



**Politecnico
di Torino**

ScuDo

Scuola di Dottorato - Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Computer Science (38th cycle)

Artificial Intelligence as an enabling technology for multiscale physical modelling

By

Mario Vozza

Supervisor(s):

Prof. Francesco Mercuri
Prof. C.D., Co-Supervisor
Prof. E.F., Co-Supervisor
Prof. G.H., Co-Supervisor

Doctoral Examination Committee:

Politecnico di Torino
2026

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Mario Vozza
2026

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

“All models are wrong, but some are useful.”

George E. P. Box

I dedicate this work to my brother, Antonio. Everyone who knows me understands how central you are to my life, and that without you none of this would have been possible. I also dedicate it to my parents, who made it possible for me to undertake this experience with stability and freedom, without having to constantly question its feasibility, allowing me to focus fully on research and personal growth.

Abstract

Modern engineered materials and technological systems exhibit strong couplings across multiple spatial and temporal scales, from atomistic structure to device-level behaviour and system-level functionality. Traditional multiscale modelling approaches, while physically rigorous, increasingly struggle to address the combined challenges of high dimensionality, computational cost, heterogeneous data representations and large design spaces. In this context, artificial intelligence offers powerful tools for surrogate modelling, representation learning and generative exploration, but its effective use requires principled integration with physical knowledge, structured data infrastructures and reproducible workflows.

This doctoral thesis develops and validates an AI-driven, data-centred multiscale modelling framework aimed at linking physical structure to functional prediction across heterogeneous domains. Rather than enforcing direct hierarchical coupling between physical models operating at different scales, the proposed approach treats data as the primary coupling layer, enabling information transfer through shared representations, automated workflows and machine learning models grounded in physical constraints.

The methodological framework is articulated through two complementary case studies. At the atomistic scale, an automated and FAIR-compliant workflow is developed for the generation of a large dataset of defective graphene nanostructures, including electronic, energetic and transport properties as well as virtual scanning tunnelling microscopy images. Gradient-boosted models and explainable AI techniques are employed to quantify structure–property correlations, assess model interpretability and identify the role of specific defect configurations. Generative models, including variational autoencoders and conditional diffusion processes, are further introduced to enable controlled exploration of the defect space aligned with target physical properties.

At the mesoscale, the thesis addresses device-level modelling of bulk heterojunction organic photovoltaic cells. Automated pipelines are constructed to generate three-dimensional morphologies and high-fidelity drift–diffusion simulations. Neural operator architectures are then employed as surrogate models to learn the underlying physical operators governing charge transport and current–voltage characteristics, significantly reducing computational cost while preserving predictive accuracy. Morphology clustering and generative strategies are used to explore the relationship between mesoscale structure and device performance.

Beyond the specific applications, the thesis proposes a unifying conceptual framework in which artificial intelligence acts as an integrative layer across scales, supported by automation, structured data management and Key Enabling Technologies. The results demonstrate that data-driven models can capture complex physical mappings, support interpretability and enable generative design when embedded within disciplined computational workflows. The framework developed in this work is designed to be transferable across domains characterised by multiscale physical complexity, offering a methodological foundation for future AI-driven modelling in materials science, energy devices and emerging technological infrastructures.

Contents

List of Figures	xii
------------------------	------------

List of Tables	xxii
-----------------------	-------------

1 Introduction	1
1.1 Context and Motivation	1
1.2 Main Objective and challenges	2
1.3 Scope and evolution of the research project	3
1.3.1 Research questions	4
1.4 Thesis structure	4
2 Physical Foundations: From Nanoscale Materials to Functional Devices	6
2.1 Introduction	6
2.2 Multiscale Modelling: From Classical to Modern Data-Driven Paradigms	8
2.3 Advances in Nanomaterials and Two-Dimensional Materials	10
2.3.1 Introduction to 1D and 2D structures	10
2.3.2 Fabrication method of nanomaterials	11
2.3.3 Engineering the graphene structure	13
2.4 Organic Photovoltaics and Bulk Heterojunction Devices	15
2.4.1 Fundamentals of organic photovoltaics	15
2.4.2 Bulk heterojunction architecture	17

2.4.3	Device scale modelling: the drift diffusion formalism	18
2.5	Data and automation foundation for computational material science	20
2.5.1	Automated simulation workflows	22
2.5.2	Methods for integrating heterogeneous data	23
2.6	Key Enabling Technologies	26
2.7	Conclusions	27
3	Machine Learning Foundations for Scientific Modelling	28
3.1	Introduction	28
3.1.1	Data-driven Modelling and Machine Learning Background .	29
3.1.2	Supervised vs Unsupervised Learning	30
3.1.3	Bias–variance tradeoff	31
3.2	Data Representations for Physical Systems	33
3.2.1	Structured and unstructured data	33
3.2.2	Feature Extraction and Latent Spaces	33
3.2.3	Dimensionality reduction and clustering overview	35
3.2.4	Convolutional Neural Networks and Object Detection Archi- tectures	37
3.2.5	Attention Mechanisms and Representation Interaction . . .	39
3.3	Gradient Boosting Models and Explainable AI	41
3.3.1	The intuition behind boosting	41
3.3.2	Decision Tree models	42
3.3.3	Extreme Gradient Boosting Algorithm	43
3.3.4	Explainability of Tree Models and Introduction to XAI . . .	45
3.4	Operator Learning for PDE Approximation	47
3.4.1	Scientific Machine Learning (SciML)	47
3.4.2	Surrogate Modelling for Physical Simulations	48

3.4.3	Fourier Neural Operator	49
3.5	Generative Modelling for Materials and Morphology Design	52
3.5.1	Variational autoencoders	53
3.5.2	Diffusion Models	55
3.5.3	Conditioning the Generative Process	60
3.6	Computational material representation	62
3.6.1	Atomistic Representation	62
3.6.2	Molecular Representation	63
3.7	Conclusion	64
4	The conceptual bridge	65
4.1	Why a Conceptual Bridge Is Needed	65
4.2	From Physical State to Informational Representation	66
4.3	Data as the Multiscale Coupling Layer	68
4.4	Key Enabling Technologies as Structural Connectors	71
4.5	Learning and Optimisation Across Scales	72
4.6	Chapter Synthesis and Transition to Case Studies	74
5	Learning the Device Physics of Organic Photovoltaics: Neural Operators, Morphology Clustering and Generative Modelling	76
5.1	The importance of organic devices in smart cities	76
5.2	Solving drift-diffusion equations with operator learning	77
5.3	Automation frameworks for device simulation	79
5.3.1	Morphology Generation for Bulk Heterojunction	80
5.3.2	Cahn-Hilliard Modelling of Phase Separation	81
5.3.3	Simulation Setup	83
5.3.4	Selection of Representative Snapshots along the J–V Curve	86
5.4	Proposed Solution	88

5.5	Results and Discussion	91
5.6	Introduction to the selection of promising morphology spaces	97
6	Atomistic Graphene Data to Generative Models: An AI Framework for Structure–Property Correlations	102
6.1	Introduction	102
6.2	Graphene as an enabling material for the city of the future	103
6.3	Automated dataset creation workflows	104
6.3.1	First steps into dataset generation	104
6.3.2	Electronic properties calculation	106
6.3.3	STM Virtual Replica	108
6.3.4	Electron Transport Simulation	110
6.3.5	STM images calculation	112
6.3.6	Data encoding and FAIR workflow	114
6.3.7	Data Validation	117
6.4	Defect–property correlations with XGBoost	121
6.4.1	General Overview	121
6.4.2	The Process of Image Generation	122
6.4.3	Defect extraction process	124
6.4.4	Feature extraction	126
6.4.5	Model Training	132
6.4.6	Results Discussion	136
6.4.7	Model Comparison with LightGBM	141
6.5	From uncontrolled to conditional generation	144
6.5.1	Motivation	144
6.5.2	Workflow	145
6.5.3	First Case: Model architecture and Training	146

6.5.4	Second Case: Conditioning with specific features	153
6.5.5	The Conditioning Workflow	154
6.6	Conclusion	160
7	Conclusions and Perspectives	163
7.1	Contributions, limitations and scope	163
7.2	Future research directions	166
7.2.1	Generative and integrative perspectives across scales	166
	References	168
	Appendix A General Background Appendix	183
A.1	Solve the Chan-Hilliard with Spectral Methods	183
A.2	Tucker Decomposition	186
A.3	Density Functional Theory	188
A.4	Density Functional Tight-Binding	189
A.5	Itô Stochastic Differential Equations	190
	Appendix B Supporting information Appendix	193
B.1	Atomistic Graphene Data to Generative Models: An AI Framework for Structure Properties Correlation	194
B.2	Learning the Device Physics of Organic Photovoltaics: Neural Oper- ators, Morphology Clustering and Generative Modelling	213

