Predicting Operational Lifetimes in Hybrid Perovskite Solar Cells: A Case Study of Machine Learning with Small Datasets

Preetham P. Sunkari

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2025

Reading Committee:

Hugh W. Hillhouse, Chair

Marina Meila

David A. C. Beck

Program Authorized to Offer Degree:

Chemical Engineering

University of Washington

Abstract

Predicting Operational Lifetimes in Hybrid Perovskite Solar Cells: A Case Study of Machine Learning with Small Datasets

Preetham P. Sunkari

Chair of the Supervisory Committee:

Hugh W. Hillhouse
Chemical Engineering

Mixed organic-inorganic halide perovskite solar cells (PSCs) have emerged as promising candidates for photovoltaic applications due to their exceptional optoelectronic properties, high defect tolerance, low cost, and ease of fabrication. However, their instability under environmental stress remains a major barrier to commercial viability. Recent studies have focused on understanding the degradation mechanisms in these solar cells under varied environmental conditions to develop predictive models for operational lifetimes.

Given the limited understanding of these mechanisms, data-driven approaches—particularly those leveraging machine learning (ML) tools guided by domain knowledge—have become central to lifetime prediction. However, due to the significant time and cost associated with generating the requisite data on PSC degradation, available datasets are typically small, often containing only hundreds, or even just dozens, of meticulously gathered trials. This scarcity of data

is a common challenge not only in PSC research but also in other experimental fields within chemistry and materials science, limiting the effectiveness of popular data-hungry ML techniques such as neural networks.

In this work, I present a case study using in-house PSC degradation datasets to explore the challenges and strategies of modeling with small data. First, I outline criteria for identifying when a dataset falls within the *small* data regime or is considered *too small* for reliable predictive modeling. I then review recent ML advances tailored to such contexts and present a modeling workflow that includes feature construction, feature selection, assessment of metrics (such as sparsity level, false-discovery and ground-truth recovery), and uncertainty quantification. Using simulated datasets generated from known ground-truth variables, I demonstrate that prediction error alone is not always the most reliable metric for feature selection. These simulations, designed to reflect real-world scientific conditions, yield heuristic insights that are then applied to real PSC datasets.

Overall, with the help of the in-house PSC degradation datasets, I present a practical framework to guide researchers in selecting appropriate machine learning methods based on data availability. This work aims to educate scientists and engineers on statistically rigorous modeling techniques for small datasets.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I am immensely grateful to all the incredible individuals who, both professionally and personally, have supported me along the way in my PhD and enabled this work. First and foremost, I would like to thank my advisor, Dr. Hugh Hillhouse, for his unwavering commitment to my growth—both in technical expertise and interpersonal skills. I deeply appreciate your guidance in developing critical thinking for sound research, which has shaped me into a more thoughtful scientist. Thank you for consistently challenging me in presentations, writing, and discussions, and for holding my work to a high standard. I would also like to thank Dr. Marina Meila for being an excellent guide and helping me navigate complex statistical concepts that were critical to this work. Your feedback and suggestions throughout my PhD encouraged me to explore topics I had never encountered before.

I would also like to thank the other members of my committee. Dr. David Beck, thank you for your kindness and encouragement over the years, whether in courses, meetings, or exams. Also, thank you for being an excellent instructor in CHEM E 546 and 547, which was the foundation of my data science and software development skills. Dr. Robert Synovec, thank you serving as the GSR on my committee and for your interest in my research. Dr. Ting Cao, thank you serving as GSR for my general exam.

I would also like to acknowledge my colleagues and collaborators in the Hillhouse group. Dr. Yuhuan Meng, thank you for being such an amazing mentor and friend throughout my PhD journey. I deeply appreciate your experimental expertise and the wisdom you shared whenever I sought your advice. Spencer Cira, I appreciate your assistance in keeping the lab running, for being a reliable teammate during DOE presentations, and for our insightful discussions. Dr. Wiley

Dunlap-Shohl, I appreciate your dedication to our joint projects, and for training me on the microscope setup and fabricating perovskite devices. I always valued your scientific insights and looked forward to our discussions. Jason Moore, thank you for developing the spray coating methodology and teaching me how to use the spray coater. Dr. Timothy Siegler, thank you for the "big picture" conversations that extended beyond project tasks and helped shape my career thinking. Dr. Ryan Stoddard, thank you for the early conversations on perovskites during the first couple months of my PhD. Hongbo Qiao, thank you for introducing me to spin coating in my initial weeks in the lab. Chang-En Tsai, thank you for being a great colleague and a close friend. I would also like to thank Dr. Yu-Chia Chen, Zhenman Yuan, and Alex Kokot for your help in applying statistical concepts on different projects. I would like to acknowledge the Department of Energy Sunshot program, University of Washington Chemical Engineering Department, and the Clean Energy Institute (via the Washington Research Foundation) for support.

I would also like to thank my personal friends for their support throughout this journey. Daniel Trujillo, Marvell Holder, Jorge Castillo, Christopher James Browne, Gilbert Shen, Vignesh Rathana Kumar, and all my friends at church—thank you for the good times and support throughout my PhD. Finally, I would like to thank my family for their unwavering support. To my parents, Kiran Kumar and Sukanya Sunkari, and my sister, Celestene Andugula—thank you for always being there for me.

DEDICATION

To God and my family, both physical and spiritual,

without whom I would not have made it this far

1 INTRODUCTION

In 2024, the global electricity demand rose by 4.3%, a sharp increase from 2.5% in 2023.[1] This surge was accompanied by a 0.8% rise in energy-related carbon-dioxide (CO_2) emissions, reaching a record high of ~38 gigatons,[1] alongside the warmest global temperatures recorded since 1850.[2] Meeting this growing energy demand while reducing CO_2 emissions underscores the urgent need to adopt clean, renewable energy technologies, with solar energy emerging as one of the most promising. In the U.S., photovoltaic (PV) systems accounted for nearly 82% of new utility-scale generation capacity in 2024, totaling around 30 GW.[3] This growth is driven by a steep decline in the levelized cost of utility-scale solar energy[a], from 23¢/kWh in 2010 to 5.7 ¢/kWh in 2023[4], though it remains slightly above the 4.4 ¢/kWh cost of the cheapest natural gas combined cycle power generation.[5] To surpass fossil fuel technologies and achieve the U.S. Department of Energy's SunShot 2030 target of 3 ¢/kWh[6], further cost reductions in PV systems are essential.

With the cost of crystalline silicon (c-Si) modules—the incumbent PV technology—plateauing, low-cost alternatives with comparable power conversion efficiencies are needed to sustain the downward trend in PV energy costs.[7] Over the past decade, mixed organic-inorganic halide perovskites, commonly referred to as hybrid perovskites (HPs), have demonstrated a remarkable increase in efficiency—from 14% in 2013 to 27% in 2025—positioning them as strong

[a] Levelized cost of energy calculates the present value of the total capital, operational and maintenance cost over an assumed lifetime divided by the total energy produced, without considering the investment tax credit. The values shown here assume a 30-year operational lifetime, and averaged across different locations in the U.S.

candidates for next-generation PV technologies.[8] HPs exhibit exceptional optoelectronic properties, including tunable band gaps, high defect tolerance, and long carrier lifetimes.[9] Being solution-processable, these are compatible with low-cost, scalable manufacturing techniques such as roll-to-roll printing[10] and vapor deposition.[11] However, HPs are prone to rapid degradation under environmental stressors such as thermal stress, light, oxygen and moisture, following degradation pathways distinct from those observed in traditional c-Si absorbers.[12–15] In perovskite solar cells (PSCs), additional factors such as perovskite crystallization kinetics at interfaces,[16] degradation of non-perovskite ETL or HTL layers or their interfaces with the perovskite,[17,18] corrosion of metallic contact interfaces,[19] ionic migration under electric fields,[20,21] and trapping of volatile reactive species due to encapsulation,[22,23] further complicate these degradation pathways.

To assess if PSCs are a viable replacement for the incumbent c-Si technologies, it is vital to evaluate not only their power conversion efficiencies but also their durability under outdoor operational conditions.[24] A commonly used metric for this purpose is the t_{80} lifetime, defined as the time it takes for a device's power conversion efficiency to decline to 80% of its value prior to stressing.[25–27] Determining this metric typically involves testing PV devices under environmental conditions that are at least as severe as those encountered outdoors,[26] continuing until they reach their t_{80} lifetimes—experiments that can span weeks or even months. Although accelerated testing protocols have been proposed, these still require several hours or even days to complete and may exhibit significant variability, necessitating multiple replicates for each test condition.[25,27] As a result, it is beneficial to have an accurate predictive model that can quickly estimate the t_{80} lifetime of a PV device based on its initial quality, the early-time dynamics of its performance and the applied stress conditions, eliminating the need for full-duration, time-consuming experiments.

One approach to building a t_{80} prediction model is to derive expressions based on a mechanistic understanding of the underlying degradation processes. For instance, as thermally activated degradation often follows Arrhenius-type behavior,[a] many studies have used the inverse of early-time degradation rates in power conversion efficiency as proxy for effective t_{80} lifetimes.[28–31] While this approach may be feasible under simple stress conditions or for pristine perovskite absorber layers,[15] it becomes highly impractical for multi-layered PSCs subjected to general stress conditions, where the degradation mechanisms often follow complex and elusive pathways.[32] In such cases, statistical methods, particularly machine learning (ML) techniques guided by domain expertise, can be effective for constructing t_{80} predictive models. However, these models still require a representative set of experiments for training. Due to the extended length of PV durability experiments, the datasets available in laboratory settings are often too small for many common ML algorithms, especially deep learning models, which require large datasets to train effectively due to the large number of parameters they fit. Nevertheless, there are appropriate ML techniques that remain suitable for such data-scarce scenarios.

A previous report[21] from our lab demonstrated the use of sparse linear regression models (specifically, LASSO[33] and a hybrid *best-subset selection* scheme known as the $\ell_0\ell_2$ method[b]) to model t_{80} lifetimes of PSCs based on methylammonium lead iodide ($MAPbI_3$), a baseline

---

[a] It describes a non-linear dependence on temperature T as follows: $e^{-E_A/T}$, where E_A represents an effective positive activation energy (in temperature units).
[b] This method is explained in detail in Chapter 3.

perovskite composition. The best-performing model achieved a prediction error of 38%,[a] which is remarkably low given that it used only the first 90 minutes of time-series data to predict t_{80} lifetimes extending to thousands of minutes, all while relying on a dataset of just 45 accelerated durability experiments.[21] Such models provide a practical means to assessing the long-term feasibility of PSCs, even when data is scarce. While readers may be familiar with headlines[34] about advances in ML with big data, there have also been significant advances[35–43] in ML with small datasets like this—though these often receive little attention in research fields where they are most needed. Data scarcity, as encountered in PSC durability testing, is a common challenge across many experimental domains in chemistry and materials science. It is therefore worthwhile for scientists and engineers in these fields to be well-informed about statistical strategies specifically designed for small datasets.

Utilizing small in-house t_{80} lifetime datasets collected on two types of perovskite solar cells—methylammonium lead iodide ($MAPbI_3$) and formamidinium-cesium lead iodo-bromide ($FA_{0.8}Cs_{0.2}Pb(I_{0.83}Br_{0.17})_3$)—as case studies, this work provides an overview of the main tools and methods that modern statistics offers for learning target-feature relationships from small data. As alternatives to popular ML methods like neural networks, I employ classical statistical modeling techniques, focusing on *linear regression* of the form $\hat{\boldsymbol{y}} = \hat{\beta}_0 + \hat{\boldsymbol{\beta}}\boldsymbol{X}$. Linear regression with small data is a rapidly evolving field in statistics. This work reviews modern advances such as *sparse*

[a] This is the median test error evaluated from multiple models trained over the available dataset using a leave-one-out testing scheme, which is outlined in detail in chapter 3.

regression,[35,38] *knockoff filtering,*[36,37] and *conformal prediction.*[43,44] These methods are presented as part of a *workflow* (or *modelling pipeline*), from constructing features and training datasets to validating the model predictions. Further, I evaluate the capabilities of these methods on *synthetic datasets* (simulated datasets generated using a model with known 'ground-truth' features). These synthetic datasets recreate some of the common challenges in modeling real-world datasets encountered in science and engineering and provide a baseline for what to expect from real data.[45]

Finally, I demonstrate the application of these methods in predicting the t_{80} lifetimes using the two in-house PSC datasets based on $MAPbI_3$ and $FA_{0.8}Cs_{0.2}Pb(I_{0.83}Br_{0.17})_3$ perovskite compositions. In summary, this work aims to provide an accessible introduction to important recent results in statistics and their applicability to researchers in science and engineering. This is crucial for faster percolation and adoption of statistically sound analysis methods of small data in the areas of research where they are pivotal.

1.1 Outline

In Chapter 2, I contrast the large and small datasets using examples and discuss the underlying causes for data scarcity in the latter. Moreover, I also discuss mitigatory strategies—such as active learning and data augmentation—often employed by scientists and engineers in parallel with data acquisition, in contrast to the traditional approach of training ML models on already-collected data. When such proactive strategies are impractical, I discuss the advantages and disadvantages of applying high-parameter ML models, such as neural networks and tree-based ensembles to these datasets. In Chapter 3, I introduce criteria for determining when a dataset can

be considered *sufficient* or *small* for a given statistical method, with a particular emphasis on the use of sparse linear regression in *small* data scenarios. Furthermore, I discuss informatic-theoretic limits which serve as heuristic guidelines for identifying sample sizes that may be *too small* even for sparse linear models.

In Chapter 4, I review statistical machine learning tasks related to sparse linear regression for each part of the modeling pipeline, including constructing features, feature selection, determining model complexity, and quantifying uncertainty. In Chapter 5, I compare various statistical methods using synthetic datasets (with known ground-truths). Seeing how the various methods perform in this idealized case will give scientists and engineers an idea of what to expect with real data from their laboratories. Furthermore, I introduce specific characteristic properties of these methods, such as false-discovery and ground-truth recovery, which require precise knowledge of the ground-truths and are therefore not directly measurable in real-world scenarios. To assess these properties in specialized synthetic datasets (e.g., those where the ground-truth variables are deliberately excluded from the feature set available for modeling), I propose modified characterization metrics that are more general and robust.

In Chapters 6 and 7, I finally apply these methods on two real-world laboratory datasets developed in our lab—the t_{80} lifetime datasets of $MAPbI_3$ and $FA_{0.8}Cs_{0.2}Pb(I_{0.83}Br_{0.17})_3$. I compare these results with the simulation outcomes in Chapter 5 to apply heuristic insights gained from synthetic data. Throughout the manuscript, I use simplified notation and terminology (wherever possible) to make the work accessible to researchers from diverse disciplines, while providing references for more details or depth.

1.2 Notation

In general, the objective of a predictive model is to predict the unknown value of a variable of interest (referred to as the *target* variable) based on the known values of descriptive variables (referred to as *features)* by training on data that includes known values of both the target variable and the features (referred to as *training data*). The tacit assumption is that the target variable depends solely on these features.[46] By capturing empirical relationships between the target variable and the features through the training data, these models aim to predict target variable values in new, previously unseen trials, which constitute the *testing* data.[46]

Throughout this manuscript, bold typeface is used to represent sets, vectors, and matrices, while a regular typeface is used for individual variables and functions. Let Y be the target variable (also called the output variable) whose value is to be inferred based on a number (given by p) of other variables called features (or, equivalently, descriptors, inputs, attributes or covariates) whose values are known. Let the features be collectively represented by the set $\boldsymbol{F} = [F_1, F_2, \dots, F_p]$. Let the predictive model for Y be denoted by $f(\boldsymbol{F})$, which is a deterministic predictor even though the measured values of Y may include noise. If f is chosen to be linear, it takes the form $f(\boldsymbol{F}) = \beta_0 + \sum_{j=1}^{p} \beta_j F_j$, where $\boldsymbol{\beta} = [\beta_1, \beta_2, \dots, \beta_p]$ is the vector of parameters or coefficients, each corresponding to a specific feature. The term β_0 is an additive constant, commonly referred to as the intercept. The values of $\boldsymbol{\beta}$ and β_0 are obtained by fitting (or, equivalently, training or learning) f over a training dataset where the values of Y along with all features are known.

In a typical modeling task, each independent measurement of the target variable y_i is paired with its corresponding feature values $\boldsymbol{x_i}$, where $\boldsymbol{x_i} = [x_{i1}, x_{i2}, \dots, x_{ip}]$ denotes the p-dimensional

row vector of feature values corresponding to y_i. The pair $(\boldsymbol{x_i}, y_i)$ is referred to as a data-point and represents the results of a single experimental (or computational) trial or run. The total number of data-points is denoted by N^{tot} and represents the *sample size*. During modeling, the collection of data-points (the dataset, or simply the data) is divided into two collections referred to as training data and test data. Here, N represents the sample size of the training dataset, and $\boldsymbol{y}$ denotes the N-dimensional column vector of values of the target variable, $\boldsymbol{y} = [y_1, y_2, \dots, y_N]^T$, where superscript T denotes matrix transpose. The row-wise concatenation of the corresponding p-dimensional feature-data row vectors, $[\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_N}]^T$, forms the *design matrix* $\boldsymbol{X}$ with dimensions $N \times p$. Each column $\boldsymbol{X_j}$ in $\boldsymbol{X}$ corresponds to the values of the feature F_j across the N data-points.

In addition to the values of the $\boldsymbol{\beta}$ and β_0 parameters, fitting the model to training data also selects a subset of features $\boldsymbol{F^S}$ from $\boldsymbol{F}$. Let $\boldsymbol{y^{pred}} = \left[y_1^{pred}, y_2^{pred}, \dots, y_N^{pred}\right]^T$ denote the N-dimensional column vector of predicted target values where each prediction y_i^{pred} is obtained by evaluating the fitted model at each $\boldsymbol{x_i}$, Each feature F_j not selected by a particular fit has a corresponding coefficient value equal to zero ($\beta_j = 0$). The size of $\boldsymbol{F^S}$ is referred to as the *level of sparsity,* denoted here by s (where s is the cardinality of the set with $1 \leq s < p$). When s is significantly smaller than p, that is, when the majority of the coefficients in $\boldsymbol{\beta}$ are zero, the model is said to be *sparse*. As $\boldsymbol{y^{pred}}$ depends only on the $\boldsymbol{F^S}$ features, a sparse model represents a situation where the target variable is predicted well by a small subset of the features. Ideally, the learned model must only select relevant features, and 'zero out' the coefficients of the remaining irrelevant ones. When N is small, sparsity becomes important to train the predictive models reliably. Selecting $\boldsymbol{F^S}$ from among a set of p features is called *feature selection*; along with sparse

regression, it is a central topic of this review. For a summary of notation, see Table 4.2 in the Supplementary Information section of Chapter 4.

2 MACHINE LEARNING WITH SMALL DATA IN SCIENTIFIC RESEARCH

For centuries, simple data-driven linear models (like linear regression) have been used in science engineering[47,48] to make predictions in situations where mechanistic models are impractical due to a limited understanding of the underlying processes[48] or the complexity of the underlying equations. Over the past several decades, the sophistication of data-driven predictive modeling has increased dramatically,[49] employing various ML tools such as neural networks and tree-based ensembles. When data is abundant, these models have proven highly effective in tackling prediction tasks across various domains of scientific research. However, many areas of scientific research continue to face persistent data scarcity, requiring the use of simpler ML tools. This scarcity is often caused by high cost or difficulty of data generation. While proactive strategies like active learning[50] and data augmentation[51] can help mitigate these challenges during data collection, they are not universally applicable. Although training ML models on small data can offer some advantages, it also introduces significant risks to model reliability. This chapter briefly explores each of these topics.

2.1 Predictive Modeling with ML across the Data Spectrum

ML has evolved as a go-to approach for predictive modeling across many scientific domains. The sophisticated *black-box* nature of many ML methods enables them to effectively capture intricate underlying predictive mechanisms.[48] However, their remarkable success across scientific disciplines mainly depends on the availability of large training datasets to satisfy their *data-hungry* nature.[52] For instance, in the field of material science, Li et al.[53] used an ML method

called *LightGBM* (Light Gradient Boosting) to predict thermoelectric material quality based on a dataset of over 5,000 materials and 57 features. In the field medicine, Arai et al.[54] trained an alternating decision tree on clinical data from approximately 27,000 patients, described by 15 features, to predict the risk of a certain complication following stem cell transplant.

In other areas like protein folding,[55] bioinformatics,[56] and meteorology[57] even larger datasets are available where the number of data-points outnumber the features by several orders of magnitude. In many of these situations *deep neural networks* may be used to make astounding predictions. For example, *AlphaFold*[34] is a neural-network-based model that predicts the *in vivo* folded structure of a protein from its linear amino acid sequence. The model was trained on approximately 10 million amino acid sequences, using over 33,000 features derived from each sequence and its homologues. In 2020, *AlphaFold* significantly outperformed other models based on mechanistic approaches and simpler ML techniques.[58] The success of the approach and the impact it has had on molecular biology was recognized by the 2024 Nobel Prize for Chemistry. The remarkable success of *AlphaFold* is an inseparable result of both the algorithm and the availability of a large training dataset generated over many decades.

However, in many scientific settings data is scarce, and there is significant time and cost associated with generating the requisite data.[51,59] As a result, it is not possible to use the data-hungry techniques mentioned above with modeling tasks that have only hundreds, or even dozens, of data-points. However, there are appropriate techniques that can be used. For instance, Avadhanula et al.[60] examined the diagnosis odds of human thyroid dysfunction in patients with *alkaptonuria*, a rare inherited disorder, using *logistic regression* with more than a dozen descriptive features. The training dataset for this study consisted of only 125 patients, due to the disease's extreme rarity. In a separate study, Meng et al.[15] developed a predictive model for the degradation

rate of perovskite photovoltaic thin films that uses *best-subset selection* to select 3 features (from a menu of 15) based on 42 controlled degradation experiments. Meanwhile, Nasonova et al.[61] predicted certain properties of aqueous soil extracts using *partial least squares linear regression* with over 1800 features derived from mid-infrared absorption spectra, collected from just 216 soil extracts due to the study's seasonal dependence and logistical constraints. The number of patients, experiments, or samples in each of these is not only small but also comparable to, or even smaller than, the number of features. As a result, data scarcity remains a significant challenge for ML in many scientific domains. In such cases, scientists and domain-experts must leverage appropriate statistical methods and computational tools to overcome this challenge.

2.2 What leads to Small Datasets?

In many scientific fields, the datasets are small because of high data collection costs, often associated with instrumentation, personnel or materials. Costs may be particularly high for long-duration experiments such as photovoltaic device reliability testing[62] that can span years, longitudinal medical studies involving tracking patients for months or years[63], and molecular simulations demanding hundreds or thousands of high-performance computing (HPC) hours.[64] Sometimes, the rate of data collection cannot be controlled; for instance, obtaining data about a rare disease or condition is limited by the number of individuals having that condition.[65]

Sometimes, labeling data, which is typically performed by humans, adds significantly to the data collection costs. While many traditional labeling tasks like simple object recognition can be handled by laypersons, specialized tasks such as identifying tumors from images require domain experts.[51] Moreover, labeling may also involve additional experiments. For instance, in material

science, researchers may need to run specific experiments to obtain the desired labels (e.g. the degradation lifetimes of a variety of photovoltaic material compositions), in addition to the experiments required to collect feature data (e.g. initial material characterizations). These added costs associated with labeling often limit the sizes of scientific datasets.

The 'curse of dimensionality' is another challenge frequently encountered in many scientific settings. Each trial, run, or experiment may generate a lot of data, and thus there may be many features (large p). For instance, a single image (from optical microscopy) or a single spectrum (from mass spectroscopy, NMR, XRD, etc.) yields a large number of potential descriptors (edges, shapes, peak positions, etc.). In these and similar situations, the number of potential features often exceeds the number of trials ($p > N$). Thus, even when N is large, such datasets may still be considered small due to the relatively large p. Reaching sound conclusions in these data-scarce situations requires careful use of statistics and appropriate computational tools.

2.3 Advantages and Risks

Given the prevalence of small datasets are in many scientific domains, it is important to understand how these can affect the modeling process. From a computational standpoint, small data is advantageous for economical data processing, modeling and visualization. Many statistical methods (such as exhaustive searches for the best subset of features from a feature menu) combined with leave-one-out cross-validation scale significantly with the size of the dataset.[46] Consequently, while these methods become infeasible for large datasets, they are easily applicable to small datasets. However, from a statistical perspective, models trained with small datasets risk

failing to generalize effectively to unseen test data. When N is small compared to p, the fitted model becomes highly sensitive to variations in the target variable (including noise), leading to poor predictions on test data. This phenomenon, known as *overfitting*, is particularly pronounced when models with a large number of parameters (such as neural networks) are trained on small datasets.[46]

2.4 Are Proactive Strategies Effective?

To address the risks associated with small datasets, many strategies have been explored, particularly in experimental domains of chemistry and materials science.[66] Rather than exploring ML techniques suitable for already-collected small datasets, experimentalists have increasingly sought strategies integrated into the data acquisition process itself (e.g., data augmentation[67,68] and active learning[50]) or designed to borrow statistical power from external datasets (e.g., transfer learning[69–72]). In some domains, these strategies have gained popularity and have been shown to improve predictive performance.[66] However, they may not be universally applicable across all domains.

2.4.1 Data Augmentation

Data augmentation techniques, which use transformative or generative algorithms to synthetically expand training datasets, are often proposed as straightforward solutions to address the small data challenge. Transformative algorithms are typically applied to image data and involve simple operations such as cropping, translation, and rotation to generate new data.[51] Generative

algorithms, on the other hand, rely on mechanisms developed using domain expertise or statistical modeling to produce artificial data. For instance, Lee et al.[68] used mechanistic simulations to artificially generate X-ray diffraction (XRD) pattern replicates for a small dataset of solid-state electrolytes by introducing slight perturbations to the lattice parameters. However, this approach becomes impractical in certain fields, such as medicine, where simulating the true governing processes is highly complex or impossible.[67] In such cases, an alternative strategy is the use of deep learning tools like generative adversarial networks (GANs), which employ two neural networks—one to generate artificial data and the other to evaluate it against real data.[67] Nevertheless, these tools require large training datasets to avoid overfitting, making them less effective for applications with limited data.

2.4.2 Data-Efficient Modeling

Many researchers have proposed *active learning*[50,70] as an effective strategy to model small datasets. In this approach, new trials are iteratively added to the training data using a Bayesian approach (called adaptive sampling) based on the predictions of the model trained from the previous iteration's data. Though this is a clever approach to sample a large prospective training data space for an optimization problem, it does not fundamentally address the issue of predictive modeling with small data. It still requires training a regression model at each iteration using the limited available data and fixed feature set.

Another common example is *transfer learning*[66,69] which involves re-adjusting the parameters of an off-the-shelf model, typically a deep neural network, that is previously trained on a large dataset to fit well over a smaller dataset at hand. This strategy is inapplicable in many situations because it requires these datasets to share identical or similar features[69,72] and

mathematical relationships.[73] Though some reports[71] show low prediction errors achieved via transfer learning, this comes at the expense of decreased model interpretability. Considering these limitations, scientists and engineers need alternative modeling techniques that are interpretable and directly applicable to already-collected datasets.

2.5 Summary

Small data is a prevalent challenge in scientific domains such as material science and chemistry, often arising from the time and cost constraints associated with data collection. Many ML methods, such as neural networks and tree-based ensembles, although widely known for their ability to capture intricate underlying predictive mechanisms,[48] require large training datasets which are unattainable in these small data settings. When applied to small datasets, they pose a high risk of overfitting. Although some researchers attempt to mitigate the small data problem through proactive strategies such as active learning and data augmentation, these approaches are not universally applicable. This necessitates the use of modeling techniques that can be effectively applied on already-collected small datasets.

In Chapter 4, I review various modeling methods and data-efficient strategies developed by statisticians to address this issue. However, before selecting an appropriate strategy or model, it is important to first assess whether the given dataset is *small* or *too small* for the intended method. In the next chapter, I introduce certain heuristic rules that can guide this process.

3 DEMARCATING THE SMALL DATA REGIME

Considering the risks associated with small datasets, it is worthwhile to identify the conditions under which a given dataset can be considered *small*. This process of "demarcating" the *small data regime* is particularly interesting to many scientists and engineers as it helps validate the applicability of a statistical method. However, there is no consensus on how this demarcation should be carried out.

From a scientific or modeling perspective, a dataset may be considered *small* when its sample size N is insufficient for a purely data-driven approach to distinguish between possible predictive models. In such cases, domain knowledge (including laws of nature, heuristics, and human expertise) is required to constrain the parametric form or the features entering the model. However, even with such knowledge, one may need to further constrain the model to be as simple as possible. For instance, using linear regression acts as a useful constraint, as it limits the number of parameters approximately to the number of features p.

Chapter 4 discusses various ways to achieve simplicity in modeling by feature construction and feature selection. In this chapter, I focus on how model simplicity—specifically, *sparsity* in the context of linear regression—is linked to the sample size requirements for reliable modeling. Based on this relationship, one can define heuristic lower bounds on N, below which the data enters the *small data regime*. Furthermore, to identify sample sizes that may be *too small*, I draw on guidelines from statistical and information-theoretic literature.[38,74–77] Throughout the chapter, the focus remains on linear regression, and the demarcating sample sizes discussed here serve as strict conservative lower limits for models with higher complexity, such as neural networks.

3.1 The Role of Effective Degrees of Freedom

From a statistical viewpoint, whether the dataset falls within the *small data regime* depends not solely on N, but on its value relative to the *number of (effective) free parameters* tuned while training the model, also known as the *effective degrees of freedom,* or *model complexity*. For example, in classical linear regression, these parameters correspond to the feature coefficients β (and the intercept β_0) whose values are estimated during model training. Let m denote the number of such degrees of freedom for a given model and let ratio N/m denote the number of data-points per degree of freedom. A high N/m ratio is statistically desirable for reliably fitting a model. Consequently, when N is small, it is important to choose a model with a small m to ensure a reasonably high N/m ratio. In any model where p features are used, the number of parameters m is at least as large as p, suggesting that fewer features in a dataset are preferable. Additionally, caution is needed when using certain ML models, such as tree-based ensembles and neural networks, where m can significantly exceed p. Instead, one can opt for a model with a lower m, such as classical linear regression where $m \approx p$ (or $m = p + 1$ when the intercept is not ignored). However, when p is large—making m relatively high—even classical linear regression may not be a viable option.[38]

Mathematically, applying a statistical model requires $N/m > 1$, a *sufficient* condition indicating that there is at least one data-point for every degree of freedom. However, as a rule of thumb, N is typically required to be at least an order of magnitude larger than m to ensure a reliable fit and prevent overfitting. Thus, $N/m = 10$ serves as a heuristic threshold, holding in most cases

unless the data contains a high level of noise, in which case N/m may need to be significantly higher. If a small N causes the ratio N/m to fall below the threshold of 10, the dataset may be insufficient to prevent overfitting unless m is reduced to raise N/m above the threshold. Keeping p constant, this reduction in m can be achieved in linear regression by using a *sparser* model, where only a subset of s features ($1 \leq s < p$) from the full feature set $\boldsymbol{F}$ are selected to influence the predictions.

Suppose the underlying generative mechanism of Y is governed by p_0 ground-truth variables, which are also present in the available feature menu $\boldsymbol{F}$. An ideal sparse model would select exactly these variables, resulting in $s = p_0$—a task often referred to as *exact sparse recovery*. By attempting to identify and use only the ground-truth variables from $\boldsymbol{F}$ for prediction, a sparse model uses a smaller value of m, denoted as m_s^0.

Figure 3.1 summarizes how a dataset may be classified as *sufficient*, *small* or *too small* for fitting a linear regression model based on its sample size N relative to $m = p$ and $m = p_0$. The dataset may be considered *sufficient* when N is large enough to prevent overfitting (which heuristically holds when $N \geq 10p$). It may be considered *small* when N is insufficient to prevent overfitting, unless a sparser model is employed (as indicated by $m_s^0 \leq N < 10p$). Within the *small data regime*, overfitting may still occur for $m_s^0 \leq N < 10m_s^0$ and intensify as N approaches m_s^0, underscoring the need to consider different types of sparse models (see Chapter 4).

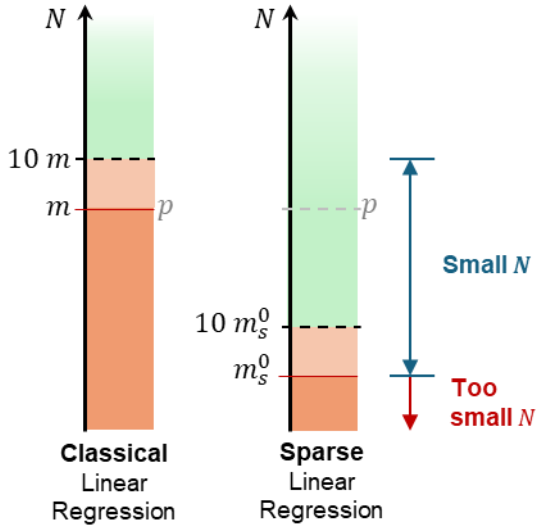

Figure 3.1. Schematic illustrating the ranges of *N* where data is considered *small* or *too small* for linear modeling. Let m and m_s^0 represent the degrees of freedom in a classical linear model and a sparse linear models performing exact sparse recovery, respectively. In the classical model, where $m = p$, the following thresholds apply: $N \geq 10m$ (green) indicates data *sufficient* for reliable modeling, $m \leq N < 10m$ (light orange) indicates data that is just sufficient but carries a risk of overfitting, and $N < m$ (dark orange) indicates data that is insufficient for modeling. The range of N values that are insufficient for reliable classical linear regression but still allow sparse linear modeling (where $m_s^0 < p$) is considered the *small data regime*. However, when $N < m_s^0$, the dataset becomes *too small* even for sparse linear modeling.

3.2 'Too small' data regime

When $N < m_s^0$, the dataset is considered *too small*, as the model is unlikely to accurately recover the ground-truth variables and may become unreliable. As m_s^0 plays a key role in demarcating this '*too small*' regime, it is an important factor to assess before performing any modeling task on a given dataset. However, to do so, one must first understand how the effective degrees of freedom m_s^0 is linked to the ideal sparsity level $s = p_0$ and the total number of features p in the dataset.

While it is straightforward to see that the effective degrees of freedom in a classical linear regression model total to p, evaluating this becomes more complex for a sparse model. Although only s features ultimately influence the predictions in a sparse model, identifying these features from the full set $\boldsymbol{F}$ requires more than s effective degrees of freedom than s.[38] Information theory provides a framework for approximately estimating the relationship between m_s^0, $s\ (= p_0)$ and p.[75] Depending on the degree of correlation among features—a property often referred to as *coherence*—and the true underlying coefficients of the ground-truth variables, various expressions have been derived to estimate m_s^0 in different studies.[74–76,78] Implicit in these analyses is the assumption of exact sparse recovery, meaning that all the ground-truth variables are present in $\boldsymbol{F}$ and are exactly recovered, such that $s = p_0$.[a] Here, I briefly present the estimates of m_s^0 derived for two generic cases: one where features in $\boldsymbol{F}$ are completely orthogonal (i.e., *incoherent*), and another where there are some correlations among the features (i.e., *coherent*). Then, I explain how these estimates can be evaluated in common real-world scenarios.

3.2.1 Incoherent Feature Data

If all possible feature subsets of size $s = p_0$ are considered from the full set $\boldsymbol{F}$ to recover the fixed ground-truth features, the sparse modeling algorithm must search over all such subsets, totaling $\binom{p}{p_0}$ combinations. Here, $\binom{p}{p_0}$ represents the number of unique ways to select p_0 features

[a] As exact sparse recovery is challenging in real-world scenarios, sparse models are often non-ideal. Nevertheless, estimating m_s^0 remains valuable, as it provides a baseline lower bound for the effective degrees of freedom that can be expected in such non-ideal models.

from a pool of p features. In information theory, the effective degrees of freedom correspond to the number of independent pieces (or *bits*) of information required to evaluate all these subsets of size $s = p_0$. Consequently, m_s^0 can be expressed as:

$$m_s^0 = \log_2 \binom{p}{p_0} \tag{1}$$

This holds only when the feature data columns in the design matrix $\boldsymbol{X}$ are strictly orthogonal, or in other words, the features in $\boldsymbol{F}$ are strictly uncorrelated or only weakly correlated to each other. In statistics, this property is known as *incoherence*.[75] When $p \ll p_0$, m_s^0 is commonly simplified as follows:

$$m_s^0 = \log_2 \binom{p}{p_0} = \log_2 \left(\frac{p!}{(p-p_0)! \cdot p_0!} \right) = \log_2 \left(\frac{p \cdot (p-1) \cdots (p-p_0+1)}{p_0!} \right)$$

$$m_s^0 \approx \log_2 \left(\frac{p^{p_0}}{p_0^{p_0}} \right) = p_0 \log_2 (p/p_0) \tag{2}$$

This is a remarkable result for very high dimensional settings (where $p \gg N$) as it indicates that a sparse linear model only requires $m_s^0 \approx p_0 \log_2 (p/p_0)$ effective degrees of freedom, instead of $m \approx p$, to recover the ground-truths. In signal processing, this approach to recovering a full signal from an under-sampled one became widely popular in the late 2000s under the name *compressed sensing*.[74] However, this requires that the design matrix $\boldsymbol{X}$ be strictly *incoherent*, a property commonly encountered in signal compression and processing. For coherent matrices, estimating m_s^0 becomes slightly more complex.

3.2.2 Coherent Feature Data

In many scientific domains of chemistry and materials science, obtaining a strictly incoherent $\boldsymbol{X}$ is nearly impossible. This is because many features—despite being correlated with multiple others—are still retained in $\boldsymbol{F}$, due to unknown and complex interdependencies with the ground-truth variables, some of which may not even be directly present in $\boldsymbol{F}$. As a result, estimating m_s^0 can be more complex than previously outlined.

To obtain generic estimates for $\boldsymbol{X}$ with non-zero correlations, Wainwright[75] used simulated design matrices with two types of correlation structures (as shown in Figure 3.2), as follows: (a) one in which all features are mutually correlated with a Pearson correlation coefficient ρ (where $\rho > 0$), and (b) another in which only the ground-truth features are correlated with each other, while the remaining features are uncorrelated with both each other and the ground-truths. Moreover, each feature follows a standard Gaussian distribution. The estimate for m_s^0 in both cases is given below:

$$m_s^0 \approx \frac{p_0 \log_2(p - p_0)}{1 - \rho} \tag{3}$$

The $(1 - \rho)$ in the denominator accounts for the effects of *coherence* in $\boldsymbol{X}$, indicating the need for more degrees of freedom to accurately identify the ground-truth variables. For both design matrices shown in Figure 3.2, the correlation between any ground-truth feature and a non-ground-truth feature never exceeded the correlation among the ground-truth features themselves. As long as this condition is satisfied, eq (3) remains valid.

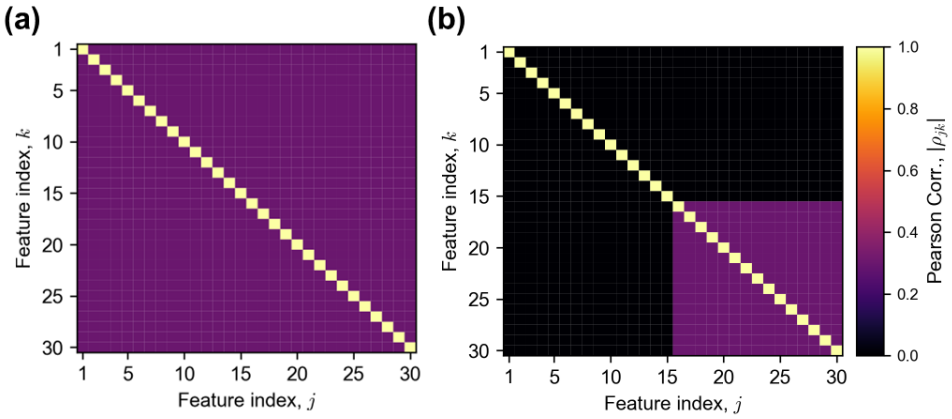

Figure 3.2. Correlation structures of the two types of design matrices used by Wainwright[75] to derive the m_s estimates as per eq (3). The m_s estimates assume an ideal sparse model capable of exact ground-truth recovery with $s = p_0$. For clear visualization, a set of $p = 30$ features are used in the figures, where the last p_0= 15 features are designated as ground-truths. **(a)** All features are mutually correlated with a Pearson correlation coefficient ρ = 0.3. **(b)** Only the ground-truth features are corelated to each other at ρ = 0.3, while the remaining features are orthogonal to one another and the ground-truths.

Table 3.1 lists the values of m_s^0 estimated using eq. (1)-(3) for varying values of p, p_0 and ρ. While all three expressions yield similar estimates for small p_0, the estimates begin to diverge as p_0 increases, with eq. (3) providing conservatively higher values. Figure 3.3 demonstrates these differences for increasing values of p_0, with p fixed at 50.

Table 3.1. Estimating the effective degrees of freedom m_s^0 used by a sparse linear model for with varying sparsity levels (s) and feature set sizes (p), using eq. (1), (2) and (3). For eq. (1) and (2), no correlations between features are assumed, while eq. (3) uses ρ, which represents the mean Pearson correlation coefficient between all possible pairs of features. All values in the table are ceiled to the nearest integer.

	$m_s \approx \log_2 \binom{p}{s}$ (eq. 1)		$m_s \approx s \log_2(p/s)$ (eq. 2)		$m_s \approx \frac{s \log_2(p-s)}{1-\rho}$ (eq. 3)			
					$\rho = 0$		$\rho = 0.2$	
	$p = 20$	$p = 50$	$p = 20$	$p = 50$	$p = 20$	$p = 50$	$p = 20$	$p = 50$
$s = 1$	5	6	5	6	5	6	6	8
$s = 2$	8	11	7	10	9	12	11	14
$s = 3$	11	15	9	13	13	17	16	21
$s = 4$	13	18	10	15	16	23	20	28
$s = 5$	14	22	10	17	20	28	25	35

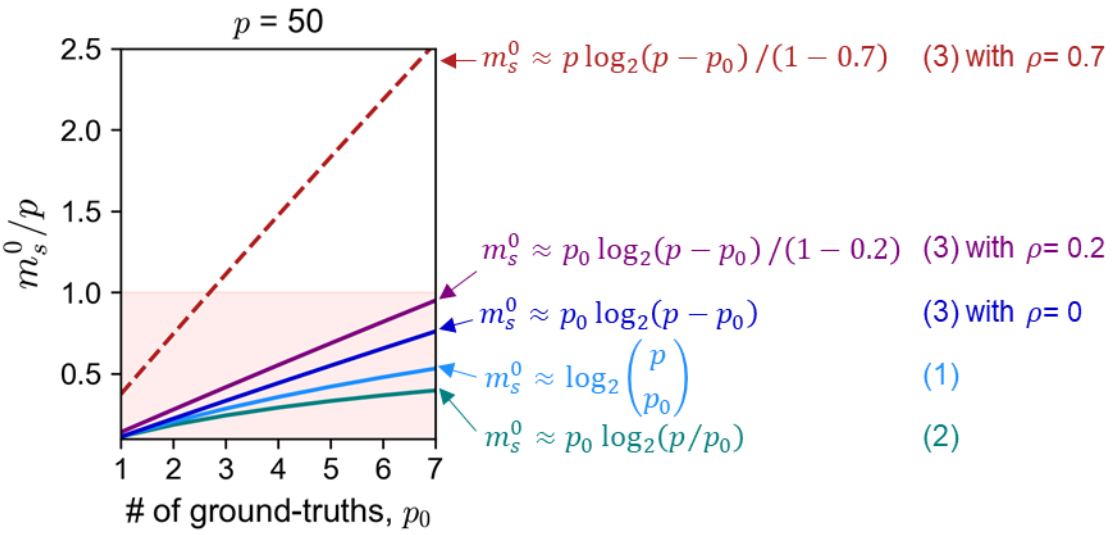

Figure 3.3. Effective degrees of freedom m_s^0 estimated using eq. (1), (2) and (3). The shaded region represents the range of m_s^0 values which are smaller than p.

3.2.3 Real-world Data

Eq. (1)-(3) provide generic estimates of m_s^0, applicable to many real-world scenarios. However, two main complexities often hinder accurate estimation: (1) the lack of knowledge about the ground-truth variables, and thereby the exact value of p_0, and (2) the presence of a complex correlation structure among features. The first issue can be addressed by assuming a plausible range of p_0 (guided by domain knowledge) and estimating m_s^0 for each value within that range. This helps assess whether the dataset is *too small* for the assumed range of p_0. To address the second issue, one must assess the distribution of pairwise correlations among the features before applying eq. (1)-(3).

In many real-world scenarios, where a small but non-negligible correlation coefficient ρ, often below 0.3, is common among features due to noise, eq. (3) provides a more reasonable and general estimate of m_s^0. In cases where a few feature pairs exhibit high correlations (say, > 0.7), this estimate still remains applicable by using a ρ value that reflects the average degree of correlation among the features. However, in many cases, the assumption of exact sparse recovery becomes inapplicable, potentially rendering eq. (3) unusable, particularly when the ground-truth variables are not directly present in $\boldsymbol{F}$, but only indirectly through correlated features. For instance, if two features in $\boldsymbol{F}$ are strongly correlated and both are predictive of Y, it becomes difficult to determine which one represents a ground-truth, as both may serve as 'proxies' for a missing ground-truth variable. In such cases, the objective shifts to identifying as many relevant features as possible in a parsimonious manner, while minimizing the inclusion of irrelevant ones. Nevertheless, eq. (3) still serves as a heuristic lower bound for m_s^0 in such cases, helping to determine the minimum sample size N before it becomes *too small*.

