4	3.7 ± 0.3	876 ± 6	10.5 ± 0.9	7.8 ± 0.1	3280 ± 48	38 ± 5
W-KRR	5.0 ± 0.2	1785 ± 2	15.1 ± 0.3	13.5 ± 0.8	809 ± 26	9.0 ± 0.1	1.8 ± 0.3	1034 ± 93	9.9 ± 0.1	4.8 ± 0.2	3364 ± 30	33 ± 2
Ny W-KRR	5.0 ± 0.1	67 ± 0.8	3.8 ± 0.1	12.9 ± 0.2	57 ± 0.2	6.2 ± 1.0	1.79 ± 0.01	5.9 ± 0.1	1.3 ± 0.2	4.8 ± 0.1	32 ± 0.5	7.2 ± 0.1