List of Figures

- 2.1 The multiscale paradigm for nanomaterials, illustrated through representative examples across length and time scales. The horizontal axis denotes the physical scale, while the vertical axis represents the characteristic time scale. 8
- 2.2 STM images of chemically reduced graphene oxide nanosheets. (a) Nanometre-scale STM image showing a smooth and straight nanosheet edge. (b) Atomic-resolution image of an individual nanosheet edge. (c) Atomic-resolution STM current image of a nanosheet corner, with panel (d) reporting the line profile extracted along the white trace shown in (c) 12
- 2.3 Graphene is a two-dimensional material and forms the fundamental building block for carbon structures of all dimensionalities. Starting from the bottom left, the figure shows the zero-dimensional fullerene, the one-dimensional carbon nanotube and three-dimensional graphite. 13
- 2.4 Schematic representation of structural defects commonly found in graphene, including (a) a Stone–Wales defect, (b) a 585 divacancy, and (c) a 555–777 divacancy. Symmetry axes are indicated by red dashed lines. 15
- 2.5 Schematic representation of the ADES model. The four pillars of Automation, Data, Environment and Sharing together define the conceptual and technological foundations required for high throughput, reproducible and community oriented computational materials science. 22

3.1	Decomposition of the generalisation error into squared bias, variance and irreducible noise as a function of model complexity. The total error exhibits a characteristic U-shape, with an optimal region between underfitting and overfitting.	32
3.2	Schematic representation of an auto-encoder network	35
3.3	Comparison of dimensionality reduction techniques applied to a non linear manifold. Top left: synthetically generated three dimensional sphere dataset illustrating an intrinsically two dimensional surface embedded in a higher dimensional ambient space. Top right: linear projection obtained through principal component analysis (PCA), which preserves global variance but cannot recover the curved geometry of the manifold. Bottom left: t distributed stochastic neighbour embedding (t-SNE), which preserves local neighbourhoods and therefore produces separated regions that reflect local structure rather than global topology. Bottom right: Uniform Manifold Approximation and Projection (UMAP), which balances local connectivity and global organisation to produce a continuous two dimensional embedding that more faithfully reflects the structure of the original manifold.	36
3.4	Illustration of the decision boundaries induced by a decision tree classifier trained on a two-dimensional synthetic dataset. The model partitions the feature space into axis-aligned rectangular regions, each associated with a constant prediction. This piecewise structure reflects the hierarchical sequence of thresholding operations performed at the internal nodes of the tree. Decision trees therefore produce interpretable and locally uniform regions whose geometry provides insight into how feature values drive classification.	43
3.5	Illustration of the additive nature of gradient boosting on a one-dimensional regression task. Each panel shows the cumulative model after successively adding weak learners f_1, f_2, f_3 , each trained to correct the residuals of the previous stage. The final boosted model approximates the underlying target function by summing a sequence of shallow regression trees, demonstrating the core weak-to-strong principle of boosting.	44

3.6	Fourier Neural Operator architecture representation.	50
3.7	Schematic architecture of a Variational Autoencoder (VAE). The encoder maps the input data distribution (left) into a low-dimensional latent representation, parameterised probabilistically. Latent samples drawn from this representation are then passed to the decoder, which reconstructs the data in the original space (right). The stochastic latent space enables smooth interpolation and generative sampling while enforcing a regularised representation of the input distribution.	53
3.8	Forward diffusion applied to an image. The sequence $x_0, x_1, \dots, x_T$ demonstrates the progressive attenuation of the signal as Gaussian noise with variance β_t ; that follow a linear schedule across the diffusion steps; is added at each step. The visible structure is rapidly lost, highlighting the tendency of the forward process to converge towards an isotropic Gaussian distribution. The diffusion procedure was implemented by the author.	56
3.9	Representation of benzene (C_6H_6) molecule (left) and sorted Coulomb matrix (right). Sorting in the coulomb matrix is essential for the invariance preservation. The calculation procedure was implemented by the author.	63
5.1	Computational time and C costant of the drift diffusion solver for grids with size 2^N	79
5.2	A) Automatic workflow for morphology generation based on Perlin noise patterns. B) Simulation workflow using a parallel pool executor and systematic dataset storage.	80
5.3	The image on the left shows a morphology obtained using a fixed seed of 22 and the parameters listed in Table A.1	83
5.4	a) Chemical structure of Y6 and PM6, materials of acceptor and donor in our simulations b) Representation of the simulation setup	84
5.5	The figure shows the full J-V curve plotted using all the snapshots provided by the simulator, alongside the approximated J-V curve obtained from the selected snapshots (red points).	87

- 5.6 Architecture of the network used in this study. In particular, we can observe that it is a two-head network: the green branch, starting from the generated morphologies, predicts the target fields, while the yellow branch concatenates the morphologies with the predictions and, through attention pooling, helps predict the 5 snapshots of the J-V curve, which are subsequently approximated. 89
- 5.7 Comparison between simulated and predicted hole current density J_p for a representative three-dimensional morphology. The top panel shows the input morphology, while the bottom panels report the simulated reference field, the neural-operator prediction and the spatial distribution of the normalised squared relative error (SRE). . 92
- 5.8 Comparison between simulated and predicted Shockley–Read–Hall (SRH) recombination rate $R_{\text{SRH}, n_{\text{free}} \rightarrow p_{\text{trap}}}$ for a representative three-dimensional morphology. The top panel shows the input morphology, while the bottom panels report the simulated reference field, the neural-operator prediction and the spatial distribution of the normalised squared relative error (SRE). 93
- 5.9 Comparison between simulated and predicted Hole Mobility μ_p for a representative three-dimensional morphology. The top panel shows the input morphology, while the bottom panels report the simulated reference field, the neural-operator prediction and the spatial distribution of the normalised squared relative error (SRE). . 94
- 5.10 Current–voltage (J–V) characteristic for the selected morphology. The solid line denotes the reference J–V curve obtained from the numerical simulation, while the dashed curve represents the model prediction. The predicted curve is reconstructed from the five voltage–current points inferred by the scalar prediction branch in a single forward pass, with a piecewise sigmoid interpolation applied to obtain a continuous J–V characteristic. The close agreement between the two curves demonstrates the ability of the model to accurately recover global device-level behaviour from morphology-dependent inputs. 95

5.11	Tortuosity extracted from a random sample in the dataset. The starting points are shown in green, the goal points in yellow, while the two extracted tortuosity paths are highlighted as the median path (blue) and the shortest path (red). The small grey dots represent the acceptor sites within the intermediate layer.	99
5.12	Cluster-wise projection of morphology and tortuosity descriptors in the two-dimensional PCA space. Each panel highlights a single cluster, while samples belonging to other clusters are shown in grey. Points within the active cluster are coloured according to discretised Fill Factor ranges, normalised by cluster size. The annotations report the cluster population and the relative distribution of low-, medium-, and high-Fill-Factor samples.	100
6.1	a) Scheme of the workflow for the generation of the HDF5 dataset; b) Distribution and corresponding kernel density estimates (KDE) of the number of atoms in the reference dataset with respect to the complete starting dataset; c) Example of a procedure for the pre-processing of the original DG structures.	105
6.2	a) Scheme of the process used to generate models for transport calculations, attaching source (red) and drain (blue) electrodes to DG samples; b) Example STM image of a DG flake, plotted with the <i>hot</i> colormap; c) Structure of the HDF5 file used to store the datasets. Data groups are represented in green, while the datasets, along with their respective attributes, are represented in red.	110
6.3	Scatter plot of the computed transport current as a function of the percentage of missing atoms in each graphene flake	111
6.4	Map of the Mulliken charges for a selected DG sample computed with the a) SCC and b) non-SCC approximations.	119
6.5	Map of the Mulliken charges for a selected DG sample computed with the a) SCC and b) non-SCC approximations.	120
6.6	Scheme of the prediction workflow applied to predict electronic and transport properties of DG samples from morphological features of defects.	121

6.7	Proposed image-based representation for defected graphene flakes .	123
6.8	Training and validation metrics for the YOLO model. The plots show the evolution of box loss (a), classification loss (b), and distribution focal loss (DFL) during training and validation (c), alongside the mAP50-mAP50-95 (d), precision (e), and recall (f) metrics. The consistent decline in losses and the increase in performance metrics indicate successful convergence of the model and robust detection performance.	125
6.9	Magnitude spectra of a graphene flake containing a single defect translated across different spatial positions. Each column corresponds to a specific translation (e.g., left, right, up, down), while the lower row shows the difference with respect to a reference configuration. The colour map highlights variations induced by defect translation, demonstrating that Fourier features are sensitive to symmetry breaking induced by defect translation, thereby indirectly encoding positional information.	131
6.10	The plots show the boosting rounds compared to RMSE errors for the train set and test set using XGBoost for Fermi level (a), ionization potential (b), electron affinity (c), band gap (d), energy per atom (e), and current (f).	135
6.11	Performance evaluation of the YOLOv10 model for defect detection in graphene structures. (a) Confusion matrix showing true positives, false positives, and false negatives for defect classification. (b) Precision curve illustrating the model's precision across varying confidence thresholds. (c) Recall curve demonstrating the model's ability to capture the majority of actual defects. (d) F1-score curve highlighting the balance between precision and recall, with the model achieving high reliability and accuracy in defect detection.	136
6.12	Defects detected in 4 different nanographene samples by the trained YOLOv10 model, along with their respective confidence values. . .	137