As a general rule when using eq. (3), the mean absolute Pearson correlation coefficient over all possible pairs of features in $\boldsymbol{F}$ can be used to assign ρ. Alternatively, to simulate an extremely conservative scenario, ρ can be set to the maximum absolute Pearson correlation coefficient. The red dashed line in Figure 3.3 illustrates how the estimates of m_s^0 increase significantly when ρ is set to 0.7. In conclusion, it is important to note that eq. (3)—and even eq. (1) and (2)—serve only as heuristic guidelines for estimating approximate values of m_s^0, and thus the lower bounds on sample sizes. Determining the exact information-theoretic limits remains challenging for real-world datasets, where features often exhibit non-Gaussian distributions and complex correlation structures.

3.3 Summary

In summary, when $N \geq 10p$, the dataset may be heuristically considered *sufficient* to prevent overfitting. If a sparse model is employed and m_s^0 is assumed to represent the effective degrees of freedom required to recover p_0 underlying ground-truth variables, then when $m_s^0 \leq N < 10p$, the dataset can be considered to fall within the *small data* regime. It is important to note that overfitting is still likely for $m_s^0 \leq N < 10m_s^0$, and the risk increases as N approaches m_s^0. This underscores the need to consider different types of sparse models (as discussed in Chapter 4). When $N \leq m_s^0$, the dataset is considered to be in the *'too small'* regime, where it becomes intractable regardless of the sparse model chosen. Eq. (3) can be used to identify this boundary.

4 WORKFLOW FOR MACHINE LEARNING WITH SMALL DATASETS

Figure 4.1 illustrates a general workflow for estimating a regression model with a small dataset. Given a target variable Y, a domain expert begins by collecting all known or measurable variables that may be relevant to inferring Y. These variables, referred to as primary features, can be used directly as features, transformed individually, or grouped and combined to create new features. This process, known as *feature construction*, is guided by domain expertise to include physically meaningful features in $\boldsymbol{F}$, without relying on the observed target values. Each data-point in the dataset now consists of the measured or calculated target value paired with its corresponding feature values.

Next, the data is divided into two groups, a training dataset and test set. A sparse modeling algorithm then uses the training data to select the most predictive subset $\boldsymbol{F^S}$, consisting of s features, from a total of p features in $\boldsymbol{F}$. This process of *feature selection* is performed either explicitly as a separate step or implicitly during the parameter estimation step. To determine an appropriate value for s, cross-validation—a procedure that repeatedly splits the dataset into distinct training and validation sets—or another tuning or regularization method is employed. A procedure similar to cross-validation can also be applied at the initial stage to split the data into training and test sets. Once s and $\boldsymbol{F^S}$ are determined, the sparse parameter vector $\boldsymbol{\beta}$ and intercept β_0 are then obtained by training an ordinary least square linear regression once more on the entire N data-points but using only the selected features $\boldsymbol{F^S}$. For a test data-point with feature values $\boldsymbol{x'} = \{x'_1, \dots, x'_p\}$, the target value is predicted using $f^*(\boldsymbol{x'})$, where f^* represents the trained model. At this stage, conformal prediction[44], a framework for estimating model uncertainty at $\boldsymbol{x'}$, can be used

to produce a confidence interval around $f^*(\boldsymbol{x}')$. Each of these steps are discussed in detail in this chapter.

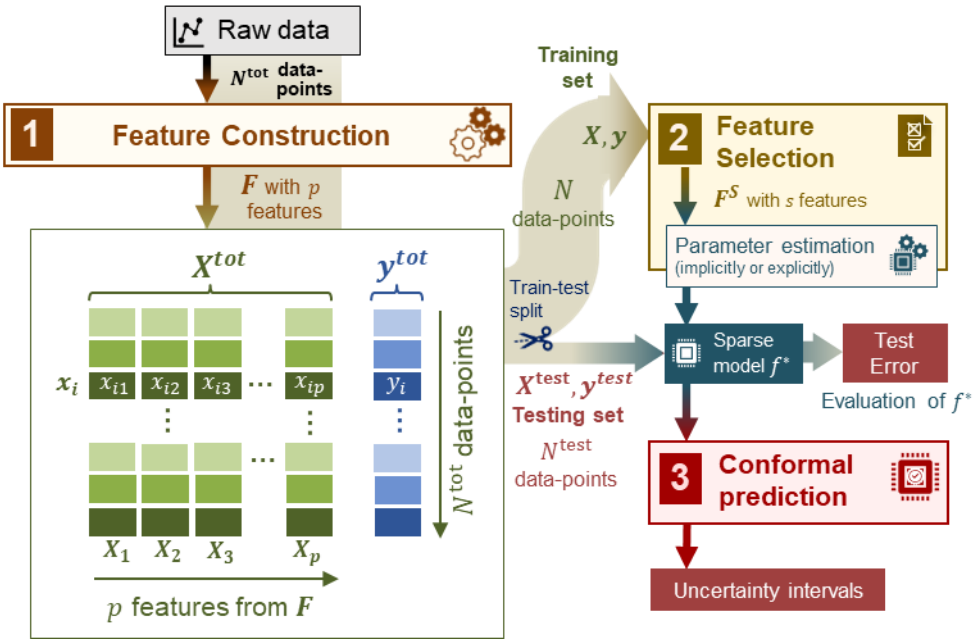

Figure 4.1. Schematic showing the general strategy for modeling small datasets.

4.1 *Feature Construction*

Feature construction encompasses all operations involved in preparing the feature menu $\boldsymbol{F}$, performed without accessing the observed values of the target variable. In each operation, leveraging domain expertise is crucial to ensure the accuracy and interpretability of the final learned model f^*. Feature construction begins with compiling a set of primary features—measurable variables that the domain expert believes can inform about Y. Subsequent statistical procedures assume that all explanatory variables that are indispensable for inferring Y have been included in this list of primary features. This leads to the tacit assumption in statistical prediction

that Y can be explained accurately up to independent noise with the available features. However, statistics alone cannot determine whether any such variables have been missed. So, it is the domain expert's responsibility to compile the list of primary features comprehensively to avoid such omissions.

For example, in material science, a scientist may attribute the variability of material properties to certain ground-truth variables describing the underlying atomic and molecular structures, local electronic environments, particle arrangements and phase boundaries. However, in practice, the values of these variables cannot be measured or calculated. Therefore, scientists typically rely on empirical or *in silico* experiments to probe seemingly relevant primary feature variables, some of which may indirectly capture the underlying relationships between the ground-truth and the target variables. While some of these features can be used directly as features for modeling, new feature variables can be created by applying specific operations or transformations to other features. In some cases, separate experiments may be required to construct physics-, chemistry-, or biology-informed features.

Often, experiments result in high-dimensional, highly correlated data-points. A typical example is image data, with each data-point containing thousands of interrelated pixel values. In large data settings, it may be possible to incorporate each pixel or measurement as a feature in the model. For instance, Lee et al.[79] accurately predicted the properties of certain inorganic materials by using powder XRD spectra, where each spectrum's 8192 intensity values, measured at specific 2θ positions, served as distinct features.

For small data, it is more practical to reduce these primary features to simple statistics or descriptors most relevant to the problem of interest. This could include initial slopes or curvatures

in monotonic time series, peak characteristics (e.g., positions, heights and widths) in intensity spectra, and pixel statistics (e.g., mean and variance) in images. For instance, Suzuki et al.[80] also used the XRD spectrum for property prediction but only considered the first ten peak positions and the number of peaks as features rather than using the entire raw spectrum. However, as many such simplifying reductions are possible for a dataset, it is essential for researchers to apply domain knowledge to extract the most *physically meaningful* features from the primary features. This process may involve specific non-linear transformations and operations, which are particularly useful for enabling a model that is linear in its features to capture non-linear patterns in the data to some extent. This was done to great effect in Menon et al.[81], a study on rheology that investigated the effects of dispersing polymers in aqueous suspensions. The authors systematically derived physics-inspired features from measurable primary variables such as polymer composition, viscosity and osmotic pressure, and subsequently fit a sparse linear model (namely LASSO[33]).

Sometimes, constructing physically meaningful features requires separate experiments that are distinct from those used solely to collect data for modeling. For instance, Dunlap-Shohl et al.[21] predicted the performance lifetimes of perovskite solar cells under varying environmental conditions using a feature (a kinetic rate expression that is a non-linear function of the experimental conditions) that was derived from separate experiments[14] on perovskite film degradation.

4.2 *Feature Selection*

Feature selection involves selecting a subset $\boldsymbol{F^S}$ (comprising s features, where $1 \leq s < p$) from the feature menu $\boldsymbol{F}$ that predicts Y more effectively than the full set. As the underlying

mechanism generating the values of Y typically depends on certain ground-truth variables, feature selection methods aim to identify these variables from $\boldsymbol{F}$. However, real-world datasets often lack *all* the ground-truth variables in their feature menus. Instead, they contain features that may influence Y indirectly through correlations with the missing ground-truth variables, which are sometimes referred to as *unmeasured confounders*.[82] This makes feature selection more challenging in real-world settings, requiring more robust schemes. In this section, I discuss various feature selection methods that may be suitable for such datasets.

In any extensive feature set $\boldsymbol{F}$, designed to incorporate as many ground-truth variables as possible, either directly or through correlated features, some features may contribute to predicting Y, while others may not. A good feature selection method accurately identifies the *relevant* features that provide high predictive power, while avoiding *irrelevant* ones that unnecessarily inflate the size of the subset $\boldsymbol{F^S}$. Feature selection schemes broadly vary based on their methodology (e.g., filter-based, wrapper and embedded methods) and the characteristics of $\boldsymbol{F^S}$ they aim to control (e.g., minimizing prediction error, selecting relevant features without redundancy, or avoiding irrelevant features).[83] Among these methods, *filter-based* approaches are the simplest. Here, features are ranked according to their ability to explain the variation observed in $\boldsymbol{y}$, and the top-ranking features are selected. Commonly used metrics for ranking include correlation metrics like the Pearson coefficient[42] and the Kendall coefficient[84], as well as information-theoretic metrics like the mutual information[85]. However, the major drawback of these filter-based methods is their tendency to overlook features that appear irrelevant when assessed individually but prove relevant in combination with other features. Additionally, there is a risk of mistaking spurious correlations for genuine ones with this method.

Wrapper methods incrementally select a feature subset $\boldsymbol{F^S}$, generally using a filter-based method at each iteration to guide the selection process. Most traditional feature selection methods, such as Orthogonal Matching Pursuit (OMP)[86], are wrapper methods that explicitly select a feature subset iteratively before separately estimating the model parameters. In contrast, an *embedded* method performs feature subset selection simultaneously with parameter estimation. Sparse linear regression, also known as penalized or regularized linear regression, is a popular example of this approach.

A hallmark of feature selection methods is the introduction of a *hyperparameter* that can be tuned to adjust the sparsity level, s. While some methods, such as OMP and best-subset selection (BSS)[46], use a hyperparameter that directly represents s, in other cases, the hyperparameter influences s indirectly. For example, a sparse linear regression selects features by solving for $\boldsymbol{\beta}$ and β_0 that minimize a penalized least squares loss function[46]:

$$\mathcal{L}_{\ell_q} = \mathcal{L}_{OLS}\left(\boldsymbol{y}, \boldsymbol{y^{pred}}\right) + \lambda \cdot \ell_q(\boldsymbol{\beta}) \tag{4}$$

The term $\mathcal{L}_{OLS}\left(\boldsymbol{y}, \boldsymbol{y^{pred}}\right)$ is known as the ordinary least squares (OLS) cost function and is defined as $\sum_{i=1}^{N}\left|y_i - y_i^{pred}\right|^2$ where $y_i^{pred} = f(\boldsymbol{x_i}) = \beta_0 + \sum_{j=1}^{p}\beta_j\, x_{ij}$. Meanwhile, $\ell_q(\boldsymbol{\beta})$ is a penalty function that penalizes the coefficients of insignificant features. Here, λ is the hyperparameter, often called the regularization parameter, that influences s by controlling the strength of penalization, similar to the maximum tree depth in tree-based models.[46] Increasing the value of s leads to a smaller $\boldsymbol{F^S}$, and vice versa. In this section, I describe basic feature selection methods including LASSO[33,46] (a method based on penalizing non-zero coefficients), BSS[46,87] (an exhaustive search method), OMP[86] (a 'greedy' iterative method that selects the top-performing

features at each iteration to achieve an overall optimal solution), and Boruta[88] (an iterative method based on a random forest regressor).

4.2.1 LASSO and Variants

The basic choice for penalization is to set $\ell_q(\boldsymbol{\beta}) = \ell_1(\boldsymbol{\beta}) = \sum_{j=1}^{p}|\beta_j|$. Consequently, this method is sometimes referred to as the ℓ_1 model (or ℓ_1 regularization).[38] It *shrinks* the coefficients of features to zero, thereby implicitly performing feature selection.

An extensive literature has been developed to examine when this method can recover the "true" sparse model generating the target values, excluding irrelevant features (i.e., exact sparse recovery).[38,77] A fundamental assumption underlying these results is that the features are not overly dependent on each other. In addition to the general information-theoretic conditions outlined in Section 3.2, this assumption is often formalized in the ℓ_1 method through conditions such as the *mutual incoherence*[38,77] (see Section 4.8.2 in the Supporting Information), which requires that the ground-truth features (if any present in $\boldsymbol{F}$) have no or only weak correlations with the remaining features. In limited data settings, conditions like these are unlikely to be satisfied, especially in scientific and engineering contexts where candidate features are often highly interrelated, as they are derived from the same underlying latent variables. In such cases, ℓ_1 penalization can still be applied if only a few features in $\boldsymbol{F}$ are correlated with potentially relevant features and can be easily removed before fitting.

To induce further 'sparsification', some reports[39,40,46] have suggested using weighted ℓ_1 models, setting $\ell_q(\boldsymbol{\beta}) = \ell_1^{\boldsymbol{w}}(\boldsymbol{\beta} \mid \boldsymbol{w}) = \sum_{j=1}^{p} w_j|\beta_j|$, where $\boldsymbol{w} = \{w_1, \ldots, w_p\}$ represent the penalizing weights. These models aim to suppress the impact of insignificant features on the

solution by independently assigning large values to their corresponding weights $\boldsymbol{w}$. The literature describes various types of weighted ℓ_1 models that assign values to $\boldsymbol{w}$ in different ways.[39,40]

For example, the adaptive ℓ_1 (ℓ_1^{AD}) method[39] first performs unpenalized OLS or ridge linear regression, and then assigns the inverses of these fitted coefficients to $\boldsymbol{w}$ for use in a subsequent weighted ℓ_1 model (see Figure 4.2a). In contrast, iteratively re-weighted ℓ_1 (ℓ_1^{IR})[40] method starts by fitting a traditional ℓ_1 model and then iteratively trains a weighted ℓ_1 model by updating $\boldsymbol{w}$ at each step with the fitted coefficients from the previous model (see Figure 4.2b). While these methods often result in sparser models in practice, they offer limited theoretical guarantees regarding their solution properties.

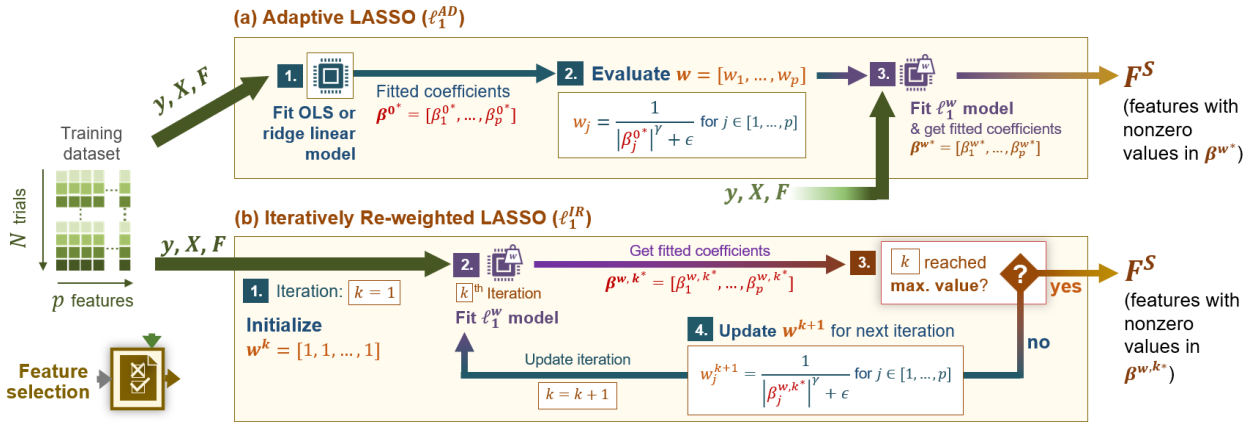

Figure 4.2. Weighted ℓ_1 methods for feature selection. (a) Adaptive ℓ_1 (ℓ_1^{AD}). The weights $\boldsymbol{w}$ in this method are typically obtained from preliminary OLS or ridge regression fitted coefficients $\boldsymbol{\beta^{0^*}}$. Ridge regression is a penalized linear regression that uses the ℓ_2 penalty: $\ell_2(\boldsymbol{\beta}) = \sum_{j=1}^{p} \beta_j^2$. In this work, we used a ridge regression that is 10-fold cross-validated over a hyperparameter list $\boldsymbol{\lambda} = [0.01, 0.1, 10]$ is utilized. In step 2, γ is set to 2 and ϵ to 10^{-25}. **(b) Iteratively reweighted $\boldsymbol{\ell_1}$ ($\boldsymbol{\ell_1^{IR}}$).** At each iteration k, an ℓ_1^w model is fitted using the feature coefficients $\boldsymbol{\beta^{w,k^*}}$ from the previous iteration as weights $\boldsymbol{w^k}$. The maximum iteration count, which is used as the stopping criteria (see step 3), is set to 10. In step 4, γ is set to 1, and ϵ is defined as $max(10^{-3},\ 0.1 \cdot \sigma_k)$, where σ_k is the standard deviation of nonzero values in $\boldsymbol{\beta^{w,k^*}}$.

4.2.2 *Best Subset Selection (BSS) and Variants*

For greater control over sparsity, one can set $\ell_q(\boldsymbol{\beta}) = \ell_0(\boldsymbol{\beta}) = \sum_{j=1}^{p} 1\{\beta_j \neq 0\}$, which simply equals the number of non-zero coefficients in $\boldsymbol{\beta}$. This approach identifies the best-performing subset across all possible combinations of features in $\boldsymbol{F}$, given a specific subset size s. Historically, this method was not a preferred choice due to the computationally expensive exhaustive search it requires. However, recent algorithmic advances have made BSS a computationally feasible option.[41,87] Additionally, recent studies[87,89–91] have shown that BSS is more resilient than the ℓ_1 method to correlated features.

In datasets with high levels of noise, adding an additional ℓ_2 penalty, defined as $\ell_2(\boldsymbol{\beta}) = \sum_{j=1}^{p} |\beta_j|^2$, to the loss function can help mitigate the effects of any potentially ill-conditioned coefficient vectors during the subset selection process.[41,92] The combination of these penalties has been coined as the $\ell_0\ell_2$ penalty, which is shown to further improve feature selection without significantly increasing computation time.[41,90,92]

4.2.3 *Orthogonal Matching Pursuit (OMP) and Variants*

OMP[86] and its variant, iterative sure-independence screening (ISIS),[42] are both wrapper-based approaches that iteratively add features to $\boldsymbol{F^S}$ (see Figures 4.3 and 4.4). At each iteration (say, the k^{th}), both methods begin employ a correlation-based screening step to choose a subset of important features (referred to as the *active set* and denoted here as $\boldsymbol{F^{A_k}}$). An ordinary least squares linear regression is then performed over these features to compute the residuals of $\boldsymbol{y}$, which are subsequently used to identify relevant features from the remaining variables to update the active set. While ISIS uses a penalized model over the residuals to select these features, OMP selects

single feature most correlated with the residuals. This procedure is repeated until the desired level of sparsity is achieved. Although OMP and ISIS are straightforward to implement, they may not yield significantly different results compared to the ℓ_1 method.

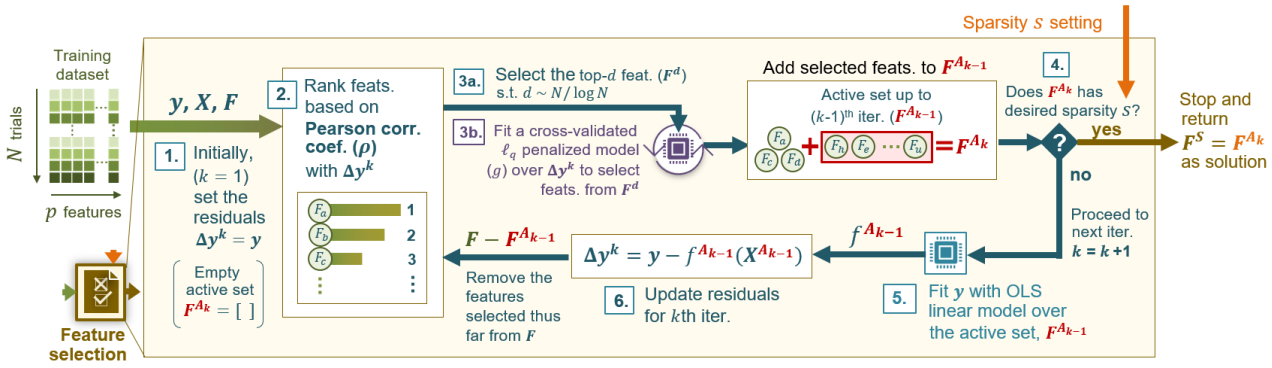

Figure 4.3. Schematic of the Iterative Sure Independence Screening (ISIS) algorithm.[42] From $\boldsymbol{F}$, features are initially ranked based on their correlation with the residuals $\boldsymbol{\Delta y^k}$ as shown in steps 1 and 2. For the first iteration $k = 1$, $\boldsymbol{\Delta y^k = y}$. In step 3, the top-d features are selected to fit $\boldsymbol{\Delta y^k}$ using a cross-validated, penalized linear model, denoted as g, typically an ℓ_1 model. The selected features are added to an initially empty set, called the active set, which expands iteratively as $\boldsymbol{F^{A_k}}$ after each k^{th} iteration. If the desired sparsity level s is reached after iteration k, the algorithm terminates, outputting the active set as the solution $\boldsymbol{F^S}$. Otherwise, an OLS regressor fits $\boldsymbol{y}$ over the active set to generate new residuals $\boldsymbol{\Delta y_k}$ (step 6). Steps 2 and 3 are then repeated over the remaining features, that is, $\boldsymbol{F - F^{A_{k-1}}}$. The sparsity level s is implicitly controlled in step 4 by regulating the number of iterations.

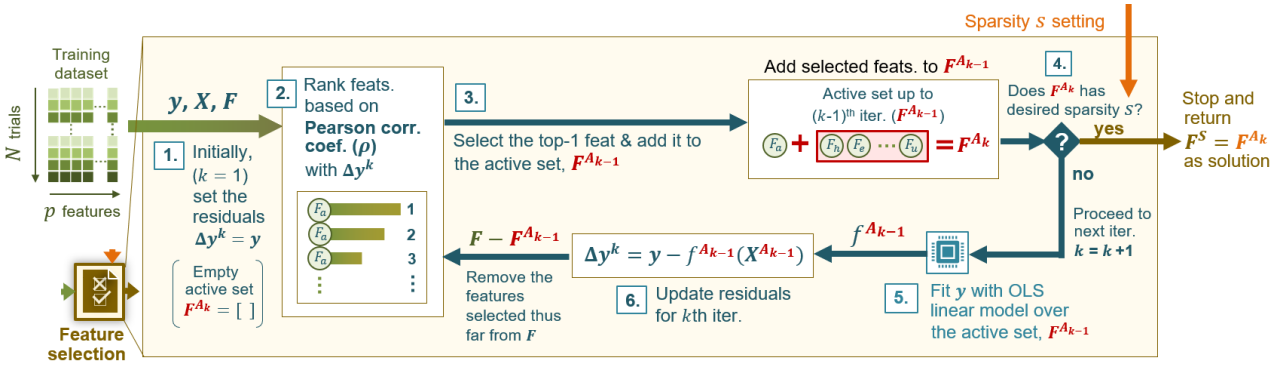

Figure 4.4. Schematic of the Orthogonal Matching Pursuit (OMP) algorithm.[86] This method is identical to ISIS (Figure 4.3), except in step 3, where the top-ranked feature from step 2 is

selected and added to the active set. Since each iteration adds only one feature, the number of iterations directly determines the sparsity level s of the solution.

4.2.4 Boruta

Unlike previous methods that focus on selecting relevant features, Boruta[88] eliminates irrelevant features (see Figure 4.5). In this method, for every feature F_j in $\boldsymbol{F}$, a *shadow* feature F_j' is introduced, with its values artificially generated by randomly shuffling those of its counterpart F_j. As a result, these shadow features are designed to be non-informative. When the combined set of original and shadow features is used to train a random forest, the shadow features serve as a negative control group.

When the impact of individual features on Y is evaluated (typically using impurity or permutation methods[46]), original features that perform similarly to their shadow counterparts are considered insignificant. Because these features offer little more predictive value than random guessing, they are eliminated from the model. A similar, but more statistically rigorous approach with theoretical guarantees, known as the *knockoffs* scheme[36,37], also utilizes a negative control group of features. However, the values of features generated by the knockoffs method exhibit more stringent properties, including correlation structures identical to those of the original features. This method is discussed in detail in Section 4.3.2.

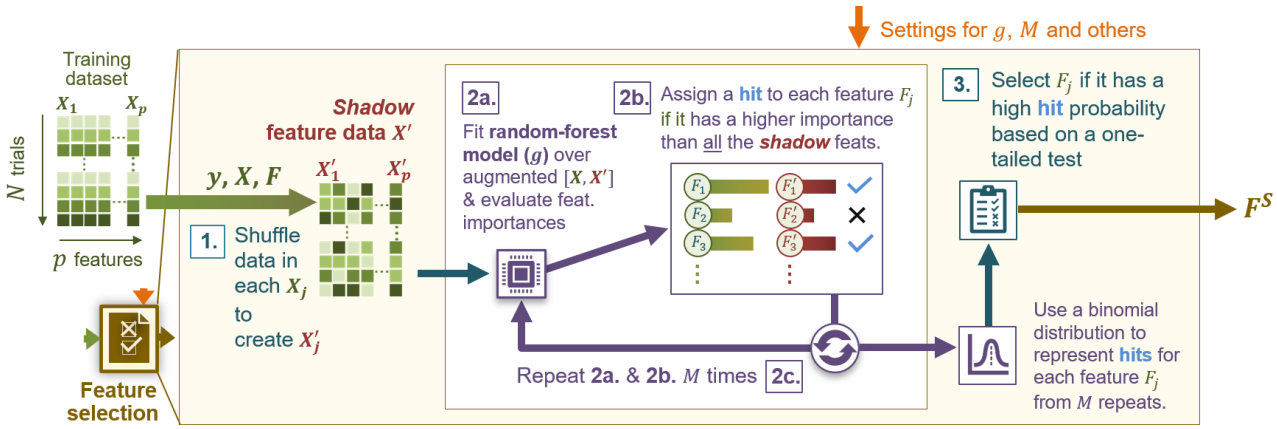

Figure 4.5. Schematic of the Boruta algorithm. In each of the M repeats, g is independently fitted, and feature importances are evaluated (step 2a) to compare each feature F_j against all shadow features. In each iteration, a feature is either be assigned a "hit" or not. After M repeats, the probability of hits for each feature is calculated. Using a binomial distribution, a one-tailed test identifies features with a significantly high probability of hits as $\boldsymbol{F^S}$.

4.2.5 Summary

In summary, the methods outlined above represent a diverse range of feature selection schemes that are suitable for small datasets. To build a sparse model after using an explicit feature selection method, such as OMP or Boruta, an unpenalized OLS linear regression can be used on the training data with the selected features to estimate the values of the sparse parameter vector $\boldsymbol{\beta}$ and the intercept β_0. In contrast, an implicit feature selection method like LASSO estimates the values of $\boldsymbol{\beta}$ and β_0 simultaneously without requiring a separate fitting procedure. Nevertheless, an OLS linear regression for $\boldsymbol{\beta}$ estimation can still be applied after implicit feature selection if the final predictions are to remain comparable across different methods on equal footing.[76]

Feature selection results in fewer degrees of freedom than direct fitting with the full feature menu, thereby reducing the risk of overfitting. On the other hand, feature selection methods generally require a separate strategy to determine the sparsity level s of the solution $\boldsymbol{F^S}$, either

directly or implicitly through hyperparameter tuning. In the next section, I explore how to choose the appropriate level of sparsity for feature selection schemes in small datasets.

4.3 Selecting the Model Sparsity

Selecting a large $\boldsymbol{F^S}$ not only captures more relevant variables but also includes many irrelevant features. Therefore, determining the optimal sparsity must balance these opposites. This is typically performed after a sparse model is independently learned for a range of hyperparameters (that either directly represent the sparsity levels or indirectly influence them like the regularization parameter λ). Here, I explore two strategies to choose the value of s—cross-validation, the traditional approach that focuses on minimizing prediction error, and the knockoffs scheme, a modern indirect approach that aims at minimizing false discoveries. While cross-validation independently estimates several versions of the same model for each hyperparameter on distinct subsets of the training data, the knockoffs method estimates these versions on the entire training data. From these versions of models, the optimal one is chosen, and its hyperparameter (and thereby the corresponding sparsity level s) is retained.

4.3.1 Cross-Validation

A common strategy is to select a hyperparameter value—and thus the value of s and the corresponding $\boldsymbol{F^S}$—that minimizes prediction error over a validation dataset set aside from the training and testing data. For each candidate hyperparameter value, a feature selection method is applied using the training data, and the validation data is then used to evaluate the selected features.

In small datasets where withholding a validation set is not affordable, K-fold cross-validation (CV)[46] is usually employed (see Figure 4.7a).

In K-fold CV, the full training dataset is first split into K disjoint subsets (or folds) of roughly equal size. Each fold can be *held out* as a validation set, while the model is trained on the remaining $K - 1$ folds. This is repeated K times, with each fold acting as the hold-out once during the training process. Thus, each candidate hyperparameter value is used to fit a model on K distinct training datasets, each producing a distinct feature subset which is then *validated* on the hold-out. The mean or median of the resulting K validation errors serves as the overall validation error for the current hyperparameter value. From a range of candidate hyperparameter values, the one that minimizes the overall validation error is then selected. This value is used with the full training data to determine the final $\boldsymbol{F^S}$, which is then used to obtain the final sparse trained model. If the features selected by the chosen hyperparameter vary significantly across the K splits and the full training set, this indicates that the feature selection method lacks robustness, and the final feature subset should be interpreted with caution.[93]

As N decreases, it is recommended to increase K to ensure that the training data in each split doesn't become too small. For datasets with significantly fewer data-points (e.g., $N \leq 50$ as a general rule of thumb), it becomes feasible to set $K = N$, where only one data-point is withheld for validation in each split, while the remaining $N - 1$ data-points are used for training. This special case of CV is known as *leave-one-out* (LOO) cross-validation.[46]

4.3.2 Knockoffs Scheme

While CV directly controls the size of $\boldsymbol{F^S}$ based on estimated prediction accuracy, the knockoffs approach follows a different principle: ensuring that $\boldsymbol{F^S}$ does not include too many irrelevant features.[37] This approach aims to minimize the false discovery rate (FDR), a metric representing the fraction of irrelevant features in $\boldsymbol{F^S}$ (see Section 5.3 and Section 4.8.3 in Supporting Information for more details). Similar to Boruta (in Section 4.2.4), it employs a negative control group of notional features called *knockoffs* (hence the method's name). However, knockoffs are assigned values through a more sophisticated process than random shuffling, ensuring that certain theoretical guarantees related to FDR estimation in the solution are upheld.

For every feature F_j, a knockoff feature $\tilde{F}_j$ (where $j \in [1, 2, \dots, p]$) is introduced as an inherently non-informative or irrelevant feature, uncorrelated with Y. Unlike the shadow features in Boruta, $\tilde{F}_j$ is additionally required to be indistinguishable from the original feature F_j in terms of its joint distribution with other features. This ensures that swapping the data of F_j with its knockoff counterpart $\tilde{F}_j$ at a given column index j in the design matrix $\boldsymbol{X}$ does not alter the pairwise correlation between that column and the remaining features (see Section 4.8.3 in Supporting Information). Despite the close resemblance between knockoff features and the original ones, the main intuition is that an original feature should be preferred over its knockoff by a significant margin during feature selection unless it is truly irrelevant.

The effectiveness of this approach depends on how well the knockoff features satisfy these properties, making the generation of knockoff data the most critical and challenging step. This process requires either a large dataset[37,94] or prior knowledge of the feature data distribution.[36,95] Consequently, this approach shows great potential for laboratory datasets where the distributions

of many features are either exactly known (e.g., specimen temperature and material composition, which are directly controllable) or can be inferred using domain expertise. For small datasets, where the available data-points are insufficient to accurately represent the true underlying distribution, approximate knockoff data generators have been proposed[36] (see Section 4.8.3 in Supporting Information).

Similar to the random forest used in Boruta, the knockoffs approach requires an embedded feature selection method to operate on a combined menu of original and knockoff features. Typically, penalized linear regression, such as the ℓ_1 method, is employed as the embedded method[37], with its sparsity level controlled by a regularization hyperparameter λ. Based on the subsets of features selected from the combined feature menu as λ is varied, a feature importance statistic W_j is calculated for each original feature F_j, quantifying its relative significance compared to its knockoff counterpart $\tilde{F}_j$. The original features whose W_j values surpass a set threshold T_q are then finally selected as $\boldsymbol{F^S}$ (see Figure 4.7b). The value of this threshold is determined using a theoretical framework that guarantees the true FDR in the solution to be less than or equal to a nominal value q (where $0 < q < 1$), specified by the user (see Figure 4.12 in Supporting Information). Lowering the value of q raises T_q, resulting in fewer features being selected into $\boldsymbol{F^S}$. Thus, q serves as the primary hyperparameter that indirectly determines the sparsity level by nominally controlling the FDR of the solution.

Although FDR can theoretically reach zero, lowering q below a certain minimum value typically results in an empty $\boldsymbol{F^S}$ solution. By selecting this minimum q, the knockoffs scheme produces a non-empty $\boldsymbol{F^S}$ with the smallest possible FDR. This value of q also serves as a conservative estimate of the true FDR, which cannot be exactly calculated in real-world datasets

where the underlying ground truths influencing Y are typically unknown. Section 4.8.3 in Supporting Information provides a more detailed technical description of this method. In conclusion, the level of sparsity and the corresponding subset of features selected are determined using the training dataset, which is used in its entirety for the knockoffs approach but divided into multiple training and validation subsets in K-fold cross-validation.

4.4 Leave-one-out Testing Scheme

Traditionally, when data is large, the training set is initially separated from a test dataset, which is ultimately used to evaluate the generalizability of the selected feature subset $\boldsymbol{F^S}$. But, if the available data is small, making it impractical to withhold a test set, a K-fold scheme, similar to K-fold cross-validation, can be employed. The full dataset with N^{tot} data-points can be divided into K disjoint folds of equal size, with each fold serving as a test set while the remaining $K-1$ folds serve as the training set (see Figure 4.5). For datasets with significantly fewer trials, a leave-one-testing (LOO) scheme can be used where each fold contains only one data-point. This approach maximizes the utilization of the available data by withholding only a single data-point as test set in each split, while the model is trained on the remaining $N^{tot}-1$ data-points. Thus, across all the LOO splits, $N^{tot}-1$ independently trained models are obtained and tested on their corresponding held-out test data-points, yielding $N^{tot}-1$ test errors. The mean or median of these errors may be used to infer the model's overall performance.

Similarly, the trained feature coefficients can be individually averaged across the LOO splits to obtain the overall trained feature coefficient vector $\boldsymbol{\beta}$. The non-zero values of $\boldsymbol{\beta}$ then

define the final selected feature subset $\boldsymbol{F^S}$. The standard deviation of the coefficients across the LOO splits reflects each feature's selectivity by the model. A small standard deviation suggests that the model consistently identifies the feature as relevant or irrelevant, by either selecting or excluding it in most splits.[93] Conversely, a large standard deviation indicates inconsistency in the model's selection of the feature, suggesting uncertainty about its relevance. However, even if all these steps are performed correctly, the inferences may change due to fluctuations in new data, which can be significant for scientific settings. In the next section, I introduce a family of methods known as Conformal Prediction (CP)[44] to evaluate a model's sensitivity to variations in data with rigorous statistical guarantees.

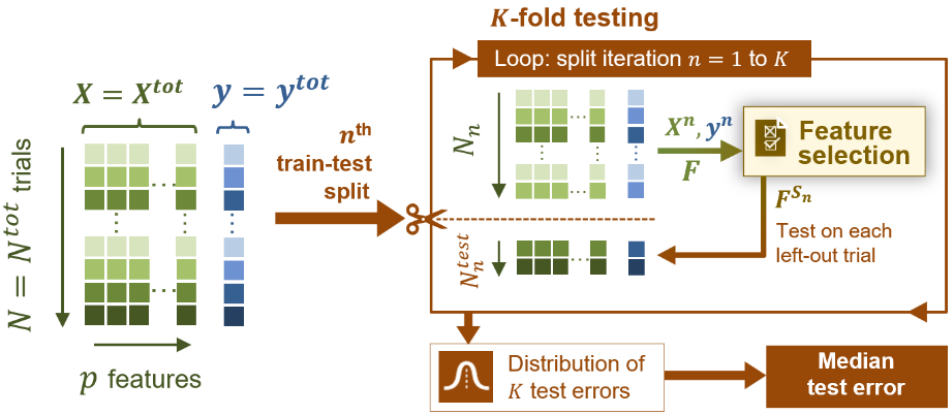

Figure 4.6. Schematic illustrating the K-fold testing scheme. For small datasets where withholding a separate test set is impractical, multiple training and testing sets can be generated using the K-fold scheme. When K is set to N_{tot}, it reduces to a leave-one-out scheme.

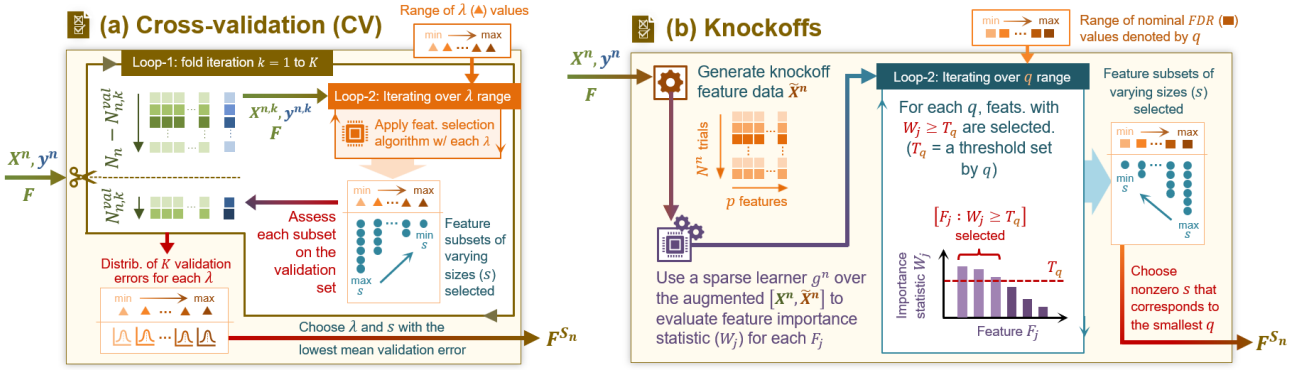

Figure 4.7. Schematic illustrating sparsity level selection for each n^{th} training set, X^n and y^n, in Figure 4.6. (a) Cross validation. In Loop-2, a feature selection algorithm (as discussed in Section 4.2) is applied several times—once over each value of the sparsity-controlling hyperparameter (λ) which varies depending on the selection algorithm. For example, in an ℓ_1 model, λ represents the regularization parameter. In Boruta, λ corresponds to the inverse of the desired sparsity level, i.e., $1/s$. **(b) Knockoffs scheme.** The nominal false-discovery rate (q) serves as the sparsity-controlling hyperparameter. See Section 4.8.3 for more details on how the threshold T_q is derived for each q.

4.5 Uncertainty Quantification (UQ)

4.5.1 Why Quantify Uncertainty?

In many scientific settings, predictions by a fitted model often deviate from newly observed values, even when experimental trials are replicated with only slight perturbations in the feature data. These deviations may stem from instrumentation or human error, as well as fluctuations in unaccounted yet influential phenomena (see Figure 4.8a). In such cases, beyond merely predicting the target value for a test data-point, it is beneficial to also provide a confidence interval that quantifies the model's uncertainty for that data-point.[44] This is especially important in high-risk fields such as medical diagnostics, where poor model reliability can lead to life-threatening situations.[44]

Confidence Intervals (CI) are a standard form of uncertainty (UQ). The user chooses a confidence level $1-\alpha$, where α $(0 < \alpha < 1)$ is a small probability representing the acceptable level of risk of the user. For a test trial $(\boldsymbol{x}', y')$, a trained predictive model aims to estimate a single real value that closely approximates y'. Given a real value α, let $C_\alpha(\boldsymbol{x}')$ denote such a confidence interval satisfying the condition:

$$\mathbb{P}(y' \in C_\alpha(\boldsymbol{x}')) \geq 1-\alpha \tag{5}$$

Here, $\mathbb{P}(y' \in C_\alpha(\boldsymbol{x}'))$ represents the *coverage*, which is the probability that y' lies within $C_\alpha(\boldsymbol{x}')$. This probability is expected to be higher than or equal to 1 – α. For instance, for α = 0.05, a 95% CI is a range of Y values that is statistically guaranteed to contain the true y' with a probability of at least 95%.

Note that such an interval is not unique, and that different statistical methods can output slightly different intervals. The coverage is the actual probability that y' lies within $C_\alpha(\boldsymbol{x}')$, thus it should always be at least $1-\alpha$. If a statistical method is used inappropriately, then the coverage can be smaller, giving the user a false sense of accuracy. For example, statistical packages provide CIs for the predicted Y values based on the assumption that the noise in the observed $y_1, \ldots, y_N$ is Gaussian. When this assumption does not hold in practice, the CIs are not correct, meaning that their coverage is below the level $1-\alpha$. Here, I describe methods that estimate CIs, with theoretical guarantees on coverage. These methods are part of an active area of statistics called Conformal Prediction (CP), and the intervals obtained by them are called prediction intervals.

4.5.2 The 'Naïve' Approach to UQ

A straightforward but "naïve" approach[43] is to use the distribution of training residual values $[R_1, R_2, \ldots, R_N]$, where each residual is evaluated as $R_i = |y_i - f^*(\boldsymbol{x_i})|$, with f^* being the predictive model trained over the full training set. For a test input $\boldsymbol{x}'$, the α^{th} and $(1-\alpha)^{\text{th}}$ quantiles of the R_i distribution can heuristically define an uncertainty interval around $f^*(\boldsymbol{x}')$, with these quantiles representing the maximum deviations below and above $f^*(\boldsymbol{x}')$ respectively. However, the intervals generated by this approach are typically smaller than required to satisfy eq. (5). This is because the predictive model f^* is specifically trained to minimize residuals on the training data. As a result, the R_i distribution does not accurately represent the true variability or deviations that may occur in new, unseen test data.

To address this issue, many CP procedures[44] rely on calculating $C_\alpha(\boldsymbol{x}')$ using a separate dataset, known as the *calibration* set, which is withheld prior to training. In small datasets,

withholding a separate set is not recommended, as it would reduce the already small size of the training data. On the other hand, with small data, it is computationally feasible to retrain the model f in a leave-one-out (LOO) procedure, which can be leveraged to estimate accurate α level confidence intervals. These are *Jackknife,* an inexact precursor method, and *Jackknife+* and *Jackknife-minimax*, two recent CP methods derived from it.[43] Figure 4.8b outlines the LOO procedure involved here.

4.5.3 4.5.3. Jackknife—an early UQ method

In these approaches, for every i^{th} data-point (where $i = 1, \ldots, N$) that is held out, let the LOO model trained on the remaining data be denoted as f^*_{-i}. The residual $|y_i - f^*_{-i}(\boldsymbol{x_i})|$, evaluated on the held-out data-point $\boldsymbol{x_i}$ using this model, is denoted as R_{-i}. These residuals, generated using unseen data, provide unbiased estimates of uncertainties for predictions. Figure 4.8b illustrates how this approach generates N models, each denoted as f^*_{-i} with its corresponding residual R_{-i}, resulting in a distribution of N residuals. To estimate the uncertainty interval for a test input $\boldsymbol{x'}$, *Jackknife* uses the α^{th} and $(1-\alpha)^{\text{th}}$ quantiles of the R_{-i} distribution as maximum deviations below and above the prediction $f^*(\boldsymbol{x'})$ respectively. Though this approach yields slightly larger intervals than the "naïve" approach, these intervals do not satisfy eq. **(2)**.[43]

4.5.4 Jackknife+ and Jackknife-minmax—CP methods derived from Jackknife

Jackknife+ and *Jackknife-minmax*, on the other hand, generate prediction intervals with coverage guarantees as outlined in eq. (5).[43] Unlike the traditional *Jackknife*, which centers the uncertainty intervals around $f^*(\boldsymbol{x'})$, these approaches use the predictions from the LOO models,

$f^*_{-i}(\boldsymbol{x}')$, to define the intervals. *Jackknife+* uses the α^{th} quantile of the distribution of $|f^*_{-i}(\boldsymbol{x}') - R_{-i}|$ values as the lower bound of the uncertainty interval and the $(1-\alpha)^{\text{th}}$ quantile of the $|f^*_{-i}(\boldsymbol{x}') + R_{-i}|$ values as the upper bound. Jackknife-minmax is a more conservative estimator that produces a confidence interval at a new observation $\boldsymbol{x}'$ using not just the quantiles of the residuals, but also the widest gap in predictions between any of the LOO estimators $f^*_{-i}(\boldsymbol{x}')$ fit on the previous N data-points. Table 1 provides a summary of the intervals obtained by each CP method discussed. Figure 4.8c illustrates how these methods can be applied using nested leave-one-out schemes to obtain prediction intervals for all data-points in the dataset, similar to the leave-one-out testing approach discussed in Section 4.4 and Figure 4.6.

Table 4.1. Estimation of uncertainty intervals using CP approaches for a test data-point $(\boldsymbol{x}', \boldsymbol{y}')$ expecting a coverage of $\boldsymbol{1-\alpha}$. f^* denotes the model trained on the full training set, and f^*_{-i} denotes the LOO model trained on data with i^{th} held out. R_i denotes the residual $|y_i - f^*(x_i)|$, and R_{-i} denotes $|y_i - f^*_{-i}(x_i)|$. Q_α denotes the α-quantile function. For example, $Q_\alpha(R_i)$ denotes the α^{th} quantile of the distribution of the R_i values.

CP method	Lower bound	Upper bound
Naïve	$f^*(\boldsymbol{x}') - Q_\alpha(R_i)$	$f^*(\boldsymbol{x}') + Q_{1-\alpha}(R_i)$
Jackknife	$f^*(\boldsymbol{x}') - Q_\alpha(R_{-i})$	$f^*(\boldsymbol{x}') + Q_{1-\alpha}(R_{-i})$
Jackknife+	$Q_\alpha(f^*_{-i}(\boldsymbol{x}') - R_{-i})$	$Q_{1-\alpha}(f^*_{-i}(\boldsymbol{x}') + R_{-i})$
Jackknife-minimax	$\min_i f^*_{-i}(\boldsymbol{x}') - Q_\alpha(R_{-i})$	$\max_i f^*_{-i}(\boldsymbol{x}') + Q_{1-\alpha}(R_{-i})$

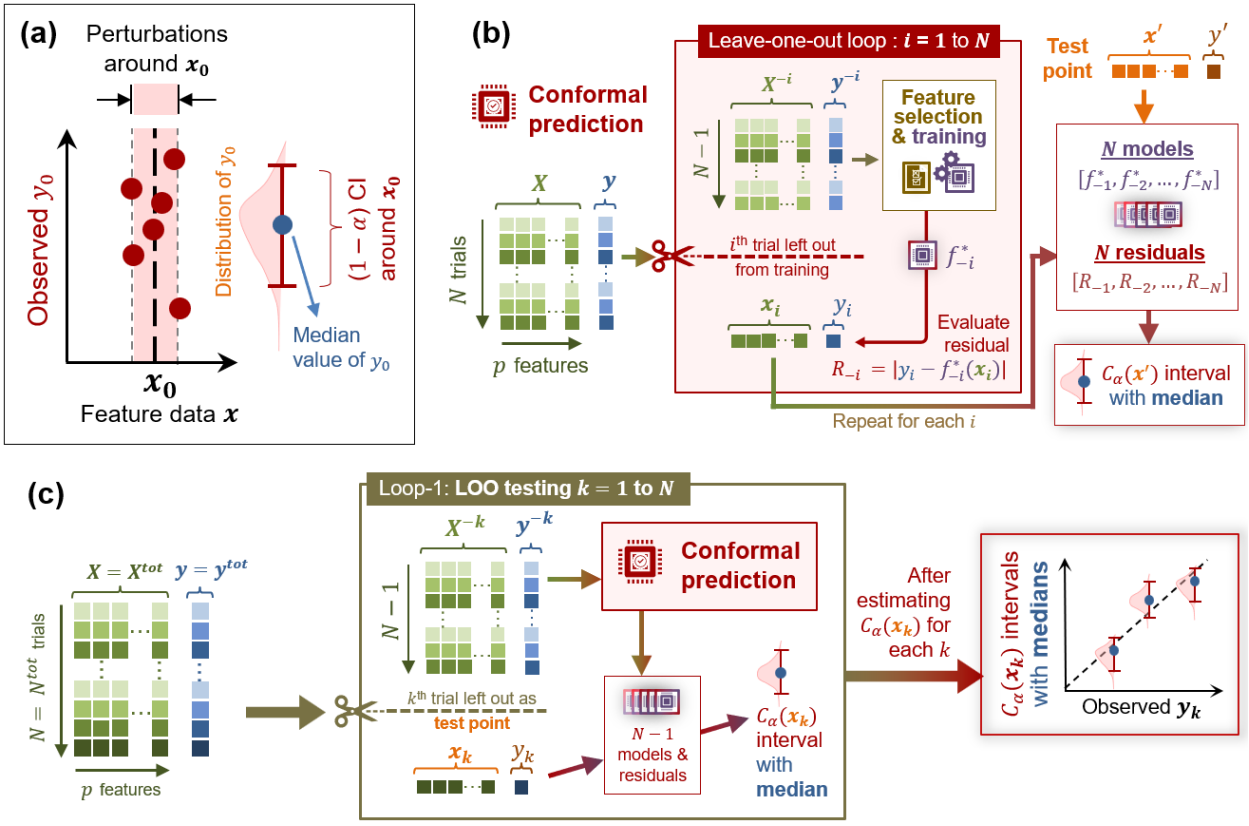

Figure 4.8. Conformal Prediction. (a) Schematic illustrating how slight perturbations in feature data can lead to large variations in observed target value. For data-points repeated for an arbitrary feature data vector $\boldsymbol{x_0}$, a distribution of y_0 may be obtained instead of a single value. Once this distribution is learned, its median and confidence interval (CI) can be determined. The parameter α $(0 < \alpha < 1)$ represents to the significance level of the confidence interval. **(b)** Schematic demonstrating how the LOO-based CP approach generates N independent models and residuals. Depending on the subsequent method used (e.g., Jackknife+ or Jackknife-minimax), these models and residuals can be utilized to construct the prediction interval $C_\alpha(\boldsymbol{x'})$ for a test data-point $(\boldsymbol{x'}, y')$. **(c)** Schematic illustrating how independent prediction intervals can be generated for all data-points using a leave-one-out testing scheme.

4.6 Prescriptive Guidelines to Modeling with Small Data

Here, I present some practical guidelines to be followed while modeling with small data, based on the concepts presented thus far:

1. Applying domain expertise during feature construction is essential in order to incorporate physically meaningful features into the models. During this process, caution must be exercised to ensure that only the primary feature data and hypotheses based on domain-knowledge are utilized, without relying on observed target values $\boldsymbol{y}$. This is to separate the information in data used for feature construction, which should depend solely on the distribution of $\boldsymbol{X}$, from the information used for regression, which should depend on the conditional distribution of $\boldsymbol{y}$ given $\boldsymbol{X}$. Moreover, creating new features from independent data sets or experiments is encouraged, as it brings valuable domain-specific prior information to the modeling workflow.

2. Feature selection methods, which yield sparse linear models when coupled with linear regression, provide a convenient approach to reducing the effective degrees of freedom m_s^0 when the dataset's sample size N is small. However, certain filter-based selection methods select or discard features based on their correlation with $\boldsymbol{y}$ and then reuse the same $(\boldsymbol{X}, \boldsymbol{y})$ data to fit the regression model. This is not valid as it reuses the information in $(\boldsymbol{X}, \boldsymbol{y})$.

3. Before beginning feature selection, it is worthwhile to evaluate whether the available dataset is not *too small*. To ensure this, the heuristic rule of $N \geq m_s^0$ should be satisfied, where m_s^0 represents the effective number of degrees of freedom required for exact sparse recovery with $s = p_0$, where p_0 denotes the number of underlying ground-truth variables (see Section 3.2). As p_0 is unknown in a real-world dataset, $m_s^0 \approx p_0 \log_2(p - p_0)/(1 - \rho)$ can be evaluated for a plausible range of p_0 (guided by domain knowledge) to determine whether the dataset is *too small* in each value in the range. For ρ, the mean absolute Pearson correlation coefficient can be used as a representative value for the pairwise correlations.

4. K-fold cross-validation is a valid statistical procedure for selecting model sparsity, as each fold ensures that the training data remains independent of the validation data. Afterwards, the scores from all folds are averaged for each candidate hyperparameter value evaluated. It is valid to use this procedure to select a particular hyperparameter, and then to use the same dataset in its entirety to fit a final model with the selected hyperparameter. Note that the selected features and the sparsity levels may vary between the individual folds and the final model.

5. Leave-one-out testing scheme is a powerful approach yielding statistically valid inferences even when datasets are small. $N \leq 50$ can be used as a rule of thumb for applying leave-one-out testing. For larger N, where this procedure can be computationally demanding, a K-fold testing scheme can be employed. When a leave-one-out or a K-fold testing scheme

produces multiple training sets, any cross-validation used to determine model sparsity must be performed independently within each training set in a nested manner.

4.7 Summary

In this chapter, I presented a sparse modeling workflow using linear models, specifically tailored for small datasets commonly encountered in scientific research. This workflow includes constructing features using domain expertise, applying a feature selection method, determining the optimal sparsity level, and validating the final model using conformal prediction. Among the feature selection methods, LASSO[33] (referred to here as the traditional ℓ_1 method) has been widely used in high-dimensional settings due to its ability to accurately identify relevant features when $\boldsymbol{y}$ has low noise and the features exhibit low mutual coherence (i.e., pairwise correlations). However, in real-world scenarios, where these conditions may not hold, variants with weighted ℓ_1 penalties, such as ℓ_1^{AD} and ℓ_1^{IR}, have been proposed.[39,40] Recently, BSS (referred to here as the ℓ_0 method) and its hybrid variant $\ell_0\ell_2$,[92] which were historically computationally expensive, have gained popularity due to algorithmic advances[41,87] that significantly reduce their time complexity. Other methods, such as OMP and ISIS, originally developed in signal processing,[42,86] have also seen increased adoption in regression tasks. However, caution must be exercised when using ℓ_0, $\ell_0\ell_2$, OMP and ISIS methods in cases with strong non-linear dependencies between the target variable and the features.

While these methods focus on minimizing prediction error, Boruta[88] aims to eliminate irrelevant features by introducing a set of shadow features that serve as a negative control group.

Similar to this, but a more statistically rigorous approach is the knockoffs method, which not only suppresses the selection of irrelevant features (i.e., false discoveries) but also provides a theoretical estimate q of the underlying false discovery rate (FDR). These estimates can be used to select a sparsity level s corresponding to the lowest value of q, in contrast to cross-validation, which selects model sparsity based on the lowest validation error. By focusing on eliminating the irrelevant features rather than attempting to identify the relevant ones, Boruta and knockoff-based feature selection methods can be effective in cases with strong non-linear dependencies between the features and the target variables, with the latter being preferable due to its theoretical guarantees.

Additionally, I discussed how a leave-one-out testing scheme can be employed to maximize data utilization when the sample sizes are small. Beyond its use for target prediction, this approach can also be used for uncertainty quantification, as it estimates prediction intervals for each observed value in $\boldsymbol{y}$ through the Jackknife+ and Jackknife-minmax conformal prediction methods (see Figure 4.8c). In the next chapter, I benchmark the methods discussed here using synthetic datasets in which the underlying ground-truth models are known.

4.8 Supporting Information

4.8.1 Summary of Notation

Table 4.2. Table of notation and description of each variable.

Symbol[a]	Description
N	Number of trials (also referred to as runs, data-points, samples, observations, calculations or experiments depending on the context) in the training dataset.
p	Number of features in the feature set $\boldsymbol{F}$ available for modeling.
s	Number of features selected from $\boldsymbol{F}$ by a feature selection method (implicitly or explicitly), and consequently the number of non-zero feature coefficients in a sparse linear model operating over $\boldsymbol{F}$**.**
p_0	Number of ground truth features influencing Y (as defined in a synthetic dataset).
Y	Target variable whose values are to be predicted.
$\boldsymbol{F}$	Set of p feature variables (each denoted by F_j for $j = 1, 2, .., p$) available for modeling.
$\boldsymbol{F^S}$	Subset of $\boldsymbol{F}$ containing s features, selected by a feature selection method.
$\boldsymbol{F'}$	Set of p shadow feature variables (each denoted by F'_j for $j = 1, 2, .., p$) used in the *Boruta* feature selection method.

[a] A bold symbol represents a set, vector, or matrix while regular font is used for variables and functions.

$\widetilde{\boldsymbol{F}}$	Set of p knockoff feature variables (each denoted by $\tilde{F}_j$ for $j = 1, 2, .., p$) used in the *knockoffs* scheme.
$\boldsymbol{x_i}$	p-dimensional array of feature values (each denoted by x_{ij} corresponding to F_j for $j = 1, 2, .., p$) of the i^{th} trial.
$\boldsymbol{X}$	$N \times p$ matrix containing feature data for N trials, where each row $\boldsymbol{x_i}$ ($i = 1, 2, \dots, N$) represents an individual trial, and each column $\boldsymbol{X_j}$ ($j = 1, 2, \dots, p$) corresponds to one of the p features. Note that $\boldsymbol{X}$ is standardized before modeling so that each column has a mean of 0 and a variance of 1.
$\boldsymbol{y}$	N-dimensional array of Y's observed values (each denoted by y_i for $i = 1, 2, \dots, N$) in the training dataset.
$\boldsymbol{y^{pred}}$	N-dimensional array of Y's predicted values (each denoted by $y_i^{pred} = f^*(\boldsymbol{x_i})$ for $i = 1, 2, \dots, N$) corresponding to trials from the training dataset.
f	Function representing the predictive model, which takes a p-dimensional feature data array as input and outputs a real-valued prediction of Y.
$\boldsymbol{\beta}$	p-dimensional vector of parameters or coefficients (each denoted by β_j corresponding to feature F_j for $j = 1, 2, .., p$) used in a linear model f.
β_0	Intercept parameter used in a linear model f.
$\boldsymbol{\beta}^*, \beta_0^*$	Optimal values of the parameters $\boldsymbol{\beta}$ and β_0 respectively, obtained after fitting f to the training data.
f^*	Fitted predictive model obtained by assigning optimal values to the parameters in f.
$C_\alpha(\boldsymbol{x}')$	Prediction interval range for a trial with feature data $\boldsymbol{x}'$ such that it contains the observed target value y' with a probability of at least $1 - \alpha$.

† A bold symbol represents a set, vector, array, or matrix while regular font is used for variables and functions.

4.8.2 Mutual Incoherence in ℓ_1 method

As discussed in Section 3.2, the information-theoretic estimations of m_s^0 provide the lower bounds on the sample size for reliable recovery of ground-truth variables that constitute the generative mechanism governing Y. However, these limits underestimate these bounds when there are correlations (or *coherence*) present between the features in the design matrix $\boldsymbol{X}$ (see Section 3.2.2). In this section, I discuss a condition called *mutual incoherence* (also referred to as *irrepresentability*)[38,77], specifically in context of the ℓ_1 method. This condition determines when the method can exactly recover all and only the ground-truth features from $\boldsymbol{F}$, depending on the degree of dependence among the features in $\boldsymbol{F}$. It assumes that among the p features in $\boldsymbol{F}$, a subset of p_0 mutually uncorrelated features ($1 \leq p_0 < p$) represents the ground truth variables. Loosely speaking, the condition requires these p_0 ground truth features to be either uncorrelated or only weakly correlated with the remaining $p - p_0$ features in $\boldsymbol{F}$.

Let $\boldsymbol{S} = [1, 2, \dots, p]$ denote the ordered set of indices corresponding to all the features in $\boldsymbol{F} = [F_1,\ F_2, \dots, F_p]$, and let $\boldsymbol{S^0}$ ($\subset \boldsymbol{S}$) represent the set of indices corresponding to the ground truth features within $\boldsymbol{F}$. Thus, $\boldsymbol{S - S^0}$ represents indices of the $p - p_0$ features in $\boldsymbol{F}$ that are not ground truths. Suppose an ordinary least squares (OLS) linear regression is performed to model each of these $p - p_0$ features (denoted as F_j, where $j \in (\boldsymbol{S - S^0})$) as a function of only the p_0 grounds truths. Then, let $\boldsymbol{\beta^{0\to j}} = \left[\beta_1^{0\to j}, \beta_2^{0\to j}, \dots, \beta_{p_0}^{0\to j}\right]$ denote the corresponding fitted coefficient values, each associated with a specific ground truth feature. The irrepresentability condition is then mathematically expressed as follows:

$$\max_{j \in (\boldsymbol{S-S^0})} \ell_1(\boldsymbol{\beta^{0\to j}}) \ < \ 1 \tag{6}$$

$$\text{where} \quad \ell_1(\boldsymbol{\beta}^{\mathbf{0}\to\boldsymbol{j}}) = \sum_{k=1}^{p_0} \left|\beta_k^{0\to j}\right| \tag{7}$$

If any of the ground truths are correlated with a feature F_j for $j \in (\boldsymbol{S} - \boldsymbol{S}^{\mathbf{0}})$, the corresponding $\boldsymbol{\beta}^{\mathbf{0}\to\boldsymbol{j}}$ will have nonzero values, resulting in a nonzero $\ell_1(\boldsymbol{\beta}^{\mathbf{0}\to\boldsymbol{j}})$ norm.

To analyze how (6) is influenced by dataset parameters such as N, p and p_0, in addition to pairwise correlations between features, I conducted simulations using synthetic data, following the approach reported by Hastie et al.[38] For each feature in $\boldsymbol{F}$, data is independently sampled from a standard Gaussian distribution. A subset of p_0 features is then randomly selected from $\boldsymbol{F}$ to form the set of ground truths and the corresponding index set $\boldsymbol{S}^{\mathbf{0}}$. Furthermore, a feature is randomly chosen from the remaining $p - p_0$ features and assigned values such that it maintains a specific non-negative Pearson correlation coefficient value (ρ) with a randomly selected ground truth feature. As a result, the left-hand side of (6) depends solely on this correlated feature pair. For a given combination of N, p and p_0, smaller values of ρ increase the likelihood of satisfying (6), suggesting that removing correlated features from $\boldsymbol{F}$ is beneficial.

Alternatively, when p, p_0 and ρ values are fixed, (6) holds only if N exceeds a certain minimum value, indicating that an insufficient number of trials may hinder selection consistency in the ℓ_1 model. Figure 4.9 presents these minimum N values (averaged over 25 replicated datasets) as curves across various combinations of p, p_0 and ρ values. The vertical dashed line, corresponding to $p_0 = 0.2 \times p$, serves as a visual guide to illustrate how the minimum N required for the exact recovery of ground-truths increases with ρ and p. In real world scenarios, the ground truths often do not appear directly in $\boldsymbol{F}$, but are instead represented through relevant features that correlate with them. In such cases, the irrepresentability condition establishes a theoretical

baseline, demonstrating that more data is typically required than what is anticipated. Here, exact ground-truth recovery can be treated as equivalent to identifying all relevant features in $\boldsymbol{F}$ while excluding all irrelevant ones. Figure 4.9 also shows minimum N required for ground-truth recovery based on the information-theoretic limit $N/m_s^0 > 1$, discussed in Section 3.2, using eq. (1)-(3).

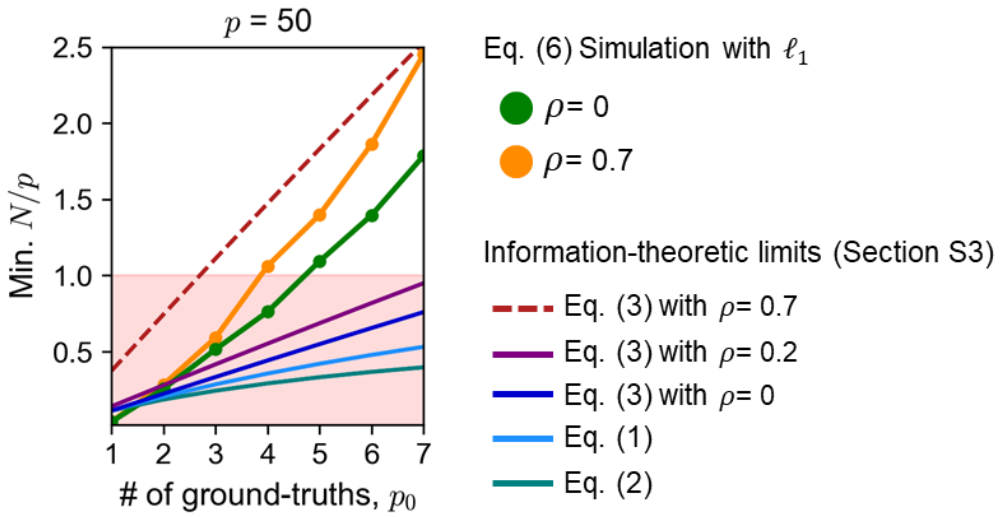

Figure 4.9. Minimum N needed for an ℓ_1 solution to be consistent according to the *mutual incoherence* condition (see eq. 6). For each data set with a total of p features in $\boldsymbol{F}$, p_0 ground-truth features are randomly chosen. Each point on the curve is a mean of minimum N values obtained from 25 simulations. For the curve corresponding to $\rho = 0$ (green), data of every feature in $\boldsymbol{F}$ is independently sampled from a standard Gaussian distribution. For curves with $\rho > 0$ (orange), a feature from $\boldsymbol{F}$ that is not a ground truth is randomly chosen and modified such that it is correlated with a randomly chosen ground truth feature with a Pearson correlation coefficient equal to ρ. The red shaded region indicates a high-dimensional region where $N < p$. Other curves are made using eq. (1)-(3), which provide information-theoretic limits to N based on the heuristic rule $N/m_s^0 < 1$. For each of these equations, s is set to p_0, assuming an ideal model that recovers all and only ground-truths.

4.8.3 Detailed Description of Knockoffs Scheme

Since its introduction by Barber et al[37] in 2015, the *knockoffs* scheme has been an effective method for controlling model sparsity by minimizing false discovery. Additionally, it provides a theoretical estimate of the false discovery rate (FDR) in real-world datasets where the ground truth features that constitute the generative mechanism for Y are unknown. Initially proposed as *Fixed-X knockoffs*[37] for low-dimensional cases where $N \geq p$, the method was later extended to other cases[36] under the name *Model-X knockoffs*, which is the focus of discussion here.

In the *Model-X* scheme, a copy $\widetilde{\boldsymbol{X}}_{\boldsymbol{j}}$ (where $j \in [1, 2, \dots, p]$) of each original feature data column $\boldsymbol{X}_{\boldsymbol{j}}$ is generated in a special manner to represent the data of an abstract feature variable $\widetilde{F}_j$ called the *knockoff* feature of F_j. Let $\widetilde{\boldsymbol{F}}$ be the set of all the knockoff features for $j \in [1, 2, \dots, p]$, and let $\widetilde{\boldsymbol{X}}$ be the corresponding knockoff feature data matrix, obtained by concatenating all $\widetilde{\boldsymbol{X}}_{\boldsymbol{j}}$ columns in order. For the method to be effective, $\widetilde{\boldsymbol{X}}$ must be designed to be independent of Y, meaning that all features in $\widetilde{\boldsymbol{F}}$ are inherently irrelevant in explaining Y. Additionally, the distribution of $\widetilde{\boldsymbol{X}}$ should match that of $\boldsymbol{X}$, ensuring that each $\boldsymbol{X}_{\boldsymbol{j}}$ and its knockoff $\widetilde{\boldsymbol{X}}_{\boldsymbol{j}}$ are statistically indistinguishable.

To achieve these properties, $\widetilde{\boldsymbol{X}}$ must satisfy the *pairwise exchangeability* property, as illustrated in Figure 4.10. In simple terms, the joint distribution of any two data columns, $\boldsymbol{X}_{\boldsymbol{j}}$ and $\boldsymbol{X}_{\boldsymbol{k}}$ ($j \neq k$), must remain unchanged even if one or both columns are swapped with their knockoff counterparts. Designing such $\widetilde{\boldsymbol{X}}$ is particularly challenging when the correlation structure is complex, as seen in the correlation heatmaps of the hypothetical example in Figure 4.10.

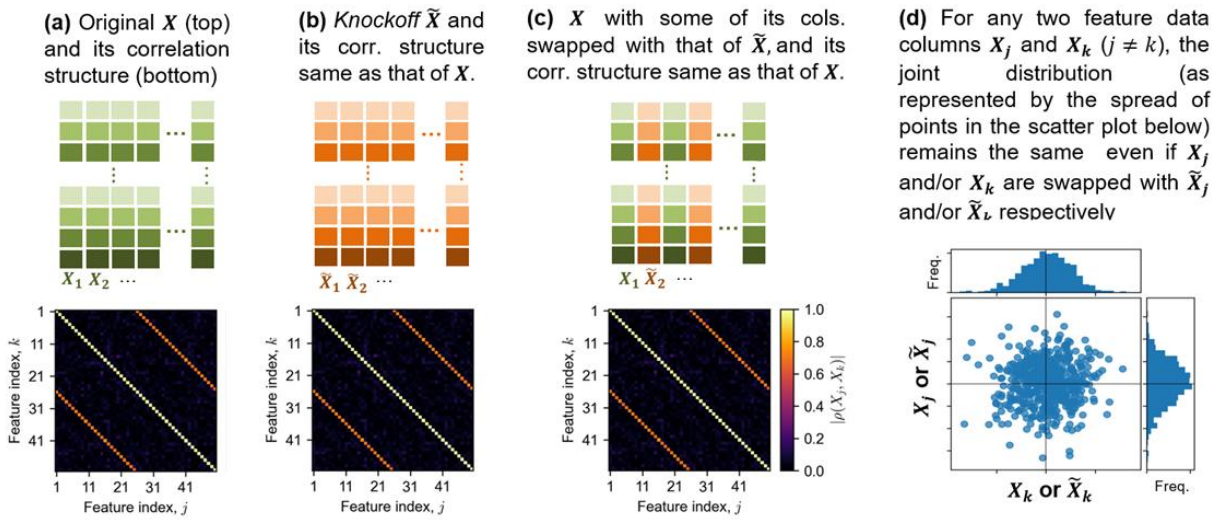

Figure 4.10. Demonstration of the pairwise exchangeability property of the knockoff feature data $\tilde{X}$. In (a)-(c), the correlation structure is displayed using a heatmap that corresponds to the absolute values of the pairwise Pearson correlation coefficients (ρ) between the features data columns.

The most challenging step in generating such $\tilde{X}$ is extracting knowledge about the distribution of X. This is difficult especially in the *small* data regime, where the available data may not be sufficiently representative to capture the true underlying distribution. As a result, only approximate $\tilde{X}$ that satisfies the exchangeability property to some extent[96] can be generated. A common strategy is to assume that X follows a multivariate Gaussian distribution and estimate its covariance matrix Σ_X, which is then used to produce a *second order approximation* of $\tilde{X}$, as shown in Figure 4.11a. If X is purely Gaussian, this method generates exact *model-X knockoff* data $\tilde{X}$. While this approach is simple and computationally efficient, it may not perform well on scientific datasets where one or more features deviate from a Gaussian distribution.

For such cases, an alternative deep learning approach using a neural network to learn the underlying distribution of X has been proposed.[94] However, this method is inefficient for small

datasets and requires slight modifications to generate a reliable approximation of $\widetilde{\boldsymbol{X}}$, known as *deep knockoff* data. Figure 4.11b illustrates a strategy for generating deep knockoff $\widetilde{\boldsymbol{X}}$ from $\boldsymbol{X}$ with minimal trials. During the intermediate step of preparing the synthetic data matrix for neural network training, domain expertise is valuable in effectively estimating the distribution of $\boldsymbol{X}$.

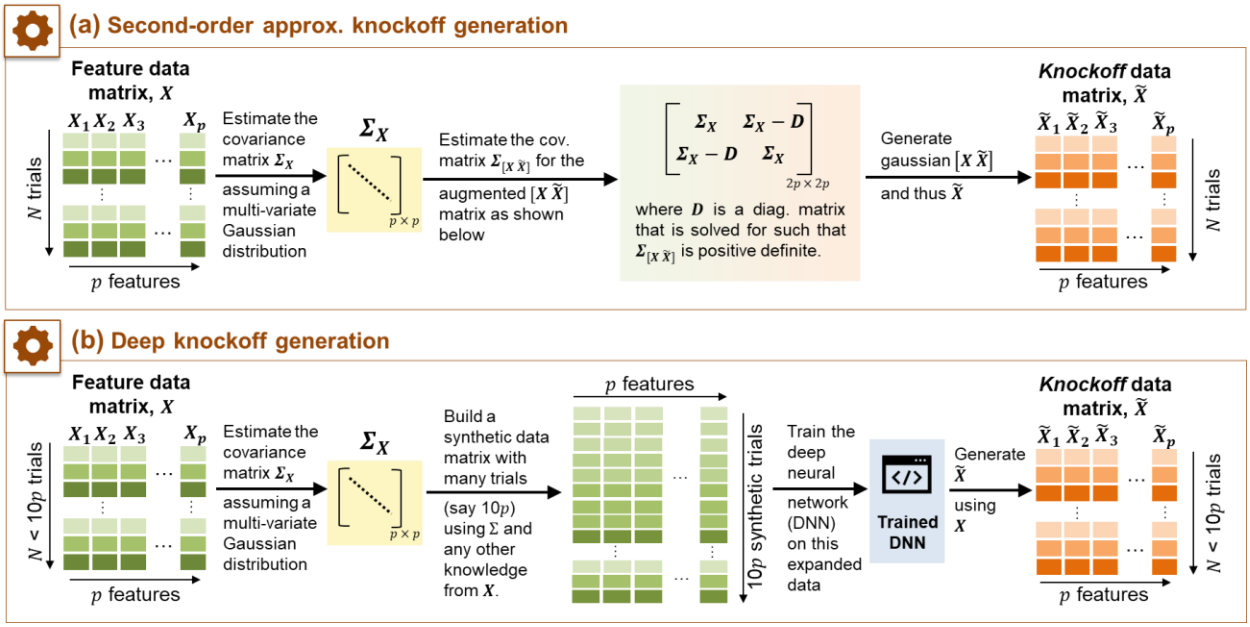

Figure 4.11. Knockoff feature data generation. (a) Second order approximation **(b)** Deep Knockoff data generation when the number of trials is small, say $N < 10p$, where p is the number of features. When N is larger than $10p$, then the deep neural network can be directly trained on the original $\boldsymbol{X}$.

After generating the knockoff data matrix $\widetilde{\boldsymbol{X}}$, the knockoffs scheme (Figure 4.12) is applied to obtain the final solution $\boldsymbol{F^S}$ with an optimal sparsity level s. Step 2 (Figure 4.12) typically involves using a sparse learner g that functions as a nested embedded feature selection method operating over the augmented $[\boldsymbol{X}, \widetilde{\boldsymbol{X}}]$ matrix. If g does not preferentially select an original feature F_j over its knockoff counterpart $\tilde{F}_j$, then F_j can be eliminated as irrelevant. To quantify the importance of F_j relative to the inherently irrelevant $\tilde{F}_j$, a feature importance statistic W_j is used.

Figure 4.13 presents the common feature importance statistics used in the *knockoffs* scheme. While originally designed for the traditional ℓ_1 model as g, these metrics can also be extended to other sparse models, such as ℓ_1^{AD}, ℓ_1^{IR}, ℓ_0, and $\ell_0\ell_2$.[97] Other types of W_j have been proposed recently[98–100], offering slight improvements over those in Figure 4.13. Steps 3-4 in the knockoffs scheme (Figure 4.12) theoretically guarantee that the false discovery rate (FDR), which is the fraction of irrelevant in the solution $\boldsymbol{F^{S_{q_k}}}$, does not exceed the nominal value q_k used in its derivation.[97,101] Thus, this method not only controls FDR but also provides a theoretical estimate, which is valuable for real-world datasets where ground truth features are typically unknown, making exact FDR evaluation impossible. To obtain the final sparsity level s and the corresponding $\boldsymbol{F^S}$, one can select the $\boldsymbol{F^{S_{q_k}}}$ with the smallest nominal FDR value q_k.

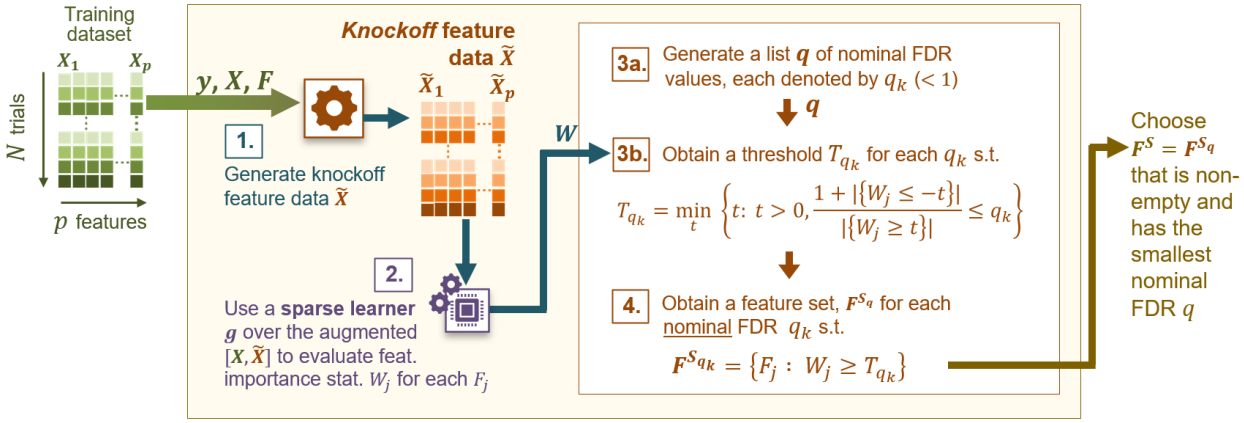

Figure 4.12. Flowchart describing the Knockoff feature selection scheme.

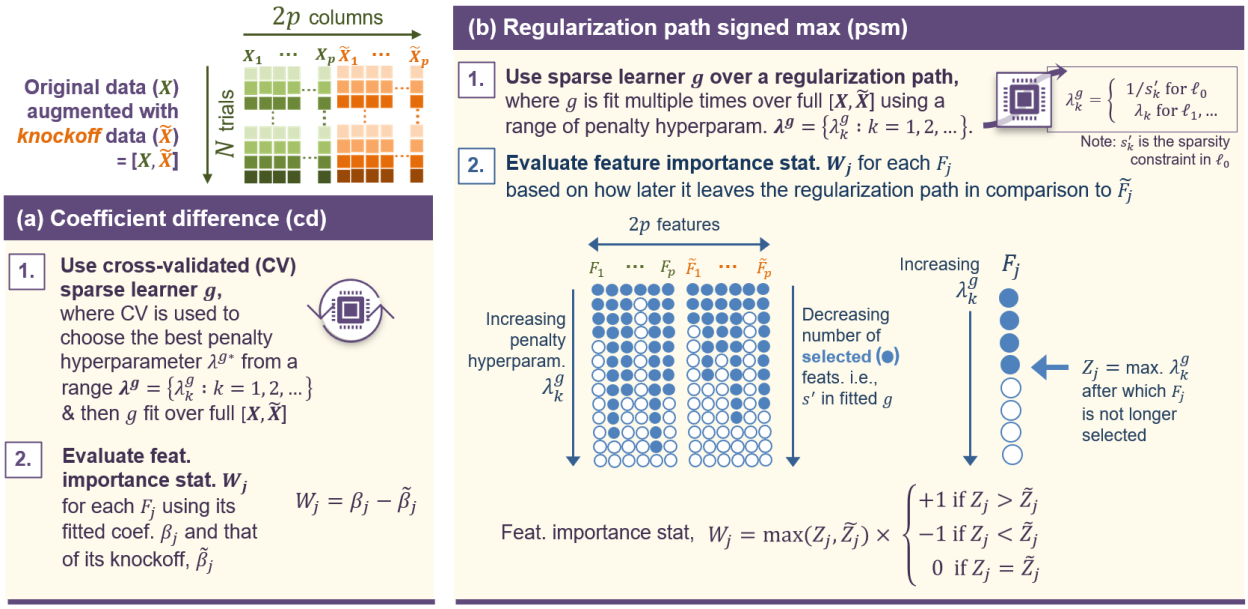

Figure 4.13. Evaluating the feature importance statistic W_j for use in knockoffs scheme.[36,37] **(a)** Coefficient difference (cd) measure **(b)** Regularization path signed maximum (psm) measure.

5 FEATURE SELECTION WITH SYNTHETIC DATA

Before applying the feature selection methods discussed in Chapter 4 to the PSC t_{80} lifetime datasets, it is important to first evaluate their efficacy. Synthetic data—where target variable data is artificially generated with relationships based on known ground truth variables—provides an ideal setting for assessing how well these methods can recover the underlying ground truth variables (either directly or through correlated 'proxy' features) from $\boldsymbol{F}$, while avoiding the irrelevant features. The enable evaluation of certain metrics such as *false-discovery rate* (FDR), *ground-truth recovery rate* (θ) and *redundancy* (ξ), which are discussed in detail in this chapter. These metrics assess how closely the selected feature subset $\boldsymbol{F^S}$ aligns with the ground-truth variables. Unlike common metrics like prediction error, which relies solely on the observed and predicted values of the target variable, accurately estimating FDR, θ and ξ requires exact knowledge of which features in $\boldsymbol{F}$ are truly relevant, irrelevant or redundant. However, this information is unavailable in real-world scenarios, which is precisely why feature selection methods are employed in the first place.

Additionally, synthetic datasets provide greater control over data properties such as N, p, noise level, probability distributions, non-linearity in target-feature dependencies, the proportion of relevant features in $\boldsymbol{F}$ and the degree of collinearity among features. These properties can be systematically varied to study their effects on the feature selection methods.[45] As the underlying mechanism for generating the values of Y can be specified in synthetic datasets, they can be designed to mimic typical datasets from a specific domain. When feature selection algorithms are evaluated on such synthetic datasets that emulate real-world conditions, the resulting inferences

can be heuristically extended to real-world scenarios.[45] These datasets are particularly valuable for selecting a "good" feature selection algorithm for a given domain, especially when real-world data is scarce or lacks diversity. In this chapter, I present several types of synthetic data generated to evaluate the feature selection methods suited for small datasets. Furthermore, I also discuss modified versions of FDR and θ metrics, which are more general and robust than their traditional counterparts. The results of these simulations provide insight into how these methods perform on the real-world t_{80} lifetime datasets discussed in Chapters 6 and 7.

5.1 Designing Synthetic Datasets

The design of a synthetic dataset is controlled by several factors, such as the number of features, parameters influencing the generative mechanism for the target values, the strength of the influence exerted by each ground truth variable within the mechanism, the relationships between the ground-truth variables and the features, and the correlations among the features. Each factor can be adjusted to generate a myriad of combinations, with each representing a distinct type of synthetic data. However, in this chapter, I focus on a small subset of these combinations that effectively capture the characteristics of small real-world datasets commonly encountered in fields such as materials science and chemistry. I present three types of synthetic data, labeled as *Type I* to *III* based on the increasing complexity of their feature correlation structures (see Figure 5.1a-c). Within each type, I further vary additional controlling factors to create datasets with different levels of complexity, as discussed later. In all types, a total of p = 50 features and four ground-truth variables (denoted as p_0 = 4) are used.

Type I is the simplest type, with a Pearson correlation coefficient of approximately 0.2 between each pair of features (see Figure 5.1d). This type of synthetic dataset, where features are only weakly correlated with each other, is commonly used for preliminary evaluation of selection algorithms. In *Type II*, $\boldsymbol{F}$ is divided into three equal-sized groups—labeled as $\boldsymbol{F^A}$, $\boldsymbol{F^B}$ and $\boldsymbol{F^C}$—which together create a distinct correlation structure that challenges feature selection methods with variables exhibiting strong dependencies among them (see Figure 5.1e). Features in $\boldsymbol{F^A}$ and $\boldsymbol{F^B}$ are designed to be only weakly correlated with each other, as in *Type I*. Meanwhile, features in $\boldsymbol{F^C}$ are individually correlated with those in $\boldsymbol{F^B}$ in a one-on-one correspondence. Out of the four ground-truth variables, two are selected from $\boldsymbol{F^A}$ and two from $\boldsymbol{F^B}$.

Type III is identical to *Type II*, except that the two ground-truth features from $\boldsymbol{F^B}$ are removed, while maintaining a total of p features in the final menu $\boldsymbol{F}$. This forces feature selection methods to rely on the correlated features from $\boldsymbol{F^C}$ to indirectly incorporate ground-truths into the solution—a possible scenario in real-world data. While synthetic datasets have been widely used in previous studies to compare feature selection methods, they are mostly of *Type I*, as introduced here, and often fail to adequately represent the real-world scenarios. The dataset types introduced in this study aim to address this limitation. For all dataset types, the target variable Y is formed by a linear combination of the ground-truths with fixed coefficient values, plus an intercept of one.

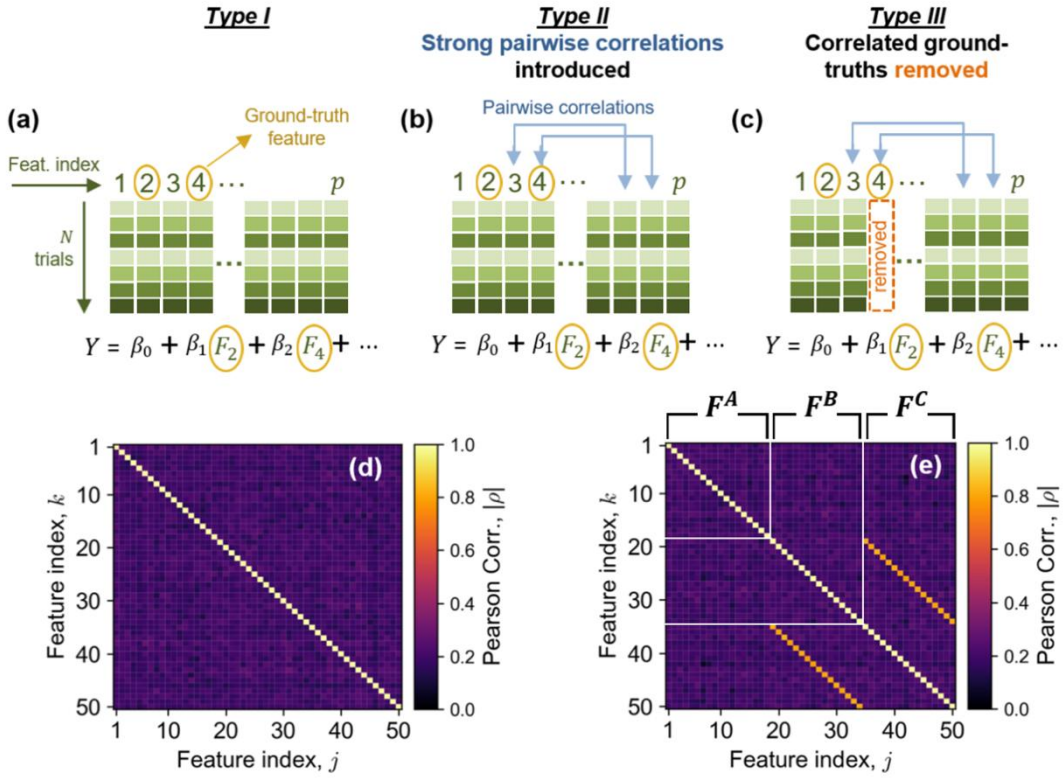

Figure 5.1. (a-c) Schematics of the three types of synthetic datasets as described in Section 5.1. **(d)** Heatmap of the correlation structure for a typical *Type I* dataset, such that the mean of the Pearson correlation coefficients of all feature pairs (F_j, F_k) (where $1 \leq j,\ k \leq 50$ and $j \neq k$) is 0.2. **(e)** Heatmap of the correlation structure for a typical *Type II* dataset, highlighting three feature groups—$\boldsymbol{F^A}$, $\boldsymbol{F^B}$ and $\boldsymbol{F^C}$—based on the pairwise Pearson correlations (maximum correlation = 0.8). Out of the four ground truth variables, two are selected from $\boldsymbol{F^A}$ and two from $\boldsymbol{F^B}$. *Type III* datasets follow the correlation structure as in (e) but are initially generated with 52 variables. Out of the four ground truth variables, the two chosen from $\boldsymbol{F^B}$ are subsequently removed to form the final feature set with $p = 50$.