6.13	Scatter plots show the correlation between the predicted and true values in the test set using XGBoost for electron affinity (a), ionization potential (b), band gap (c), Fermi level (d), energy per atom (e), and current (f). The lines (orange) represent the ideal $y = x$ correlation. Marginal histograms display the distribution of predicted and true values for each property, indicating the performance of the predictive model across different datasets.	139
6.14	SHAP value summary for the current target	140
6.15	Comparison of the top 10 features based on permutation importance. XGBoost results are shown in blue, while LightGBM results are shown in orange.	143
6.16	Samples of graphene with defects image used in the training of VAEs	145
6.17	Workflow for the generation of synthetic datasets containing graphene nanoflakes with small distributed defect	147
6.18	Train and validation loss for the proposed architecture	149
6.19	Example of a) not valid structure and b) valid structure.	151
6.20	a) Learning curves showing training and validation loss over epochs; b) Confusion matrix obtained from evaluating the trained binary classifier on the test set.	152
6.21	Comparison between the original dataset (OD) and the synthetically generated dataset (GD). (a) Distribution of atom counts. Frequency-domain analysis of the nanoflakes, including (b) maximum frequency, (c) entropy, and (d) frequency contrast.	153
6.22	Clusters of the training samples used in Section 6.4. Features are projected into a two-dimensional PCA space, with kernel density estimation (KDE) indicating the density of training samples. Cluster boundaries are approximated via convex hulls. Coloured points denote synthetically generated samples, conditioned on the high-current cluster and coloured according to their predicted current values.	156

6.23	Different diffusion steps of the conditional generation process, illustrating the reverse trajectory from the initial noise state to the final synthesised sample.	159
6.24	Discrete Bernoulli diffusion framework for conditional binary structure generation. The top panel illustrates the forward diffusion process, in which a clean binary sample x_0 is progressively corrupted through discrete Bernoulli pixel replacement, producing increasingly noisy states x_t until pure Bernoulli noise x_T is reached. The central panel depicts the training phase: given a noised sample x_t , a conditioning vector, and the corresponding timestep embedding, a FiLM-conditioned U-Net predicts the logits of the clean binary image x_0 . The model is optimised using a binary cross-entropy reconstruction loss combined with a lattice regularisation term enforcing local structural consistency. The bottom panel shows the reverse (generative) process. Starting from random Bernoulli noise x_T , the U-Net iteratively predicts logits for the clean image, which are explicitly sampled via a Bernoulli distribution to obtain a candidate clean estimate $\hat{x}_0$. This estimate is then stochastically re-corrupted according to the forward diffusion rule to produce x_{t-1} , ensuring consistency with the discrete diffusion dynamics. The procedure is repeated for $t = T \rightarrow 1$, yielding the final generated binary sample at $t = 0$	162
B.1	Residual errors for the predictions of: a) electron affinity, b) ionization potential, c) band gap, d) Fermi level, e) energy per atom, and f) current using the XGBoost model.	196
B.2	SHAP value summary trained on XGBoost for the bandgap target .	197
B.3	SHAP value summary trained on XGBoost for the electron affinity target	198
B.4	SHAP value summary trained on XGBoost for the energy per atom target	199
B.5	SHAP value summary trained on XGBoost for the Fermi level target	200

B.6	SHAP value summary trained on XGBoost for the Ionisation potential level target	201
B.7	The plots show the boosting rounds compared to RMSE errors for the train set and test set using LightGBM for Fermi level (a), ionization potential (b), electron affinity (c), band gap (d), energy per atom (e), and current (f).	202
B.8	Scatter plots show the correlation between the predicted and true values in the test set using LightGBM for electron affinity (a), ionization potential (b), band gap (c), Fermi level (d), energy per atom (e), and current (f). The lines (orange) represent the ideal $y = x$ correlation. Marginal histograms display the distribution of predicted and true values for each property, indicating the performance of the predictive model across different datasets	203
B.9	Residual errors for the predictions of: a) electron affinity, b) ionization potential, c) band gap, d) Fermi level, e) energy per atom, and f) current using the LightGBM model.	204
B.10	XGBoost feature importance visualised as a heatmap. Each column corresponds to a different target property, while each row represents a distinct feature.	205
B.11	LightGBM feature importance visualised as a heatmap. Each column corresponds to a different target property, while each row represents a distinct feature.	206
B.12	SHAP value summary trained on LightGBM for the electron affinity target	207
B.13	SHAP value summary trained on LightGBM for the energy per atom target	208
B.14	SHAP value summary trained on LightGBM for the Fermi level target	209
B.15	SHAP value summary trained on LightGBM for the Ionisation potential level target	210
B.16	SHAP value summary trained on LightGBM for the bandgap target .	211
B.17	SHAP value summary trained on LightGBM for the current target .	212

B.18 Training vs validation curve for the E_c Target	213
B.19 Training vs validation curve for the J_p Target	214
B.20 Training vs validation curve for the μ_n Target	214
B.21 Training vs validation curve for the μ_p Target	215
B.22 Training vs validation curve for the $Q_{nfree\&nttrap}$ Target	215
B.23 Training vs validation curve for the Q_{nfree} Target	216
B.24 Training vs validation curve for the Q_{nttrap} Target	216
B.25 Training vs validation curve for the Q_{pfree} Target	217
B.26 Training vs validation curve for the Q_{ptrap} Target	217
B.27 Training vs validation curve for the Recombination rate from n_{free} to p_{trap} Target	218
B.28 Training vs validation curve for the Recombination rate from p_{free} to n_{trap} Target	218
B.29 Comparison between simulated and predicted E_c for a representative three-dimensional morphology.	219
B.30 Comparison between simulated and predicted μ_n for a representative three-dimensional morphology.	219
B.31 Comparison between simulated and predicted n_{free} for a representa- tive three-dimensional morphology.	220
B.32 Comparison between simulated and predicted n_{trap} for a representa- tive three-dimensional morphology.	220
B.33 Comparison between simulated and predicted p_{free} for a representa- tive three-dimensional morphology.	221
B.34 Comparison between simulated and predicted p_{trap} for a representa- tive three-dimensional morphology.	221
B.35 Comparison between simulated and predicted $R_{srh_pfree_nttrap}$ for a representative three-dimensional morphology.	222

List of Tables

5.1	Initial guesses and bounds for the piecewise sigmoid fitting parameters.	88
5.2	Hyperparameters of the TFNO3d network with scalar head.	90
5.3	Combined Spatial and Scalar errors statistics	97
6.1	Properties included in the HDF5 dataset. Each property is denoted by a symbol, which includes its units and dimensions (n_A is the number of atoms in the DG sample; n_E is the number of atoms in the sample including the electrodes), and can be located within the HDF5 files using the corresponding HDF5 keys. Properties are categorized into distinct types: structural (S), molecular (M), atom-in-a-molecule (A), ground-state (G), and response (R).	115
6.2	Root-level metadata attributes contained in the dataset file, with their descriptions.	117
6.3	Dataset properties analysis in terms of Mean, STD, Max, Min, Median, 3rd Moment (Skewness) and 4th Moment (Kurtosis).	118
6.4	XGBoost parameters optimized using the Optuna framework	133
6.5	Performance metrics for the XGBoost model across different target properties.	138
6.6	Performance metrics for selected targets using the LightGBM model.	142
A.1	Simulation parameters used for the Cahn-Hilliard equation	185
B.1	Ranges of XGBoost parameters used in Optuna optimization.	194

B.2	Percentage difference between the error metrics of XGBoost and LightGBM. Positive values indicate that XGBoost yields lower error metrics (e.g., MSE and MAE) compared to LightGBM, whereas negative values indicate that XGBoost achieves higher values in metrics such as R^2 , relative to LightGBM.	195
-----	---	-----

Chapter 1

Introduction

1.1 Context and Motivation

Over the past two decades, the increasing complexity of engineered materials and technological systems has exposed the structural limitations of traditional modelling paradigms. Advances in nanomaterials, energy devices and digital infrastructures have enabled unprecedented levels of functional performance, yet they have also introduced strong coupling across spatial and temporal scales that cannot be addressed through isolated or monolithic simulation approaches.

At the atomistic scale, quantum-mechanical descriptions are required to capture electronic structure, defect formation and transport phenomena. At the device scale, mesoscale models govern charge transport, morphology evolution and collective effects emerging from heterogeneous material arrangements. At the system level, these devices operate within broader technological ecosystems, where performance, reliability and optimisation are influenced by data flows, automation pipelines and infrastructure-level constraints. While each of these scales has traditionally been studied in relative isolation, contemporary applications increasingly demand a unified perspective.