5.2 Simulation Setup

Each feature selection method is applied to the three types of synthetic datasets introduced in Section 5.1. Fixing $p = 50$ and $p_0 = 4$, the sample size N and the target signal-to-noise ratio (SNR) are varied within each dataset type. Within each dataset, 50 data-points are initially set aside

as the test set, while the training set varies with varying sample sizes N. For each combination of N, SNR and dataset type, 20 unique datasets are generated and fitted separately to obtain averaged results. Across these replicates, the correlation structure of the feature data matrix $\boldsymbol{X}$, the ground-truth variables, and their corresponding coefficient values remain unchanged. Thus, any variation observed in feature selection across these datasets reflects each method's sensitivity to fluctuations in $\boldsymbol{X}$. The feature selection methods used in the simulations are listed in Table 5.1. Methods that employ the knockoffs scheme to determine the model's sparsity level are labeled with a "*/ko+*" suffix, while those using cross-validation (CV) retain their original names.

For the simulations in Section 5.4, the four ground truth variables are assigned fixed coefficients of 0.9, 0.1, 0.9 and 0.1 respectively. In *Types II* and *III*, each ground-truth pair selected from $\boldsymbol{F^A}$ and $\boldsymbol{F^B}$ receives the coefficients [0.9, 0.1] respectively. Variations in coefficient magnitudes pose a challenge for feature selection methods in recovering ground-truth variables with smaller coefficients. As a control study, I repeated these simulations with all four ground-truth coefficients fixed at 0.5 (see Section 5.6.2 in the Supporting Information for results).

All features in the datasets from Sections 5.4 and 5.6.2 follow a Gaussian distribution—a common simplifying approximation in synthetic datasets to mimic unimodal distributions observed in real data. However, many real-world datasets also contain features with flat or uniform distributions. For instance, a stimulus variable such as temperature may be uniformly varied by an experimenter to observe its effects on the target variable. To evaluate how feature selection methods respond to such non-Gaussian features, I conducted another set of simulations where all features were uniformly distributed, and the ground-truth coefficients were fixed at [0.9, 0.1, 0.9,

0.1] (see Section 5.6.3 in the Supporting Information for results). The full list of control parameters and their values are listed in Table 5.1.

Given that the noise levels are not excessively high, all datasets fall within the *small data* regime (as $N < 10p$ = 500). Moreover, no dataset is *too small* for modeling, as all N values listed in Table 5.1 exceed the estimated m_s^0, given by $m_s^0 \approx p_0 \log_2(p - p_0)/(1 - \rho) \sim 28$ (see Section 3.2 for more details). Here, $s = p_0$ represents the ideal sparsity level when a model performs exact sparse recovery, and m_s^0 represents the corresponding effective degrees of freedom. For the estimation of m_s^0, ρ is set to 0.2, which is the mean absolute Pearson correlation coefficient for all dataset *Types* (see Figure 5.1d-e).

Table 5.1. Values of the control parameters used for evaluation of the feature selection methods.

Control parameter	**Control values**	
Type of dataset	*Type - I, II, III*	
# of selectable features (p)	50	
# of ground-truth features (p_0)[a]	4	
Training data size (N)	50, 75, 100, 300, 1000	
Signal-to-noise ratio (SNR)[b]	4, 6, 8, 12, 512	
Feature selection methods[c] (Method)	CV	ℓ_1, ℓ_1^{AD}, ℓ_1^{IR}, ℓ_0, $\ell_0\ell_2$, ISIS, OMP, Boruta
	Knockoffs[d]	ℓ_1/ko+, ℓ_1^{AD}/ko+, ℓ_1^{IR}/ko+
Ground-truth coefficients	[0.9, 0.1, 0.9, 0.1]	(Section 5.4 and 5.6.3)
	[0.5, 0.5, 0.5, 0.5]	(Section 5.6.2)
Feature data distribution	Gaussian	(Section 5.4 and 5.6.2)
	Uniform	(Section 5.6.3)

[a] Feature indices designated as ground-truths are fixed. For *Types II* and *III*, two out of four ground-truths are taken from $\boldsymbol{F^A}$, while the remaining two are from $\boldsymbol{F^B}$ (see Figure 5.1e).
[b] SNR is the ratio of variance of true $\boldsymbol{y}$ to that of the added gaussian noise, indicating that a lower SNR corresponds to higher noise levels.
[c] ℓ_1^{AD} = Adaptive ℓ_1; ℓ_1^{IR} = Iteratively reweighted ℓ_1; ISIS = Iterative Sure Independence Screening, OMP = Orthogonal Matching Pursuit.
[d] *Knockoffs* scheme here uses second-order approximation for knockoff feature data construction, and regularization path signed maximum (psm) as the feature importance statistic (see Section 4.8.3 for more details).

5.3 Characterization Metrics for Simulations

Test prediction errors and goodness-of-fit are widely used preliminary metrics for evaluating the effectiveness of a feature selection method. However, in various scientific contexts, the selected feature subset $\boldsymbol{F^S}$ is also expected to be condensed, comprising only a few features that are physically meaningful and sufficiently predictive. As a result, a smaller $\boldsymbol{F^S}$ (one with a small s) is often preferred over the subset with the lowest test error when the difference in error is marginal. In synthetic datasets, the key advantage is the precise knowledge of ground-truth variables, enabling the evaluation of additional properties of $\boldsymbol{F^S}$—such as false-discovery, redundancy and ground-truth recovery—characteristics that are otherwise infeasible to assess in real-world settings. These are explained below.

The tendency of a feature selection method to select irrelevant features is commonly known as *false discovery*. Among the s features in $\boldsymbol{F^S}$, let s_ϕ denote the number of irrelevant features. The *false discovery rate* (FDR) is then defined as follows:

$$FDR = \frac{s_\phi}{s} \tag{6}$$

This represents the proportion of selected features that are *falsely* identified as relevant. Additionally, correlations may exist among the remaining $s - s_\phi$ features that are relevant, rendering some *redundant* for predicting Y. Let s_ξ denote the number of such redundant features in $\boldsymbol{F^S}$, which do not provide any new information about Y but unnecessarily inflate s. The metric ξ is then defined as follows:

$$\xi = \frac{s_\xi}{s} \tag{7}$$

It represents the proportion of selected features that are relevant but redundant.

Incorporating the underlying ground truth variables (which may be indirectly present in $\boldsymbol{F}$ through correlated relevant features) is crucial for a feature selection method to achieve high predictive power. The ability of a feature selection method to accurately identify such relevant features from $\boldsymbol{F}$, referred to here as *ground truth recovery*, can be assessed using a metric θ. Let p_0 denote the number of ground truth variables that constitute the underlying mechanism generating the values of Y. Then, θ is defined as the fraction of ground truth variables incorporated into $\boldsymbol{F^S}$, calculated as follows:

$$\theta = \frac{p_0^S}{p_0} \tag{8}$$

Here, p_0^S represents the selected ground truth variables appearing in $\boldsymbol{F^S}$ either directly or indirectly through correlated features, out of a total of p_0.

Highly sparse solutions (that is, those with very small s) risk excluding ground truth variables, thereby reducing θ. Conversely, solutions with many selected features may increase θ, but also inflate FDR and ξ by incorporating many irrelevant and redundant features. An effective feature selection method aims to achieve a solution with a small value of s and low prediction error while maintaining a good balance of reasonably high θ, low FDR, and low ξ.

Calculating the test error and s is straightforward, but determining other metrics can be challenging depending on the design of the synthetic datasets. Metrics like FDR and θ work well for *Type I and II* datasets. However, for the *Type III* datasets, where the ground truth variables are excluded from $\boldsymbol{F}$ and are only accessible indirectly through correlated features, these metrics become ineffective. In such cases, FDR is invariably one, as any feature that is not a ground truth

is categorized as wholly irrelevant, even if it is correlated with a ground truth. Similarly, θ is invariably zero, as no ground truth is considered fully recovered unless it is directly present in $\boldsymbol{F}$ and selected.

To address this issue, I developed new metrics FDR^+ and θ^+, which count features "fractionally" rather than wholly as in their traditional counterparts. FDR^+ employs mutual information[85] (MI)—an information-theoretic metric that quantifies the information shared between the data of two variables. This allows features in $\boldsymbol{F}$ to be considered partially relevant or irrelevant based on the amount of information they share with the ground truths through correlations. Rather than categorizing each feature that is not a ground truth as wholly irrelevant, FDR^+ accounts for the fractional contribution of each feature based on its deviation from the ground truth variables. Meanwhile, θ^+ simply counts every ground truth variable that is selected—whether wholly or partially through correlated features. Section 5.6.1 explains these metrics in more detail.

5.4 Simulation Results

In this section, I discuss the simulation results for synthetic datasets in which all features follow Gaussian distributions and the ground-truth coefficients are fixed at [0.9, 0.1, 0.9, 0.1]. Results for the other synthetic datasets—one with Gaussian-distributed features and ground-truth coefficients are fixed at [0.5, 0.5, 0.5, 0.5], and another with uniformly-distributed features and ground-truth coefficients are fixed at [0.9, 0.1, 0.9, 0.1]—are presented in Sections 5.6.2 and 5.6.3 respectively

in the Supporting Information. As these mostly show only minor differences from the results discussed here, they are referenced briefly, with attention drawn only to the major differences.

5.4.1 Prediction Errors and Sparsity Levels

Figure 5.2 displays the outcomes of applying the feature selection methods (outlined in Sections 4.2 and 4.3) for $N = 50$ and $SNR = 6$. While the mean squared test errors (normalized to the variance of noise) of all methods do not change significantly from *Type I* to *II*, they increase from *Type II* to *III* (Figure 5.2a-c), as expected, due to removal of the ground-truth variables from $\boldsymbol{F^B}$ in *Type III* datasets. Similarly, the test R^2 scores (Figure 5.2d-f), which quantify the alignment of test data along the observed versus predicted parity line, remain unchanged from *Type I* to *II* but decline from *Type II* to *III*. However, the variation of errors and R^2 scores across the selection methods remains insignificant within each dataset type.

In contrast, the sizes of $\boldsymbol{F^S}$ (that is, the sparsity levels) for certain methods—ℓ_1, ISIS, and OMP—are significantly higher than those of others (Figure 5.2g-i). This underscores that prediction error alone is not a reliable indicator for selecting an optimal feature selection method. To understand why some methods can produce sparser solutions without a substantial loss in prediction accuracy, it is worthwhile to evaluate their selectivity towards irrelevant, relevant, and redundant features using the FDR^+, θ^+ and ξ metrics respectively.

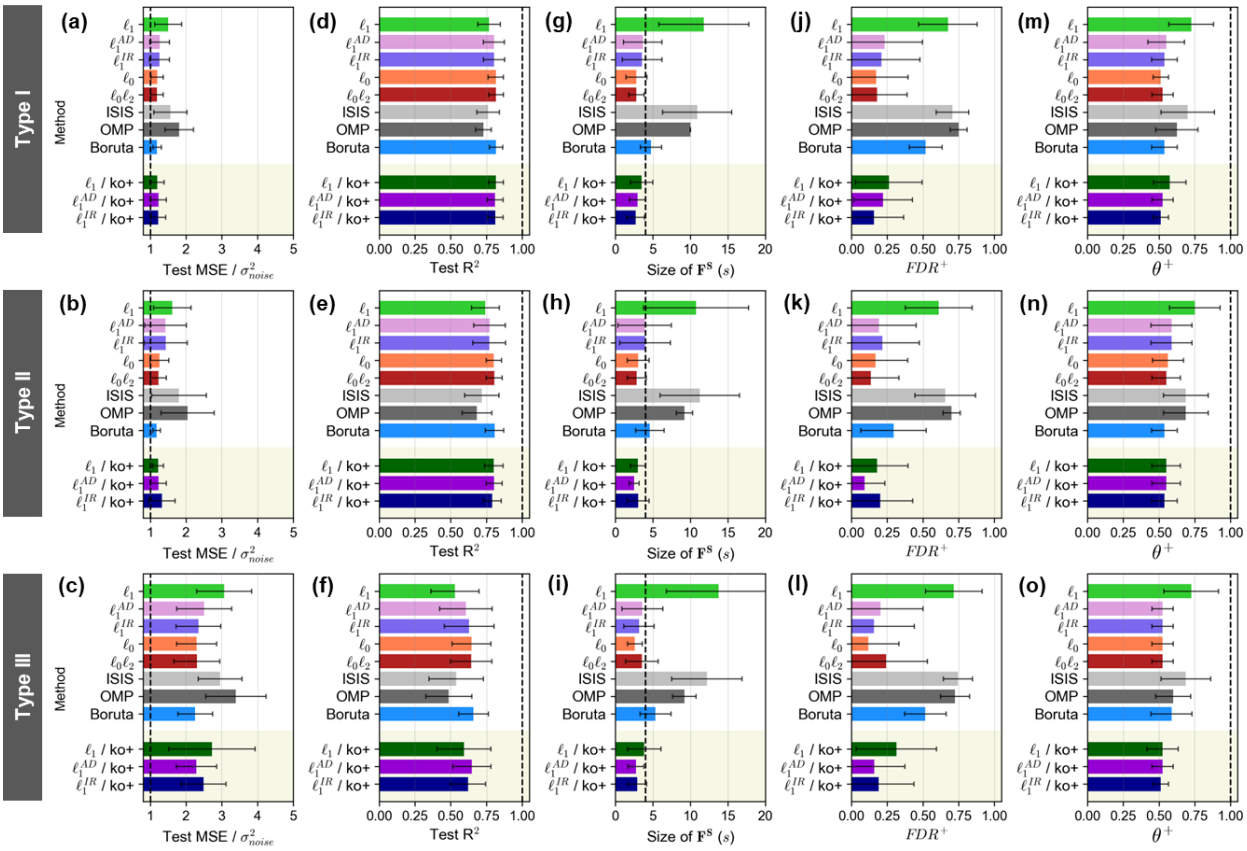

Figure 5.2. Simulation results for N = 50 and SNR = 6 for *Types I-III*, where p = 50 and p_0 = 4 , and ground-truth coefficients = [0.9, 0.1, 0.9, 0.1]. (a-c) Mean squared test error (MSE) normalized with the variance of the added gaussian noise (σ^2_{noise}) **(d-f)** R^2 score between observed and predicted $\boldsymbol{y}$ values of the test data **(g-i)** Size (s) of the selected feature subset $\boldsymbol{F^S}$ **(j-l)** Fractional false discovery rate FDR^+ **(m-o)** Fractional ground-truth recovery θ^+. For each metric, the bar value represents its mean, while the error bar represents its standard deviation across the 20 artificial datasets used for the (N, SNR, dataset *Type*) combination. The dashed vertical line represents the test MSE = σ^2_{noise} line in (a-c), the R^2 = 1 line in (d-f), the $s = p_0 = 4$ line in (h-j), and $\theta^+ = 1$ line in (k-m). Methods with the suffix 'ko+' (highlighted by a gray shade) use the knockoffs scheme, as outlined in Section 4.3.2 for sparsity level control. The rest use leave-one-out cross-validation to determine the sparsity level. All features follow a Gaussian distribution.

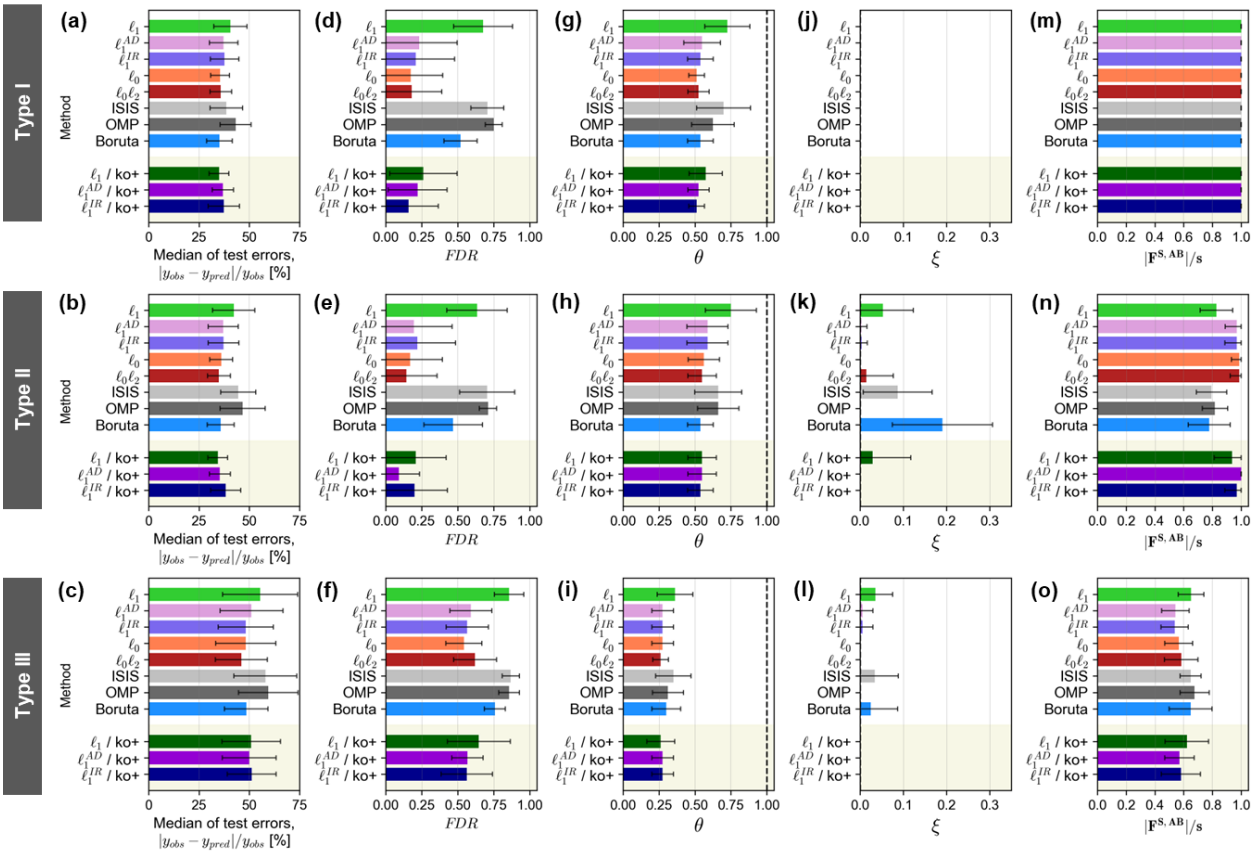

Figure 5.3. Simulation results for N = 50 and SNR = 6 for *Types I-III*, where p = 50 and p_0 = 4 , and ground-truth coefficients = [0.9, 0.1, 0.9, 0.1]. (a-c) Median percentage test error **(d-f)** Traditional false discovery rate FDR **(g-i)** Traditional ground-truth recovery rate θ **(j-l)** Redundancy ξ **(m-o)** Fraction of selected features belonging to the $\boldsymbol{F^A}$ and $\boldsymbol{F^B}$ groups (represented as $\boldsymbol{F^{S,AB}}$). In *Type I*, all features belong to the $\boldsymbol{F^A}$ group, making this fraction always equal to one. For each metric, the bar value represents its mean, while the error bar represents its standard deviation across the 20 artificial datasets used for the (N, SNR, dataset *Type*) combination. The dashed vertical line represents the θ = 1 line in (g-i). Methods with the suffix 'ko+' use the knockoffs scheme (highlighted by a gray shade), as outlined in Section 4.3.2 for sparsity level control. The rest use leave-one-out cross-validation to determine the sparsity level. For *Type III*, θ and FDR invariably take 0 and 1, respectively as shown in (i) and (l). For *Type I*, ξ invariably takes 0 as shown in (j).

5.4.2 False-Discovery and Ground-Truth Recovery

As the ground-truth features are known for the synthetic datasets, the results of FDR^+ and θ^+ are also available for the simulations (Figure 5.2j-o). For *Types I* and *II*, these metrics exhibit values similar to their traditional counterparts, FDR and θ (Figure 5.3d-i). However, for *Type III*, where the two ground-truth variables from $\boldsymbol{F^B}$ are removed, θ values drop significantly across all methods, while FDR values rise abruptly—both reaching levels that exhibit reduced variation across methods. Despite these shifts, FDR^+ and θ^+ remain stable, continuing to exhibit noticeable variation across methods. This illustrates how the refined versions of these metrics are particularly useful to compare these methods in specialized synthetic data scenarios like *Type III*.

Among the methods, ℓ_1, ISIS, and OMP consistently exhibit slightly larger θ^+. When recovering ground-truth variables with smaller coefficients, these methods perform noticeably better than the rest (see Figure 5.4), which explains their larger θ^+ values. However, when all ground-truth variables are assigned the same coefficient values, θ^+ approaches unity across all methods due to the easy recovery of all four ground-truth variables (see Figure 5.8 in Supporting Information). Meanwhile, regardless of the ground-truth coefficient values, ℓ_1, ISIS, and OMP consistently tend to select numerous irrelevant features, as evidenced by their significantly high FDR^+ values, ultimately undermining the advantage of their higher θ^+ values. These methods, characterized by low prediction errors but high s, and consequently high FDR^+ values, are best suited for purely predictive applications where obtaining a compact feature subset with only a few meaningful variables is not a priority.

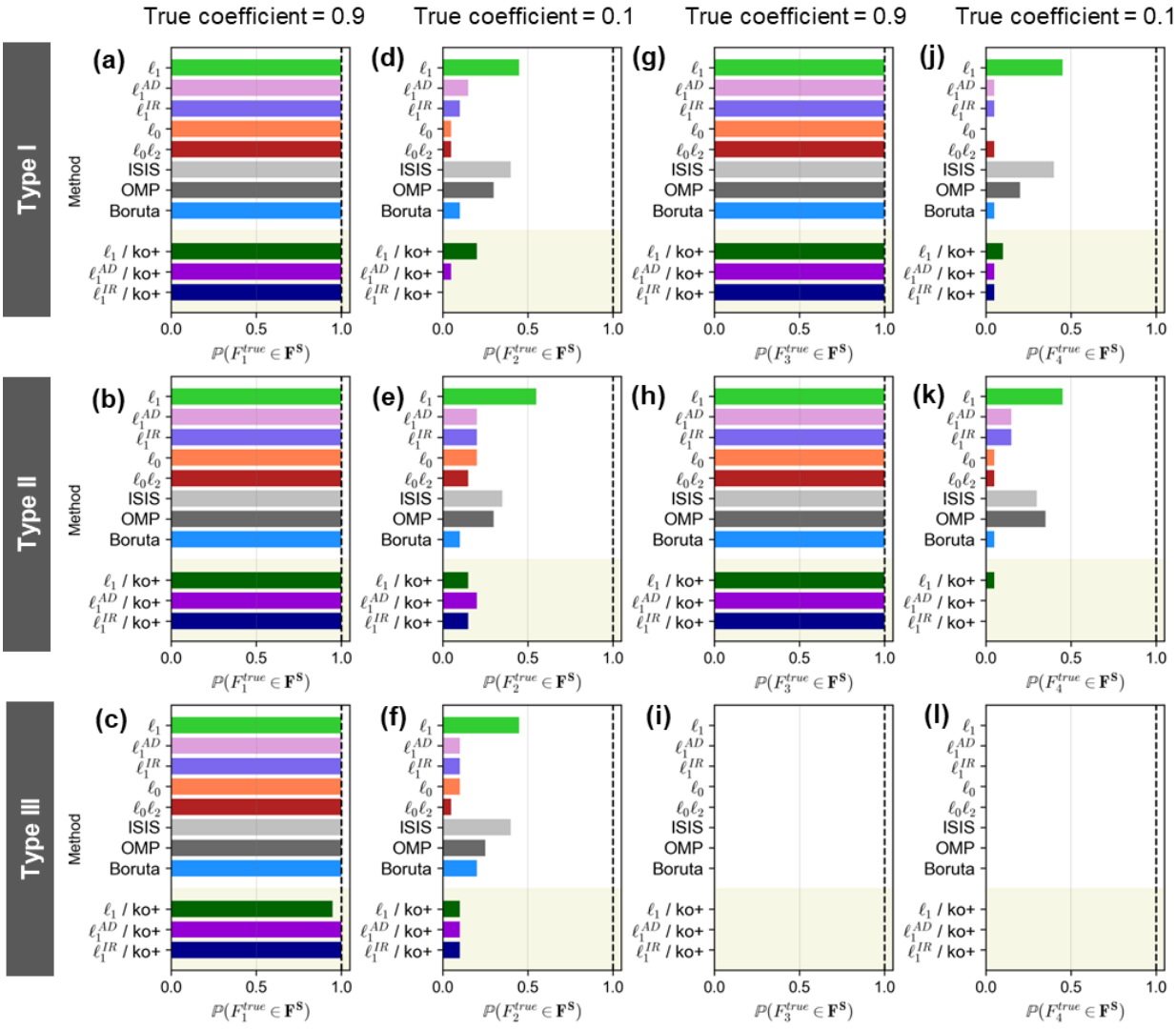

Figure 5.4. Simulation results for N = 50 and SNR = 6 for *Types I-III*, where p = 50 and p_0 = 4, and ground-truth coefficients = [0.9, 0.1, 0.9, 0.1]. Selection probabilities of ground-truth variables whose coefficients in the generative mechanism are set to **(a-c)** 0.9 **(d-f)** 0.1 **(g-i)** 0.9 **(j-l)** 0.1. For each (N, SNR, dataset *Type*) combination, the selection probability $\mathbb{P}(F_k^{true} \in \boldsymbol{F^S})$ of a given ground-truth feature F_k^{true} (where $1 \leq k \leq p_0$) is estimated by the fraction of the number of times the feature is selected among the 20 datasets generated with the combination. The dashed vertical lines represent the probability of one. In *Type III*, where F_3^{true} and F_4^{true} are absent from $\boldsymbol{F}$, their selection probabilities are invariably zero. Methods with the suffix 'ko+' use the knockoffs scheme (highlighted by a gray shade), as outlined in Section 4.3.2 for sparsity level control. The rest use leave-one-out cross-validation to determine the sparsity level.

Analyzing the correlation patterns in the *Type II* and *III* datasets—comprising the partitions $\boldsymbol{F^A}$, $\boldsymbol{F^B}$ and $\boldsymbol{F^C}$—provides insight into some of the variations across these methods (see Figure 5.1e). In *Type III*, little variation is observed between the methods regarding their selectivity towards features from $\boldsymbol{F^A}$ and $\boldsymbol{F^B}$ (see Figure 5.3o). However, in *Type II*, the presence of both ground-truth variables from $\boldsymbol{F^B}$ and their correlated counterparts from $\boldsymbol{F^C}$ poses a challenge to this behavior. The weighted ℓ_1 methods (i.e., ℓ_1^{AD} and ℓ_1^{IR}), ℓ_0, $\ell_0\ell_2$, and knockoffs-based methods preferentially select features from $\boldsymbol{F^A}$ and $\boldsymbol{F^B}$, while avoiding those from $\boldsymbol{F^C}$ as desired, demonstrating their resilience to correlations among features (see Figure 5.3n). This explains the small s and FDR^+ values of these methods, in contrast to ℓ_1, ISIS and OMP methods, which tend to select a considerable number of features from $\boldsymbol{F^C}$. Meanwhile, Boruta is a unique case; although it yields small s values by restricting its selection to ground-truth variables, it also redundantly selects the correlated counterparts from $\boldsymbol{F^C}$, which explains its large ξ value (Figure 5.3l).

5.4.3 Sensitivity to Fluctuations in $\boldsymbol{X}$

When a sparse solution (i.e., one with a small s) is desired, the weighted ℓ_1 methods, ℓ_0, $\ell_0\ell_2$, and knockoffs-based methods yield desirable results based on their averages across the 20 replicate datasets. However, it is also important to consider the error bars in Figure 5.2d-f, which represent the standard deviations of these results, while assessing these methods. As the correlation structure of the feature data matrix $\boldsymbol{X}$, the ground-truth variables, and their corresponding coefficient values remain unchanged across these replicate datasets, these standard deviations reflect each method's sensitivity to fluctuations in $\boldsymbol{X}$. For the sparsity level s, although the weighted

ℓ_1 methods exhibit lower averaged values, their standard deviations are much higher than those of ℓ_0, $\ell_0\ell_2$ and the knockoffs-based methods, indicating greater sensitivity (Figure 5.2g-i).

Figure 5.5 illustrates how the results of Figure 5.2 vary with N. According to the heuristic rules introduced in Section 3.2, although N = 50 doesn't fall within the *too small* regime (because $N > m_s^0 \approx 28$), it still remains below $10m_s^0$ (≈280), suggesting a risk of overfitting with certain sparse modeling methods, while others may perform adequately. Such overfitting may explain the large inconsistencies in s values observed for the weighted ℓ_1 methods across the replicate datasets, as evidenced by their substantial standard deviations. As shown in *Types I* and *II* (Figure 5.5g-h), the error bars around the mean s values shrink as these values converge towards p_0 with increasing N, whereas convergence cannot be guaranteed in *Type III*. ℓ_1, ISIS and OMP exhibit slow convergence at the current noise level (SNR = 6) but converge more rapidly at a lower noise level (SNR = 512, as shown in Figure 5.11 in the Supporting Information).

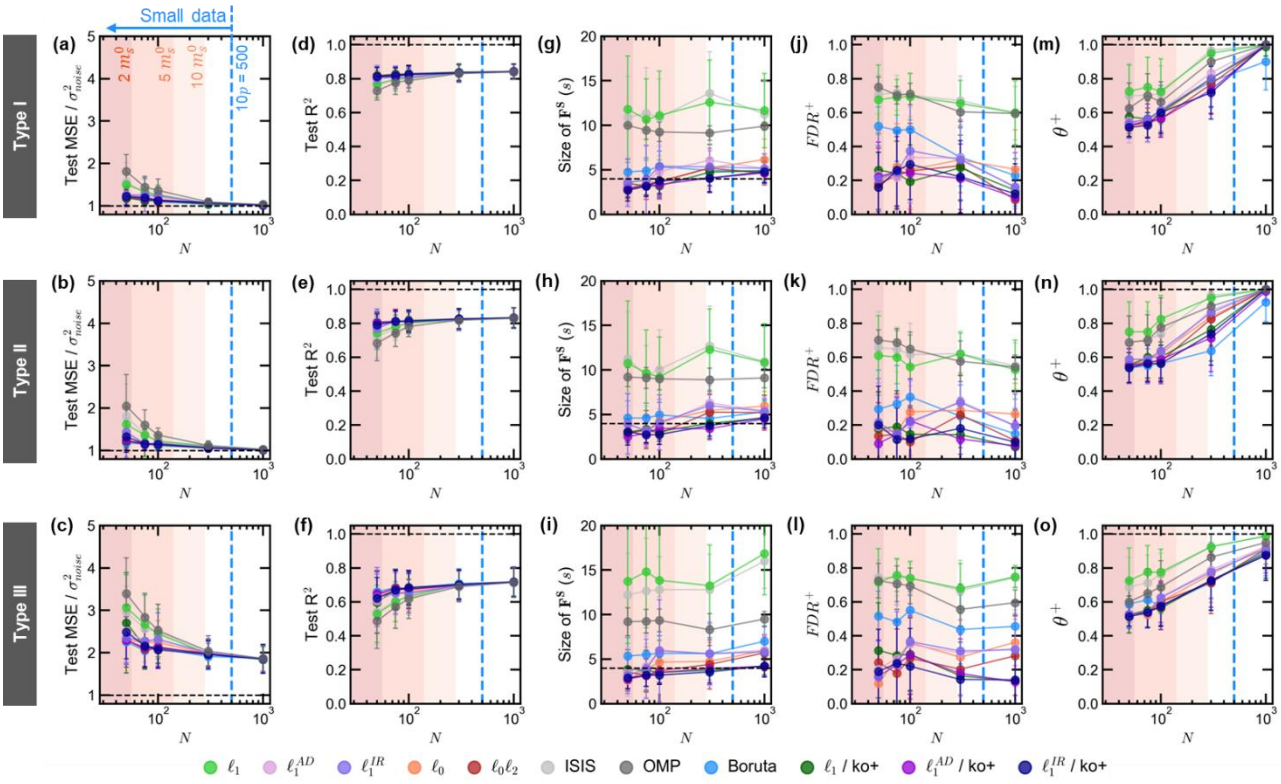

Figure 5.5. Simulation results for $N = [50, 75, 100, 300, 1000]$ and SNR = 6 for *Types I-III*, where $p = 50$ and $p_0 = 4$, and ground-truth coefficients = [0.9, 0.1, 0.9, 0.1]. (a-c) Mean squared test error (MSE) normalized with the variance of the added gaussian noise (σ^2_{noise}) **(d-f)** R^2 score between observed and predicted $\boldsymbol{y}$ values of the test data **(g-i)** Size (s) of the selected feature subset $\boldsymbol{F^S}$ **(j-l)** Fractional false discovery rate FDR^+ **(m-o)** Fractional ground-truth recovery θ^+. For each metric, the bar value represents its mean, while the error bar represents its standard deviation across the 20 artificial datasets used for the (N, SNR, dataset *Type*) combination. The dashed horizontal line represents the test MSE = σ^2_{noise} line in (a-c), the $R^2 = 1$ line in (d-f), the $s = p_0 = 4$ line in (h-j), and $\theta^+ = 1$ line in (k-m). Methods with the suffix 'ko+' use the knockoffs scheme, as outlined in Section 4.3.2 for sparsity level control. The rest use leave-one-out cross-validation to determine the sparsity level. The legend describing the markers is shown below the plots. All features follow a Gaussian distribution. The vertical blue dashed line represents $N = 10p = 500$ below which the *small data* regime begins. For a sparse model performing exact sparse recovery such that $s = p_0$, the degrees of freedom m_s^0 is given by $p_0 \log_2(p - p_0)/(1 - \rho) \approx 28$, with ρ set to 0.2 (see Section 3.2). The boundaries of the red shaded regions represent values of N corresponding to 2 m_s^0 (≈56), 5 m_s^0 (≈140) and 10 m_s^0 (≈280), as indicated by the labels.

5.4.4 Summary

In summary, ℓ_0, $\ell_0\ell_2$, and knockoffs-based methods display a good balance of low s, low FDR^+ and reasonably high θ^+ values without compromising predictive power, while demonstrating minimal sensitivity to fluctuations in feature data. When simulations are repeated in Section 5.6.3 in the Supporting Information with uniformly distributed features instead of Gaussian ones, the standard deviations of prediction errors and sparsity levels—particularly for ℓ_1^{AD} and ℓ_1 in *Type II*—increase (see Figure 5.9 in the Supporting Information), displaying these methods' high sensitivity to non-Gaussian distributions. In contrast, ℓ_0, $\ell_0\ell_2$, and the knockoffs-based methods remain stable, exhibiting reasonably small standard deviations.

5.5 Conclusions

Synthetic datasets, where ground-truth variables are known, enable certain evaluations of feature selection methods that are otherwise infeasible with real-world datasets. Many of the feature selection techniques discussed in Chapter 4 may underperform when applied to real-world datasets commonly encountered in fields of chemistry and materials science, due to several inherent complexities of these datasets. These include diverse feature data distributions including non-Gaussian ones, pairwise correlations within $\boldsymbol{F}$, the absence of ground truth variables in $\boldsymbol{F}$, and differences in the magnitudes of their underlying coefficients. As a result, synthetic datasets intended to evaluate the applicability of feature selection methods to real scientific data must account for such complexities—especially the possibility of ground truths being absent from $\boldsymbol{F}$, a factor often overlooked by the statistics and signal processing communities. This chapter begins

by designing three types of datasets with increasing complexities. To address cases where ground truths are absent from the feature set, I introduced two new metrics—FDR^+ and θ^+—which account for the fractional contributions of features to being relevant or irrelevant.

Among the feature selection methods discussed, ℓ_0, $\ell_o\ell_2$ and the knockoff-based methods emerged as promising alternatives to traditional approaches like ℓ_1 and OMP. These methods not only produced accurate and sparse solutions but also low FDR^+ and reasonably high θ^+ values—metrics that are impossible to evaluate in real-world scenarios. However, caution must be exercised when using ℓ_0 and $\ell_0\ell_2$ in cases with strong non-linear dependencies between the target variable and the features. Overall, these heuristic insights are valuable when applying these methods to practical scientific problems.

In the next two chapters, I evaluate these feature selection methods on two real-world datasets, where the features deviate from a Gaussian distribution and exhibit more complex correlation structures. While real-world datasets can be compared to *Types I* and *II* by analyzing their respective correlation structures, *Type III* cannot be directly compared to real world data. This is because, in practice, it is generally unknown whether the ground-truth variables are directly present in a real-world feature set. Nevertheless, *Type III* provides insight into how challenging a dataset can become, relative to *Type II*, if any ground-truths are missing from $\boldsymbol{F}$.

5.6 Supporting Information

5.6.1 Evaluation of FDR^+ and θ^+

Beyond test prediction error, metrics such as FDR and θ (as introduced in Section 5.3) can be precisely evaluated in simulations with synthetic data, where the ground-truth variables are known. In this section, I demonstrate how FDR and θ have been traditionally defined in statistics. Additionally, I discuss the refinements introduced in this work to generalize them.

Suppose there are no correlations between the features in $\boldsymbol{F}$. Let $\boldsymbol{F^0}$ represent the set of p_0 ground truth features that truly govern Y, such that $\boldsymbol{F^0} \subseteq \boldsymbol{F}$ and $p_0 < p$. Each feature in the selected subset $\boldsymbol{F^S}$ (of size s) is classified as a *true-positive* if it belongs to $\boldsymbol{F^0}$ and a *false-positive* otherwise. Traditionally, the false-discovery rate (FDR) is defined as the fraction of features in $\boldsymbol{F^S}$ that are *false-positives* and is evaluated as:

$$FDR = \frac{\sum_{k=1}^{s} \alpha_k^{\mathrm{FP}}}{s} \tag{9}$$

where α_k^{FP} takes a value of 1 if the corresponding F_k^S is a *false-positive* and 0 otherwise. Thus, $\boldsymbol{\alpha^{\mathrm{FP}}} = [\alpha_1^{\mathrm{FP}}, \alpha_2^{\mathrm{FP}}, \dots, \alpha_s^{\mathrm{FP}}]$ forms a binary status set indicating whether each feature in $\boldsymbol{F^S}$ is a *false-positive*. In non-statistical fields, *false-positives* are often loosely referred to as *irrelevant* features. Let s_ϕ represent the number of such irrelevant features. Then, $s_\phi = \sum_{j=1}^{s} \alpha_j^{\mathrm{FP}}$ and $FDR = s_\phi / s$ (see Section 5.3).

Besides false discovery, the ability of a features selection method in recovering the ground truths from $\boldsymbol{F}$ is assessed by a separate metric, represented here by θ, which is the fraction of ground truths from $\boldsymbol{F^0}$ that are selected into $\boldsymbol{F^S}$ (see Section 5.3) and is calculated as:

$$\theta = \frac{\left|\boldsymbol{F}_{\boldsymbol{\theta}}^{\boldsymbol{0}}\right|}{p_0} = \frac{p_0^S}{p_0} \tag{10}$$

where $\boldsymbol{F}_{\boldsymbol{\theta}}^{\boldsymbol{0}}$ represents the set of ground-truth features selected into $\boldsymbol{F}^{\boldsymbol{S}}$ (and thus classified as *true-positives*), and $p_0^S = \left|\boldsymbol{F}_{\boldsymbol{\theta}}^{\boldsymbol{0}}\right|$ is the number of elements in it. FDR and θ are fractions related to $\boldsymbol{F}^{\boldsymbol{S}}$ and $\boldsymbol{F}^{\boldsymbol{0}}$ respectively and have values between 0 and 1. An effective feature selection method produces $\boldsymbol{F}^{\boldsymbol{S}}$ whose FDR is close to zero and θ close to one, consequently resulting in a low prediction error.

Note that the FDR and θ defined earlier were originally proposed for datasets used in sparse signal recovery applications, where $\boldsymbol{F}^{\boldsymbol{0}} \subseteq \boldsymbol{F}$, and all features in F are independent. However, these definitions are not well-suited for application to the *Type III* synthetic datasets introduced in this work (see Section 5.1). In these datasets, ground truths are not directly present in $\boldsymbol{F}$ but are instead included indirectly through correlated 'proxy' features. As a result, any selected feature—despite being correlated with a ground truth—is classified as a *false-positive* under the traditional definition, as it does not belong to $\boldsymbol{F}^{\boldsymbol{0}}$. Consequently, the traditional FDR and θ metrics invariably yield values of 1 and 0, respectively. To address this, I introduce a generalized version of the traditional FDR, called the *fractional* false discovery rate (FDR^+), defined as follows:

$$FDR^+ = \frac{\sum_{k=1}^{s} \alpha_k^{\mathrm{FP}+}}{s} \tag{8}$$

where $\alpha_k^{\mathrm{FP}+}$ takes a real value between 0 and 1, representing the 'fractional' contribution of the selected feature F_k^S to being a *false-positive*. Similar to $\boldsymbol{\alpha}^{\mathbf{FP}}$, all values of $\alpha_k^{\mathrm{FP}+}$ for $k \in [1, 2, \ldots, s]$ collectively form the status set $\boldsymbol{\alpha}^{\mathbf{FP}+}$. For datasets where $\boldsymbol{F}^{\boldsymbol{0}} \subseteq \boldsymbol{F}$ and the features in $\boldsymbol{F}$ are uncorrelated to each other, $\boldsymbol{\alpha}^{\mathbf{FP}+}$ reduces to $\boldsymbol{\alpha}^{\mathbf{FP}}$, making FDR^+ equivalent to FDR.

To evaluate $\boldsymbol{\alpha}^{\mathbf{FP+}}$, I leverage the information-theoretic concept of mutual information (MI), that quantifies the information shared between the data of two variables. The higher the value of MI between two variables, the more strongly they are correlated. Figure 5.6 illustrates this concept using a hypothetical example where a selected feature F_j^S is correlated with one ground truth F_k^0 and two irrelevant features, F_m and F_n, which do not share any information with any ground truth. In this example, the fractional contribution of F_j^S to the count of false positives, $\alpha_k^{\mathrm{FP+}}$, can be expressed as:

$$\alpha_j^{\mathrm{FP+}} = \frac{MI(F_n, F_j^S) + MI(F_m, F_j^S)}{MI(F_n, F_j^S) + MI(F_m, F_j^S) + MI(F_n, F_k^0)} \tag{9}$$

Thus, I interpret the fractional contribution $\alpha_j^{\mathrm{FP+}}$ as the proportion of information in the data of F_j^S (excluding noise) that is irrelevant for explaining Y.

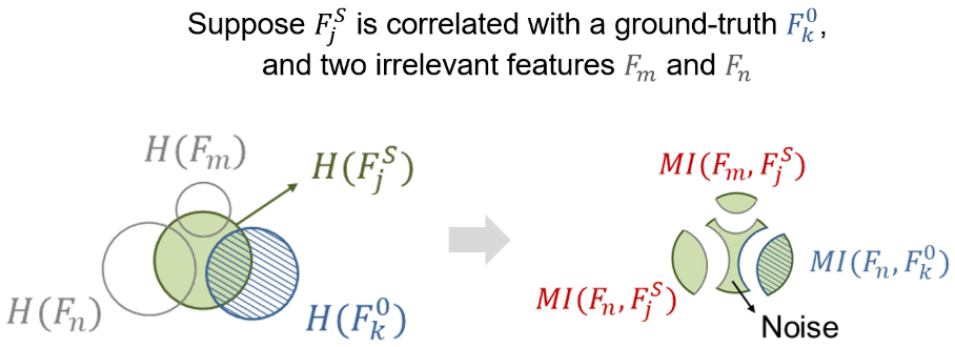

Figure 5.6. Schematic illustrating a hypothetical example to explain mutual information. For an arbitrary feature variable F_j, $H(F_j)$ represents the amount of information, also known as entropy in information theory, contained in its data. Suppose F_k is another arbitrary variable, then $MI(F_j, F_k)$ denotes the mutual information between F_j and F_k, which quantifies the information shared between their data. In a hypothetical dataset, let F_j^S be a selected feature that is uncorrelated with all features except the ground truth F_k^0 and two irrelevant features, F_m and F_n, which do not share any information with any ground truth. In the figure, the Venn diagram on the left illustrates how the information in F_j^S is shared with F_m, F_n and F_k^0. Each circle represents the entropy $H(...)$ of the corresponding variable, while the degree of overlap between the circles indicates the mutual

information $MI(...)$ shared between them. The diagram on the right explains how the information in the data of F_j^S can be decomposed based on the mutual information it shares with other variables.

Analogously, I introduce θ^+ as the fraction of ground truths from $\boldsymbol{F^0}$ that are selected into $\boldsymbol{F^S}$, either directly or via correlated proxy features, and define it as:

$$\theta^+ = \frac{\left|\boldsymbol{F^0_{\theta+}}\right|}{p_0} \tag{10}$$

where $\boldsymbol{F^0_{\theta+}}$ represents the set of ground truth features from $\boldsymbol{F^0}$ that are selected into $\boldsymbol{F^S}$ either directly or through correlated features, and $\left|\boldsymbol{F^0_{\theta+}}\right|$ denotes the number of elements in this set. By considering not only ground truths explicitly selected but also those correlated with any selected feature, θ^+ serves as a useful metric for evaluating the effectiveness of a feature selection method, even in datasets of *Type III* where ground truths are absent in $\boldsymbol{F}$. In this paper, to estimate MI, I used the *mutual_info_regression()* function from the open-source Python package *sklearn.feature_selection*. This function employs a certain non-parametric approach to estimate entropy for data of continuous feature variables.[102,103]

5.6.2 *Simulation Results: Synthetic Data with Ground-Truth Coefficients = [0.5, 0.5, 0.5, 0.5] and Gaussian-distributed Features*

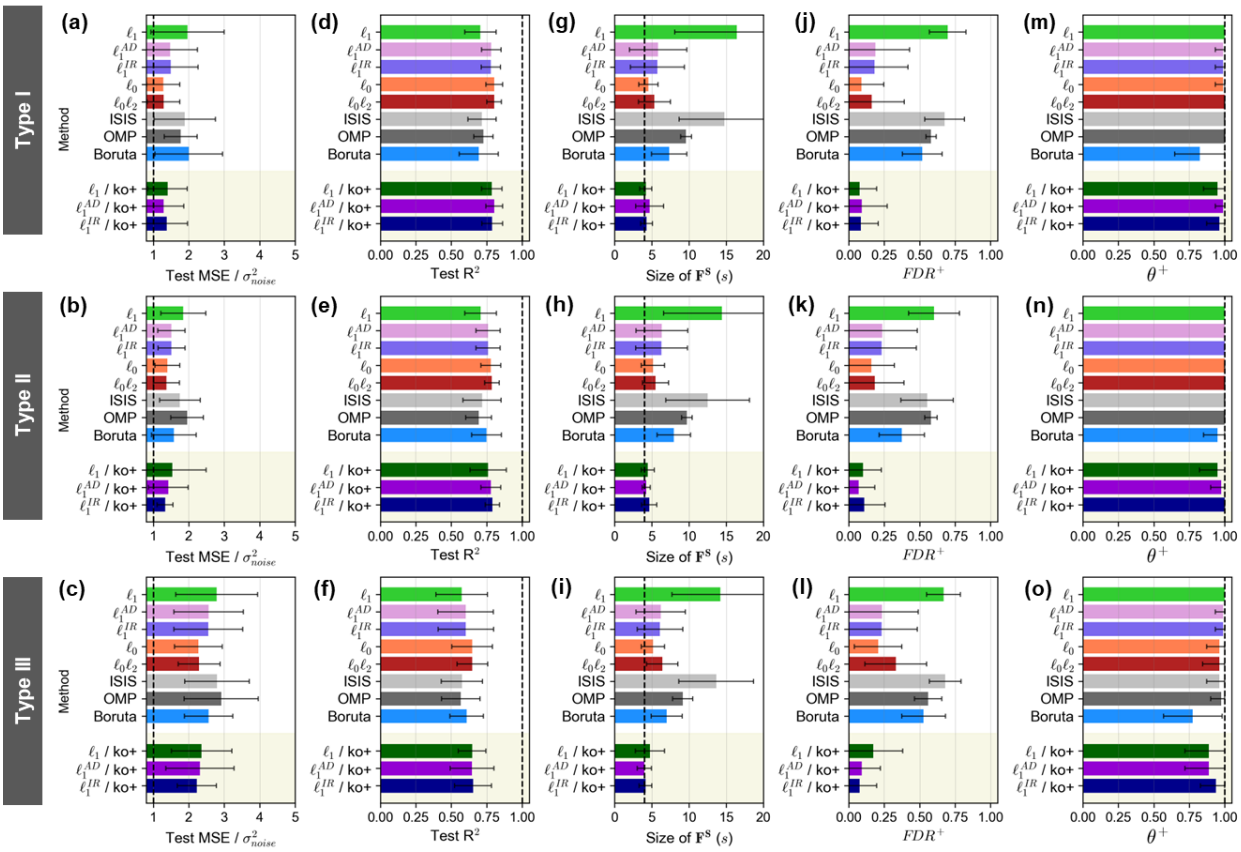

Figure 5.7. Simulation results for N = 50 and SNR = 6 for *Types I-III*, where p = 50 and p_0 = 4, and ground-truth coefficients = [0.5, 0.5, 0.5, 0.5]. (a-c) Mean squared test error (MSE) normalized with the variance of the added gaussian noise (σ^2_{noise}) **(d-f)** R^2 score between observed and predicted $\boldsymbol{y}$ values of the test data **(g-i)** Size (s) of the selected feature subset $\boldsymbol{F^S}$ **(j-l)** Fractional false discovery rate FDR^+ **(m-o)** Fractional ground-truth recovery θ^+. For each metric, the bar value represents its mean, while the error bar represents its standard deviation across the 20 artificial datasets used for the (N, SNR, dataset *Type*) combination. The dashed vertical line represents the test MSE = σ^2_{noise} line in (a-c), the R^2=1 line in (d-f), the $s = p_0$ = 4 line in (h-j), and $\theta^+ = 1$ line in (k-m). Methods with the suffix 'ko+' use the knockoffs scheme (highlighted by a grey shade), as outlined in Section 4.3.2 for sparsity level control. The rest use leave-one-out cross-validation to determine the sparsity level. All features follow Gaussian distribution.

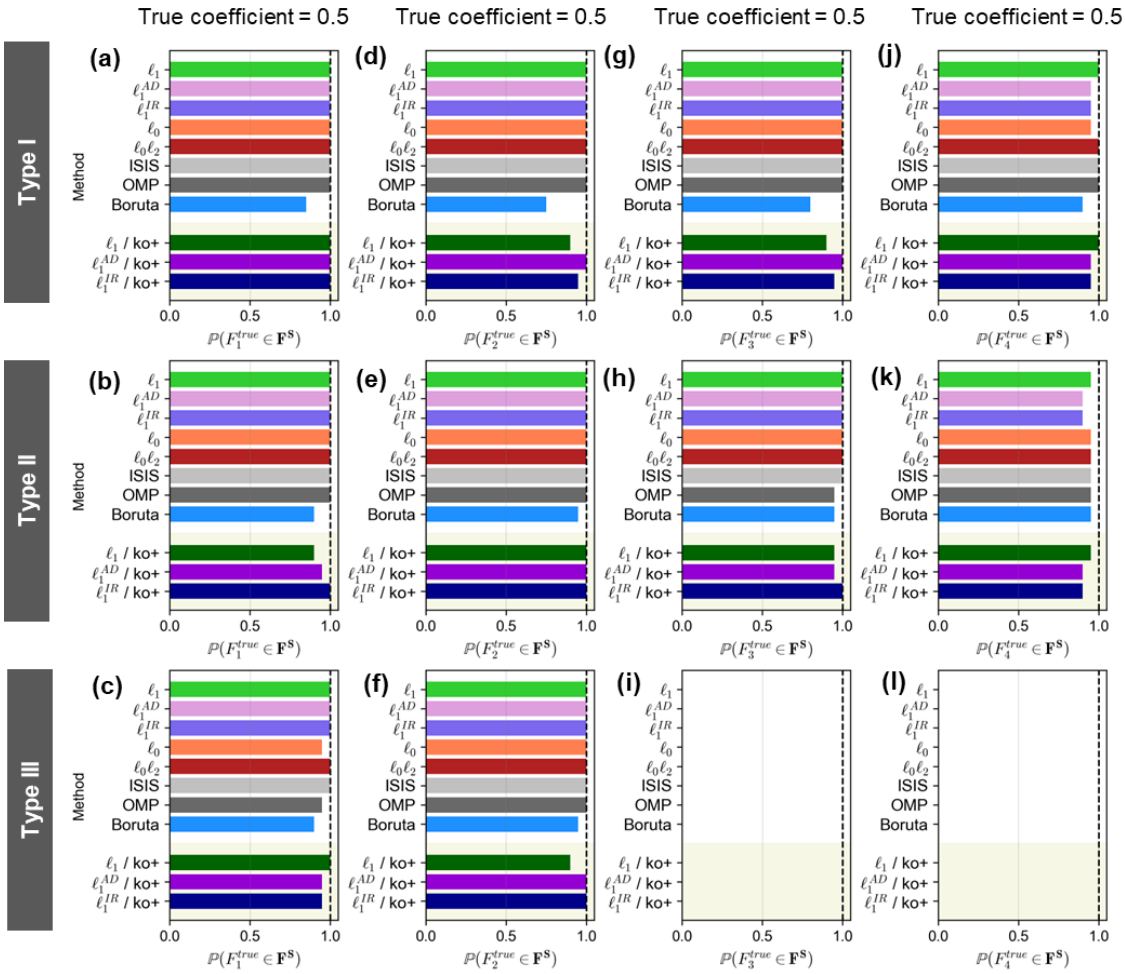

Figure 5.8. Simulation results for $N = 50$ and SNR = 6 for *Types I-III*, where $p = 50$ and $p_0 = 4$, and ground-truth coefficients = [0.5, 0.5, 0.5, 0.5]. Selection probabilities of ground-truth features whose coefficients in the generative mechanism are set to **(a-c)** 0.9 **(d-f)** 0.1 **(g-i)** 0.9 **(j-l)** 0.1. For each (N, SNR, dataset *Type*) combination, the selection probability $\mathbb{P}(F_k^{true} \in \boldsymbol{F^S})$ of a given ground-truth feature F_k^{true} (where $1 \leq k \leq p_0$) is estimated by the fraction of the number of times the feature is selected among the 20 datasets generated with the combination. The dashed vertical lines represent the probability of one. In *Type III*, where F_3^{true} and F_4^{true} are absent from $\boldsymbol{F}$, their selection probabilities are invariably zero. Methods with the suffix 'ko+' use the knockoffs scheme (highlighted by a gray shade), as outlined in Section 4.3.2 for sparsity level control. The rest use leave-one-out cross-validation to determine the sparsity level.

5.6.3 *Simulation Results: Synthetic Data with Ground-Truth Coefficients = [0.9, 0.1, 0.9, 0.1] and Uniformly-Distributed Features*

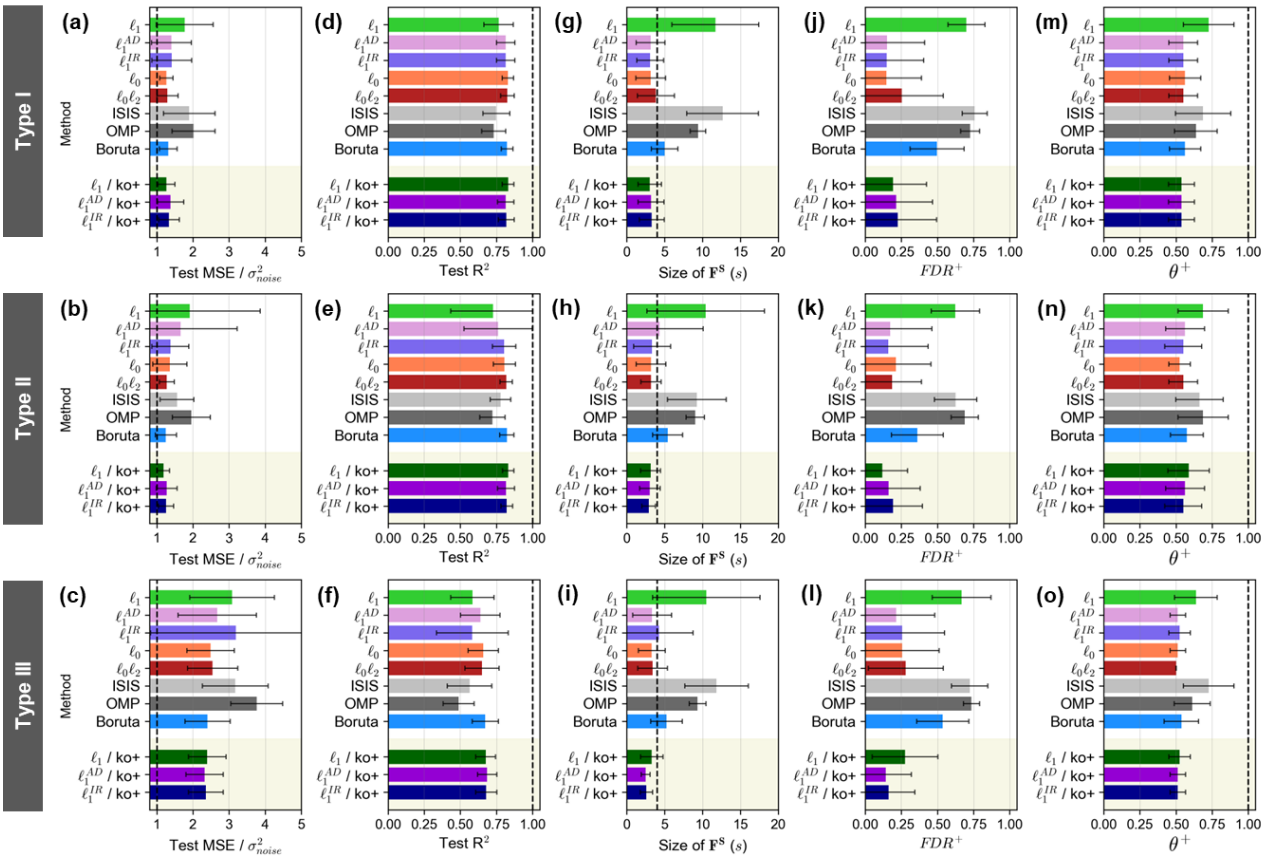

Figure 5.9. Simulation results for N = 50 and SNR = 6 for *Types I-III* (with uniformly distributed feature data columns), where p = 50 and p_0 = 4 , and ground-truth coefficients = [0.9, 0.1, 0.9, 0.1]. (a-c) Mean squared test error (MSE) normalized with the variance of the added gaussian noise (σ^2_{noise}) **(d-f)** R^2 score between observed and predicted $\boldsymbol{y}$ values of the test data **(g-i)** Size (s) of the selected feature subset $\boldsymbol{F^S}$ **(j-l)** Fractional false discovery rate FDR^+ **(m-o)** Fractional ground-truth recovery θ^+. For each metric, the bar value represents its mean, while the error bar represents its standard deviation across the 20 artificial datasets used for the (N, SNR, dataset *Type*) combination. The dashed vertical line represents the test MSE = σ^2_{noise} line in (a-c), the R^2=1 line in (d-f), the $s = p_0$ = 4 line in (h-j), and $\theta^+ = 1$ line in (k-m). Methods with the suffix 'ko+' use the knockoffs scheme (highlighted by a gray shade), as outlined in Section 4.3.2 for sparsity level control. The rest use leave-one-out cross-validation to determine the sparsity level. All features follow a uniform distribution.

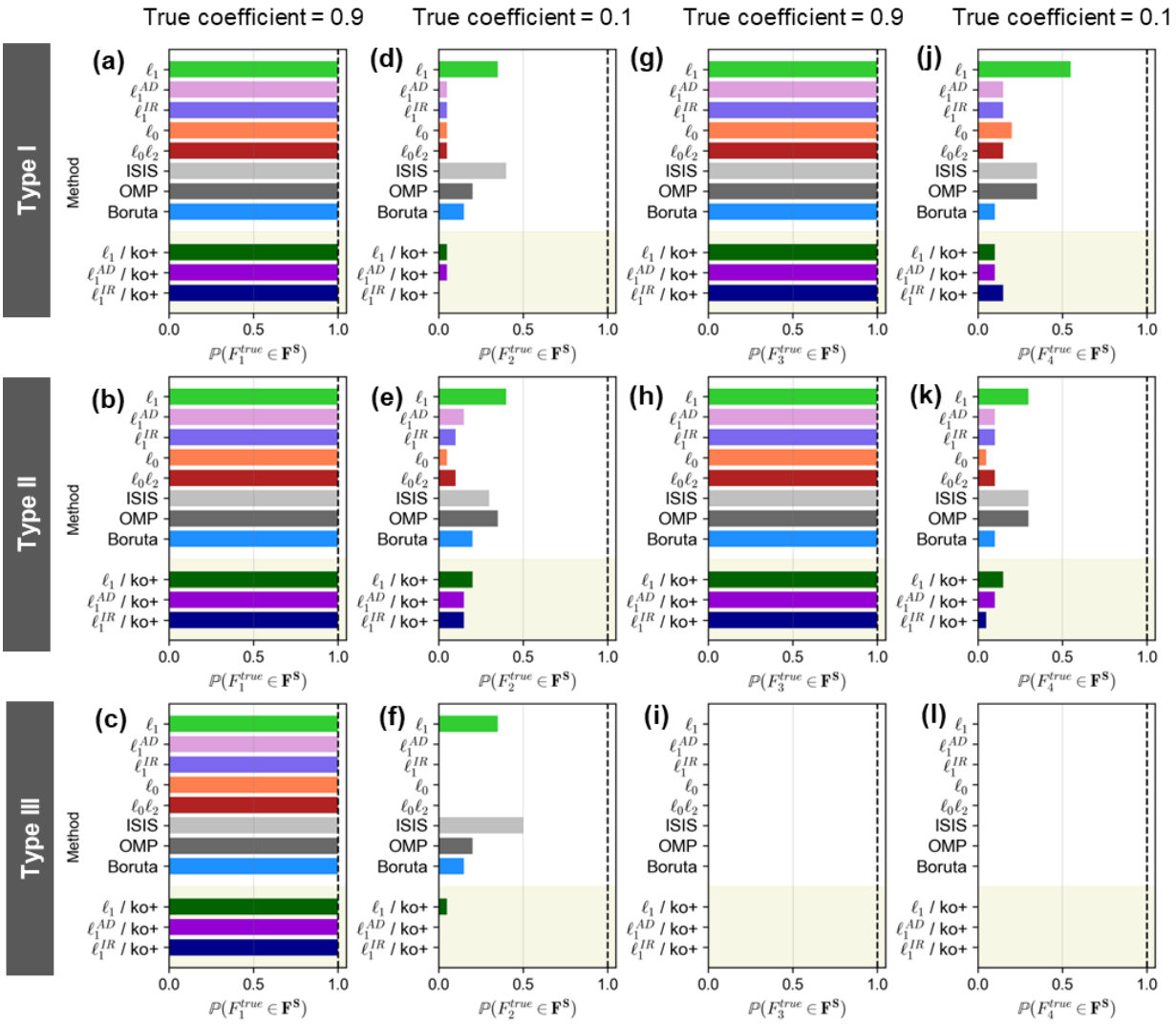

Figure 5.10. Simulation results for N = 50 and SNR = 6 for *Types I-III* (with uniformly distributed feature data columns), where p = 50 and p_0 = 4, and ground-truth coefficients = [0.9, 0.1, 0.9, 0.1]. Selection probabilities of ground-truth features whose coefficients in the generative mechanism are set to **(a-c)** 0.9 **(d-f)** 0.1 **(g-i)** 0.9 **(j-l)** 0.1. For each (N, SNR, dataset *Type*) combination, the selection probability $\mathbb{P}(F_k^{true} \in \boldsymbol{F^S})$ of a given ground-truth feature F_k^{true} (where $1 \leq k \leq p_0$) is estimated by the fraction of the number of times the feature is selected among the 20 datasets generated with the combination. The dashed vertical lines represent the probability of one. In *Type III*, where F_3^{true} and F_4^{true} are absent from $\boldsymbol{F}$, their selection probabilities are invariably zero. Methods with the suffix 'ko+' use the knockoffs scheme, as outlined in Section 4.3.2 for sparsity level control. The rest use leave-one-out cross-validation to determine the sparsity level.

5.6.4 *Supplementary Figures*

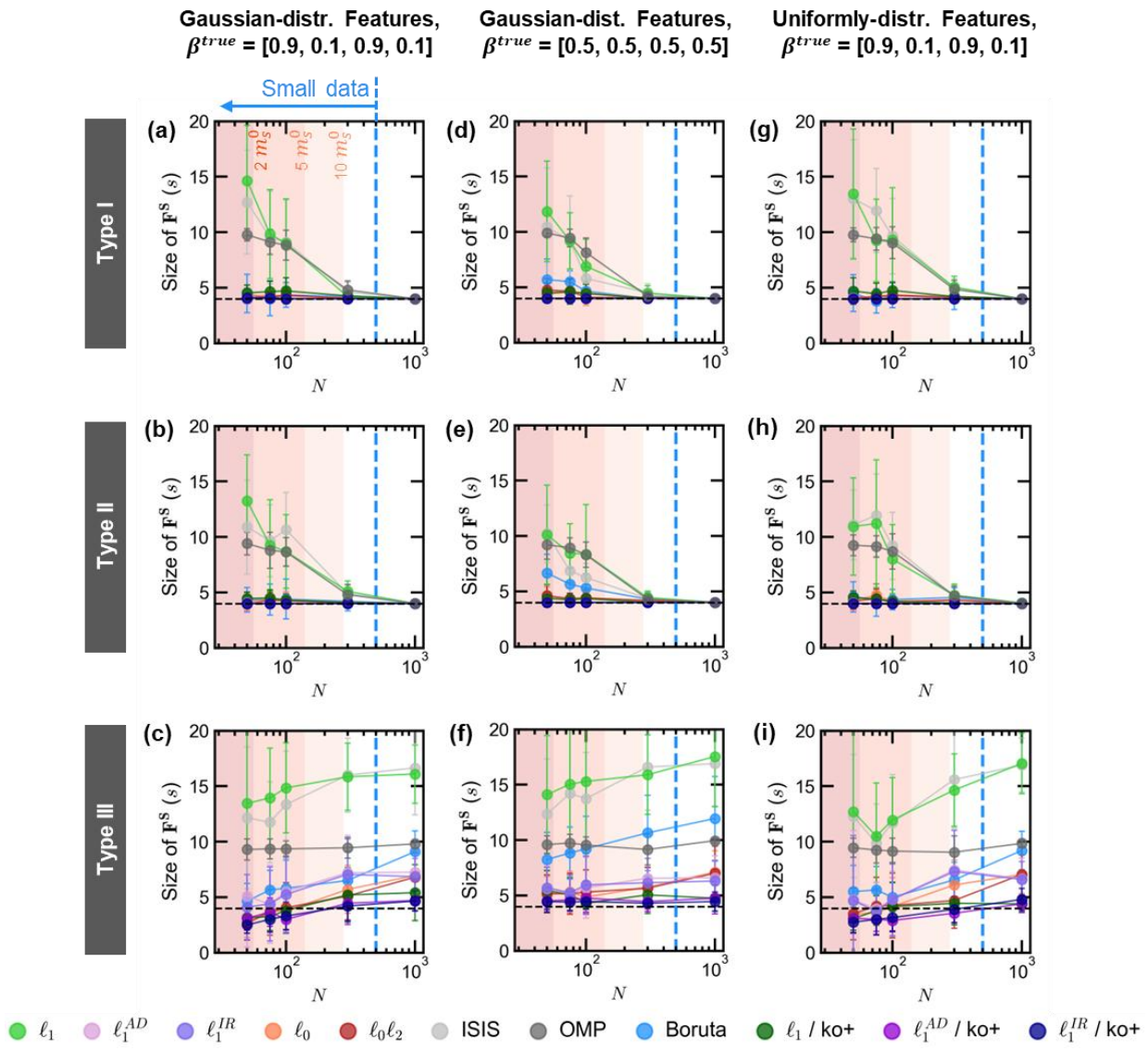

Figure 5.11. Selected feature subset sizes s obtained for N = [50, 75, 100, 300, 1000], SNR = 512, p = 50, and p_0 = 4 for *Types I-III* in three simulation studies. (a-c) Gaussian-distributed feature data columns with ground-truth coefficients ($\boldsymbol{\beta}^{\boldsymbol{true}}$) = [0.9, 0.1, 0.9, 0.1], as presented in Section 5.4. **(d-f)** Gaussian-distributed feature data columns with ground-truth coefficients = [0.5, 0.5, 0.5, 0.5], as presented in Section 5.6.2. **(g-i)** Uniformly-distributed feature data columns with ground-truth coefficients = [0.9, 0.1, 0.9, 0.1], as presented in Section 5.6.3. At a given (N, SNR, dataset *Type*) combination, the marker point represents the mean subset size, and the error bar represents its standard deviation across the 20 artificial datasets used with the combination. The dashed horizontal line represents the $s = p_0$ = 4 line. Methods with the suffix 'ko+' use the knockoffs scheme, as outlined in Section 4.3.2 for sparsity level control. The rest use leave-one-out cross-validation to determine the sparsity level. The legend describing the markers

is shown below the plots. The vertical blue dashed line represents $N = 10p = 500$ below which the small data regime begins. For a sparse model performing exact sparse recovery such that $s = p_0$, the degrees of freedom m_s^0 is given by $p_0 \log_2(p - p_0)/(1 - \rho) \approx 28$, with ρ set to 0.2 (see Section 3.2). The boundaries of the red shaded regions represent values of N corresponding to 2 m_s^0 (≈56), 5 m_s^0 (≈140) and 10 m_s^0 (≈280), as indicated by the labels.

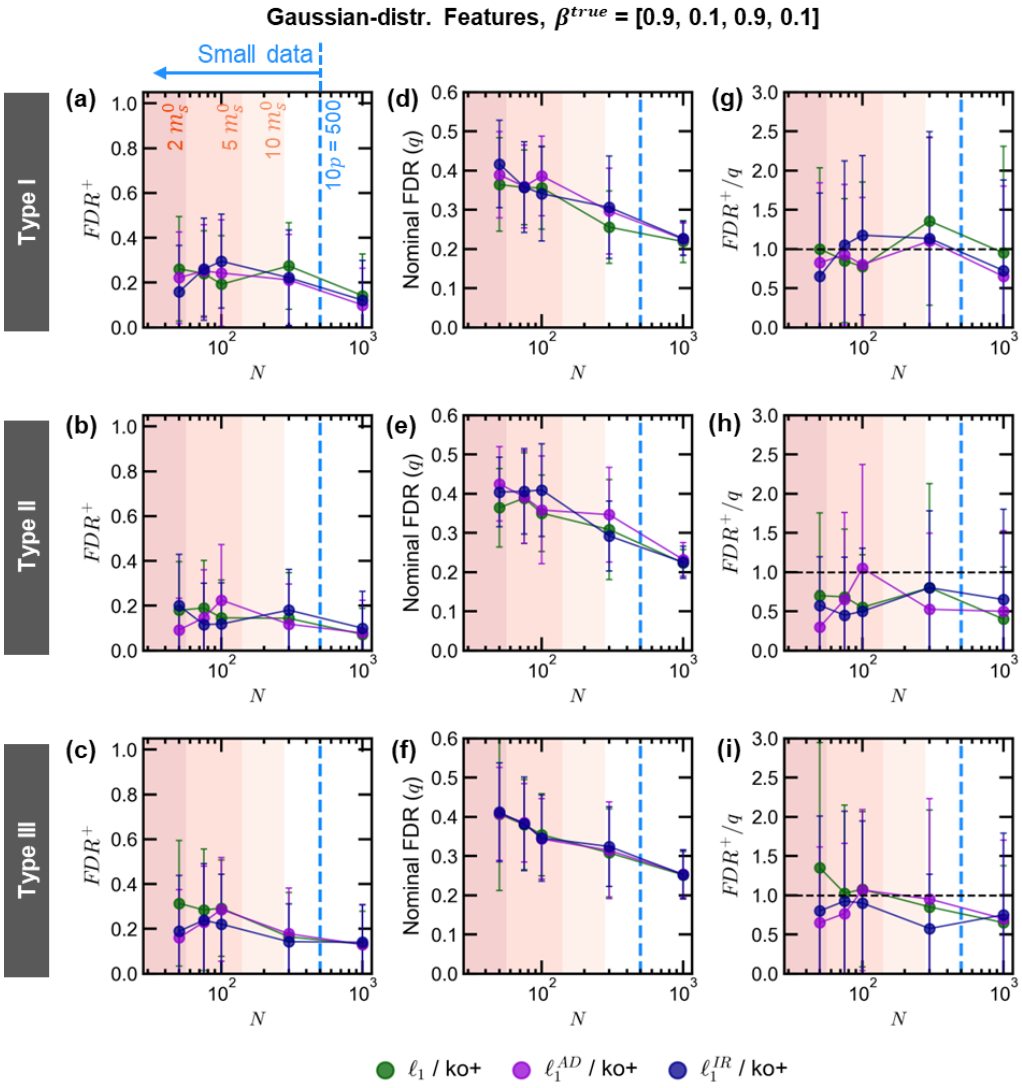

Figure 5.12. Comparison of the observed FDR^+ and predicted nominal false-discovery rates q in knockoffs-based methods for *Types I* to *III* with p = 50, p_0 = 4, SNR = 6, Gaussian-distributed features and ground-truth coefficients = [0.9, 0.1, 0.9, 0.1]. (a-c) Observed fractional false-discovery rate (FDR^+) **(d-f)** Nominal false-discovery rate (q) as predicted by the knockoffs scheme. **(g-i)** Ratio of FDR^+ to q. For each metric, the marker represents its mean, while the error bar represents its standard deviation across the 20 artificial datasets used for the (N, SNR, dataset *Type*) combination. The dashed horizontal line in (e-f) represents the $FDR^+/q = 1$ line. The legend describing the markers is shown below the plots. The vertical blue dashed line

represents $N = 10p = 500$ below which the small data regime begins. For a sparse model performing exact sparse recovery such that $s = p_0$, the degrees of freedom m_s^0 is given by $p_0 \log_2(p - p_0)/(1 - \rho) \approx 28$, with ρ set to 0.2 (see Section 3.2). The boundaries of the red shaded regions represent values of N corresponding to 2 m_s^0 (≈56), 5 m_s^0 (≈140) and 10 m_s^0 (≈280), as indicated by the labels.

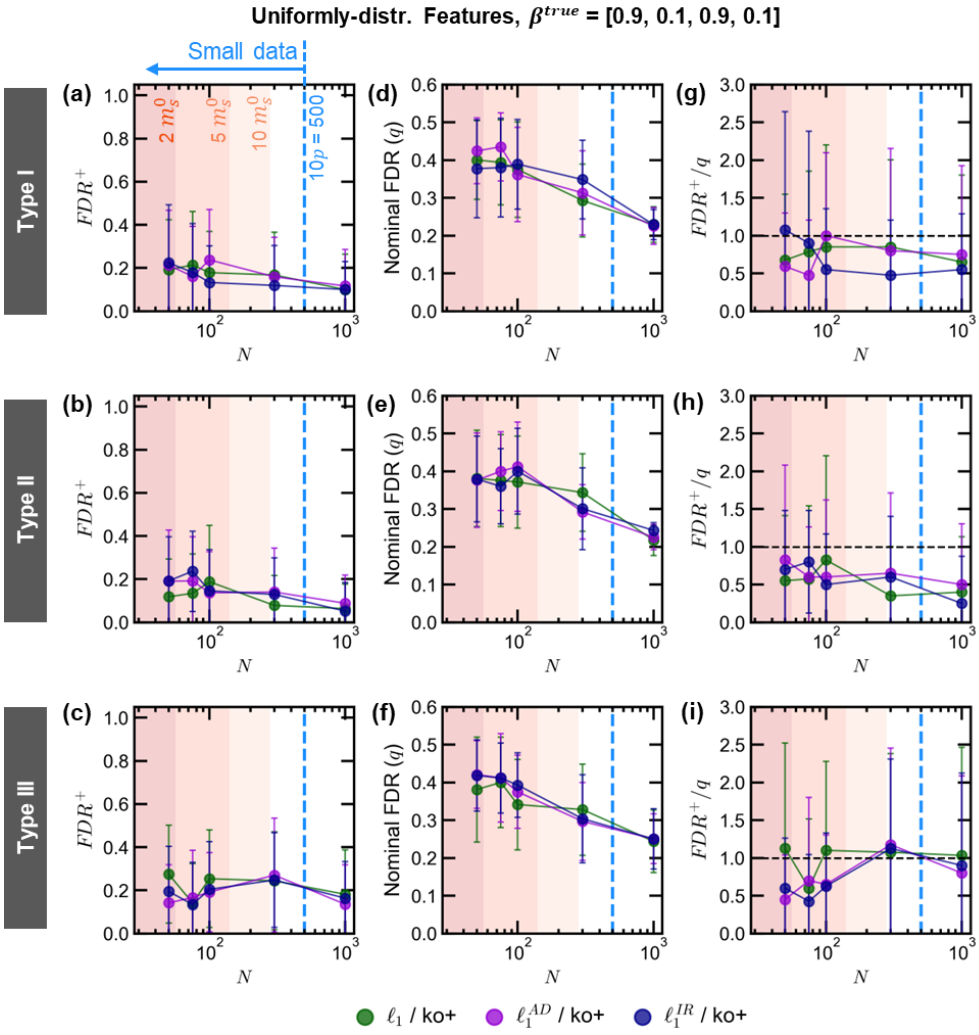

Figure 5.13. Comparison of the observed FDR^+ and predicted nominal false-discovery rates q in knockoffs-based methods for *Types I* to *III* with p = 50, p_0 = 4, SNR = 6, uniformly-distributed features and ground-truth coefficients = [0.9, 0.1, 0.9, 0.1]. (a-c) Observed fractional false-discovery rate (FDR^+) **(d-f)** Nominal false-discovery rate (q) as predicted by the knockoffs scheme. **(g-i)** Ratio of FDR^+ to q. For each metric, the marker represents its mean, while the error bar represents its standard deviation across the 20 artificial datasets used for the (N, SNR, dataset *Type*) combination. The dashed horizontal line in (e-f) represents the $FDR^+/q = 1$ line. The legend describing the markers is shown below the plots. The vertical blue dashed line

represents $N = 10p$ = 500 below which the small data regime begins. For a sparse model performing exact sparse recovery such that $s = p_0$, the degrees of freedom m_s^0 is given by $p_0 \log_2(p - p_0)/(1 - \rho) \approx 28$, with ρ set to 0.2 (see Section 3.2). The boundaries of the red shaded regions represent values of N corresponding to 2 m_s^0 (≈56), 5 m_s^0 (≈140) and 10 m_s^0 (≈280), as indicated by the labels.

6 PREDICTION OF t_{80} LIFETIMES IN METHYLAMMONIUM LEAD IODIDE SOLAR CELLS

Despite the recent popularity of hybrid perovskite materials as promising candidates for photovoltaic applications, their instability under environmental stress has hindered their commercialization.[5,24] Methylammonium lead iodide ($MAPbI_3$), often abbreviated as MAPI, is the simplest perovskite composition, whose degradation behavior under exposure to environmental stressors, including elevated temperatures, light, oxygen, and moisture, has been widely studied.[13,14,17,23,104] $MAPbI_3$ is known to decompose at elevated temperatures due to the escape of volatile species,[104] and to react with moisture and oxygen in the presence of light, forming photo-inactive compounds.[14] When incorporated into a solar cell alongside other non-perovskite layers, additional degradation pathways emerge, including degradation of non-perovskite ETL or HTL layers or their interfaces with the perovskite,[105,106] corrosion of metallic contact interfaces,[19] ionic migration under electric fields,[20,21] and reactions with trapped volatile species due to encapsulation.[22,23] These factors collectively contribute to the degradation of the device leading to a gradual decline in power conversion efficiency (PCE) over time.

The resilience of a solar cell to environmental stress is commonly quantified by its t_{80} lifetime, defined as the time it takes for the cell's power conversion efficiency (PCE) to decline to 80% of its initial value.[21,25,26] As reaching t_{80} through testing can be time-consuming, often requiring weeks or even months of experiment time,[25,27] it would be useful to predict this lifetime based on the device's initial quality, early-time dynamics of its performance, and the environmental stress conditions, without conducting full-duration experiments. Given the complexity of the

underlying degradation mechanisms in a $MAPbI_3$ solar cell, it is preferable to use a physics-informed statistical model to predict t_{80}, rather than attempting to derive a fully mechanistic model. However, statistical models still require a representative set of experiments for training, which is particularly challenging in this context due to the slow-pace of experimentation. As a result, this leads to a *small data* scenario for modeling purposes.

To address this, Dunlap-Shohl et al.[21] (which includes contributions from me) previously developed a preliminary ML model using an in-house dataset containing t_{80} lifetimes of only N = 45 $MAPbI_3$ solar cells stressed in air under 1 sun illumination, and varying cell temperatures and moisture levels. The dataset includes numerous features extracted from time-series measurements of current-voltage (J-V) characteristics, wide-field photoluminescence (PL), and dark-field (DF) images.[21] In this chapter, using this small dataset as a case study, I apply the various modeling strategies discussed in Chapter 4 to more comprehensively evaluate the development of ML models for t_{80} prediction in $MAPbI_3$ solar cells. This includes outlining the various stages of the modeling workflow—feature construction, feature selection and model fitting, and uncertainty quantification—as introduced in Chapter 4. I also present results that incorporate insights obtained from simulations on synthetic data (Chapter 5) and provide a physical interpretation of the selected features. Overall, this dataset serves as a valuable real-world example for applying the small data modeling techniques discussed throughout this work.

Depending on the ambient conditions, the t_{80} lifetime of a perovskite solar cell in operation can vary by several orders of magnitude—from several months under inert, encapsulated conditions[27] to just a few hours under accelerated stress in oxygen- or moisture- rich environments at elevated temperatures[21]. The thermally-activated nature of the underlying physicochemical

processes (their dependence of a factor of e^{E_A/k_bT}),[31] causes much of this variation. To account for this non-linearity, I assign the base-10 logarithm of the lifetime ($\log_{10} t_{80}$), with t_{80} expressed in minutes, as the target variable Y, thereby linearizing its dependence on the input features.[107] This serves as an example of how relatively simple domain knowledge may be used to simplify the machine learning task.

6.1 Feature Construction

6.1.1 A priori Known Features

As discussed in Section 4.1, the first step in the modeling workflow is feature construction. Temperature and humidity level (expressed in partial pressure units)—both controlled variables known to influence the t_{80} lifetime—are the first features added to the feature menu $\boldsymbol{F}$. However, the exact relationship between these features and the t_{80} lifetime is not fully known. Some reports have suggested estimating the early-time degradation rates in PCE using an Arrhenius-type relationship[a] with temperature and using the inverse of these rates as proxies for effective t_{80} lifetimes. While this approach may be viable under simple stress conditions, it becomes impractical when the underlying degradation mechanisms are more complex, requiring the development of more sophisticated, physics-informed features.[32]

---

[a] It describes a non-linear dependence on temperature T as follows: $e^{-E_A/T}$, where E_A represents an effective positive activation energy (in temperature units).

Using pristine films, Siegler et al.[14] (our group) derived a non-linear function of temperature, moisture, illumination intensity and oxygen concentration that approximates the decomposition rate of the $MAPbI_3$ perovskite absorber layer when there are no mass transport limitations on the supply of reactants or the removal of volatile products. To incorporate this physics and chemistry domain knowledge, I included this rate function (r_{MAPI}) in the feature menu $\boldsymbol{F}$, defined as follows:

$$\begin{aligned} r_{MAPI} \;=\; & k_{0,WPO} \exp\left(-\frac{E_{\mathrm{W,DPO}}^{\mathrm{eff}}}{k_B T}\right) \frac{P_{O_2} P_{H_2O} I_{in}^{0.7}}{\left(1 + K_{2W} P_{O_2}\left(1 + K_{3W} I_{in}^{0.7}\right)\right)^2} \\ & + \; k_{0,DPO} \exp\left(-\frac{E_{\mathrm{A,DPO}}^{\mathrm{eff}}}{k_B T}\right) \frac{P_{O_2} I_{in}^{0.7}}{1 + K_{2D} P_{O_2}\left(1 + K_{3D} I_{in}^{0.7}\right)} \\ & + \; k_{hum} \exp\left(-\frac{E_{\mathrm{A,hum}}^{\mathrm{eff}}}{k_B T}\right) + \; k_{therm} \exp\left(-\frac{E_{\mathrm{A,therm}}^{\mathrm{eff}}}{k_B T}\right) \end{aligned} \tag{9}$$

Here, T, P_{H_2O}, P_{O_2} and I_{in} represent the temperature, partial pressures of water and oxygen, and the incident illumination intensity, respectively. The remaining variables correspond to various constants, such as the effective rate constants and activation energies associated with the different steps in the $MAPbI_3$ degradation mechanism (see Siegler et. al.[14] for more details). Eq. (9) illustrates how a highly complex, non-linear dependence of early-time degradation dynamics on environmental conditions can be captured through a physics-informed feature built using domain expertise.

6.1.2 Experimental Features

Features such as temperature, humidity, illumination intensity, or the idealized chemical degradation rate (r_{MAPI}), whose values are known or can be calculated prior to the start of the

experiment, capture some of the apparent factors that may potentially affect the t_{80} lifetimes. However, these features cannot account for variations in the t_{80} lifetime arising from elusive underlying phenomena in multi-layered devices, such as the defects that arise from small variations in processing conditions and crystallization kinetics at interfaces,[16] or degradation of the non-perovskite layers or their interfaces with the perovskite,[105,106] or defect migration driven by electric fields.[20,21] To incorporate these sample-dependent effects into t_{80} prediction, additional complementary characterization measurements that can be performed quickly *in situ* alongside PCE measurements are necessary.

Photoluminescence quantum yield (PLQY) is one such measurement that reflects the fraction of photo-excited charge carriers undergoing radiative recombination, as opposed to following destructive non-radiative pathways through deep electronic defects in the bulk or at the interfaces of the perovskite photo-absorber layer.[108,109] This measurement can be easily integrated into the characterization sequence by illuminating the device under an optical microscope during operation.[21] By tracking the changes in PLQY with time, one can record the accumulation of defects in the bulk or at the interfaces of the perovskite layer due to operational degradation, which correlates with a decline in PCE. Wide-field PL imaging enables spatial mapping of PLQY, while capturing several of these images within a short time interval (5 seconds) allows for analysis of its temporal dynamics at each spatial location. Additionally, dark-field (DF) imaging can also be incorporated into these time-series microscopy measurements.[21,107] DF records the scattered light resulting from spatial variations in refractive index at grain or phase boundaries, or from gradients in film morphology.[21] As non-perovskite phases form and grow as degradation products, DF imaging provides spatially resolved insights into this process. Finally, additional current-voltage

(J-V) measurements, including stabilized measurements under short-circuit and open-circuit conditions, and J-V sweeps, can be integrated into the characterization sequence to more effectively capture underlying physicochemical and electronic phenomena that may contribute to the decline in PCE. For instance, the short-circuit current serves as a proxy to capturing photo-absorption and carrier mobility losses, while the open-circuit voltage is linked to the losses in photo-excited charge carrier lifetimes due to bulk and interfacial degradation in the perovskite layer.[21,28] Figure 6.1 illustrates the experimental setup used in this work.

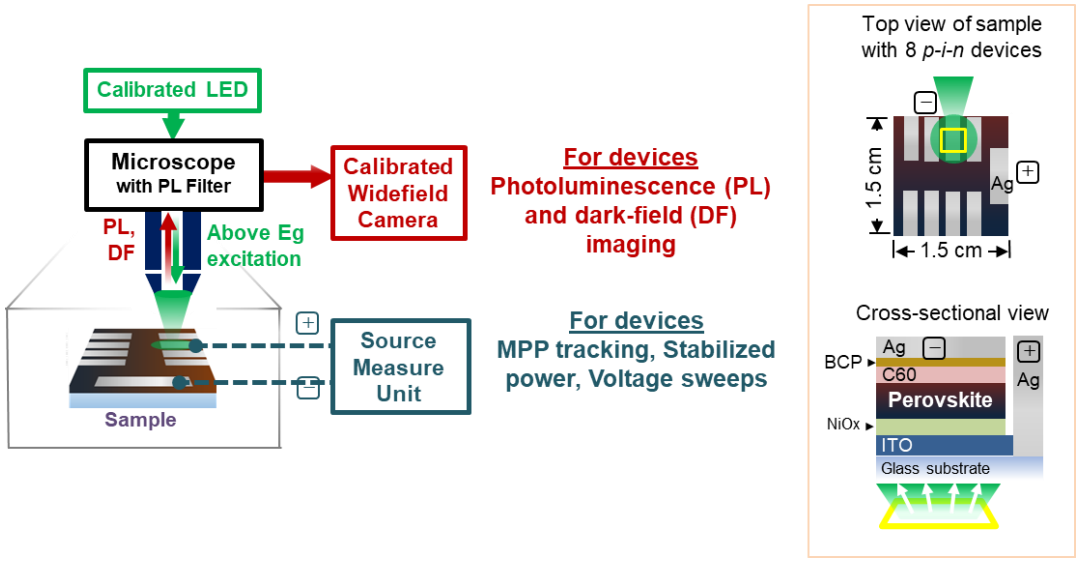

Figure 6.1. Experimental setup used for the degradation studies of perovskite solar cells in this work. The sample is placed in an environmental chamber and illuminated with 1 sun above-bandgap equivalent photon-flux under an upright optical microscope for both stressing and probing. During the stressing process, *in situ* wide-field PL, DF and J-V measurements are taken periodically. Each sample (1.5 cm x 1.5 cm) contains 8 *p-i-n* devices (each with an area of ~ 0.067 cm^2), only one of which is stressed and characterized in each experiment. The inset on the right shows the top view and the cross-sectional view of the illuminated device within each sample. See Section 6.6.2 in the Supporting Information for more details.