Classical multiscale modelling strategies typically rely on hierarchical coupling or parameter passing between models operating at different resolutions. Although powerful, such approaches become increasingly fragile as system complexity grows, particularly when large design spaces, stochastic variability and high-dimensional outputs are involved. In this context, the bottleneck is no longer the formulation of

physical models alone, but the ability to integrate, manage and exploit heterogeneous data produced across scales.

Artificial intelligence, and in particular data-driven modelling, has emerged as a viable response to these challenges. Machine learning methods enable the construction of surrogate models that approximate expensive simulations, extract latent structure from high-dimensional data and support inverse design and generative exploration. However, the effective deployment of AI in scientific contexts requires more than algorithmic performance: it demands structured data representations, reproducible workflows and a principled integration with physical knowledge.

This thesis is motivated by the need to move beyond isolated applications of AI toward a coherent, data-centred multiscale framework. Rather than treating artificial intelligence as a replacement for physics-based modelling, the work explores how AI can act as an integrative layer, enabling information transfer across atomistic, mesoscale and system-level domains through shared representations and computational principles.

1.2 Main Objective and challenges

The primary objective of this doctoral research is to develop and validate an AI-driven multiscale modelling framework capable of linking physical structure to functional prediction across heterogeneous domains. The focus is not on a single material system or modelling technique, but on the construction of a generalisable workflow that integrates simulation, data management and learning.

To this end, the thesis pursues three tightly coupled goals. First, it aims to design automated pipelines for the generation of high-quality, physics-consistent datasets at different scales, ranging from atomistic simulations of defective graphene to mesoscale device-level simulations of organic photovoltaic morphologies. Second, it seeks to develop machine learning models, including neural operators and gradient-boosted methods, that can learn complex physical mappings while preserving interpretability and physical relevance. Third, it investigates how these components can be integrated into a unified methodological framework that supports scalability, reproducibility and cross-domain transfer.

Several challenges arise in pursuing these objectives. From a modelling perspective, the systems under consideration exhibit high dimensionality, strong nonlinearities and sensitivity to structural variability. From a data perspective, the resulting datasets are heterogeneous, combining scalar, vectorial and field-based quantities with different resolutions and semantic meanings. From a methodological standpoint, ensuring that AI models remain grounded in physical constraints while retaining flexibility and generalisation capability is non-trivial.

Addressing these challenges requires a careful balance between physical modelling, data engineering and machine learning design. The thesis therefore adopts a data-centred view, in which the structure, provenance and interoperability of data are treated as first-class elements of the modelling process.

1.3 Scope and evolution of the research project

The doctoral project was originally framed within the context of automation frameworks for transportation systems and logistics. During the course of the research, however, the focus progressively shifted from a domain-specific perspective to a more fundamental methodological investigation. This evolution was driven by the recognition that many of the critical challenges encountered in transportation and logistics systems, including scale heterogeneity, computational intractability, data fragmentation and model interoperability are not specific to a single application domain. Rather, they are structural properties of complex technological systems. To address these challenges in a controlled and scientifically rigorous manner, the research adopted advanced materials and device-level systems as primary case studies. These systems provide a tractable yet representative setting in which multi-scale modelling, automation, and data-driven integration can be systematically developed and validated. The resulting methodological framework is therefore not confined to materials science, nor to transportation and logistics, but is designed to be transferable across domains in which complex physical processes, data-driven abstraction and system-level decision-making must be coherently integrated. Within this perspective, materials and devices act as enabling substrates through which broader technological and infrastructural systems, including smart cities, can be analysed and designed.

1.3.1 Research questions

The work presented in this thesis is guided by a set of interconnected research questions:

- How can heterogeneous physical simulations across atomistic, mesoscale and system-level domains be represented within a unified, data-driven framework?
- To what extent can machine learning models, particularly neural operators and surrogate models, capture the underlying physical operators governing complex material and device behaviour?
- How can automated workflows and FAIR-compliant data infrastructures enable reproducible and scalable multiscale modelling?
- What are the limits of data-driven approaches in preserving physical interpretability, and how can explainability techniques mitigate these limitations?
- Can conditional generative modelling be employed to steer material and morphological design towards desired target properties, particularly when such targets are defined by higher-scale functional requirements?

These questions are addressed through a combination of methodological development, case studies and critical evaluation, with the aim of extracting transferable insights rather than system-specific solutions.

1.4 Thesis structure

The structure of this thesis has been designed to reflect a coherent progression across the different scales at which modern materials and device science operate. Rather than presenting the atomistic, mesoscale and system level components as isolated case studies, the thesis develops a unified narrative in which physical modelling, automated data generation and artificial intelligence form an integrated methodological framework. This perspective supports a multiscale interpretation of materials and device behaviour, in which information flows not through direct numerical coupling but through structured data representations and shared computational principles.

Chapter 2 introduces the physical and technological foundations of the work. It provides an overview of the relevant materials systems, including defective graphene and bulk heterojunction organic photovoltaic devices, together with the modelling principles that govern their behaviour at different scales. The chapter also presents the methodological basis for data centred research, covering automated workflows, structured data infrastructures and the broader context of Key Enabling Technologies.

Chapter 3 develops the conceptual and technical foundations of artificial intelligence used throughout the thesis. It outlines the principles of representation learning, neural operators and generative models, placing emphasis on their role in modelling complex physical systems and in enabling the transition from physics based to data driven methodologies.

Chapter 4 acts as a conceptual bridge between physical modelling and artificial intelligence. It examines how data integration, workflow automation and AI based surrogate modelling allow the passage from atomistic descriptions to mesoscale device behaviour and system level considerations. This chapter provides the overarching framework that connects the different scales explored in the thesis. Chapter 5 examines the mesoscale domain and develops neural operator models for device level simulations of bulk heterojunction organic photovoltaics. The chapter discusses morphology clustering, surrogate modelling of drift diffusion simulations and generative approaches for exploring the space of device morphologies.

Chapter 6 focuses on the atomistic domain and presents the AI driven modelling of defective graphene. It introduces the automated construction of the graphene dataset, the associated FAIR aligned data infrastructure and the machine learning models employed for predicting structural and electronic properties as well as for generative exploration of defect configurations.

Chapter 7 concludes the thesis by synthesising the contributions across the different scales and discussing the implications of data centred modelling for materials, devices and emerging technological infrastructures. It also outlines possible directions for extending the multiscale and AI oriented framework introduced in this work.

Chapter 2

Physical Foundations: From Nanoscale Materials to Functional Devices

2.1 Introduction

The purpose of this chapter is to establish the scientific and technological foundations that underpin the remainder of this thesis. While the following chapters will explore advanced machine-learning methodologies and their application to complex physical systems, the present discussion focuses on the physical domains themselves: advanced materials at the nanoscale, organic optoelectronic devices at the mesoscale, and the technological landscape into which such innovations are increasingly deployed. The intent is not to provide an exhaustive survey of each field, but rather to outline the essential concepts, mechanisms and challenges that motivate the need for data-driven modelling and automated computational frameworks.

Modern engineering systems, from energy-harvesting devices to large-scale digital infrastructures, rely heavily on materials whose behavior is governed by structural features spanning several orders of magnitude. At the atomic level, the emergence of two-dimensional materials has transformed our understanding of structure–property relationships. Graphene, in particular, exemplifies how local atomic arrangements, defect configurations, and nanoscale topology determine macroscopic electrical and mechanical behavior. Capturing and predicting such effects requires a combination

of accurate atomistic simulations and principled data structures capable of expressing high-dimensional physical information.

At the device scale, the development of high-performance organic solar cells illustrates a parallel challenge. Bulk-heterojunction (BHJ) morphologies exhibit complex, interpenetrating donor–acceptor networks whose spatial organisation shapes exciton dissociation, charge transport, and overall device efficiency. The inherently multiscale nature of these phenomena, coupled with the high cost of physics-based simulation workflows, makes BHJ systems a compelling domain for methodological innovation in modelling, automation and surrogate learning. Understanding the physical context of these devices is therefore essential before introducing the computational strategies employed later in this thesis.

Beyond materials and devices, contemporary technological ecosystems increasingly depend on digitally interconnected infrastructures, particularly in mobility, energy distribution and smart cities. These large-scale systems place stringent demands on efficiency, resilience and sustainability. Their evolution is driven in part by advances in enabling technologies, including nanomaterials, high-performance computing and artificial intelligence. While this thesis does not focus on smart-city optimisation per se, the broader context is relevant: the same technological forces that underpin modern urban infrastructures also shape the research agenda in materials science and device engineering. This motivates a unified perspective grounded in European Key Enabling Technologies (KETs), which offer a coherent conceptual lens through which innovations in materials, devices and automated computational frameworks can be understood.