To leverage the sample-specific information obtained from these measurements, I derive features whose values are determined based on the initial dynamics of their time-series data. For

example, from time series data consisting of scalar measurements, such as the current and voltage across the cell, recorded at regular intervals, I computed the early-time slope and curvature, and used them as features. For spatially-resolved measurements, such as PL and DF images acquired over time, I applied statistical operations (e.g., mean or standard deviation across pixels) to reduce each two-dimensional image to a scalar value at each time point, yielding a corresponding scalar time series. The early-time slope and curvature of this time-series were then calculated and used as additional features. Using the dataset from Dunlap-Shohl et al.[21] and the methods described above, I constructed a comprehensive feature menu consisting of p = 34 features. This was achieved by additionally eliminating features that exhibited very high correlations (i.e., absolute Pearson correlation coefficient greater than 0.9) with other features and retaining those believed to influence Y based on domain knowledge.

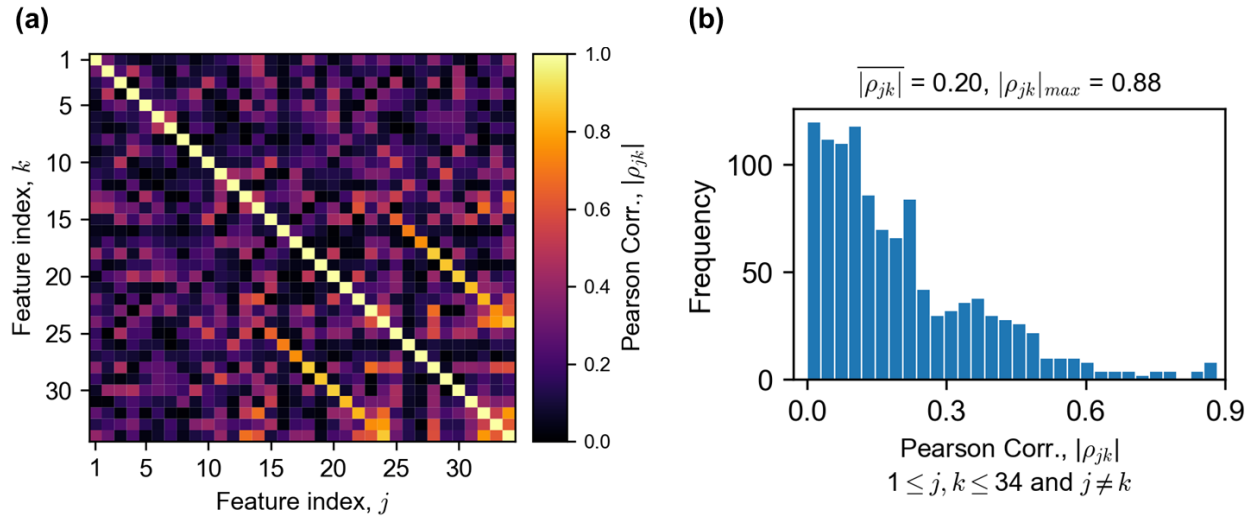

Figure 6.2. (a) Absolute Pearson correlation structure of the $MAPbI_3$ perovskite solar cell t_{80} degradation lifetime dataset. A feature set of p = 34 is obtained after removing highly-correlated features that exceeded an absolute Pearson correlation value of 0.9. **(b) Distribution of absolute Pearson correlation coefficients.** $|\rho_{jk}|$ represents the mean value and $|\rho_{jk}|_{max}$ represents the maximum value.

6.2 Model Setup and Testing

Given the complexity and the time-intensive nature of each experiment, this dataset comprises only $N = 45$ data-points, which is below the $10p$ (=340) heuristic value, placing it within the *small* data regime. To ensure that the dataset doesn't fall into the *too small* regime, N must exceed $m_s^0 \approx p_0 \log_2(p - p_0)/(1 - \rho)$, which represents the effective degrees of freedom required by a sparse model for exact sparse recovery of all p_0 ground-truth variables (see Section 3.2). For ρ, the mean value of the absolute Pearson correlation coefficients (= 0.2) is used, as it serves as a representative value for the majority of pairwise correlations among the features (see Figure 6.2). Since p_0 is unknown, I consider a reasonable range of values between 1 and 7. Within this range, the estimated m_s^0 lies between 7 and 42, satisfying the condition $N > m_s^0$. Thus, while the current dataset is not *too small* for modeling, it still poses a risk of overfitting for certain methods, as $N < 10m_s^0$.

I applied the feature selection schemes (as outlined in Section 4.2 and 4.3) and fit an OLS linear model using the selected features. A leave-one-out (LOO) testing scheme (as discussed in Section 4.4) was used to evaluate the median error by predicting $Y = \log_{10}(t_{80})$ (with t_{80} expressed in minutes) for each left-out trial using the model trained on the remaining trials. Additionally, the R^2 between the predicted t_{80} values at the left-out test trials and their respective observed values, denoted as R^2_{test}, was calculated for each method. A high R^2_{test} value indicates good model generalizability. Finally, Jackknife+ and Jackknife-minmax methods were employed

to estimate the confidence intervals for predictions (as discussed in Section 4.5) by the best-performing feature selection method.

6.3 t_{80} Prediction Results

6.3.1 Observations

Figure 6.3a shows the feature subsets selected by each selection method, along with their corresponding median prediction error and R^2_{test} values. Although ℓ_1/ko+ achieves the highest R^2 value, ℓ_1^{IR}/ko+ exhibits the lowest median error while maintaining a similar R^2_{test} (of 0.71) and sparsity level s as the former. Figure 6.3b displays how well the observed t_{80} values, left out in each LOO iteration, align with their predictions by the ℓ_1^{IR}/ko+ method. Figure 6.3c depicts the selection of features in individual LOO test iterations, while Figure 6.3d displays the nonzero mean values of the coefficients and their standard deviations evaluated across the N = 45 LOO iterations. Among the features with nonzero mean coefficients, four features—T (K), r_{MAPI}, DF_{med} (t=0), d^2DF_{std}/dt^2 and dFF/dt—display smaller standard deviations (in relative to their means) due to being frequently selected by ℓ_1^{IR}/ko+ across the LOO iterations (Figure 6.3c). T (K) and r_{MAPI}, in particular, are selected by all feature selection methods (Figure 6.3a), indicating their significance in predicting the t_{80} value.

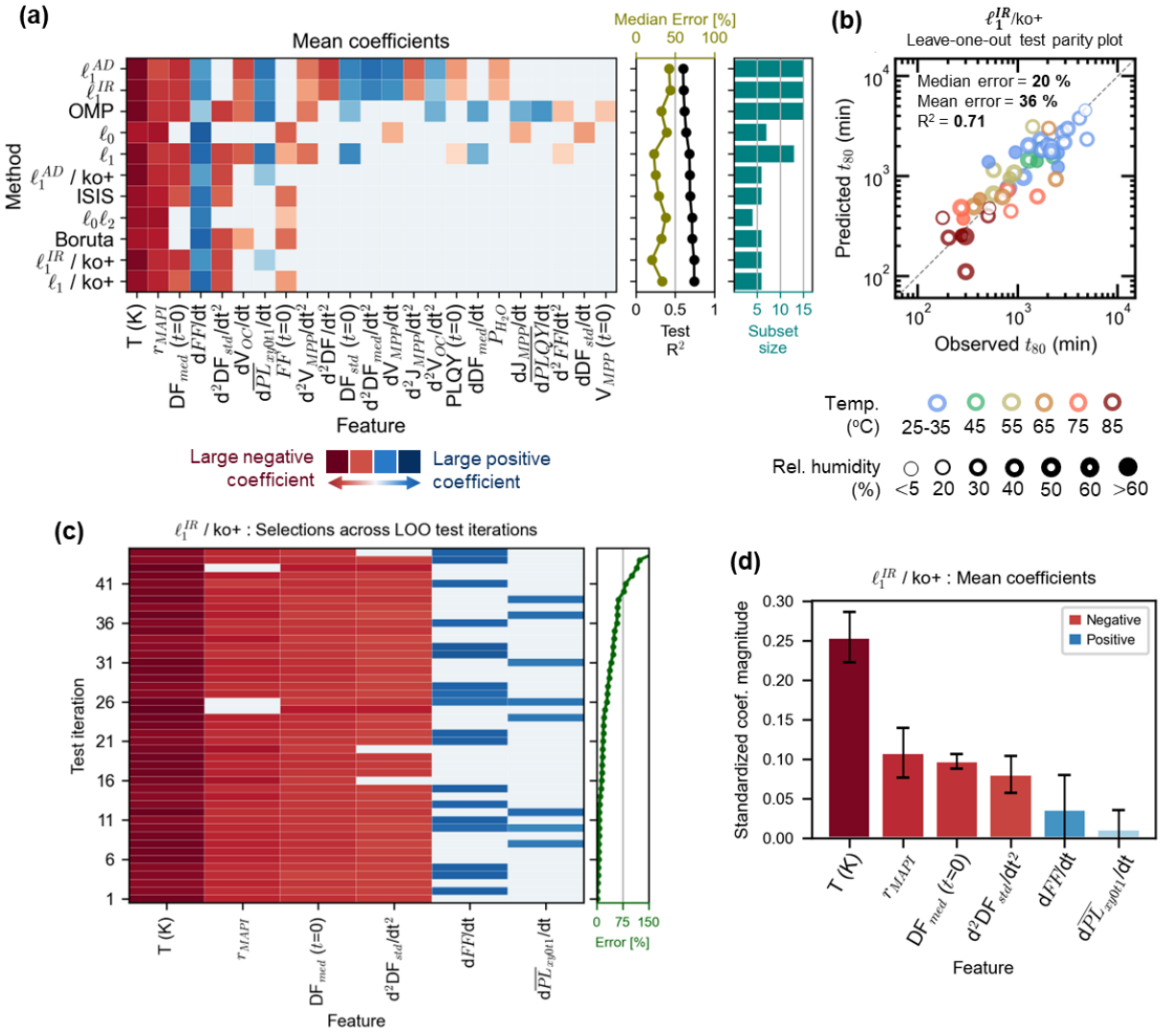

Figure 6.3. t_{80} degradation lifetime prediction in $MAPbI_3$ perovskite solar cells. (a) Heatmap displaying mean coefficients of features selected from a list of $p = 34$ by each selection method. The intensity of colors represents the magnitude of the mean feature coefficients, evaluated over $N = 45$ leave-one-out (LOO) train-test splits. Red and blue hues indicate negative and positive coefficients, respectively. The green graph on the right shows the median percentage test errors (i.e., |observed t_{80} − predicted t_{80}|/ observed t_{80} %), evaluated over the LOO train-test splits. The black graph shows the test R^2 value between the observed and the predicted t_{80} values of the LOO test points. The bar chart shows the overall sparsity of each method, evaluated based on the number of non-zero mean coefficients, obtained from the LOO models. **(b)** Parity plot comparing the observed and the predicted t_{80} values at the LOO test points for the ℓ_1^{IR}/ko+ method, which has the best balance of parsimony, median test error and R^2 value as shown in (a). The dashed line in (b) represents the $x = y$ parity line. The legend for the markers is provided below the parity plot. **(c)** Heatmap displaying coefficients selected by the ℓ_1^{IR}/ko+ method across the LOO test-train splits. The red and blue hues indicate negative and positive coefficients respectively, while the

intensity of the color indicates the magnitude. The green graph on the right shows the percentage test errors (i.e., |observed t_{80} − predicted t_{80}|/ observed t_{80} %) evaluated over each LOO test point. **(d)** Mean coefficients of the features selected by the ℓ_1^{IR}/ko+ method, evaluated from the N = 45 LOO fits. The error bars indicate the standard deviations along the LOO test-train splits.

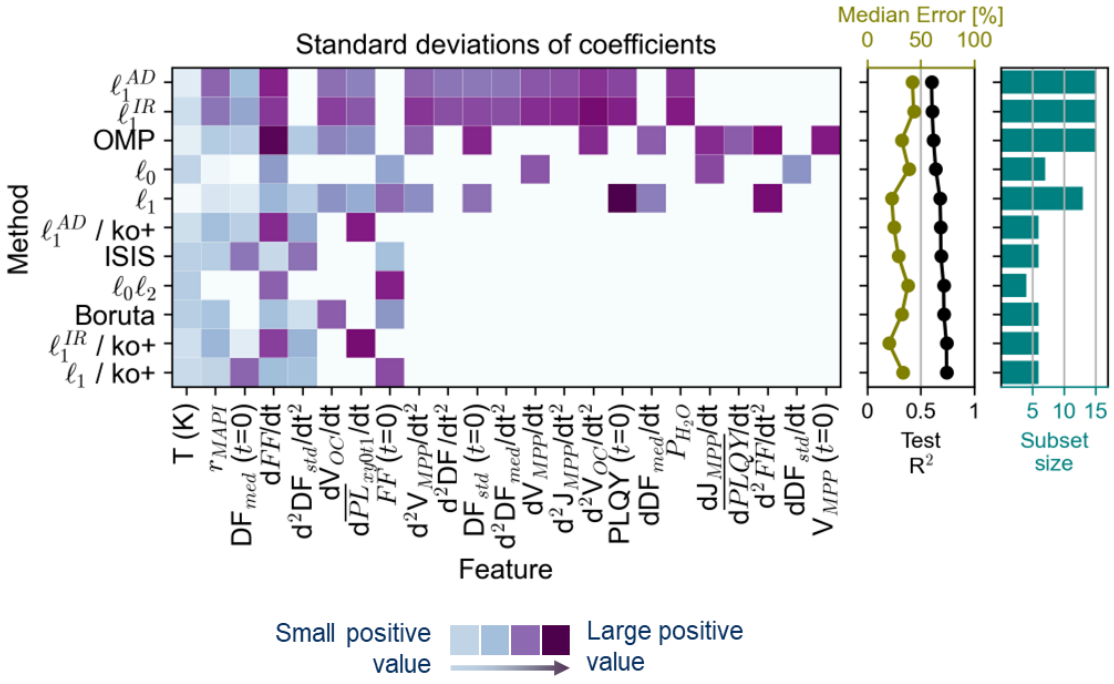

Figure 6.4. Heatmap of standard deviations of feature coefficients during t_{80} degradation lifetime prediction in $MAPbI_3$ perovskite solar cells. The intensity of colors represents the standard deviation of the feature coefficients evaluated over N=45 leave-one-out (LOO) train-test splits. The green graph on the right shows the median percentage test errors (i.e., |observed t_{80} − predicted t_{80}|/ observed t_{80} %), evaluated over the LOO train-test splits. The black graph shows the test R^2 value between the observed and the predicted t_{80} values of the LOO test points. The bar chart shows the overall sparsity of each method, evaluated based on the number of non-zero median coefficients, obtained from the LOO models.

6.3.2 Discussion

Similar to the synthetic datasets in Section 5.4, the sample size N of the current dataset falls below the $10\,m_s^0$ heuristic threshold, posing a risk of overfitting. I estimated the signal-to-noise (SNR) ratio of the current dataset to be approximately 7 (see Figure 6.9 in the Supporting Information), which makes the synthetic datasets in Section 5.4 with SNR = 6 to be considered at least as hard as the current dataset. Figure 6.2a demonstrates the correlation structure of feature data in the current dataset, which closely resembles that of the *Type II* synthetic datasets. Although the current dataset is more complex with non-Gaussian feature data distributions (see Figure 6.6 in the Supporting Information) and non-linear feature-target dependencies, certain inferences obtained from the generically designed synthetic datasets still remain extendable to this dataset.

Across the feature selection methods, ℓ_1, OMP and the weighted ℓ_1 (i.e., ℓ_1^{AD} and ℓ_1^{IR}) methods yield large feature subsets for the current dataset, as evidenced by the large number of non-zero mean coefficients (see Figure 6.3a). As shown in Figure 6.4, most features selected by these methods exhibit large standard deviations across the LOO splits, possibly due to overfitting within each LOO training set. Figure 6.7b (in the Supporting Information) illustrates how feature selection by ℓ_1^{AD} varies drastically across the LOO test iterations, despite the fact that the training sets in any two iterations differ only by a couple data-points, leading to large number of features with nonzero mean coefficients. The weighted ℓ_1 methods exhibited similar behavior in *Type II* simulations in Section 5.4, showing sensitivity to small fluctuations across the replicate datasets, which then intensified when the features became uniformly distributed in Section 5.6.3. Owing to complex non-Gaussian distributions and non-linearity in the current dataset, the variability of these methods across LOO iterations has further intensified.

Figure 6.4 shows that each of the knockoffs-based methods—ℓ_1/ko+, ℓ_1^{AD}/ko+, and ℓ_1^{IR}/ko+— always selects four features (which are not necessarily the same across methods due to correlations among features) with small standard deviations, along with two features exhibiting large standard deviations. This suggests that four specific features are frequently selected across LOO iterations by each method, while two features are selected only occasionally, leading to six features with non-zero mean coefficients (see Figure 6.3c and 6.3d for ℓ_1^{IR}/ko+'s results). Additionally, the strength of these methods lies in their ability not only to suppress false discoveries but also to provide estimates (q) of the underlying false discovery rates (see Section 4.3.2), even without the precise knowledge of the ground-truth variables. As these estimates closely matched the observed values on average for N = 50 in the *Type II* datasets (see Figures 5.12 and 5.13), it is reasonable to assume similar behavior for the current dataset, which exhibits a much lower average q-estimate of 0.2 (see Figure 6.10 in the Supporting Information). This suggests that the two inconsistently selected features out of the total six may be irrelevant. Meanwhile, $\ell_0\ell_2$ method provides the most parsimonious solution without significant selecting two features—T (K) and r_{MAPI}—consistently across the LOO iterations, and two features occasionally, leading to a total of four nonzero mean coefficients.

The behaviors of the knockoffs-based methods and the $\ell_0\ell_2$ method are similar to those observed in *Type II* datasets, where these methods yielded highly sparse solutions with little variation across the replicate datasets. Meanwhile, ℓ_0, which performed similarly to $\ell_0\ell_2$ in the synthetic datasets, exhibit substantial variations in its feature selection across the LOO splits, possibly due to the increased complexity of the current dataset. The presence of an additional ℓ_2 penalty in the $\ell_0\ell_2$ method might have helped mitigate this. In summary, supported by results from

generically designed synthetic datasets, the knockoffs-based and the $\ell_0\ell_2$ methods, serve as effective feature selection methods for t_{80} prediction in $MAPbI_3$ solar cells.

6.3.3 Physical Interpretation of Selected Features

Selection of the four features—T (K), r_{MAPI}, DF_{med} (t=0), d^2DF_{std}/dt^2 and dFF/dt—is consistent with previous reports explaining the decay of PCE in $MAPbI_3$ solar cells. Dunlap-Shohl et al.[21] identified the degradation of the perovskite layer in $MAPbI_3$ (characterized by r_{MAPI}) as the dominant cause of PCE decay, which is further accelerated by the degradation of other layers at high temperatures.[17,18,21] Degradation of $MAPbI_3$ results in the formation of photo-inactive phases such as lead iodide, its hydroxide derivatives, and/or hydrates.[14] This process typically begins at the interfaces with adjacent layers, such as the ETL and HTL, where the ingress of oxygen and moisture, as well as the escape of volatile degradation products, is most pronounced.[21] *In situ* dark-field (DF) microscopy, which measures the diffusely-reflected light from surface of the solar cell, has been previously used to observe this phenomenon by capturing light scattered at phase boundaries and from the film roughness that accompanies degradation.[21,107] As the solar cell degrades over time, changes in the film morphology, grain boundary structure, or phase distributions, generally lead to increased light scattering, resulting in higher DF signal intensity.[107] In the models presented here, the median pixel intensity of the DF image taken at time 0 (i.e., DF_{med} (t=0)) and the curvature of the standard deviation of pixel intensities from DF images taken within the first 90 minutes of stressing (i.e., d^2DF_{std}/dt^2) serve as effective proxies for capturing these changes. As these DF-derived features are evaluated at time 0 or during early-time

degradation, they reflect the sample-specific variations in PCE decay arising from initial device quality.

Furthermore, dFF/dt, despite exhibiting a coefficient with a small nonzero mean and a large standard deviation with the ℓ_1^{IR}/ko+ method (see Figure 8d), is selected as a relevant feature by every selection method. This feature represents the initial rate of change of the solar cell fill factor (FF)—a metric indicating how close the solar cell is to ideal operation. The positive coefficient is believed to link the early-time rise in FF to improved t_{80} lifetimes, suggesting potential interfacial or bulk trap passivation during early light exposure, or the alleviation of the inherently 'built-in' electric field under maximum power point biasing.[21]

6.4 Uncertainty Quantification in t_{80} Predictions

Finally, I estimate the uncertainties around the t_{80} predictions caused by perturbations in data using Jackknife+ conformal prediction (CP), as discussed in Section 4.5. Figure 6.5 shows the median t_{80} predictions and their confidence intervals with $\alpha = 0.1$ (corresponding to a theoretical coverage of 90%), obtained using the ℓ_1^{IR}/ko+ method. In Figure 6.5a, an apparent coverage estimate of 96% is determined by counting the median t_{80} predictions that fall within their respective CP intervals. Unlike Figure 6.3b, the parity plot in Figure 6.5a provides a median-centered prediction interval rather than a single prediction for each observed t_{80} value, allowing for an assessment of the sensitivity of the feature selection method to data fluctuations.

Figure 6.5b shows that the normalized sizes of these intervals are nearly identical across the range of observed t_{80} values, with only a subtle downward trend, potentially attributable to

slightly increased fluctuations in data at lower t_{80} values. This suggests that ℓ_1^{IR}/ko+ more accurately captures the model's uncertainty behavior by producing larger intervals for more challenging data. This property of producing varying interval sizes, known as *adaptivity*, is desirable in CP as it goes beyond merely generating small intervals[44]. Meanwhile, the CP intervals obtained using the Jackknife-minimax scheme are much larger, indicating more conservative estimates with higher empirical coverages (see Figure 6.11 in the Supporting Information). Overall, by assessing model uncertainty using CP, one can not only provide prediction intervals with statistical guarantees but also identify regions in data that are challenging for prediction.

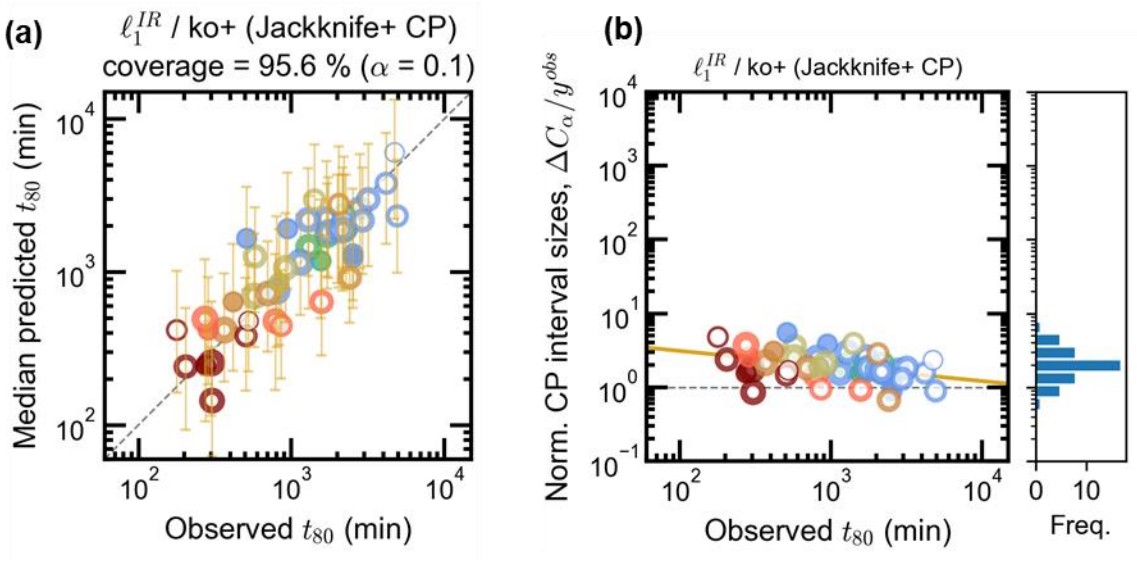

Figure 6.5. Conformal Prediction (CP) of t_{80} degradation lifetimes in $MAPbI_3$ perovskite solar cells using Jackknife+ with ℓ_1^{IR}/ko+ (a) Parity plot showing CP intervals and the median predictions at each observed t_{80} value, generated with $\alpha = 0.1$ using scheme outlined in Figure 4.8c. Markers represent the median predicted t_{80} values, while the error bars around them represent the CP intervals. The empirical coverage (of ~96%) indicates the percentage of observed t_{80} values lying within their corresponding CP intervals **(b)** Variation of normalized CP interval sizes (i.e., CP interval sizes ΔC_α divided by their respective observed t_{80} values) in (a) with the observed t_{80} values. The orange line is linearly fit trend line to guide the eye. The horizontal dashed line indicates the cases when the CP interval sizes are equal to their respective observed t_{80} values. The histogram on the right displays the distribution of the CP interval sizes.

6.5 Conclusions

Measuring t_{80} lifetimes requires time-consuming experimentation, which has limited the size of the current dataset. Starting with feature construction and culminating in uncertainty quantification using conformal prediction, I provided a comprehensive demonstration of the overall modeling strategy that predicts the t_{80} lifetimes, along with confidence intervals, in $MAPbI_3$ solar cells, despite the small nature of the dataset. First, I discussed the importance of constructing features that are physically meaningful using domain expertise. Besides the *a priori* known features such as temperature (T), partial pressure of water (p_{H_2O}) and the kinetically-modeled $MAPbI_3$ degradation rate (r_{MAPI}), experimental features derived from PL, DF and J-V measurements, performed after the start of the experiment, provide sample-specific information that is crucial for accurate t_{80} prediction.

The use of a leave-one-out testing scheme enabled maximal utilization of the small dataset for modeling. Among all features, T, r_{MAPI}, certain early-time features derived from time series of DF images, and the initial rate of change in solar cell fill factor (FF) were consistently selected as important by most feature selection methods. These observations align with previous qualitative analyses performed by our group[21] and others,[13,17,18] which identified degradation of the perovskite layer as the dominant driver of PCE decay in unencapsulated $MAPbI_3$ devices. Changes in DF intensity, which track the formation of non-perovskite phases, further confirm this hypothesis. Early-time passivation effects, such as charge carrier trap passivation and alleviation of inherently built-in electric fields, are reflected in the initial rise in FF.

Among the models used, ℓ_1^{IR}/ko+, which applies a knockoffs scheme to the ℓ_1^{IR} feature selection method, displayed the best predictive performance with a median test prediction error of 20%. This is significantly lower than the 36% previously reported by our group (Dunlap-Shohl et al.[21]) on the same dataset, despite using fewer features. This result highlights how carefully choosing a feature selection method, guided by domain expertise and backed by insights from synthetic data, can yield a high predictive accuracy, even with a small dataset. Additionally, the confidence intervals obtained through conformal prediction—evidenced by their large sizes at smaller t_{80} lifetimes—indicate the increased difficulty of predicting t_{80} lifetimes for more aggressive environmental conditions such as high temperature and humidity. This is expected, as the rapid degradation dynamics under such conditions make t_{80} lifetimes more sensitive to noise from unaccounted factors, such as the delays during manual experiment setup, lag time between the onset of stressing and the start of data acquisition, etc. Overall, the prediction of t_{80} prediction in $MAPbI_3$ solar cells serves as a valuable real-world case study to demonstrate the applicability of techniques discussed in this work.

6.6 Supporting Information

6.6.1 Solar Cell Fabrication Methods

The perovskite composition $MAPbI_3$ has a bandgap of ~1.61 eV, as reported by Dunlap-Shohl et al,[21] and the same fabrication protocol is outlined here. The solar cells used in this chapter have a *p-i-n* architecture as follows: ITO / NiOx / $MAPbI_3$ (300 nm) / C60 (40 nm) / BCP (7 nm) / Ag (100 nm). 1.0 M perovskite ink was prepared using methylammonium iodide (GreatCell Solar) and PbI_2 (Alfa Aesar, 99 999%, ultra-dry), dissolved in a 7:3 v/v mixture of γ-butyrolactone and dimethyl sulfoxide (both solvents anhydrous grade from Sigma Aldrich), and stored for 1-2 hours before use. ITO-coated glass slides (1.5 × 1.5 cm, 15 Ω sq−1 , Yingkou Shangneng Photoelectric Material Co.) were sonicated in the following solutions: (1) Alconox detergent solution, (2) deionized water, (3) acetone, and (4) isopropanol. Each sonication step was performed for 10 minutes, rinsing in deionized water in between each step. After this, the slides were dried under flowing nitrogen and cleaned under in argon plasma for 10 min. The substrates were then transferred to a nitrogen-filled glovebox (maintained at temperature between 25 – 27 ºC) for spin-coating with a solution of 0.1 M $Ni(OAc)_2.4H_2O$ (Sigma Aldrich, 99.998% trace metals basis) and 0.1 M ethanolamine (Sigma Aldrich, 99.5%) in ethanol (Sigma Aldrich, anhydrous) at 3000 rpm for one minute. The substrates were then removed from the glovebox and annealed at 300 °C for 60 min in air to form the NiO_x hole transport layer. The substrates were then returned to the glovebox for perovskite deposition, where 50 uL of perovskite ink was spin-cast onto the NiO_x-coated substrates at 4000 rpm for 45 s. 15 seconds prior to the end of the spin step, 580 mL of toluene (Sigma Aldrich, anhydrous grade) was poured onto the substrate to promote nucleation of the perovskite precursors. After the spin step, the films were annealed on a hot plate at 100 °C for

10 min. During perovskite deposition, the glovebox was continuously purged with flowing nitrogen to avoid solvent buildup. After the perovskite deposition, the substrates were transferred to a separate glovebox with a thermal evaporator (Angstrom Engineering Nexdep.) for C60 (Lumtec) and bathocuproine (BCP, Sigma Aldrich, sublimed grade) evaporation at maximum deposition rates of 0.5 and 0.3 Å s^{-1}, respectively. The substrates were then withdrawn from the evaporator, placed beneath a shadow mask (device area ~ 0.067 cm^2), and returned to the evaporator to deposit patterned Ag (Kurt Lesker, 99.99%) contacts at a maximum rate of 2 Å s^{-1}. All evaporation steps were conducted at a base pressure of 5×10^{-6} Torr or lower. After Ag deposition, completed devices were stored in the dark in a nitrogen-filled glovebox until use.

6.6.2 In situ Degradation Experiments

The same experimental procedure reported by Dunlap-Shohl et al[21] is outlined here. Each device was placed in a home-built sealable environmental chamber equipped with electrical contacts, gas connections, glass window for illumination, and electrical heater, connected to a closed-loop thermocouple temperature control system. The atmosphere was controlled by a mixture of N_2 and dry air, allowing the O_2 content to be varied while maintaining constant total gas flow set at 2.0 L min^{-1}. The humidity of the chamber is controlled by flowing the gas mixture through an appropriately mixed solution of glycerol and water, before feeding it into the environmental chamber. The device was illuminated using an upright microscope (Olympus BX53M, equipped with a 5x Mitutoyo Plan Apo NIR HR objective lens) fitted with a Lumencor Spectra X Light Engine LED light source. A wavelength of 542 nm green LED was used, whose intensity was calibrated to yield a photon flux equivalent to that absorbed by the perovskite band gap under AM1.5G illumination. The microscope allows capture of photoluminescence (PL) and

dark-field (DF) images by switching between appropriate filter cubes in the light path. For PL, a dichroic mirror with 665 nm cutoff (Semrock FF665-Di02-25x36) was used; a long-pass filter on the emission side of the cube with 664 nm cutoff (Semrock BLP01-664R-25) further attenuates spurious signal due to reflected excitation light. For DF, a standard Olympus U-MDF filter cube was used. During degradation experiments, the PL and DF cubes were switched automatically using a homemade drive mechanism controlled by an Arduino Uno microcontroller. Electrical measurements were made using Keithley 2400 source/measure unit, with the PL cube in place (1 sun equivalent illumination) every 15 minutes, until at least t_{80} was observed. For each measurement, the following characterization sequence is followed: (1) wide-field PL measurements under 1 sun illumination and open-circuit (OC) conditions; (2) DF measurements under ~0.01 sun illumination and open-circuit conditions; (3) steady V_{OC} measurement for 10 s; (4) a short J–V sweep to determine the maximum power point followed by steady measurement at maximum power point (MPP) for 10 seconds; (5) steady short-circuit current (J_{SC}) measurement for 10 s; (6) reverse and forward light J–V sweeps taken at 0.25 V s^{-1} ; and (7) reverse and forward dark J–V sweeps taken at 0.25 V s^{-1}. Between consecutive measurements, the device was maintained at the voltage corresponding to the most recently determined MPP under 1 sun illumination. Data acquisition was controlled by a house-developed control script written in Python.

6.6.3 *Supplementary Figures*

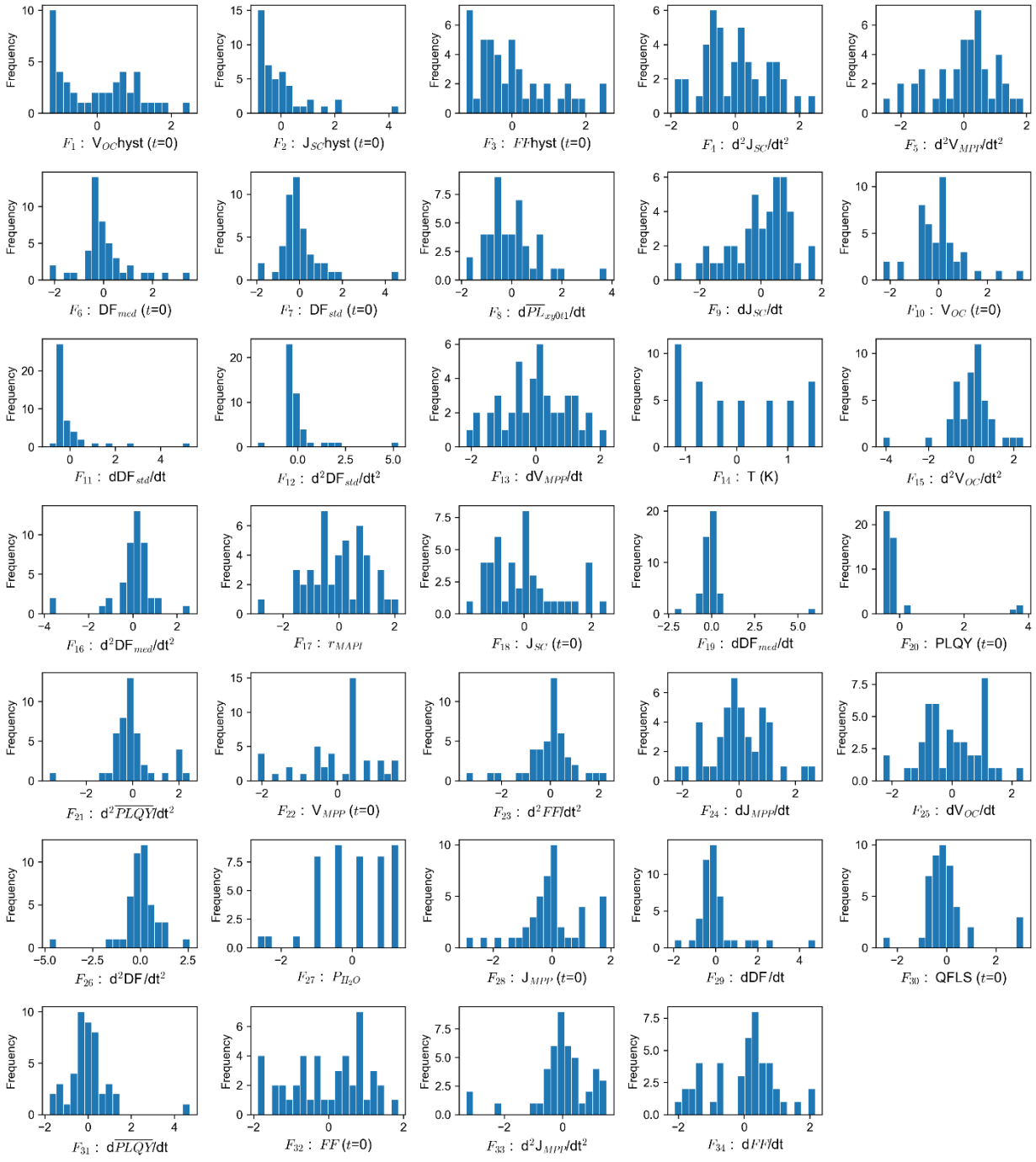

Figure 6.6. Individual data distributions of features in the $MAPbI_3$ perovskite solar cell t_{80} lifetime dataset.

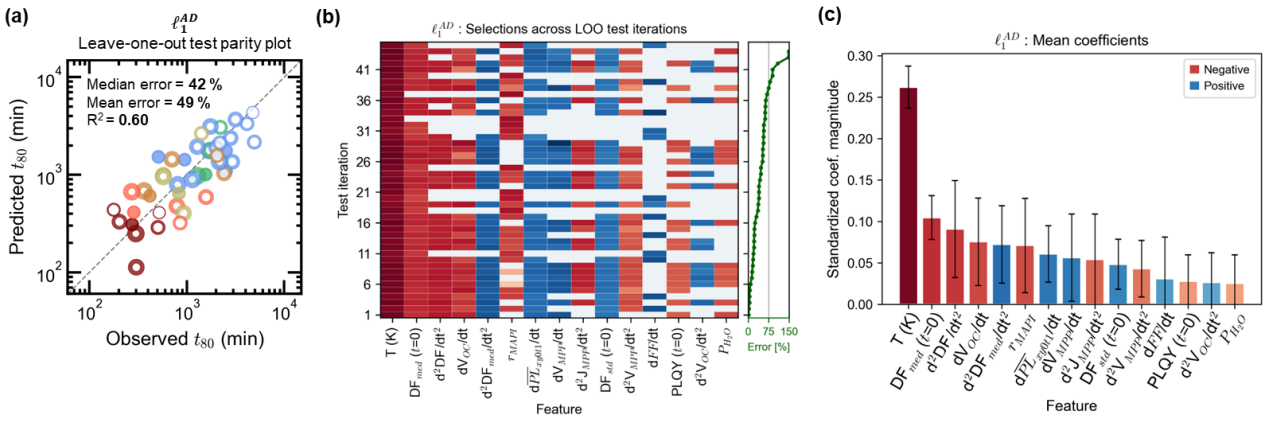

Figure 6.7.t_{80} degradation lifetime prediction in $MAPbI_3$ perovskite solar cells using ℓ_1^{AD} method (least parsimonious model). (a) Parity plot comparing the observed and the predicted t_{80} values at the leave-one-out (LOO) test data-points for the ℓ_1^{AD} method, which yields the largest s value as shown in Figure 6.3a. **(b)** Heatmap displaying coefficients selected by the ℓ_1^{AD} method (as shown in Figure 6.3a) across the LOO test-train splits. The red and blue hues indicate negative and positive coefficients respectively, while the intensity of the color indicates the magnitude. The green graph on the right shows the percentage test errors (i.e., |observed t_{80} − predicted t_{80}|/ observed t_{80} %) evaluated over each LOO test point. **(c)** Mean coefficients of the features selected by the ℓ_1^{AD} method, evaluated from the N = 45 LOO fits. The error bars indicate the standard deviations along the LOO test-train splits.

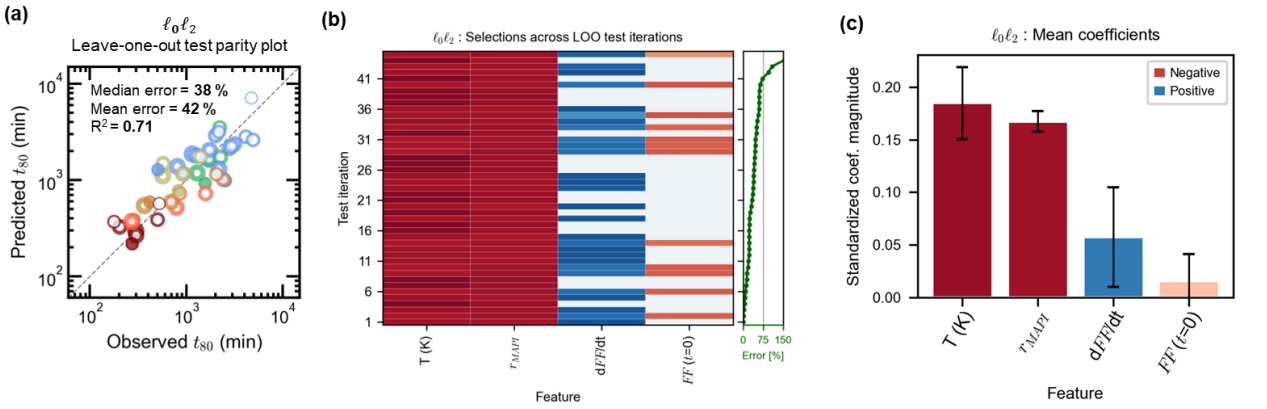

Figure 6.8. t_{80} degradation lifetime prediction in $MAPbI_3$ perovskite solar cells using $\ell_0\ell_2$ method (most parsimonious model). (a) Parity plot comparing the observed and the predicted t_{80} values at the leave-one-out (LOO) test data-points for the $\ell_0\ell_2$ method, which yields the smallest s value as shown in Figure 6.3. **(b)** Heatmap displaying coefficients selected by the $\ell_0\ell_2$ method (as shown in Figure 6.3a) across the LOO test-train splits. The red and blue hues indicate negative and positive coefficients respectively, while the intensity of the color indicates the magnitude. The green graph on the right shows the percentage test errors (i.e., |observed t_{80} − predicted t_{80}|/ observed t_{80} %) evaluated over each LOO test point. **(c)** Mean coefficients of the

features selected by the $\ell_0\ell_2$ method, evaluated from the $N = 45$ LOO fits. The error bars indicate the standard deviations along the LOO test-train splits.

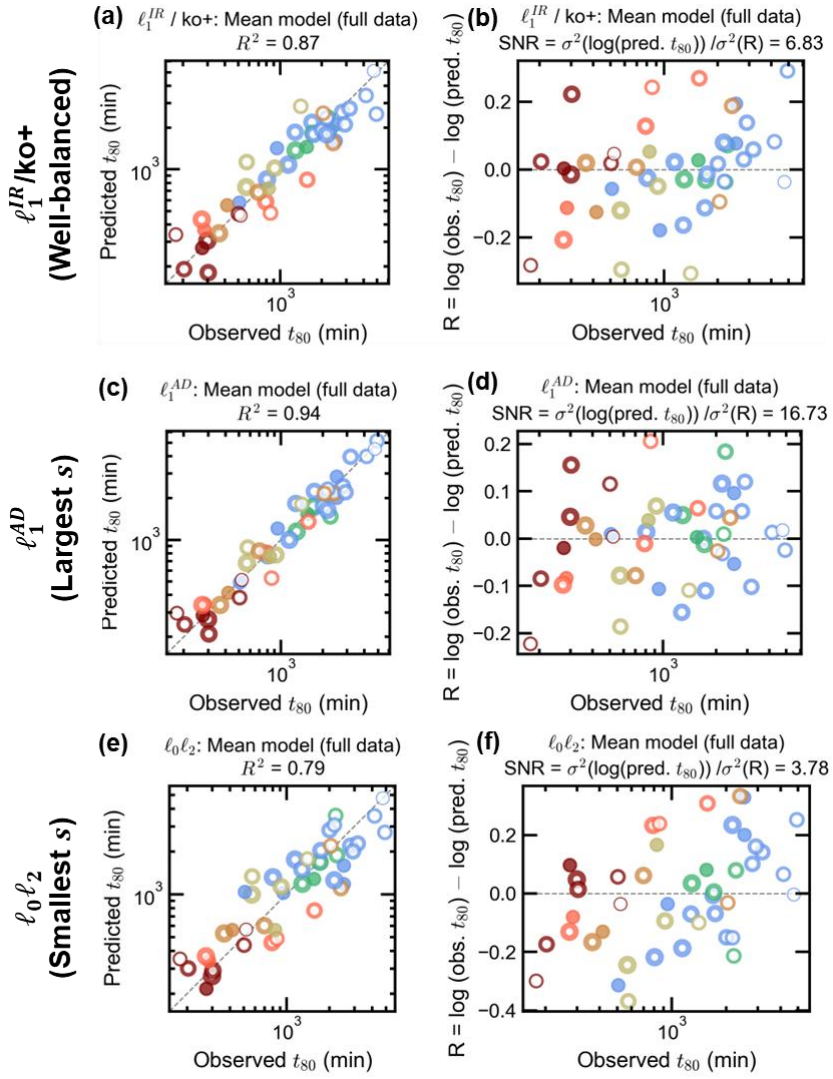

Figure 6.9. *SNR* estimation for the $MAPbI_3$ perovskite solar cell t_{80} lifetime degradation dataset. Three models are taken based on the characteristics of their solutions. **(a-b)** ℓ_1^{IR} /ko+ is the best-performing solution with the best balance of parsimony and error. **(c-d)** ℓ_1^{AD} is the least parsimonious model with the largest s, **(e-f)** $\ell_0\ell_2$ is the most parsimonious model with the smallest s. Plots a, b and c show the predicted t_{80} values obtained by fitting the full dataset using the features selected across the LOO splits (as shown in Figure 6.3a), and plots b, d, and f show the residuals $\boldsymbol{R}$ obtained from these predictions, where $R_i = \log(obs.\ t_{80})_i - \log(pred.\ t_{80})_i$ for $i = 1, \ldots, 45$. From these residuals, assuming the trained predictor as a good approximation of the underlying 'signal', we can estimate SNR as $\sigma^2(\log(pred.\ t_{80}))/\sigma(\boldsymbol{R})$ where $\sigma^2(\ldots)$ indicates variance. Here, $\ell_0\ell_2$ underestimates SNR as ~4 because the model might be omitting some ground-truths due to its parsimonious nature, whereas ℓ_1^{AD} overestimates SNR as ~17 because the model might be fitting even the noise as signal. As a result, the underlying SNR value can be

assumed to be between ~4 and ~17. ℓ_1^{IR}/ko+, which is a well-balanced model, gives an estimate of ~7.