In this chapter, we therefore examine three interconnected domains. We begin with the fundamental principles and structural characteristics of nanomaterials, with emphasis on two-dimensional systems such as graphene. We then discuss the physical mechanisms underlying organic photovoltaic devices, focusing on morphology formation and charge-transport processes that are central to BHJ operation. Finally, we situate these scientific developments within the wider technological landscape, highlighting how advanced materials and automated computational approaches contribute to the digital and energy infrastructures that define modern societies.

2.2 Multiscale Modelling: From Classical to Modern Data-Driven Paradigms

Physical, chemical and engineering systems rarely operate at a single characteristic scale. Their behaviour emerges from interactions extending from the atomic structure of matter to macroscopic device performance and system-level functionality. Classical multiscale modelling frameworks were developed to address this complexity by coupling different physical descriptions in a hierarchical manner, enabling information to flow from fine-grained levels such as quantum mechanics or molecular dynamics to coarse-grained continuum representations [1].

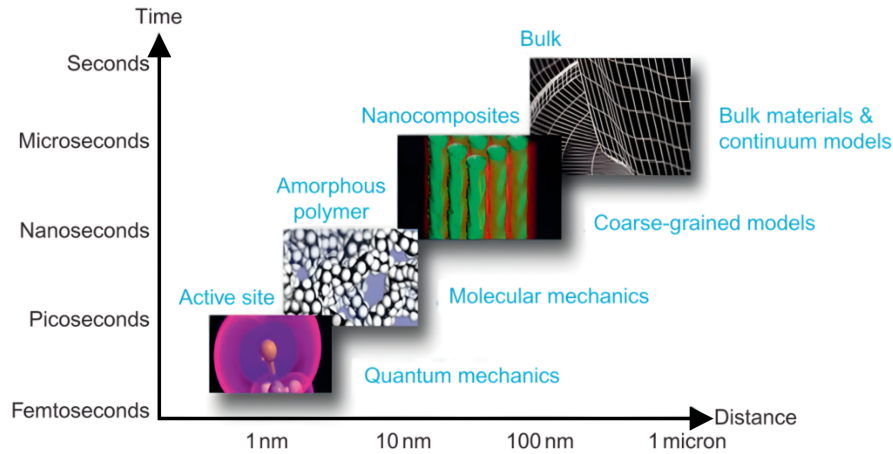


Fig. 2.1 The multiscale paradigm for nanomaterials, illustrated through representative examples across length and time scales. The horizontal axis denotes the physical scale, while the vertical axis represents the characteristic time scale. Image is reproduced from the work of Fitzgerald et al. [2]

Traditional multiscale strategies are typically categorised into three families. Bottom-up approaches begin with atomistic descriptions and propagate their effects upwards, using coarse-graining or homogenisation techniques to derive effective models at larger scales. Top-down methods start from macroscopic governing laws and introduce microscale corrections only where required. Bridging or concurrent methods combine simulations at two or more scales, allowing localised fine-scale computations to inform global behaviour. These methodologies have enabled significant advances in materials science, structural engineering, soft matter physics and fluid mechanics. Despite their success, classical multiscale workflows face well-

known challenges. Coupling models across different scales requires strict assumptions about scale separation and transferability of physical quantities. High-fidelity models at fine scales are computationally expensive, making iterative coupling prohibitive for large design spaces. Moreover, deriving consistent interfaces between scales is difficult when the underlying physics changes qualitatively from quantum effects to statistical mechanics to continuum transport [3].

In recent years, the expansion of data driven scientific computing has reshaped the way multiscale modelling is conceptualised. Classical multiscale frameworks emphasised the direct transfer of physical information across distinct levels of description, often linking quantum mechanical calculations with atomistic simulations, mesoscale formulations and continuum theories. While this hierarchical view remains central in many areas of physics and engineering, its practical implementation is frequently limited by the high computational cost of fine scale models, the difficulty of defining interfaces between disparate physical regimes and the restrictive assumptions required for rigorous scale separation.

Modern approaches have introduced a broader and more flexible interpretation of the multiscale concept. Rather than relying on explicit coupling between models, contemporary scientific practice increasingly incorporates machine learning as a common computational layer that supports the analysis of systems operating at different resolutions. Data driven representations have proved effective in capturing complex structure property relationships, encoding the essential geometric and chemical information of atomistic configurations [4], and providing accurate surrogate models for computationally intensive simulations across a wide range of physical domains. Recent reviews have highlighted how machine learning can facilitate multiscale reasoning by offering tools for efficient homogenisation, reduced order modelling and exploration of high dimensional design spaces, even when no deterministic exchange of information occurs between scales [5].

This broader understanding aligns closely with the methodological perspective adopted in this thesis. The two systems studied here, defective graphene at the atomistic level and bulk heterojunction organic photovoltaics at the mesoscale, do not form a classical multiscale chain. They are governed by distinct physical processes and simulated through independent numerical frameworks, and no direct coupling is enforced between their respective models. Nevertheless, both domains present challenges that are characteristic of multiscale scientific problems: structural

complexity that spans many degrees of freedom, high fidelity simulation workflows that are often too expensive for exhaustive exploration and parameter spaces that require principled methods for representation, prediction and design.

What unifies these systems is therefore not a flow of physical information from one scale to another, but a shared computational methodology. Automated data generation, the use of FAIR principles to ensure transparency and reproducibility, the development of machine learning surrogate models such as operator based approximators and the adoption of generative approaches for design space exploration all provide a common framework for addressing scientific questions at different levels of physical abstraction. Within this perspective, multiscale modelling becomes a question of methodological coherence rather than hierarchical coupling, and machine learning serves as the connecting logic that enables the treatment of heterogeneous systems within a unified computational landscape.

2.3 Advances in Nanomaterials and Two-Dimensional Materials

2.3.1 Introduction to 1D and 2D structures

By definition, a nanomaterial is a substance that possesses at least one dimension within the nanoscale range, as stated for instance in the definition adopted by Rauscher et al. [6]. This fundamental property endows the material with unique characteristics compared to its bulk counterpart, leading to significant applications in various fields including medicine [7], electronic devices [8], and smart city technologies [9].

Beyond their diverse applications, the distinctive behaviour of nanomaterials arises from the interplay between size effects, surface phenomena, and quantum confinement. As the characteristic dimensions of matter approach the nanometre scale, classical descriptions of physical and electronic behaviour lose validity, and quantum effects begin to dominate. In this regime, properties such as conductivity, optical absorption, or catalytic activity become size- and shape-dependent, enabling a level of control unattainable in conventional materials. This shift represents a fundamental transformation in materials science: nanomaterials are no longer

defined solely by composition, but by architecture and dimensionality, allowing the intentional tuning of their functionalities across scales. Such tunability bridges the quantum and macroscopic worlds, establishing nanomaterials as a cornerstone of modern technological innovation [10].

When a nanomaterial exhibits size-dependent behaviour in zero, one or two dimensions, it is classified as a zero-dimensional (0D), one-dimensional (1D) or two-dimensional (2D) system. Such materials occupy an intermediate regime between conventional bulk matter and fully confined nanoscale structures, where quantum effects, surface dominance and dimensional constraints collectively determine their physical properties.

In this thesis, the focus is primarily on 2D materials, with particular attention to graphene and to how its properties are influenced by atomic vacancies and structural defects. Graphene, a one-atom-thick allotrope of carbon arranged in a honeycomb lattice, is universally regarded as a strictly two-dimensional crystal. It represents the fundamental building block for all other carbon allotropes, including zero-dimensional fullerenes, one-dimensional carbon nanotubes obtained by rolling a graphene sheet and three-dimensional graphite formed by stacking multiple graphene layers, as illustrated in Figure 2.3.

The existence of graphene challenged earlier theoretical predictions that strictly 2D crystals would be thermodynamically unstable. Its stability is now understood to derive from strong covalent bonding together with intrinsic out-of-plane corrugations, such as gentle ripples or slight crumpling, which stabilise the lattice without undermining its two-dimensional nature.

2.3.2 Fabrication method of nanomaterials

Nanomaterials can be produced through top-down or bottom-up strategies, each introducing different degrees of control over size, defect concentration, crystallinity and surface chemistry. Top-down approaches, such as mechanical exfoliation, lithography or milling, derive nanoscale structures by removing material from a bulk phase. These methods offer reproducibility and integration with existing manufacturing lines, but tend to introduce structural disorder and have limited scalability for complex architectures.