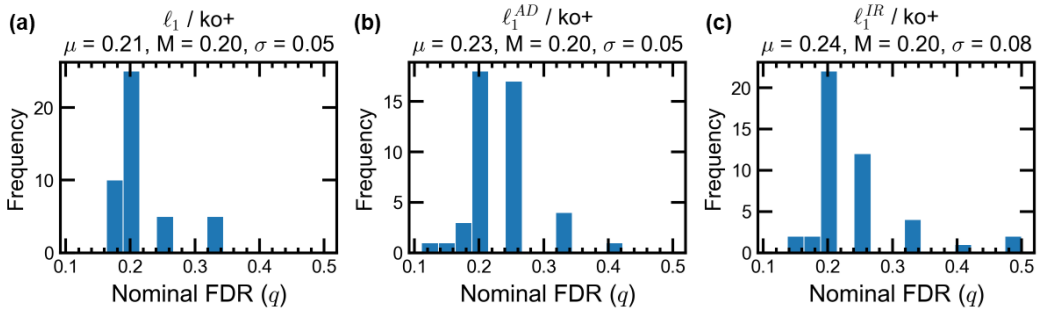

Figure 6.10. Distribution of nominal false-discovery rates q corresponding to the selected feature susbets across the leave-one-out (LOO) iterations, as predicted by **(a)** ℓ_1/ko+ **(b)** ℓ_1^{AD}/ko+ **(c)** ℓ_1^{IR}/ko+ for t_{80} degradation lifetime prediction in $MAPbI_3$ perovskite solar cells. Here, μ, M and σ represents the mean, median and standard deviation of q values across the $N = 45$ LOO iterations.

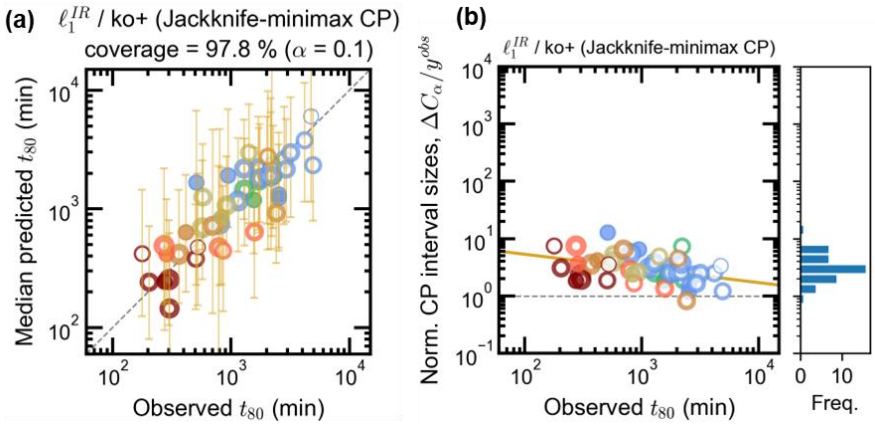

Figure 6.11. Conformal Prediction (CP) of t_{80} degradation lifetimes in $MAPbI_3$ perovskite solar cells using Jackknife-minmax with ℓ_1^{IR}/ko+ (a) Parity plot showing CP intervals and the median predictions at each observed t_{80} value, generated with $\alpha = 0.1$ using scheme outlined in Figure 4.8c. Markers represent the median predicted t_{80} values, while the error bars around them represent the CP intervals. The empirical coverage (of 98%) indicates the percentage of observed t_{80} values lying within their corresponding CP intervals **(b)** Variation of normalized CP interval sizes (i.e., CP interval sizes ΔC_α divided by their respective observed t_{80} values) in (a) with the observed t_{80} values. The orange line is linearly fit trend line to guide the eye. The horizontal dashed line indicates the cases when the CP interval sizes are equal to their respective observed t_{80} values. The histogram on the right displays the distribution of the CP interval sizes.

7 PREDICTION OF t_{80} LIFETIMES IN FORMAMIDINIUM-CESIUM LEAD IODO-BROMIDE SOLAR CELLS

The long-term stability of solar cells based on methylammonium lead iodide ($MAPbI_3$)—the archetypal perovskite composition—remains a critical challenge due to the volatility of the methylammonium (MA^+) cation and the iodide (I^-) anion. To address this, alternative perovskite compositions have been explored to enhance stability while preserving desirable optoelectronic properties.[110] A popular alternative is formamidinium-cesium lead iodo-bromide ($FA_xCs_{1-x}Pb(I_yBr_{1-y})_3$), in which a mixture of formamidinium (FA^+) and cesium (Cs^+) cations replaces the MA^+, and bromide (Br^-) anions substitute for a portion of the I^- in the perovskite lattice.[110–112] In addition to enhancing entropic stability,[113] the incorporation of large FA^+ and the small Cs^+ cations—along with Br^- anions replacing some of the I^-—in the right proportions induces optimal tilting of the PbX_6 octahedra within the perovskite lattice, resulting in improved crystal packing.[111,114] Consequently, $FA_xCs_{1-x}Pb(I_yBr_{1-y})_3$ compositions have demonstrated significant improvements in both performance and stability compared to the traditional $MAPbI_3$ composition,[13,112] making them promising candidates for commercial PV applications. Additionally, tuning the cation and anion composition enables precise control over the bandgap, key advantage for tandem PV applications.[110,112] However, these compositions still degrade under illumination in the presence of oxygen and moisture,[115] and even in inert atmospheres,[12] raising concerns about their long-term durability.

As a result, similar to the $MAPbI_3$ solar cells discussed in Chapter 6, a comprehensive evaluation of the stability of $FA_xCs_{1-x}Pb(I_yBr_{1-y})_3$ solar cells, using metrics such as the t_{80}

lifetimes, is crucial. For this, it would be beneficial to have a predictive model capable of estimating t_{80} based on the device's initial quality, early-time dynamics of its performance, and the environmental stress conditions, without requiring full-duration durability experiments, which are typically slow-paced and can take days or even months. Following the approach used for $MAPbI_3$ in Chapter 6, ML tools are preferable to fully mechanistic models for this purpose, given the complexity of the degradation mechanisms in $FA_xCs_{1-x}Pb(I_yBr_{1-y})_3$ solar cells, which are even more complicated than those in $MAPbI_3$. However, the datasets available for training such models are typically small in laboratory settings due to the slow-pace of the durability experiments, resulting in a *small data* scenario.

In this chapter, I present an in-house dataset of $N = 51$ durability experiments conducted on $FA_{0.8}Cs_{0.2}Pb(I_{0.83}Br_{0.17})_3$ solar cells, which were stressed under varying temperatures, humidity levels, oxygen concentrations and biasing conditions, until they reached their t_{80} lifetimes. The dataset is small due to the long duration of these experiments, which can extend up to a week. For each experiment, the same setup described in Chapter 6 was used to collect time-series measurements of current-voltage (J-V) characteristics, wide-field photoluminescence (PL), and dark-field (DF) images. With the goal of predicting t_{80} lifetimes in these solar cells, I implement the modeling workflow introduced in Chapter 4, comprising feature construction, feature selection and model fitting, and uncertainty quantification. In addition to incorporating insights from simulations on synthetic data (Chapter 5), I also provide a physical interpretation of the selected features. Overall, this dataset serves as another valuable real-world case study for demonstrating the small data modeling techniques discussed throughout this work.

Due to the thermally-activated nature of the underlying physicochemical processes (their dependence of a factor of e^{E_A/k_bT}), the t_{80} lifetime of a $FA_{0.8}Cs_{0.2}Pb(I_{0.83}Br_{0.17})_3$ solar cell in operation can vary by several orders of magnitude—from several months under inert, encapsulated conditions[27] to just a few hours under accelerated stress in oxygen- or moisture- rich environments at elevated temperatures[21]. To account for this non-linearity, I assign the base-10 logarithm of the lifetime ($\log_{10} t_{80}$), with t_{80} expressed in hours, as the target variable Y, thereby linearizing its dependence on the input features.[107]

7.1 Feature Construction

Temperature, humidity level, and oxygen concentration—controlled variables known to influence the t_{80} lifetime—are the first features added to the feature menu $\boldsymbol{F}$. The humidity level and the oxygen concentration are expressed in partial pressure units. As the exact relationship between these features and underlying degradation processes is not fully known, they are included directly, without applying non-linear transformations or estimating degradation rates or t_{80} lifetimes, as was done in the case of $MAPbI_3$ (see Section 6.1.1). Temperature (T), however, is incorporated in its inverse form ($1/T$), based on the assumption that t_{80} is governed by thermally-activated degradation processes, following a dependence of the form e^{E_A/k_bT}. Given that the target variable is the base-10 logarithm of t_{80}, this exponential dependence simplifies to a linear relationship with $1/T$, justifying its use as a feature in place of T. To incorporate variable biasing conditions, two stress modes are used: one under illumination and maximum power point bias, and another under dark and short-circuit bias (i.e., zero voltage at the contacts). A binary-valued feature

labeled as $Nsuns$ is used to represent this condition, with a value of 1 for illumination and 0 for dark conditions.

To account for variations in the t_{80} lifetimes arising from sample-specific factors—such as crystallization kinetics at interfaces,[16] degradation of the non-perovskite layers or their interfaces with the perovskite,[105,106] or defect migration driven by electric fields,[20,21]—characterization measurements including wide-field PLQY and DF microscopy, and J-V measurements are performed. These can be performed quickly *in situ* alongside PCE measurements. Similar to the $MAPbI_3$ case discussed in Section 6.1.2, features derived from the initial dynamics (observed in the first 60 mins) of these time-series measurements are included in the feature set $\boldsymbol{F}$. Due to the high interdependencies observed among features of this dataset, those exhibiting high mutual correlations (i.e., absolute Pearson correlation coefficient greater than 0.7) were removed, resulting in a final feature set with $p = 9$.

7.2 Model Setup and Testing

Given the complexity and the time-intensive nature of each experiment, this dataset comprises only N = 51 data-points. Although the number of features is lower in this case compared to Chapter 6, the fact that N falls below the $10p$ (=90) heuristic value still places it within the *small* data regime. To ensure that the dataset doesn't fall into the *too small* regime, N must exceed $m_s^0 \approx p_0 \log_2(p - p_0)/(1 - \rho)$, which represents the effective degrees of freedom required by a sparse model for exact sparse recovery of all p_0 ground-truth variables (see Section 3.2). For ρ, the mean value of the absolute Pearson correlation coefficients (= 0.2) is used, as it serves as a

representative value for the majority of pairwise correlations among the features (see Figure 7.1). Since p_0 is unknown, I consider a reasonable range of values between 1 and 5. Within this range, the estimated m_s^0 lies between 4 and 13, satisfying the condition $N > m_s^0$. Thus, while the current dataset is not *too small* for modeling, it still poses a risk of overfitting for certain methods, as $N < 10m_s^0$ for most assumed p_0 values.

I applied the feature selection schemes (as outlined in Section 4.2 and 4.3) and fit an OLS linear model using the selected features. As in Chapter 6, a leave-one-out (LOO) testing scheme (as discussed in Section 4.4) was used to evaluate the median error by predicting $Y = \log_{10}(t_{80})$ (with t_{80} expressed in hours) for each left-out trial using the model trained on the remaining trials. Additionally, the R^2 between the predicted t_{80} values at the left-out test trials and their respective observed values, denoted as R^2_{test}, was calculated for each method. A high R^2_{test} value indicates good model generalizability. Finally, Jackknife+ and Jackknife-minmax methods were employed to estimate the confidence intervals for predictions (as discussed in Section 4.5) by the best-performing feature selection method.

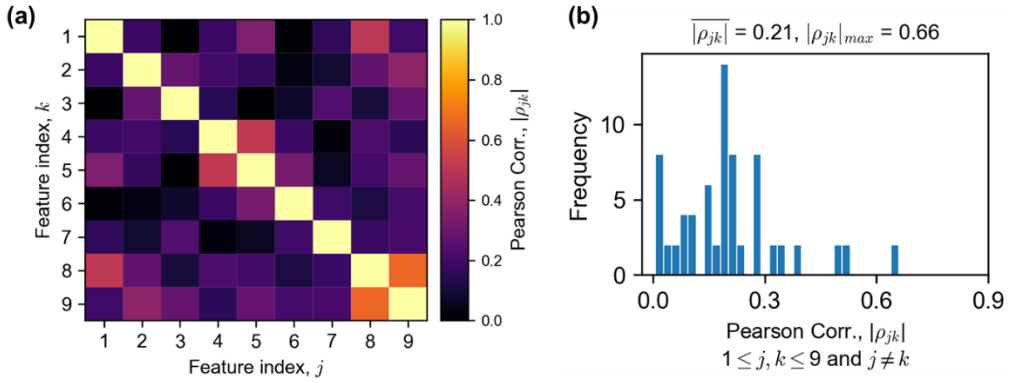

Figure 7.1. (a) Absolute Pearson correlation structure of the $FA_{0.8}Cs_{0.2}Pb(I_{0.83}Br_{0.17})_3$ perovskite solar cell t_{80} degradation lifetime dataset. A feature set of $p = 9$ is obtained after removing highly-correlated features that exceeded an absolute Pearson correlation value of 0.7. **(b) Distribution of absolute Pearson correlation coefficients.** $|\rho_{jk}|$ represents the mean value and $|\rho_{jk}|_{max}$ represents the maximum value.

7.3 t_{80} Prediction Results

7.3.1 Observations

Figure 7.2a shows the feature subsets selected by each selection method, along with their corresponding median prediction error and R^2_{test} values. Notably, the top five features—dJ_{SC}/dt, $f_{PL,bright}$ (t=0), Nsuns, DF (t=0), and P_{H_2O}—are consistently selected across nearly all methods, with similar mean coefficient values, highlighting their importance in explaining the variation in t_{80}. In contrast to the $MAPbI_3$ case, where the weighted ℓ_1 models (i.e., ℓ_1^{AD} and ℓ_1^{IR}) performed poorly by selecting many irrelevant features (see Figure 6.3a), these methods performed similar to their knockoff versions, selecting fewer features. Surprisingly, the ℓ_0 and $\ell_0\ell_2$ methods underperformed, selecting large feature subsets. Meanwhile, ℓ_1 and ISIS selected large feature subsets as expected, consistent with the trends observed in the synthetic data simulations.

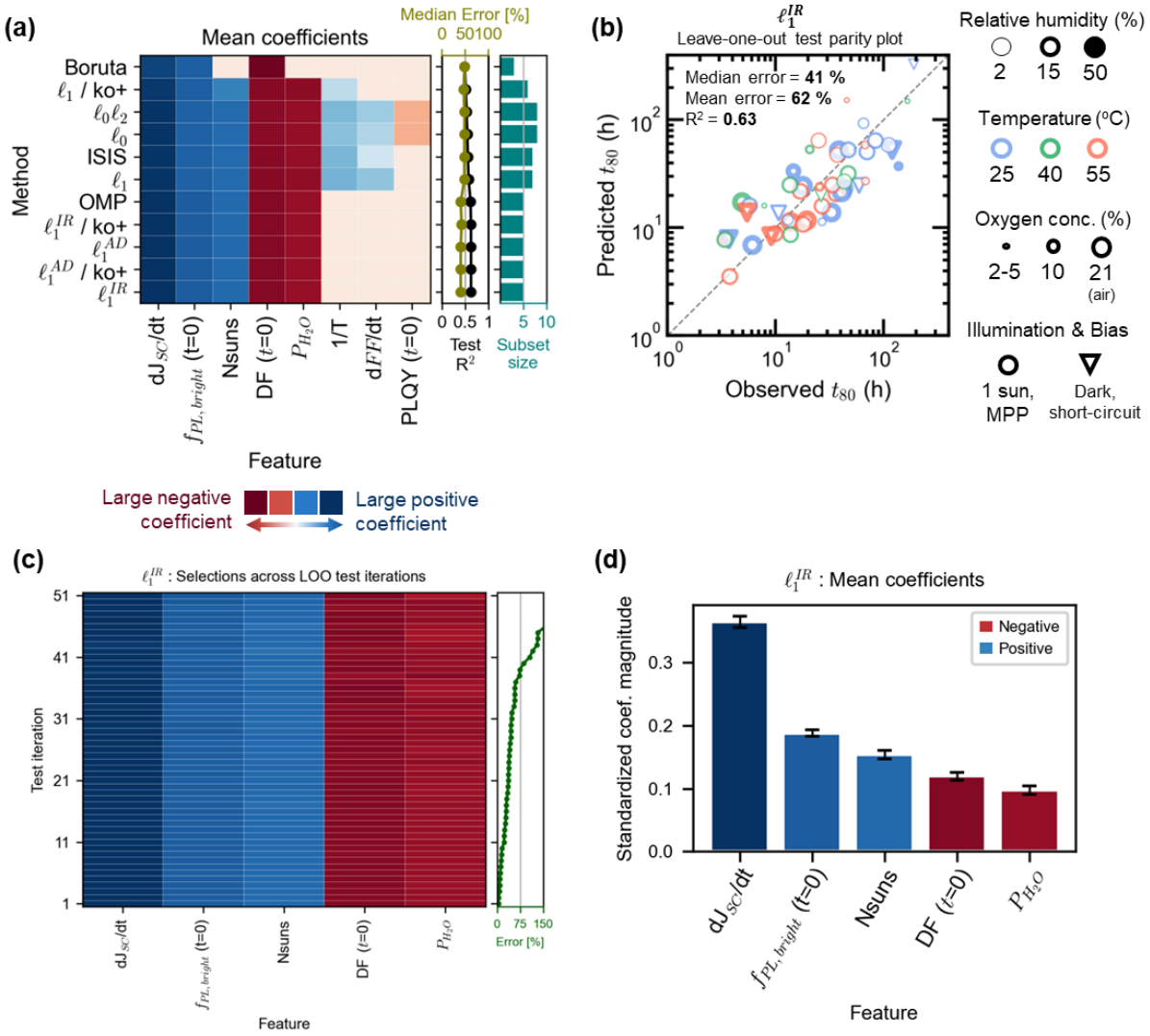

Figure 7.2. t_{80} degradation lifetime prediction in $FA_{0.8}Cs_{0.2}Pb(I_{0.83}Br_{0.17})_3$ perovskite solar cells. (a) Heatmap displaying mean coefficients of features selected from a list of p = 9 by each selection method. The intensity of colors represents the magnitude of the mean feature coefficients, evaluated over N=51 leave-one-out (LOO) train-test splits. Red and blue hues indicate negative and positive coefficients, respectively. The green graph on the right shows the median percentage test errors (i.e., |observed t_{80} − predicted t_{80}|/ observed t_{80} %), evaluated over the LOO train-test splits. The black graph shows the test R^2 value between the observed and the predicted t_{80} values of the LOO test points. The bar chart shows the overall sparsity of each method, evaluated based on the number of non-zero mean coefficients obtained from the LOO models. **(b)** Parity plot comparing the observed and the predicted t_{80} values at the LOO test points for the ℓ_1^{IR} method, which has the best balance of parsimony, median test error and R^2 value as shown in (a). The dashed line in (b) represents the $x = y$ parity line. The legend for the markers is provided next to the parity plot. **(c)** Heatmap displaying coefficients selected by the ℓ_1^{IR} method across the LOO test-train splits. The red and blue hues indicate negative and positive coefficients respectively,

while the intensity of the color indicates the magnitude. The green graph on the right shows the percentage test errors (i.e., |observed t_{80} − predicted t_{80}|/ observed t_{80} %) evaluated over each LOO test point. **(d)** Mean coefficients of the features selected by the ℓ_1^{IR} method, evaluated from the N = 51 LOO fits. The error bars indicate the standard deviations along the LOO test-train splits.

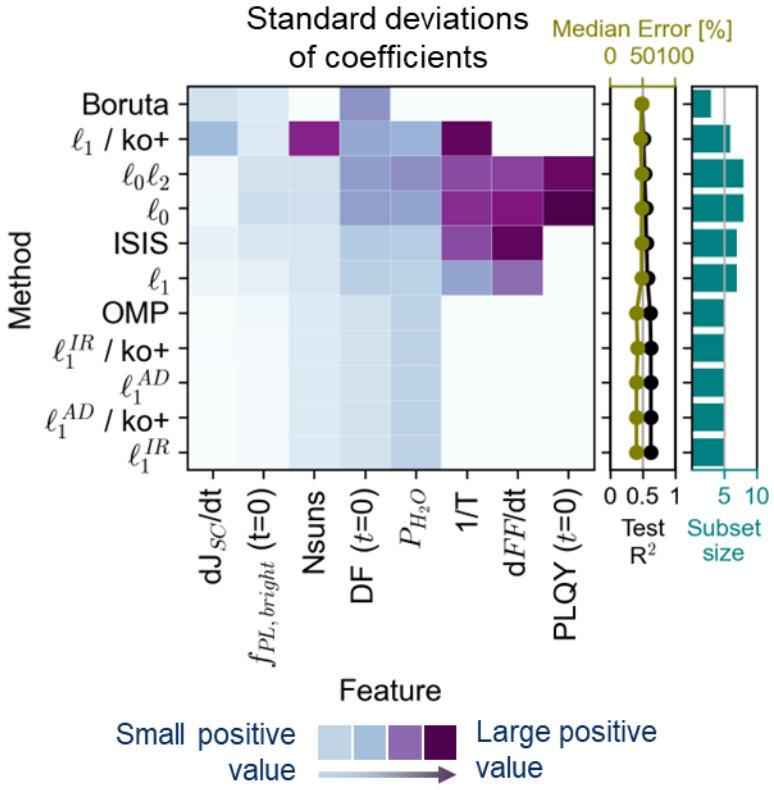

Figure 7.3. Heatmap of standard deviations of feature coefficients during t_{80} degradation lifetime prediction in $FA_{0.8}Cs_{0.2}Pb(I_{0.83}Br_{0.17})_3$ perovskite solar cells. The intensity of colors represents the standard deviation of the feature coefficients evaluated over N=51 leave-one-out (LOO) train-test splits. The green graph on the right shows the median percentage test errors (i.e., |observed t_{80} − predicted t_{80}|/ observed t_{80} %), evaluated over the LOO train-test splits. The black graph shows the test R^2 value between the observed and the predicted t_{80} values of the LOO test points. The bar chart shows the overall sparsity of each method, evaluated based on the number of non-zero median coefficients, obtained from the LOO models.

Figure 7.2b displays how closely the observed t_{80} values, left out in each LOO iteration, align with their predictions by the best-performing ℓ_1^{IR} method, which yielded a median test error of 41%. This is remarkably low given that the model used only the first 60 minutes of time-series data to predict t_{80} lifetimes extending to hundreds of hours, all while relying on a dataset of just 51 experiments. Figure 7.2c depicts the selection of features in individual LOO test iterations, while Figure 7.2d displays the nonzero mean values of the coefficients and their standard deviations evaluated across the N = 45 LOO iterations. The small standard deviations (in relative to their means) indicate that the selected features consistently have similar coefficient values across the LOO iterations (Figure 7.2c).

7.3.2 Discussion

Similar to the synthetic datasets in Section 5.4, the sample size N of the current dataset falls below the heuristic threshold of $10\ m_s^0$, posing a risk of overfitting. Figure 7.1a demonstrates the correlation structure of feature data in the current dataset, which closely resembles that of the *Type I* synthetic datasets. As shown in Figures 5.2a and 5.2d, *Type I* datasets exhibited smaller differences between the errors and R_{test}^2 values across the methods. However, substantial differences were observed in the sizes of the selected feature subsets across the methods. The subset sizes of the ℓ_1, OMP and ISIS methods remained large, whereas those for the remaining methods converged close to p_0.

Similarly, in the current dataset, the ℓ_1 and ISIS methods consistently select more than five features across the LOO iterations, while the weighted ℓ_1 (i.e., ℓ_1^{AD} and ℓ_1^{IR}) and their knockoffs versions select exactly five features. Contrary to expectations, OMP also selects only five

features—similar to the weighted ℓ_1 methods—while the BSS-based methods (ℓ_0 and $\ell_0\ell_2$) select even more features than ℓ_1 and ISIS, as indicated by the number of non-zero mean coefficients (see Figure 7.2a). To better understand this behavior, it is helpful to examine the standard deviations of the feature coefficients across the LOO iterations, as shown in Figure 7.3.

Figure 7.3 shows small standard deviations of the coefficients for the top five features selected by the weighted ℓ_1 methods, their knockoffs versions, and OMP, indicating that these features are consistently selected across the LOO iterations (see also Figure 7.2c). While the standard deviations for $\mathrm{d}J_{SC}/\mathrm{d}t$, $f_{PL,bright}$ (t=0), and Nsuns remain low, those for DF (t=0) and P_{H_2O} are slightly higher, suggesting some variability in their coefficient estimates after selection. In contrast, the remaining methods exhibit significantly higher standard deviations for the coefficients of features beyond the top five, reflecting inconsistent selection across LOO iterations. This also explains the relatively low mean coefficient values of these features in comparison to the top five. For ℓ_1 and ISIS, this high sensitivity to fluctuations in data was also observed in the *Type I* synthetic dataset, as evidenced by the large standard deviations in subset sizes (represented by error bars) in Figure 5.2g. Meanwhile, OMP showed a small standard deviation in subset sizes in the *Type I* datasets, which aligns with its reduced variability in the current dataset.

Interestingly, ℓ_0 and $\ell_0\ell_2$ demonstrate large variation in the features they select across the LOO iterations (see Figure 7.7b in the Supporting Information), which is contrary to the results observed in *Type I* and $MAPbI_3$ datasets (in Figure 5.2d and 6.3a respectively). This may be due to a strong underlying non-linear relationship between the target and the features, which these methods—designed to exhaustively search feature subsets under the assumption of linear dependence—fail to capture. This highlights the importance of incorporating physics-informed

non-linear features, such as r_{MAPI} discussed in Chapter 6, for datasets like this. Moreover, the high level of noise in this dataset, estimated at $SNR \approx 2.5$ (see Figure 7.9 in the Supporting Information), may have further amplified the variability in feature selection by these methods across the LOO iterations. Meanwhile, the small number of features and weak correlations among the features may have enabled ℓ_1^{IR} and ℓ_1^{AD} to perform well on this dataset, unlike in the $MAPbI_3$ dataset discussed in Chapter 6. Their knockoffs versions—ℓ_1^{AD}/ko+ and ℓ_1^{IR}/ko+—also performed well, consistent with their behavior in the $MAPbI_3$ case. Additionally, these methods provide estimates (q) of the underlying false discovery rates (see Section 4.4.2), even without the precise knowledge of the ground-truth variables. Since these estimates closely matched the observed values on average for the *Type I* datasets (see Figures 5.12 and 5.13), it is reasonable to assume similar behavior for the current dataset, which exhibits a much lower average q-estimate of 0.2 (Figure 7.10 in the Supporting Information).

In summary, the weighted ℓ_1 methods and their knockoff versions consistently select the same five features across the LOO iterations, demonstrating robustness to fluctuations in data. Although the remaining methods select features beyond these five, they are selected inconsistently across the LOO iterations, leading to large standard deviations in their coefficient estimates. Nevertheless, the mean coefficients of these additional features remain significantly lower than those of the top five, indicating the usefulness of these top five features in predicting t_{80}.

7.3.3 Physical Interpretation of Selected Features

Among the top five selected features, two—Nsuns and P_{H_2O}—are *a priori* known variables representing the environmental conditions. This aligns with previous studies,[12,13,17,115] which

reported accelerated degradation of unencapsulated $FA_{0.8}Cs_{0.2}Pb(I_{0.83}Br_{0.17})_3$ when exposed to light and moisture. The formation of hexagonal photo-inactive phases has been observed when stressed under light in moisture-rich environments,[115] while exposure to light in an inert atmosphere has been linked to the formation of reduced-lead containing species, which are detrimental to charge carrier transport.[12]

The remaining features— dJ_{SC}/dt, $f_{PL,bright}$ (t=0), and DF (t=0)—are sample-specific features, with values obtained after the start of the experiments. Unlike the t_{80} predictions in $MAPbI_3$ solar cells, where temperature (T) and the kinetically-modeled rate of the perovskite absorber (r_{MAPI}) were dominant descriptors,[21] the top two features in the current case are sample-specific. One possible reason for this distinction is the absence of a strong physics-informed feature, like r_{MAPI},[21] that effectively captures the non-linear dependence of t_{80} on environmental conditions. Alternatively, this may be a result of sample-to-sample variation contributing more significantly to the variation in t_{80} than the controlled environmental conditions. Nevertheless, a low median test error of ~40% indicates that these experimental features effectively capture this variation. Several studies[111,116,117] have reported the high sensitivity of $FA_xCs_{1-x}Pb(I_yBr_{1-y})_3$ solar cell stability to fluctuations in processing conditions, which likely explains the observed sample-to-sample variation. As a result, the selected experimental features serve as proxies for the initial qualities of these devices.

The top-ranked feature, $dJsc/dt$, captures the rate of change in the short circuit current within the first 60 minutes of stressing. This feature reflects the early-time 'burn-in' behavior (referring to the steep concave drop in J_{sc}), likely caused by the redistribution of accumulated ionic defects near the interface between the perovskite layer and the ETL or HTL.[118–120] This hypothesis

was validated through a separate experiment in which a low quality $FA_{0.8}Cs_{0.2}Pb(I_{0.83}Br_{0.17})_3$ device was stressed under 25 ºC, 2% relative humidity (RH), 2% v/v oxygen (O_2) concentration, 1 sun equivalent illumination, and maximum power point (MPP) bias (see Figure 7.4). The device was periodically probed using the following measure sequence: (1) measure stabilized short-circuit current (I_{SC}) under 1 sun, (2) apply a 0.5V bias in the dark for 2 minutes, and (3) measure the discharge current (I_{dis}) under dark and short-circuit conditions for 45 seconds. Figure 7.4a and 7.4b show the dynamics of I_{SC} and the subsequent I_{dis} measurements at different stress times. Each I_{dis} curve was individually integrated over the probing time period to obtain the corresponding ionic charge (Q) accumulated in the perovskite layer near the interfaces during dark biasing. Figure 7.4c demonstrates a correlation between the initial drop in stabilized I_{SC} and Q, thereby linking early-time carrier transport losses to initial ionic defect concentration—an indicator of device quality.

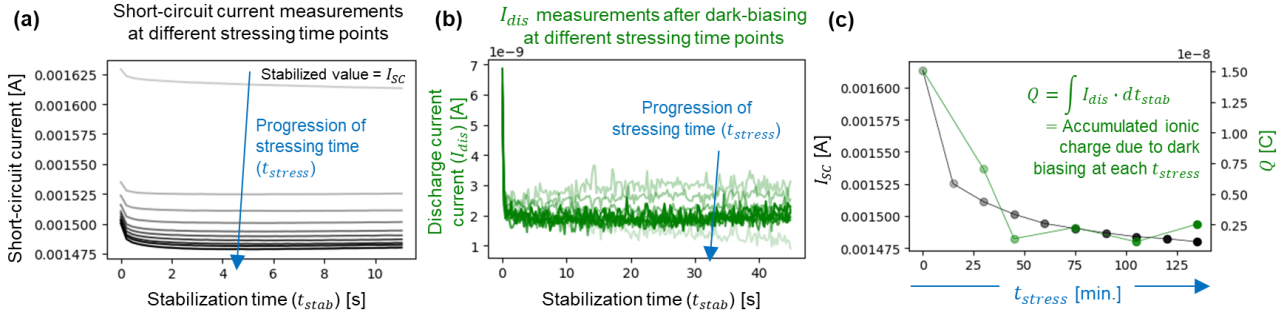

Figure 7.4. Demonstrating correlation between the initial drop in J_{SC} and the early-time ionic defect redistribution in a low-quality $FA_{0.8}Cs_{0.2}Pb(I_{0.83}Br_{0.17})_3$ photovoltaic device stressed under 25 ºC, 2% RH, 2% v/v O_2 concentration, 1 sun equivalent illumination, and MPP bias. (a) Stabilized short-circuit current (I_{SC}) measurement recorded at each stressing time. (b) Discharge current measurements taken under short-circuit, dark conditions for 45 s, immediately following a 2-minute dark bias at 0.5 V. The area under each curve represents the accumulated ionic charge during dark biasing, denoted as Q. (c) Plot of I_{SC} and Q versus stressing time, illustrating a correlation between their early-time declines.

The next dominant feature, $f_{PL,bright}$ (t=0), represents the fraction of pixels in the time 0 wide-field PL video that exhibit photo-brightening (i.e., an increase in PLQY) for 5 seconds. This feature effectively captures spatial heterogeneity by probing the spatial variation in PL photo-brightening, which is also linked to the initial ionic defect concentration and its redistribution upon illumination.[120,121] Similarly, DF (t=0)—another sample-specific experimental feature—also reflects spatial heterogeneity. It represents the mean pixel intensity of DF image captured at time 0, which corresponds to light scattering from grain or phase boundaries, and film roughness. As the initial quality of $FA_xCs_{1-x}Pb(I_yBr_{1-y})_3$ solar cells is also known to be influenced by the distribution of non-perovskite phase impurities formed during processing, this DF-based feature serves as an effective proxy.[116,117] Thus, incorporating sample-specific information related to the initial device quality is valuable for t_{80} prediction, as poor initial quality is a known driver of degradation in $FA_xCs_{1-x}Pb(I_yBr_{1-y})_3$ solar cells.[116,117]

7.4 Uncertainty Quantification in t_{80} Predictions

Finally, I estimate the uncertainties in the t_{80} predictions caused by perturbations in data using Jackknife+ conformal prediction (CP), as discussed in Section 4.5. Figure 7.5 presents the median t_{80} predictions and their confidence intervals with $\alpha = 0.1$ (corresponding to a theoretical coverage of 90%), obtained using the ℓ_1^{IR} method. In Figure 7.5a, the apparent coverage closely matches the theoretical value of 90%.

Similar to the $MAPbI_3$ case (see Figure 6.5), the sizes of the confidence interval (relative to the observed t_{80} values) in Figure 7.5b show a downward trend with increasing t_{80}, due to

increased fluctuations in data with low t_{80} values. Moreover, the interval sizes in this dataset are larger than those observed for $MAPbI_3$, due to three factors that make t_{80} prediction more challenging in $FA_xCs_{1-x}Pb(I_yBr_{1-y})_3$ solar cells: (1) prediction of much longer t_{80} lifetimes, up to several hundred hours, from only the first 60 minutes of time-series data; (2) substantial sample-to-sample variation due to differences in initial device qualities; and (3) the absence of a strong physics-informed feature like r_{MAPI}, which captures the non-linear dependence of t_{80} on environmental conditions. Figure 7.11 (in the Supporting Information) displays uncertainty quantification using the Jackknife-minmax method, which demonstrates similar confidence intervals as Jackknife+ but with slightly higher coverage. Overall, the CP methods effectively estimate confidence intervals and help identify regions in the data that are particularly challenging for prediction.

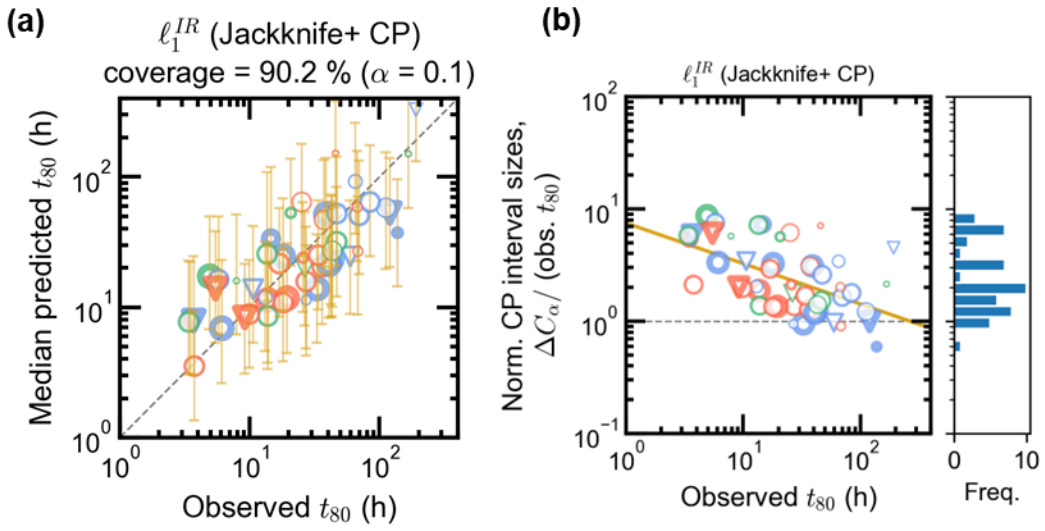

Figure 7.5. Conformal Prediction (CP) of t_{80} degradation lifetimes in $FA_{0.8}Cs_{0.2}Pb(I_{0.83}Br_{0.17})_3$ perovskite solar cells using Jackknife+ with ℓ_1^{IR} (a) Parity plot showing CP intervals and the median predictions at each observed t_{80} value, generated with $\alpha = 0.1$ using scheme outlined in Figure 4.8c. Markers represent the median predicted t_{80} values, while the error bars around them represent the CP intervals. The empirical coverage (of 90%) indicates the percentage of observed t_{80} values lying within their corresponding CP intervals **(b)** Variation

of normalized CP interval sizes (i.e., CP interval sizes ΔC_α divided by their respective observed t_{80} values) in (a) with the observed t_{80} values. The orange line is linearly fit trend line to guide the eye. The horizontal dashed line indicates the cases when the CP interval sizes are equal to their respective observed t_{80} values. The histogram on the right displays the distribution of the CP interval sizes.

7.5 Conclusions

The t_{80} lifetime dataset for $FA_{0.8}Cs_{0.2}Pb(I_{0.83}Br_{0.17})_3$ solar cells is small due to the long durations of the degradation experiments. For building a predictive model to estimate t_{80} in these solar cells, all stages of the modeling workflow including feature construction, feature selection, and uncertainty quantification, were covered. A leave-on-out testing scheme was employed to maximize the use of the available data. Unlike $MAPbI_3$ solar cells, the t_{80} prediction models in this chapter preferably select sample-specific experimental features derived from J-V characteristics, wide-field PL, and DF measurements. This indicates substantial sample-to-sample variation in the dataset, likely due to the high sensitivity of device quality to fluctuations in fabrication conditions.

Contrary to expectations, both ℓ_0 and $\ell_0\ell_2$ selected large subsets of features, likely due to a strong non-linear underlying relationship between the target and the features, as well as the absence of a physics-informed non-linear features such as r_{MAPI}. Nevertheless, the best-performing ℓ_1^{IR} model (whose results were identical to those of ℓ_1^{AD}, ℓ_1^{IR}/ko+, and ℓ_1^{AD}/ko+) yielded a median test error of 41 %—a notable result given that it predicts t_{80} values spanning over two orders of magnitude (up to several hundreds of hours) using only 60 mins of early-time data. This highlights the importance of choosing an appropriate feature selection method to achieve high predictive accuracy in a parsimonious manner, even with a small, noisy dataset. Additionally, the

confidence intervals obtained through conformal prediction—large than those observed for $MAPbI_3$—reflect the increased difficulty of predicting t_{80} lifetimes for this perovskite composition, due to both sample-to-sample variation and slower degradation rates. Meanwhile, both datasets exhibit a similar downward trend in interval size with increasing t_{80} values, suggesting the need for more data with shorter t_{80} values. Overall, the prediction of t_{80} in $FA_{0.8}Cs_{0.2}Pb(I_{0.83}Br_{0.17})_3$ solar cells serve as a valuable real-world case study, demonstrating the practical applicability of the techniques discussed throughout this work.

7.6 Supporting Information

7.6.1 Solar Cell Fabrication Methods

The perovskite composition $FA_{0.8}Cs_{0.2}Pb(I_{0.83}Br_{0.17})_3$ has a bandgap of ~1.65 eV. The solar cells used in this chapter have a *p-i-n* architecture as follows: ITO / poly-TPD / PFN-P2 (~1 nm) / $FA_{0.8}Cs_{0.2}Pb(I_{0.83}Br_{0.17})_3$ (300 nm) / C60 (50 nm) / BCP (7 nm) / Ag (100 nm). 1.0 M perovskite ink was prepared using PbI_2 (TCI America, 99.99%, trace metals basis), $PbBr_2$ (TCI America, >98.0%), FAI (Greatcell Solar Materials, >99.99%), and CsI (Fisher Scientific, 99.998%) in appropriate amounts to obtain the desired composition, in a 1:1 (v/v) mixture of N,N-Dimethylformamide (DMF) and N-Methyl pyrrolidone (NMP), and stored overnight at room temperature in a nitrogen-filled glovebox. ITO-coated glass slides (1.5 × 1.5 cm, 15 Ω sq^{-1}, Yingkou Shangneng Photoelectric Material Co.) were sonicated in Alconox detergent solution, deionized water, acetone, and isopropanol for 10 min each, rinsing in deionized water in between each step. The slides were then blow-dried in nitrogen and plasma-cleaned in argon for 10 min.

The substrates were then transferred to a nitrogen-filled glovebox (maintained at temperature between 25 – 27 ºC) for spin-coating with a solution of Poly-TPD (Poly[N,N′-bis(4-butylphenyl) -N,N′-bis(phenyl)-benzidine], Sigma Aldrich, MW >20,000 g/mol) in chlorobenzene (1 mg/mL concentration). After dripping 60 uL of solution, the substrate was spun at 4000 rpm (reached by an acceleration of 2000 rpm s^{-1}) for 30s, immediately annealed on a hot plate at 60 ºC for 10 mins and cooled at 25 ºC for 5 mins. Next, a 50 uL of PFN-P2 (Poly(9,9-bis(3'-(N,N-dimethyl)-N-ethylammoinium-propyl-2,7-fluorene)-alt-2,7-(9,9- dioctylfluorene))dibromide, Sigma-Aldrich) solution (0.5 mg/mL in methanol) was dropped within 3 s on the rotating substrate at 5000 rpm (max acceleration) for 20s, and the substrates were dried at 25 ºC for 28-32 mins. Finally, 100 uL of the perovskite ink was spin-coated at 4000 rpm (reached by an acceleration of 2000 rpm s^{-1}) for 45 s, during which 700 uL of toluene (Sigma Aldrich, anhydrous grade) was dripped (within 5-6 seconds) with 15 s remaining. The substrates were then annealed at 120 C for 15 mins. During perovskite deposition, the glovebox was continuously purged with flowing nitrogen to avoid solvent buildup. After the perovskite deposition, the substrates were transferred to a separate glovebox with a thermal evaporator (Angstrom Engineering Nexdep.) for C60 (Lumtec) and bathocuproine (BCP, Sigma Aldrich, sublimed grade) evaporation at maximum deposition rates of 0.5 and 0.3 Å s^{-1}, respectively. The substrates were then withdrawn from the evaporator, placed beneath a shadow mask (device area ~ 0.0453 cm^2), and returned to the evaporator to deposit patterned Ag (Kurt Lesker, 99.99%) contacts at a maximum rate of 2 Å s^{-1}. All evaporation steps were conducted at a base pressure of 5×10^{-6} Torr or lower. After Ag deposition, completed devices were stored in the dark in a nitrogen-filled glovebox until use.

7.6.2 *In situ Degradation Experiments*

The same experimental setup and characterization protocol described in Section 6.6.2 was used here.

7.6.3 *Supplementary Figures*

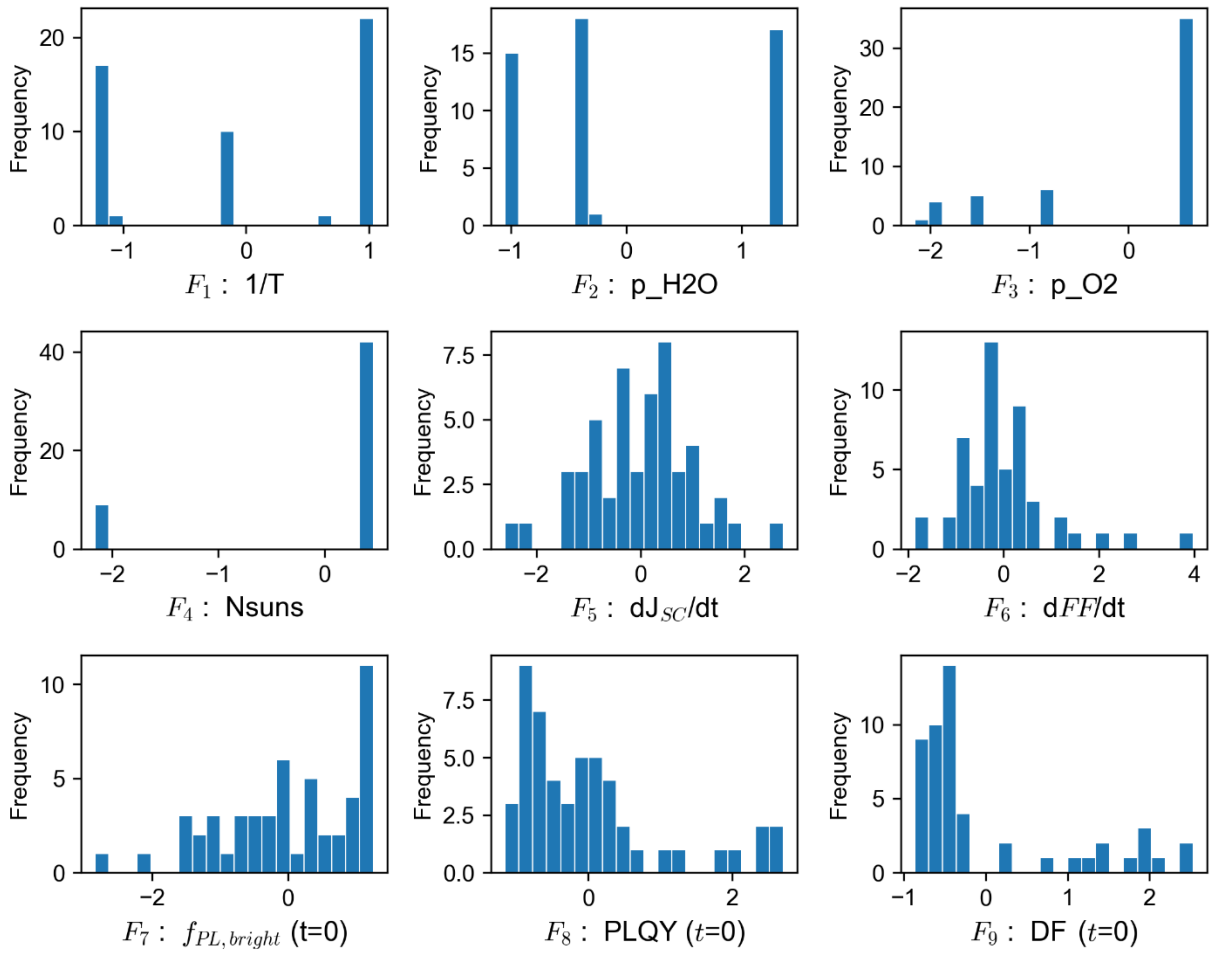

Figure 7.6. Individual data distributions of features in the $FA_{0.8}Cs_{0.2}Pb(I_{0.83}Br_{0.17})_3$ perovskite solar cell t_{80} lifetime dataset.

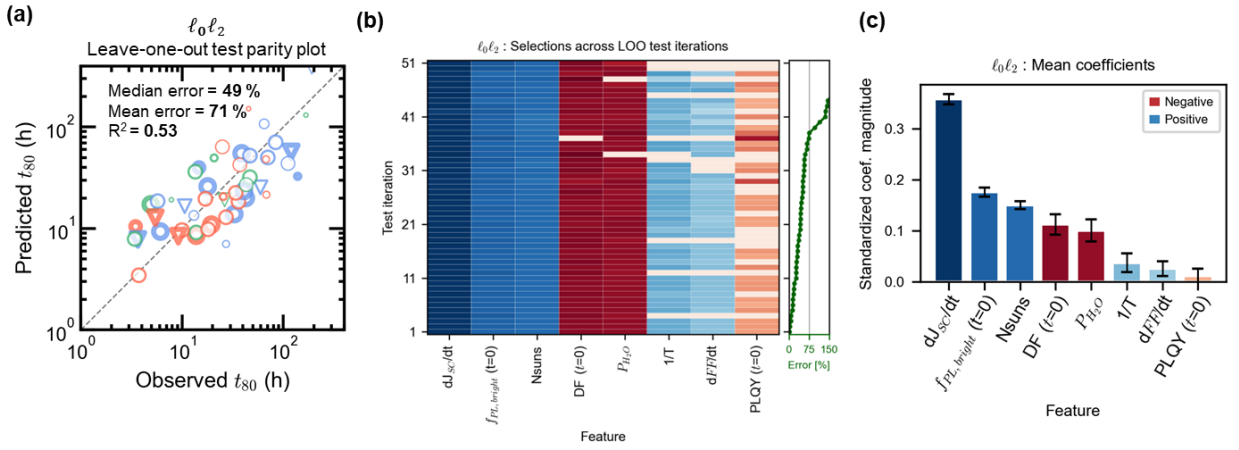

Figure 7.7. t_{80} degradation lifetime prediction in $FA_{0.8}Cs_{0.2}Pb(I_{0.83}Br_{0.17})_3$ perovskite solar cells using $\ell_0\ell_2$ method (least parsimonious model). (a) Parity plot comparing the observed and the predicted t_{80} values at the leave-one-out (LOO) test data-points for the $\ell_0\ell_2$ method, which yields the smallest s value as shown in Figure 7.2a. **(b)** Heatmap displaying coefficients selected by the $\ell_0\ell_2$ method (as shown in Figure 7.2a) across the LOO test-train splits. The red and blue hues indicate negative and positive coefficients respectively, while the intensity of the color indicates the magnitude. The green graph on the right shows the percentage test errors (i.e., |observed t_{80} − predicted t_{80}|/ observed t_{80} %) evaluated over each LOO test point. **(c)** Mean coefficients of the features selected by the $\ell_0\ell_2$ method, evaluated from the N = 51 LOO fits. The error bars indicate the standard deviations along the LOO test-train splits.

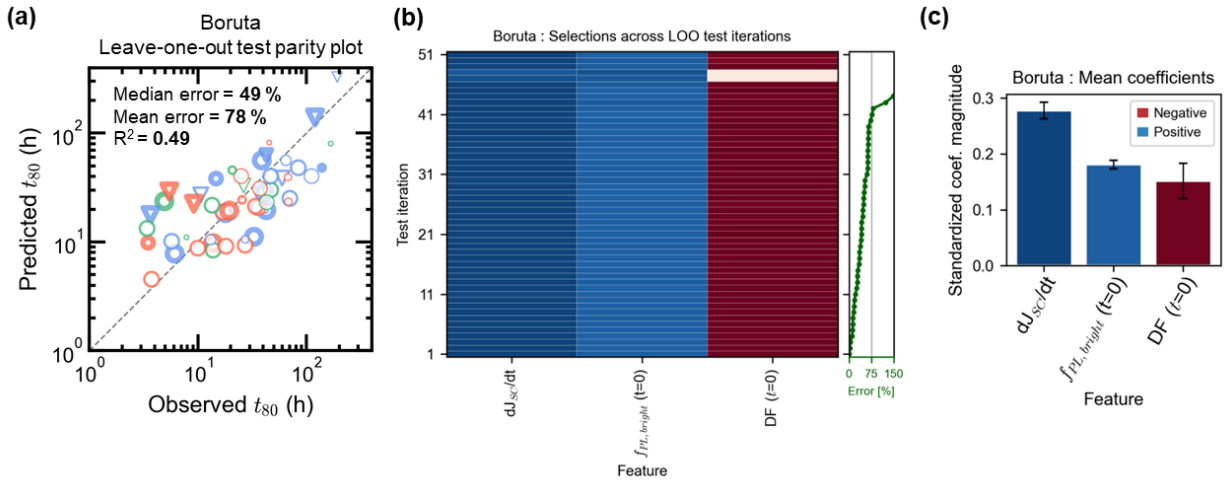

Figure 7.8. t_{80} degradation lifetime prediction in $FA_{0.8}Cs_{0.2}Pb(I_{0.83}Br_{0.17})_3$ perovskite solar cells using Boruta method (most parsimonious model). (a) Parity plot comparing the observed and the predicted t_{80} values at the leave-one-out (LOO) test data-points for the Boruta method, which yields the smallest s value as shown in Figure 7.2a. **(b)** Heatmap displaying coefficients selected by the Boruta method (as shown in Figure 7.2a) across the LOO test-train splits. The red and blue hues indicate negative and positive coefficients respectively, while the intensity of the color indicates the magnitude. The green graph on the right shows the percentage test errors (i.e., |observed t_{80} − predicted t_{80}|/ observed t_{80} %) evaluated over each LOO test point. **(c)** Mean coefficients of the features selected by the Boruta method, evaluated from the N = 51 LOO fits. The error bars indicate the standard deviations along the LOO test-train splits.

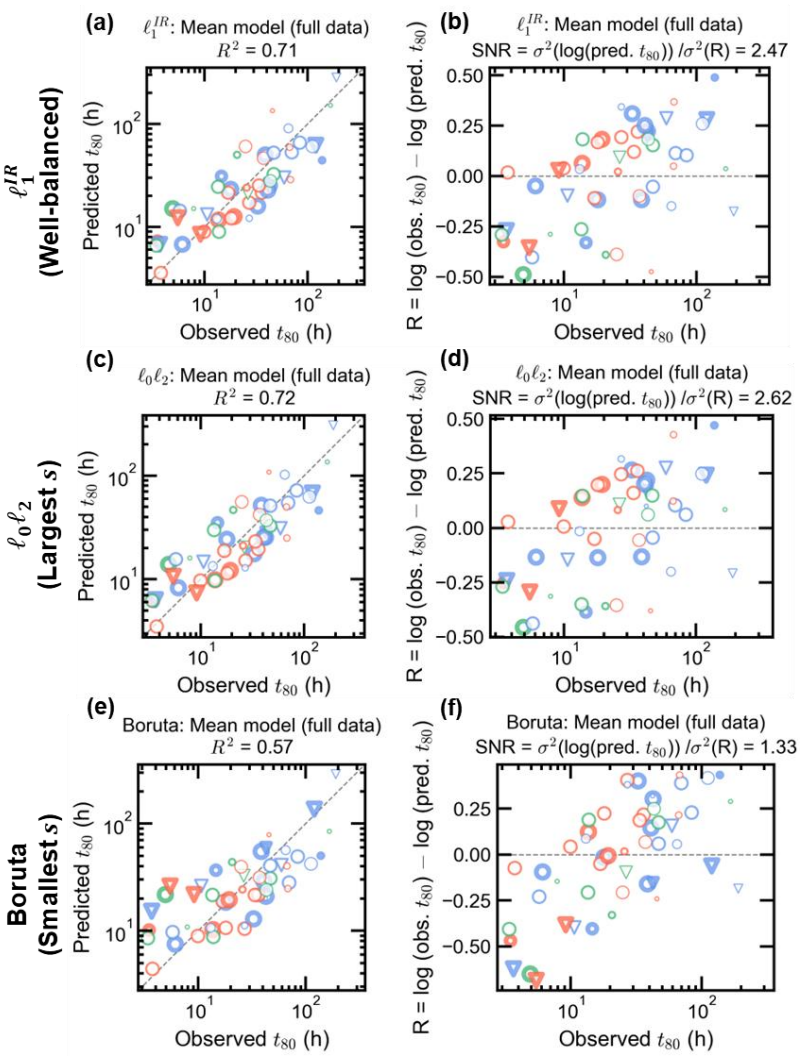

Figure 7.9. *SNR* estimation for the $FA_{0.8}Cs_{0.2}Pb(I_{0.83}Br_{0.17})_3$ perovskite solar cell t_{80} lifetime degradation dataset. Three models are taken based on the characteristics of their solutions. **(a-b)** ℓ_1^{IR} is the best-performing solution with the best balance of parsimony and error. **(c-d)** $\ell_0\ell_2$ is the least parsimonious model with the largest s, **(e-f)** Boruta is the most parsimonious model with the smallest s. Plots a, b and c show the predicted t_{80} values obtained by fitting the full dataset using the features selected across the LOO splits (as shown in Figure 7a), and plots b, d, and f show the residuals $\boldsymbol{R}$ obtained from these predictions, where $R_i = \log(obs.\ t_{80})_i - \log(pred.\ t_{80})_i$ for $i = 1, \ldots, 51$. From these residuals, assuming the trained predictor as a good approximation of the underlying 'signal', we can estimate SNR as $\sigma^2(\log(pred.\ t_{80}))/\sigma(\boldsymbol{R})$ where $\sigma^2(\ldots)$ indicates variance. Here, Boruta underestimates SNR as ~1.3 because the model might be omitting some ground-truths due to its parsimonious nature, whereas $\ell_0\ell_2$ overestimates SNR as ~2.6 because the model might be fitting even the noise as signal. As a result, the underlying SNR value can be assumed to be between ~1.3 and ~2.6. ℓ_1^{IR}, which is a well-balanced model, gives an estimate of ~2.5.

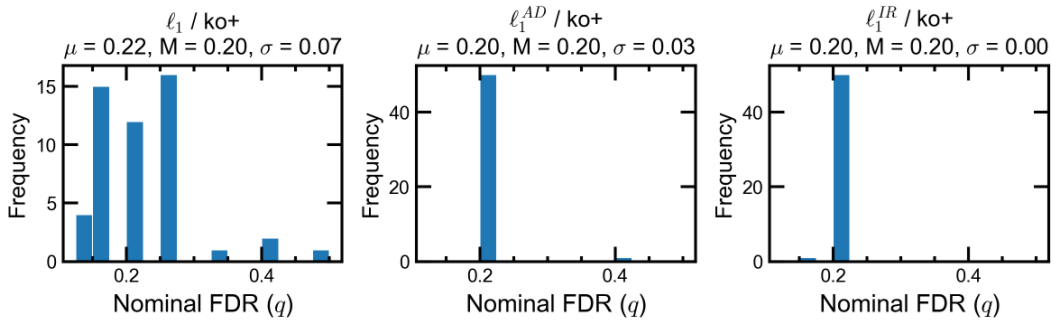

Figure 7.10. Distribution of nominal false-discovery rates q corresponding to the selected feature susbets across the leave-one-out (LOO) iterations, as predicted by **(a)** ℓ_1/ko+ **(b)** ℓ_1^{AD}/ko+ **(c)** ℓ_1^{IR}/ko+ for t_{80} degradation lifetime prediction in $FA_{0.8}Cs_{0.2}Pb(I_{0.83}Br_{0.17})_3$ perovskite solar cells. Here, μ, M and σ represents the mean, median and standard deviation of q values across the $N = 51$ LOO iterations.

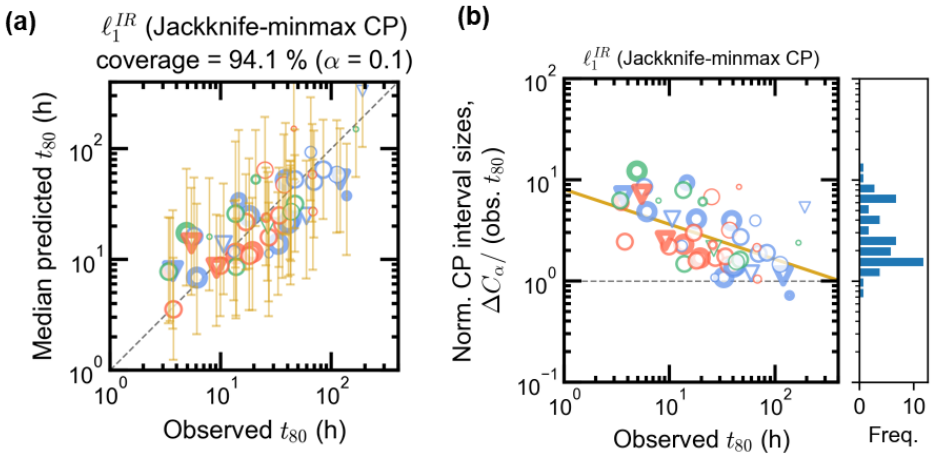

Figure 7.11. Conformal Prediction (CP) of t_{80} degradation lifetimes in $FA_{0.8}Cs_{0.2}Pb(I_{0.83}Br_{0.17})_3$ perovskite solar cells using Jackknife-minmax with ℓ_1^{IR} (a) Parity plot showing CP intervals and the median predictions at each observed t_{80} value, generated with $\alpha = 0.1$ using scheme outlined in Figure 4.8c. Markers represent the median predicted t_{80} values, while the error bars around them represent the CP intervals. The empirical coverage (of 94%) indicates the percentage of observed t_{80} values lying within their corresponding CP intervals **(b)** Variation of normalized CP interval sizes (i.e., CP interval sizes ΔC_α divided by their respective observed t_{80} values) in (a) with the observed t_{80} values. The orange line is linearly fit trend line to guide the eye. The horizontal dashed line indicates the cases when the CP interval sizes are equal to their respective observed t_{80} values. The histogram on the right displays the distribution of the CP interval sizes.

8 CONCLUSIONS AND FUTURE OUTLOOK

This work offers an educational perspective on identifying and modeling small datasets, which are common in science and engineering. While the prediction of operational lifetimes in hybrid perovskite solar cells is used as a case study, the insights presented here are broadly applicable across various scientific domains that rely on small laboratory datasets. The work outlines several key principles essential for applying machine learning to small datasets, providing guidance not only for researchers in the perovskite community but also for those in other fields facing similar data limitations:

- Small data remains a persistent challenge in scientific research, primarily due to the high costs and complexities associated with data collection. These limitations often cannot be fully addressed through proactive strategies such as active learning or data augmentation (Chapter 2).
- Statistical and information-theoretic concepts offer heuristic guidelines for determining when a dataset qualifies as *small*. Sparse linear modeling—through explicit or implicit feature selection—provides a practical approach by controlling the effective degrees of freedom used in model fitting. Depending on the sample size relative to these degrees of freedom, datasets can be categorized into regimes of *sufficient*, *small*, or *too small* data (Chapter 3).
- For small datasets, particularly those encountered in scientific domains, a structured modeling workflow using linear models is recommended. This workflow, guided by domain knowledge and statistical tools, includes constructing features based on domain

expertise, applying feature selection methods, determining the optimal sparsity level, and validating the final model using conformal prediction. Feature selection is especially challenging in real-world datasets, where feature sets often do not contain the ground truth variables that directly influence the target Y, but instead contain features correlated with these ground truths. This underscores the need for careful implementation of the workflow (Chapter 4).

- Synthetic datasets, designed to mimic real-world scientific data and equipped with known ground truths, offer a practical means to evaluate the effectiveness of these modeling techniques. BSS-based methods (ℓ_0 and $\ell_0\ell_2$), weighted ℓ_1 methods (ℓ_1^{AD} and ℓ_1^{IR}) and knockoffs-based methods have shown a good balance of low FDR^+ and high θ^+, while producing parsimonius solutions with a low prediction error (Chapter 5). However, in datasets with many features and complex correlation structures, the weighted ℓ_1 methods may fail to yield parsimonius solutions (Chapter 6). On the other hand, when there is a strong non-linear dependence of the target variable on the features, the BSS-based methods may struggle to yield a parsimonius solution (Chapter 7). In contrast, knockoffs-based approaches remain robust under these complexities.
- Durability research in the perovskite community exemplifies a data-scarce domain, where experiments are time-intensive and data acquisition is slow. For tasks such as t_{80} prediction, the statistical principles outlined in this work—combined with domain expertise and insights from synthetic data simulations—are invaluable. For instance, predicting t_{80} values in $MAPbI_3$ solar cells, which span several orders of magnitude up to thousands of minutes, has been achieved with a test error of just 20%. This was

accomplished using a knockoffs-based variant of LASSO (i.e., ℓ_1^{IR}/ko+), relying solely on features derived from stressing conditions, the first 90 minutes of data, and a limited number of training experiments (Chapter 6).

- Even when datasets are both small and noisy, the principles discussed remain applicable. For example, predicting t_{80} values in $FA_{0.8}Cs_{0.2}Pb(I_{0.83}Br_{0.17})_3$ solar cells—ranging up to several hundred hours—has been achieved with a test error of only 41%. This was done using LASSO variants (ℓ_1^{IR} and ℓ_1^{AD}) and their knockoffs counterparts (ℓ_1^{IR}/ko+ and ℓ_1^{AD}/ko+), using just the first 60 minutes of feature data and a small number of training experiments (Chapter 7).

During my Ph.D., perovskite materials gained widespread attention as promising candidates for PV applications due to their excellent optoelectronic properties. However, their instability has kept the field in its early stages, necessitating further investigation into their long-term durability. Compounding this challenge is the slow pace of durability experiments, which significantly limits dataset sizes in this domain. This highlights the importance of the modeling techniques presented in this work, which help sustain the pace of research despite data scarcity. The perovskite datasets used here thus serve as valuable real-world examples of small data, effectively demonstrating the applicability of these methods.

Similarly, several emerging fields in materials science and chemistry—such as photocatalysts, quantum materials, solid-state battery materials, and biomimetic materials—also suffer from limited data availability. The methods presented here are particularly valuable in such contexts. While computational studies in many domains within materials science and chemistry

often employ common machine learning techniques like neural networks, such methods are frequently unsuitable for small data scenarios, leaving many scientists and engineers without effective tools. Although the statistics literature does cover modeling strategies for small data, it often assumes a high level of prior knowledge required to navigate the field-specific jargon of academic statistics, making it difficult for researchers from other disciplines to engage with these ideas.

To support continued scientific progress in these data-scarce domains, the following thrusts can be useful: (1) effective collaboration with statisticians to quickly adopt ML tools tailored for small datasets, (2) proactive educational publishing by data scientists with backgrounds in chemistry and materials science to bridge the gap between theory and practice, and to introduce these tools to research areas where they are most needed, and (3) development of open-source software with user-friendly documentation, enabling scientists and engineers from non-programming backgrounds to apply these tools effectively.

9 DATA AND CODE AVAILABILITY

The code and data related to simulations on the synthetic datasets (Chapter 5) and real-world data modeling (Chapters 6 and 7) are available on www.github.com/hillhouse-group/small-data-ml. For specialized feature selection methods, the following open-source Python packages have been used: (1) *celer*[122,123] for minimizing the ℓ_1 (and the weighted ℓ_1) regularized loss functions (2) *abess*[41] to fit the ℓ_0 and $\ell_0\ell_2$ models (3) *boruta_py* from scikit-learn-contrib repository on GitHub, contributed by Kursa et al.[88], for the Boruta method, (4) *knockpy*[124] for knockoff feature construction.

10 VITA

Preetham P. Sunkari earned a Bachelors in Technology in Chemical Engineering with a Minor in Materials Science and Engineering at the Indian Institute of Technology, Kanpur, India. After graduation, Preetham started his M.S. and Ph.D. studies at the University of Washington in Fall of 2019, where he focused on developing machine learning tools for forecasting degradation in perovskite solar cells. Preetham currently lives in Seattle, WA, and enjoys running and cooking Indian food.

11 BIBLIOGRAPHY

1. International Energy Agency (IEA). *Global Energy Review 2025*. www.iea.org (2025).
2. Roxana Bardan (NASA). Temperatures Rising: NASA Confirms 2024 Warmest Year on Record. *https://www.nasa.gov/news-release/temperatures-rising-nasa-confirms-2024-warmest-year-on-record/* (2025).
3. Public Power Association, A. *America's Electricity Generation Capacity Update*. www.PublicPower.org (2025).
4. International Renewable Energy Agency (IRENA). *Renewable Power Generation Costs in 2023*. (2024).
5. Wilson, G. M. *et al.* The 2020 photovoltaic technologies roadmap. *Journal of Physics D: Applied Physics* vol. 53 Preprint at https://doi.org/10.1088/1361-6463/ab9c6a (2020).
6. U.S. Department of Energy, S. E. T. O. (SETO). SunShot 2030. *https://www.energy.gov/eere/solar/sunshot-2030* (2016).
7. Berry, J. J. *et al.* Perovskite Photovoltaics: The Path to a Printable Terawatt-Scale Technology. *ACS Energy Letters* vol. 2 2540–2544 Preprint at https://doi.org/10.1021/acsenergylett.7b00964 (2017).
8. National Renewable Energy Laboratory (NREL). Best Research-Cell Efficiency Chart. *https://www.nrel.gov/pv/cell-efficiency* (2025).
9. Dong, H. *et al.* Metal Halide Perovskite for next-generation optoelectronics: progresses and prospects. *eLight* **3**, 3 (2023).
10. Weerasinghe, H. C. *et al.* The first demonstration of entirely roll-to-roll fabricated perovskite solar cell modules under ambient room conditions. *Nat Commun* **15**, 1656 (2024).
11. Abzieher, T. *et al.* Vapor phase deposition of perovskite photovoltaics: short track to commercialization? *Energy Environ Sci* **17**, 1645–1663 (2024).
12. Cira, S. G. *et al.* Light-induced degradation of mixed-cation, mixed-halide perovskite: observed rates and influence of oxygen. *J Mater Chem A Mater* **13**, 5033–5044 (2025).
13. Boyd, C. C., Cheacharoen, R., Leijtens, T. & McGehee, M. D. Understanding Degradation Mechanisms and Improving Stability of Perovskite Photovoltaics. *Chem Rev* **119**, 3418–3451 (2019).
14. Siegler, T. D. *et al.* Water-Accelerated Photooxidation of $CH_3NH_3PbI_3$ Perovskite. *J Am Chem Soc* **144**, 5552–5561 (2022).
15. Meng, Y., Sunkari, P. P., Meilă, M. & Hillhouse, H. W. Chemical Reaction Kinetics of the Decomposition of Low-Bandgap Tin-Lead Halide Perovskite Films and the Effect on the Ambipolar Diffusion Length. *ACS Energy Lett* **8**, 1688–1696 (2023).
16. Saliba, M. *et al.* How to Make over 20% Efficient Perovskite Solar Cells in Regular (*n–i–p*) and Inverted (*p–i–n*) Architectures. *Chemistry of Materials* **30**, 4193–4201 (2018).
17. Mazumdar, S., Zhao, Y. & Zhang, X. Stability of Perovskite Solar Cells: Degradation Mechanisms and Remedies. *Frontiers in Electronics* **2**, (2021).
18. Dunfield, S. P. *et al.* From Defects to Degradation: A Mechanistic Understanding of Degradation in Perovskite Solar Cell Devices and Modules. *Adv Energy Mater* **10**, (2020).

19. Domanski, K. *et al.* Not All That Glitters Is Gold: Metal-Migration-Induced Degradation in Perovskite Solar Cells. *ACS Nano* **10**, 6306–6314 (2016).
20. Bae, S. *et al.* Electric-Field-Induced Degradation of Methylammonium Lead Iodide Perovskite Solar Cells. *J Phys Chem Lett* **7**, 3091–3096 (2016).
21. Dunlap-Shohl, W. A. *et al.* Physiochemical machine learning models predict operational lifetimes of CH3NH3PbI3 perovskite solar cells. *J Mater Chem A Mater* **12**, 9730–9746 (2024).
22. Shi, L. *et al.* Gas chromatography–mass spectrometry analyses of encapsulated stable perovskite solar cells. *Science (1979)* **368**, (2020).
23. Wang, S., Jiang, Y., Juarez-Perez, E. J., Ono, L. K. & Qi, Y. Accelerated degradation of methylammonium lead iodide perovskites induced by exposure to iodine vapour. *Nat Energy* **2**, 16195 (2016).
24. Zhu, H. *et al.* Long-term operating stability in perovskite photovoltaics. *Nature Reviews Materials* vol. 8 569–586 Preprint at https://doi.org/10.1038/s41578-023-00582-w (2023).
25. Saliba, M., Stolterfoht, M., Wolff, C. M., Neher, D. & Abate, A. Measuring Aging Stability of Perovskite Solar Cells. *Joule* vol. 2 1019–1024 Preprint at https://doi.org/10.1016/j.joule.2018.05.005 (2018).
26. Khenkin, M. V. *et al.* Consensus statement for stability assessment and reporting for perovskite photovoltaics based on ISOS procedures. *Nat Energy* **5**, 35–49 (2020).
27. Jiang, Q. *et al.* Towards linking lab and field lifetimes of perovskite solar cells. *Nature* **623**, 313–318 (2023).
28. Julien, A., Puel, J.-B. & Guillemoles, J.-F. Distinction of mechanisms causing experimental degradation of perovskite solar cells by simulating associated pathways. *Energy Environ Sci* **16**, 190–200 (2023).
29. Lim, J. *et al.* Kinetics of light-induced degradation in semi-transparent perovskite solar cells. *Solar Energy Materials and Solar Cells* **219**, 110776 (2021).
30. Mavlonov, A. *et al.* Thermal stability test on flexible perovskite solar cell modules to estimate activation energy of degradation on temperature. *Solar Energy Materials and Solar Cells* **277**, 113148 (2024).
31. Kim, J. *et al.* An effective method of predicting perovskite solar cell lifetime–Case study on planar CH 3 NH 3 PbI 3 and HC(NH 2) 2 PbI 3 perovskite solar cells and hole transfer materials of spiro-OMeTAD and PTAA. *Solar Energy Materials and Solar Cells* **162**, 41–46 (2017).
32. Graniero, P. *et al.* The challenge of studying perovskite solar cells' stability with machine learning. *Front Energy Res* **11**, (2023).
33. Tibshirani, R. Regression Shrinkage and Selection Via the Lasso. *J R Stat Soc Series B Stat Methodol* **58**, 267–288 (1996).
34. Jumper, J. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).
35. Tibshirani, R. *Regression Shrinkage and Selection via the Lasso*. *Source: Journal of the Royal Statistical Society. Series B (Methodological)* vol. 58 (1996).
36. Candes, E., Fan, Y., Janson, L. & Lv, J. Panning for Gold: Model-X Knockoffs for High-dimensional Controlled Variable Selection. (2016).
37. Barber, R. F. & Candés, E. J. Controlling the false discovery rate via knockoffs. *Ann Stat* **43**, 2055–2085 (2015).

38. Hastie, T., Tibshirani, R. & Wainwright, M. *Statistical Learning with Sparsity*. (Chapman and Hall/CRC, 2015). doi:10.1201/b18401.
39. Zou, H. The adaptive lasso and its oracle properties. *J Am Stat Assoc* **101**, 1418–1429 (2006).
40. Candes, E. J., Wakin, M. B. & Boyd, S. P. Enhancing Sparsity by Reweighted L1 Minimization. (2007).
41. Zhu, J. *et al.* A polynomial algorithm for best-subset selection problem. *PNAS* **117**, (2014).
42. Fan, J. & Lv, J. Sure Independence Screening for Ultrahigh Dimensional Feature Space. *J R Stat Soc Series B Stat Methodol* **70**, 849–911 (2008).
43. Barber, R. F., Candès, E. J., Ramdas, A. & Tibshirani, R. J. Predictive inference with the jackknife+. *The Annals of Statistics* **49**, (2021).
44. Angelopoulos, A. N. & Bates, S. A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification. (2021).
45. Bolón-Canedo, V., Sánchez-Maroño, N. & Alonso-Betanzos, A. A review of feature selection methods on synthetic data. *Knowl Inf Syst* **34**, 483–519 (2013).
46. Hastie, Trevor, Tibshirani, Robert, Friedman, J. *The Elements of Statistical Learning The Elements of Statistical LearningData Mining, Inference, and Prediction, Second Edition*. *Springer series in statistics* (2009).
47. Montáns, F. J., Chinesta, F., Gómez-Bombarelli, R. & Kutz, J. N. Data-driven modeling and learning in science and engineering. *Comptes Rendus - Mecanique* vol. 347 845–855 Preprint at https://doi.org/10.1016/j.crme.2019.11.009 (2019).
48. Breiman, L. *Statistical Modeling: The Two Cultures*. *Statistical Science* vol. 16 (2001).
49. Dhar, V. Data science and prediction. *Commun ACM* **56**, 64–73 (2013).
50. Lookman, T., Balachandran, P. V., Xue, D. & Yuan, R. Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design. *npj Computational Materials* vol. 5 Preprint at https://doi.org/10.1038/s41524-019-0153-8 (2019).
51. Adadi, A. A survey on data-efficient algorithms in big data era. *J Big Data* **8**, (2021).
52. Efron, B. Prediction, Estimation, and Attribution. *J Am Stat Assoc* **115**, 636–655 (2020).
53. Li, Y. *et al.* Large Data Set-Driven Machine Learning Models for Accurate Prediction of the Thermoelectric Figure of Merit. *ACS Appl Mater Interfaces* **14**, 55517–55527 (2022).
54. Arai, Y. *et al.* Using a machine learning algorithm to predict acute graft-versus-host disease following allogeneic transplantation. *Blood Adv* **3**, 3626–3634 (2019).
55. Khakzad, H. *et al.* A new age in protein design empowered by deep learning. *Cell Systems* vol. 14 925–939 Preprint at https://doi.org/10.1016/j.cels.2023.10.006 (2023).
56. Li, R., Li, L., Xu, Y. & Yang, J. Machine learning meets omics: Applications and perspectives. *Briefings in Bioinformatics* vol. 23 Preprint at https://doi.org/10.1093/bib/bbab460 (2022).
57. Bouallègue, Z. Ben *et al.* The Rise of Data-Driven Weather Forecasting A First Statistical Assessment of Machine Learning–Based Weather Forecasts in an Operational-Like Context. *Bull Am Meteorol Soc* **105**, E864–E883 (2024).
58. Kryshtafovych, A., Schwede, T., Topf, M., Fidelis, K. & Moult, J. Critical assessment of methods of protein structure prediction (CASP)—Round <scp>XIV</scp>. *Proteins: Structure, Function, and Bioinformatics* **89**, 1607–1617 (2021).

59. Ramalli, E. & Pernici, B. Challenges of a Data Ecosystem for scientific data. *Data Knowl Eng* **148**, (2023).
60. Avadhanula, S. *et al.* Assessment of Thyroid Function in Patients With Alkaptonuria. *JAMA Netw Open* **3**, E201357 (2020).
61. Nasonova, A., Levy, G. J., Rinot, O., Eshel, G. & Borisover, M. Organic matter in aqueous soil extracts: Prediction of compositional attributes from bulk soil mid-IR spectra using partial least square regressions. *Geoderma* **411**, (2022).
62. Del Pero, C., Aste, N., Leonforte, F. & Sfolcini, F. Long-term reliability of photovoltaic c-Si modules – A detailed assessment based on the first Italian BIPV project. *Solar Energy* **264**, (2023).
63. Caruana, E. J., Roman, M., Hernández-Sánchez, J. & Solli, P. Longitudinal studies. *J Thorac Dis* **7**, E537–E540 (2015).
64. Liu, T. *et al.* Applying high-performance computing in drug discovery and molecular simulation. *National Science Review* vol. 3 49–63 Preprint at https://doi.org/10.1093/nsr/nww003 (2016).
65. Mitani, A. A. & Haneuse, S. Small Data Challenges of Studying Rare Diseases. *JAMA Network Open* vol. 3 Preprint at https://doi.org/10.1001/jamanetworkopen.2020.1965 (2020).
66. Xu, P., Ji, X., Li, M. & Lu, W. Small data machine learning in materials science. *npj Computational Materials* vol. 9 Preprint at https://doi.org/10.1038/s41524-023-01000-z (2023).
67. Chen, R. J., Lu, M. Y., Chen, T. Y., Williamson, D. F. K. & Mahmood, F. Synthetic data in machine learning for medicine and healthcare. *Nature Biomedical Engineering* vol. 5 493–497 Preprint at https://doi.org/10.1038/s41551-021-00751-8 (2021).
68. Lee, J. W. *et al.* A data-driven XRD analysis protocol for phase identification and phase-fraction prediction of multiphase inorganic compounds. *Inorg Chem Front* **8**, 2492–2504 (2021).
69. Yamada, H. *et al.* Predicting Materials Properties with Little Data Using Shotgun Transfer Learning. *ACS Cent Sci* **5**, 1717–1730 (2019).
70. Shim, E. *et al.* Predicting reaction conditions from limited data through active transfer learning. *Chem Sci* **13**, 6655–6668 (2022).
71. Lansford, J. L., Barnes, B. C., Rice, B. M. & Jensen, K. F. Building Chemical Property Models for Energetic Materials from Small Datasets Using a Transfer Learning Approach. *J Chem Inf Model* **62**, 5397–5410 (2022).
72. De Breuck, P. P., Hautier, G. & Rignanese, G. M. Materials property prediction for limited datasets enabled by feature selection and joint learning with MODNet. *NPJ Comput Mater* **7**, (2021).
73. Gupta, V. *et al.* Cross-property deep transfer learning framework for enhanced predictive analytics on small materials data. *Nat Commun* **12**, (2021).
74. Candes, E. J. & Wakin, M. B. An introduction to compressive sampling: A sensing/sampling paradigm that goes against the common knowledge in data acquisition. *IEEE Signal Process Mag* **25**, 21–30 (2008).
75. Wainwright, M. J. Information-theoretic limits on sparsity recovery in the high-dimensional and noisy setting. *IEEE Trans Inf Theory* **55**, 5728–5741 (2009).

76. Wainwright, M. J. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ1-constrained quadratic programming (Lasso). *IEEE Trans Inf Theory* **55**, 2183–2202 (2009).
77. Zhao, P. & Edu, B. B. *On Model Selection Consistency of Lasso Bin Yu*. *Journal of Machine Learning Research* vol. 7 https://dl.acm.org/doi/10.5555/1248547.1248637 (2006).
78. Candès, E. & Romberg, J. Sparsity and incoherence in compressive sampling. *Inverse Probl* **23**, 969–985 (2007).
79. Lee, B. Do *et al.* Powder X-Ray Diffraction Pattern Is All You Need for Machine-Learning-Based Symmetry Identification and Property Prediction. *Advanced Intelligent Systems* **4**, (2022).
80. Suzuki, Y. *et al.* Symmetry prediction and knowledge discovery from X-ray diffraction patterns using an interpretable machine learning approach. *Sci Rep* **10**, 21790 (2020).
81. Menon, A. *et al.* Elucidating multi-physics interactions in suspensions for the design of polymeric dispersants: a hierarchical machine learning approach. *Mol Syst Des Eng* **2**, 263–273 (2017).
82. Pourhoseingholi, M. A., Baghestani, A. R. & Vahedi, M. *Gastroenterology and Hepatology From Bed to Bench*. *Gastroenterol Hepatol Bed Bench* vol. 5 (2012).
83. Kumar, V. Feature Selection: A literature Review. *The Smart Computing Review* **4**, (2014).
84. Li, G., Peng, H., Zhang, J. & Zhu, L. Robust rank correlation based screening. *The Annals of Statistics* **40**, (2012).
85. Vergara, J. R. & Estévez, P. A. A review of feature selection methods based on mutual information. *Neural Comput Appl* **24**, 175–186 (2014).
86. Cai, T. T. & Wang, L. Orthogonal Matching Pursuit for Sparse Signal Recovery With Noise. *IEEE Trans Inf Theory* **57**, 4680–4688 (2011).
87. Bertsimas, D., King, A. & Mazumder, R. Best subset selection via a modern optimization lens. *The Annals of Statistics* **44**, (2016).
88. Kursa, M. B. & Rudnicki, W. R. Feature Selection with the Boruta Package. *J Stat Softw* **36**, (2010).
89. Guo, Y., Zhu, Z. & Fan, J. Best subset selection is robust against design dependence. (2020).
90. Hazimeh, H. & Mazumder, R. Fast Best Subset Selection: Coordinate Descent and Local Combinatorial Optimization Algorithms. *Oper Res* **68**, 1517–1537 (2020).
91. Hazimeh, H., Mazumder, R. & Saab, A. Sparse regression at scale: branch-and-bound rooted in first-order optimization. *Math Program* **196**, 347–388 (2022).
92. Mazumder, R., Radchenko, P. & Dedieu, A. Subset Selection with Shrinkage: Sparse Linear Modeling When the SNR Is Low. *Oper Res* **71**, 129–147 (2023).
93. Soloff, J. A., Barber, R. F. & Willett, R. Bagging Provides Assumption-free Stability. (2023).
94. Romano, Y., Sesia, M. & Candès, E. Deep Knockoffs. *J Am Stat Assoc* **115**, 1861–1872 (2020).
95. Bates, S., Candès, E., Janson, L. & Wang, W. Metropolized Knockoff Sampling. *J Am Stat Assoc* **116**, 1413–1427 (2021).
96. Barber, R. F., Candès, E. J. & Samworth, R. J. Robust inference with knockoffs. *The Annals of Statistics* **48**, (2020).
97. Barber, R. F. & Candès, E. J. Controlling the false discovery rate via knockoffs. *The Annals of Statistics* **43**, (2015).

98. Gimenez, J. R., Ghorbani, A. & Zou, J. Knockoffs for the mass: new feature importance statistics with false discovery guarantees. (2018).
99. Spector, A. & Fithian, W. Asymptotically Optimal Knockoff Statistics via the Masked Likelihood Ratio. (2022).
100. Lu, Y. Y., Fan, Y., Lv, J. & Noble, W. S. DeepPINK: reproducible feature selection in deep neural networks. (2018).
101. Candès, E., Fan, Y., Janson, L. & Lv, J. Panning for Gold: 'Model-X' Knockoffs for High Dimensional Controlled Variable Selection. *J R Stat Soc Series B Stat Methodol* **80**, 551–577 (2018).
102. Ross, B. C. Mutual Information between Discrete and Continuous Data Sets. *PLoS One* **9**, e87357 (2014).
103. Kraskov, A., Stögbauer, H. & Grassberger, P. Estimating mutual information. *Phys Rev E* **69**, 066138 (2004).
104. Juarez-Perez, E. J., Hawash, Z., Raga, S. R., Ono, L. K. & Qi, Y. Thermal degradation of $CH_3NH_3PbI_3$ perovskite into NH_3 and CH_3I gases observed by coupled thermogravimetry–mass spectrometry analysis. *Energy Environ Sci* **9**, 3406–3410 (2016).
105. Dunlap-Shohl, W. A., Li, T. & Mitzi, D. B. Interfacial Effects during Rapid Lamination within $MAPbI_3$ Thin Films and Solar Cells. *ACS Appl Energy Mater* **2**, 5083–5093 (2019).
106. Thampy, S., Zhang, B., Hong, K. H., Cho, K. & Hsu, J. W. P. Altered Stability and Degradation Pathway of CH3NH3PbI3 in Contact with Metal Oxide. *ACS Energy Lett* **5**, 1147–1152 (2020).
107. Stoddard, R. J. *et al.* Forecasting the Decay of Hybrid Perovskite Performance Using Optical Transmittance or Reflected Dark-Field Imaging. *ACS Energy Lett* **5**, 946–954 (2020).
108. Stoddard, R. J., Eickemeyer, F. T., Katahara, J. K. & Hillhouse, H. W. Correlation between Photoluminescence and Carrier Transport and a Simple In Situ Passivation Method for High-Bandgap Hybrid Perovskites. *J Phys Chem Lett* **8**, 3289–3298 (2017).
109. Braly, I. L., Stoddard, R. J., Rajagopal, A., Jen, A. K.-Y.-. & Hillhouse, H. W. Photoluminescence and Photoconductivity to Assess Maximum Open-Circuit Voltage and Carrier Transport in Hybrid Perovskites and Other Photovoltaic Materials. *J Phys Chem Lett* **9**, 3779–3792 (2018).
110. Correa-Baena, J.-P. *et al.* Promises and challenges of perovskite solar cells. *Science (1979)* **358**, 739–744 (2017).
111. Schelhas, L. T. *et al.* Insights into operational stability and processing of halide perovskite active layers. *Energy Environ Sci* **12**, 1341–1348 (2019).
112. McMeekin, D. P. *et al.* A mixed-cation lead mixed-halide perovskite absorber for tandem solar cells. *Science (1979)* **351**, 151–155 (2016).
113. Yi, C. *et al.* Entropic stabilization of mixed A-cation ABX_3 metal halide perovskites for high performance perovskite solar cells. *Energy Environ Sci* **9**, 656–662 (2016).
114. Barrier, J. *et al.* Compositional heterogeneity in $Cs_yFA_{1-y}Pb(Br_xI_{1-x})_3$ perovskite films and its impact on phase behavior. *Energy Environ Sci* **14**, 6394–6405 (2021).
115. Marchezi, P. E. *et al.* Degradation mechanisms in mixed-cation and mixed-halide $Cs_xFA_{1-x}Pb(Br_yI_{1-y})_3$ perovskite films under ambient conditions. *J Mater Chem A Mater* **8**, 9302–9312 (2020).

116. Mundt, L. E. *et al.* Mixing Matters: Nanoscale Heterogeneity and Stability in Metal Halide Perovskite Solar Cells. *ACS Energy Lett* **7**, 471–480 (2022).
117. Macpherson, S. *et al.* Local nanoscale phase impurities are degradation sites in halide perovskites. *Nature* **607**, 294–300 (2022).
118. Thiesbrummel, J. *et al.* Ion-induced field screening as a dominant factor in perovskite solar cell operational stability. *Nat Energy* **9**, 664–676 (2024).
119. Erdil, U. *et al.* Mimicking Outdoor Ion Migration in Perovskite Solar Cells: A Forward Bias, No-Light Accelerated Aging Approach. *ACS Energy Lett* 1529–1537 (2025) doi:10.1021/acsenergylett.5c00376.
120. DeQuilettes, D. W. *et al.* Photo-induced halide redistribution in organic-inorganic perovskite films. *Nat Commun* **7**, (2016).
121. Deng, X. *et al.* Dynamic study of the light soaking effect on perovskite solar cells by in-situ photoluminescence microscopy. *Nano Energy* **46**, 356–364 (2018).
122. Massias, M., Gramfort, A. & Salmon, J. Celer: a Fast Solver for the Lasso with Dual Extrapolation. (2018).
123. Massias, M., Vaiter, S., Gramfort, A. & Salmon, J. Dual Extrapolation for Sparse Generalized Linear Models. (2019).
124. Spector, A. & Janson, L. Powerful Knockoffs via Minimizing Reconstructability. (2020).