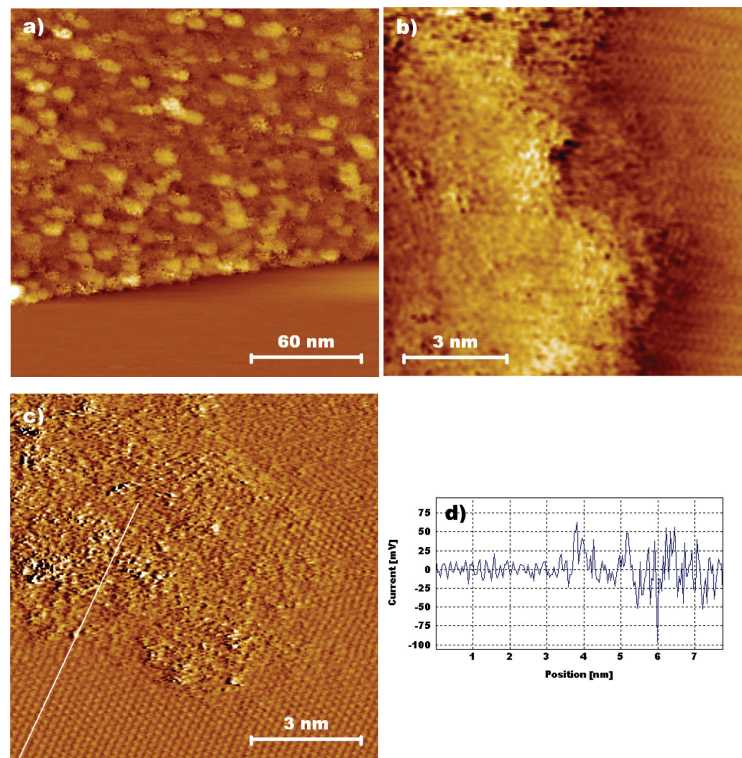


Fig. 2.2 STM images of chemically reduced graphene oxide nanosheets. (a) Nanometre-scale STM image showing a smooth and straight nanosheet edge. (b) Atomic-resolution image of an individual nanosheet edge. (c) Atomic-resolution STM current image of a nanosheet corner, with panel (d) reporting the line profile extracted along the white trace shown in (c). Image and caption reprinted from [11].

Bottom-up methodologies assemble nanomaterials from atoms or molecular precursors. Chemical vapour deposition, sol-gel chemistry, colloidal synthesis and atomic layer deposition are common routes for producing uniform, compositionally controlled nanostructures. These techniques enable precise tuning of grain size, doping profiles, surface functionalisation and hybrid interfaces, which makes them preferable when the target application requires engineered optoelectronic, catalytic or transport properties [12, 13]. The design of a nanomaterial typically follows a property-driven workflow. Functional requirements such as bandgap, carrier mobility, mechanical stiffness, catalytic activity, hydrophilicity are translated into structural descriptors such as morphology, aspect ratio, defect density, crystal orientation, or surface terminations. Synthesis parameters (temperature, precursor chemistry, pressure, plasma power, solvent environment) are then selected to drive the system toward the desired structure-property regime. Iterative characterisation, using tech-

niques like TEM, AFM, Raman or XPS or STM closes the loop between design, fabrication and validation. An example of graphene oxide visualisation is provided in Fig. 2.2

2.3.3 Engineering the graphene structure

Defect engineering in graphene refers to the deliberate introduction, control and repair of structural imperfections within its honeycomb lattice in order to tailor its electronic, chemical, magnetic and mechanical properties. Although pristine graphene exhibits outstanding intrinsic properties, structural defects are often introduced inadvertently during synthesis or processing. Rather than being purely detrimental, these defects can be exploited to achieve new functionalities that are otherwise unattainable in perfect graphene [14].

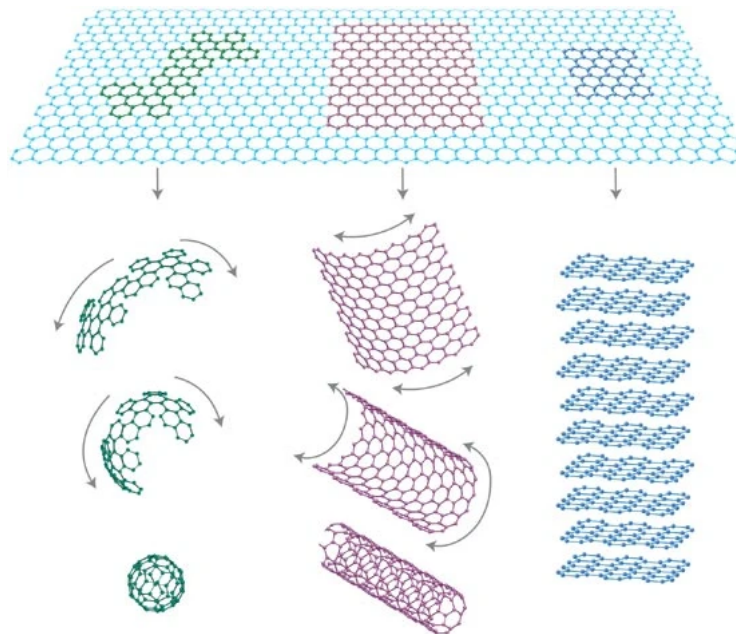


Fig. 2.3 Graphene is a two-dimensional material and forms the fundamental building block for carbon structures of all dimensionalities. Starting from the bottom left, the figure shows the zero-dimensional fullerene, the one-dimensional carbon nanotube and three-dimensional graphite. Image reprinted from the work of Geim et al. [15]

Defects in graphene are generally classified according to their dimensionality. Point defects (0D), such as single and double vacancies or Stone–Wales defects, locally distort the lattice and can induce magnetism, modify the local density of

states and open a tunable electronic band gap. Line defects (1D), including grain boundaries or tilt boundaries, act as extended structural disruptions that influence electronic transport and mechanical integrity. Edge defects, present at the termination of graphene sheets, also play a crucial role in defining local electronic states, chemical reactivity and overall stability.

The generation of defects is achieved through various non-equilibrium methods. These include controlled particle irradiation, which can selectively remove carbon atoms to create vacancies or adatoms, chemical treatments, such as oxidation or plasma exposure that introduce functional groups, and growth-induced defects arising during large-scale synthesis techniques like chemical vapour deposition. Conversely, defect healing can be accomplished through thermal annealing, which allows reconstruction of the lattice; metal-assisted repair, where interactions with a catalytic substrate facilitate defect closure; and chemical absorption, in which molecules selectively fill vacancies and can simultaneously introduce dopants.

The careful balance between defect introduction and repair enables precise tuning of graphene properties [14]. Electronically, defects can open a band gap and generate resonant scattering states, which are essential for semiconductor-like behaviour and device applications. Chemically, defects increase local reactivity, allowing selective functionalization. Magnetically, vacancies and certain reconstructed defects induce localized magnetic moments. Mechanically, while defects generally reduce tensile strength and stiffness, efficient healing can restore mechanical integrity. By controlling both the creation and repair of defects, graphene can be transformed from a nearly perfect two-dimensional conductor into a highly versatile material with tailored multifunctional properties.

Fundamental studies, and in particular the work of Nakaharai et al. [16] and Lherbier et al. [17], have shown that the electronic transport of graphene can be modulated not only through chemical functionalisation but also by introducing a controlled density of structural defects directly into the lattice. Low-dose irradiation techniques, such as accelerated helium ion beams, generate vacancies, divacancies and Stone-Wales defects (see Fig. 2.4) at concentrations of approximately 0.2 to 1.3% while preserving the overall crystallinity of the sheet. These imperfections act as local perturbations of the potential landscape and give rise to distinct resonance energies that appear as peaks in the density of states. At these energies, scattering becomes particularly intense: the mean free path is reduced to a few nanometres, the

semiclassical conductivity approaches its universal minimum, and the conductance decays exponentially with the length of the defective region. Both experimental observations and large-scale simulations indicate that this pronounced suppression of transport originates from strong Anderson localisation rather than from the opening of a conventional bandgap. The localisation is selective, emerging only at the defect-induced resonance energies, with each defect type producing its own characteristic signature, while other regions of the spectrum remain less affected. Defect engineering therefore reshapes the informational landscape through which carriers propagate, guiding transport via disorder-induced localisation rather than through chemical alteration of the intrinsic electronic structure, and enabling controlled modulation of conductivity even at room temperature.

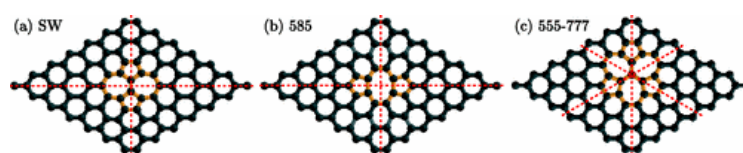


Fig. 2.4 Schematic representation of structural defects commonly found in graphene, including (a) a Stone–Wales defect, (b) a 585 divacancy, and (c) a 555–777 divacancy. Symmetry axes are indicated by red dashed lines. The figure is adapted from Lherbier et al. [17].

2.4 Organic Photovoltaics and Bulk Heterojunction Devices

2.4.1 Fundamentals of organic photovoltaics

Organic photovoltaics have emerged as a distinct class of solar energy conversion devices with properties that set them apart from traditional inorganic technologies. Their development has been driven not only by scientific interest in the photophysics of molecular semiconductors, but also by the potential for entirely new production paradigms and application domains. As highlighted in the foundation work of Brabec [18], organic photovoltaic devices can be processed from solution, deposited on flexible substrates and fabricated using continuous printing and coating techniques. These characteristics make them promising candidates for low cost, lightweight and mechanically compliant solar technologies.

In contrast to crystalline and thin film inorganic photovoltaics, where device fabrication relies on high cost vacuum processes and rigid substrates, organic photovoltaic devices can be manufactured through reel to reel deposition using well established printing methods such as screen, inkjet or flexographic printing. This change in processing strategy has profound implications for scalability and production cost. It opens the possibility of large area modules, integration into curved or lightweight surfaces and incorporation into products where conventional photovoltaic technologies are impractical. The technological attractiveness of organic photovoltaics therefore rests on three main factors: low temperature solution processing, compatibility with flexible substrates and the potential for significant cost reduction in watts per peak power [18].

However, these advantages do not eliminate the fundamental performance constraints that any photovoltaic technology must meet. Efficiency, stability and cost together determine the commercial viability of solar energy conversion systems, and the same holds for organic devices. Early studies demonstrated that competitive performance cannot be achieved by improving only one or two of these factors independently [18]. The efficiency of organic devices must be balanced with operational lifetime and with the economic benefits of low cost processing. This interplay defines the technological space in which organic photovoltaics can compete, and it continues to guide current research directions.

From a scientific perspective, the operation of organic photovoltaic devices differs significantly from that of inorganic semiconductors. Photoexcitation in molecular materials typically produces strongly bound excitons rather than free carriers, owing to low dielectric screening and localised electronic states. Efficient device operation therefore requires the controlled management of exciton diffusion, dissociation at donor acceptor interfaces and the subsequent transport of separated charges. These processes depend sensitively on molecular energetics, electronic couplings and mesoscale morphology, which makes organic photovoltaics a rich platform for studying the relationship between structure and function in soft electronic materials.

Taken together, these characteristics define the foundations of organic photovoltaics as both a scientific and technological domain. They also motivate the bulk heterojunction architecture, discussed in the following section, which was developed to overcome the limitations of exciton diffusion and to maximise interfacial area in solution processed organic solar cells.

2.4.2 Bulk heterojunction architecture

The bulk heterojunction (BHJ) architecture was introduced as a solution to the fundamental limitations imposed by the short exciton diffusion lengths that characterise organic semiconductors. In contrast to inorganic materials, where photogenerated carriers are created as free charges, photoexcitation in molecular semiconductors produces tightly bound excitons with diffusion lengths typically on the order of ten nanometres. Early bilayer organic photovoltaic devices suffered from limited interfacial area and therefore low charge generation efficiency, since only excitons that reached the donor acceptor interface within their lifetime contributed to photocurrent. The BHJ concept, originally proposed in the mid 1990s and established through later developments in polymer and fullerene based systems, addresses this limitation by creating a finely intermixed donor acceptor network with a vast internal interface [18, 19].

In a BHJ device, the active layer consists of a bicontinuous interpenetrating morphology in which donor rich and acceptor rich domains form a nanoscale network. The characteristic domain sizes are comparable to the exciton diffusion length, which ensures that most excitons can reach a heterojunction interface before recombining. This increase in interfacial area dramatically enhances the probability of exciton dissociation into free charges. The efficiency of this process depends not only on the morphological structure but also on the energetic alignment between donor and acceptor materials. The offsets between the highest occupied and lowest unoccupied molecular orbitals must be sufficient to drive charge separation, while the interfacial microstructure influences the pathways for charge transfer and the formation of charge transfer states [20].

The three dimensional morphology plays a decisive role in device operation. The connectivity and purity of donor and acceptor domains determine the pathways for electron and hole transport, while the local composition influences recombination and charge trapping. Numerous studies have shown that optimal device performance arises from a delicate balance between domain size, percolation, tortuosity and interfacial sharpness [21, 22]. Excessive mixing can hinder charge transport by disrupting continuous pathways, whereas overly large domains reduce exciton harvesting. The processing conditions used to fabricate the BHJ layer, including solvent selection, drying kinetics and thermal or solvent annealing, strongly influence the final morphology and therefore the resulting device characteristics.

From a modelling perspective, the BHJ architecture introduces significant challenges. The active layer cannot be treated as a uniform effective medium; instead, its nanoscale structure leads to spatially varying generation rates, electric fields and transport pathways. The mesoscale morphology is therefore intimately linked to the device level behaviour described by the drift diffusion formalism. Predicting device performance requires models capable of capturing the heterogeneous and complex structure of the active layer, as well as its impact on exciton dissociation, carrier motion and recombination.

Taken together, the bulk heterojunction architecture represents a paradigmatic example of how nanoscale morphology governs macroscopic device response [23]. Its scientific and technological importance extends beyond organic photovoltaics, as it exemplifies the broader principle that functional device behaviour emerges from the interplay between molecular energetics, interfacial structure and mesoscale morphology. These concepts motivate the modelling strategy adopted in the following section, where the drift diffusion equations are introduced as the framework for describing charge transport in complex heterogeneous media.

2.4.3 Device scale modelling: the drift diffusion formalism

The operation of bulk heterojunction organic solar cells at the device scale is commonly described by the drift diffusion formalism, which provides a continuum model for charge transport, generation and recombination in semiconductor devices [24–26]. In this framework, the active layer is treated as a medium in which the local carrier densities and electrostatic potential evolve according to coupled partial differential equations. Although organic semiconductors differ from crystalline inorganic materials in terms of energetic disorder, hopping transport and excitonic generation, the drift diffusion approach has proven to be a versatile tool for interpreting current–voltage characteristics, loss mechanisms and internal field distributions in organic photovoltaic devices [27, 28].

At the core of the model are the continuity equations for electrons and holes,

$$\frac{\partial n}{\partial t} = \frac{1}{q} \nabla \cdot \mathbf{J}_n + G - R, \quad (2.1)$$

$$\frac{\partial p}{\partial t} = -\frac{1}{q} \nabla \cdot \mathbf{J}_p + G - R, \quad (2.2)$$

where $n(\mathbf{r}, t)$ and $p(\mathbf{r}, t)$ are the electron and hole densities, q is the elementary charge, $\mathbf{J}_n$ and $\mathbf{J}_p$ are the corresponding current densities, $G(\mathbf{r}, t)$ denotes the generation rate of free charges and $R(\mathbf{r}, t)$ represents the recombination rate. In steady state simulations, which are typically used to compute current–voltage characteristics, the time derivatives vanish and the continuity equations reduce to balance relations between current divergence, generation and recombination.

The current densities are written as the sum of drift and diffusion contributions,

$$\mathbf{J}_n = q\mu_n n \mathbf{E} + qD_n \nabla n, \quad (2.3)$$

$$\mathbf{J}_p = q\mu_p p \mathbf{E} - qD_p \nabla p, \quad (2.4)$$

where μ_n and μ_p are the electron and hole mobilities, D_n and D_p are the corresponding diffusion coefficients, and $\mathbf{E} = -\nabla\phi$ is the electric field derived from the electrostatic potential $\phi(\mathbf{r})$. The diffusion coefficients are often related to the mobilities through the Einstein relation,

$$D_n = \frac{k_B T}{q} \mu_n, \quad D_p = \frac{k_B T}{q} \mu_p, \quad (2.5)$$

where k_B is the Boltzmann constant and T the temperature. In organic semiconductors this relation is an approximation, as transport proceeds via thermally activated hopping in a disordered energy landscape, but it remains a widely used assumption in device scale modelling [25].

The electrostatic potential is obtained from the Poisson equation,

$$-\nabla \cdot (\epsilon(\mathbf{r}) \nabla \phi) = q [p(\mathbf{r}) - n(\mathbf{r}) + \rho_{\text{fixed}}(\mathbf{r})], \quad (2.6)$$

where $\epsilon(\mathbf{r})$ is the local permittivity and ρ_{fixed} accounts for any fixed charge, such as dopants or interfacial charges. In many bulk heterojunction devices the active layer is nominally intrinsic, and the term ρ_{fixed} can often be neglected, although interfacial dipoles and contact induced band bending may still play a role.

The quantities G and R encode the photophysical processes specific to organic solar cells. In most drift diffusion models, the generation term is taken to represent the rate of free carrier formation following exciton dissociation, which may depend on position, field and morphology. Recombination is frequently described by a

bimolecular Langevin type expression,

$$R = \gamma(np - n_i^2), \quad \gamma = \frac{q}{\epsilon}(\mu_n + \mu_p), \quad (2.7)$$

where γ is the Langevin recombination coefficient and n_i is the intrinsic carrier density. Deviations from the Langevin limit, which are often observed in high performance polymer–fullerene and non fullerene systems, can be captured by introducing a reduction factor or by employing more refined recombination models [25, 26].

Boundary conditions complete the device model. At the contacts, selective or non selective injection is represented through appropriate relations between carrier densities, quasi Fermi levels and the local potential. Contact properties, such as injection barriers and surface recombination velocities, strongly influence the open circuit voltage, fill factor and overall device performance [27]. Within the bulk, spatial variations in mobility, permittivity and generation rate can be introduced to reflect the underlying bulk heterojunction morphology, making the drift diffusion approach sensitive to the three dimensional structure described in the previous section.

In summary, the drift diffusion formalism provides a continuum level description that links the nanoscale morphology and material parameters of bulk heterojunction devices to their macroscopic current–voltage response. In this thesis, it serves both as a reference model for simulating device behaviour and as a source of high fidelity data for training surrogate models based on machine learning, which will be discussed in the following chapters and in particular in the case study in the chapter 5

2.5 Data and automation foundation for computational material science

Modern materials research increasingly depends on the ability to manage, transform and analyse large and heterogeneous datasets. In practical scientific workflows, data originate from diverse sources, include multiple physical observables and often span a wide range of numerical formats and levels of complexity. This heterogeneity

makes structured organisation essential. Standardisation of formats, metadata and conventions enables coherent analysis and prevents the loss of critical scientific information that may arise when data are produced or interpreted in inconsistent ways.

Within computational materials science, several challenges are well recognised. Issues of veracity and completeness often arise when datasets combine high fidelity simulations with sparse or inconsistent experimental measurements. Many existing data infrastructures place strong emphasis on computational outputs, while comprehensive and standardised experimental datasets remain comparatively scarce. Assessing the reliability of data is further complicated by methodological differences across studies, numerical biases and the absence of uniform community standards, all of which hinder large scale integration [29]. These challenges underscore the need for disciplined data management practices that support reproducibility and effective reuse.

Despite the specificity of materials science, the problem of integrating heterogeneous data is not unique to this field. In any scientific domain where data are generated through multiple processes and represented across incompatible formats, meaningful analysis requires the harmonisation of metadata, units and structural conventions. Automated workflows play a central role in this process, providing systematic procedures for validation, preprocessing and format alignment at scale. They ensure that large datasets can be transformed into coherent analytical structures suitable for downstream modelling and quantitative interpretation.

Adopting FAIR principles further strengthens this process by enforcing transparent provenance, standardised metadata and clearly defined conditions for findability, accessibility, interoperability and reusability [30]. When applied to computational materials science, FAIR compliant practices make it possible to integrate simulation results, derived descriptors and experimental data into unified and machine readable representations. This, in turn, enables the construction of predictive and data driven models capable of capturing structure property relationships across diverse materials systems. As highlighted in recent studies on scientific data infrastructures [31], standardisation and automation together provide the conceptual foundation that bridges heterogeneous sources and supports the development of reliable insights from complex, multi source datasets.

2.5.1 Automated simulation workflows

The increasing scale and complexity of computational studies in materials science have made automation a fundamental requirement for modern scientific practice. At the atomistic level, systems such as defective graphene span extremely large configurational spaces. Variations in defect geometry, local bonding, strain distributions or chemical termination can produce significant changes in electronic and transport behaviour. Exploring these degrees of freedom demands the ability to generate, simulate and analyse thousands of structures in a systematic and reproducible manner. Manual pipelines are not adequate for this purpose. Automated workflows are required to manage each step of the process, from structure generation to simulation execution, post processing and data archiving.

A comprehensive perspective on this transformation was formalised by Pizzi and collaborators through the ADES framework [32]. The ADES model identifies four interdependent pillars that underpin modern computational research: *Automation*, which abstracts away low level execution details and enables high throughput studies; *Data*, which ensures structured storage, provenance and queryability; *Environment*, which provides a controlled workspace for composing and running complex scientific workflows; and *Sharing*, which promotes reproducibility, dissemination of workflows and the adoption of standard formats within the scientific community. Figure 2.5 summarises these principles and illustrates how automation must be understood not as an isolated technical tool, but as part of an integrated methodological ecosystem that supports scalable and reproducible scientific inquiry.

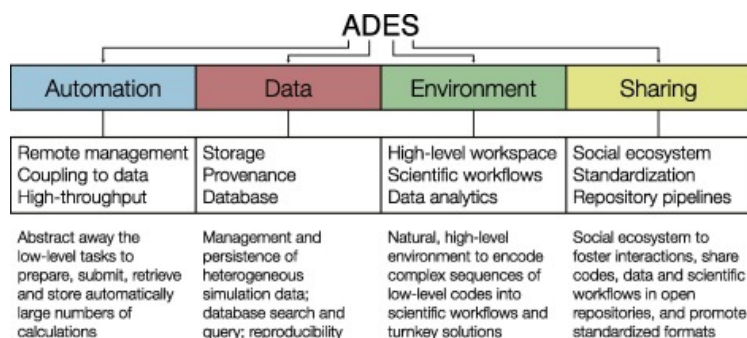


Fig. 2.5 Schematic representation of the ADES model. The four pillars of Automation, Data, Environment and Sharing together define the conceptual and technological foundations required for high throughput, reproducible and community oriented computational materials science. Image reprinted from [32]

Automation frameworks developed within the computational materials community give concrete form to these ideas. AiiDA provides a formalised environment for workflow management, provenance tracking and the execution of large simulation campaigns [32], ensuring that each computational step is recorded and reproducible. The NOMAD infrastructure establishes standards for storing and disseminating electronic structure data, emphasising transparency, traceability, and machine readability [33]. Tools such as ASE complement these platforms by enabling high throughput generation and manipulation of atomic configurations. Together, these systems demonstrate how automated workflow design can transform atomistic modelling from a collection of isolated simulations into a coherent, data centred methodology.

Comparable requirements arise at the mesoscale, particularly in the modelling of bulk heterojunction organic photovoltaic devices. Drift diffusion simulations require repeated numerical solutions across multiple time resolved snapshots and morphological realisations. Each simulation produces detailed spatial fields, including electrostatic potentials, carrier densities and current distributions. As in atomistic studies, automation ensures that these simulations are executed consistently, that numerical parameters remain controlled and that the resulting data are stored in a structured form suitable for large scale analysis. Recent work in the computational study of surfaces and interfaces has shown that automated pipelines allow machine learning models to capture subtle physicochemical trends that would remain inaccessible through manual approaches [34].

The scientific role of automation therefore extends well beyond computational efficiency. Automated workflows enforce methodological consistency, reduce human driven variability and create the foundations for reproducible scientific research [35]. They also enable the construction of large, homogeneous datasets that support the development of predictive and generative models, which are central to the data driven methodologies introduced in the following chapters.

2.5.2 Methods for integrating heterogeneous data

As previously introduced, working with unstructured or highly heterogeneous data poses distinct challenges compared to data that can be efficiently managed through relational databases or tabular formats. In real-world applications, datasets often coexist in multiple formats, including XML or JSON files, images, audio recordings,

and historical time series. Processing these diverse data types requires approaches that are fundamentally different from one another, and in many cases, bespoke workflows are necessary to achieve what can be described as complete integration [36, 37].

A further critical challenge arises from semantic inconsistencies. It is common to encounter two datasets describing the same phenomena but employing different units of measurement or terminologies, complicating the construction of a unified workflow. Maintaining data quality is also a substantial concern: crowdsourced or externally sourced data often demand extensive preprocessing, which can be time-consuming and may not align with the intended use of the data [37].

Finally, legal, ethical, and regulatory considerations add another layer of complexity. Compliance with frameworks such as GDPR, as well as with specific licensing agreements, requires robust metadata management and access controls. Balancing the necessity of data sharing with privacy and intellectual property protections remains a critical challenge in contemporary data integration efforts [30]. Having outlined the challenges associated with data integration, it is essential to examine the methodologies developed to address them. In particular, tackling the challenges discussed above requires robust data integration frameworks. Current state-of-the-art studies increasingly emphasize the synergy between artificial intelligence (AI) techniques and data integration frameworks. As highlighted in review works [36, 37], data-driven approaches are employed to correlate entries that may represent the same entity and to add a semantic layer to the data. These methods have been shown to outperform rule-based applications both in terms of scalability and classification accuracy.

Other notable approaches focus on the organisational aspects of datasets. For example, Google Dataset Search enables the cataloguing of publicly available datasets, making them easily discoverable, while platforms such as DataHub combine versioning capabilities with enriched metadata, facilitating a clearer understanding of the dataset. While these solutions improve dataset organisation and reduce some of the existing barriers, they primarily address structural challenges. When it comes to data reuse and accessibility, the FAIR framework [30] provides a comprehensive approach. Originally proposed as guidelines for publishing datasets alongside scholarly articles, FAIR has evolved into the de facto standard for data management,

particularly when datasets are intended to be accessible and reusable across diverse scientific communities.

In essence, the FAIR principles, Findability, Accessibility, Interoperability, and Reusability, aim to make data available and usable for both humans and machines in a wide variety of contexts. The individual principles, as discussed in [30], are summarised as follows:

- **Findability:** Findability ensures that datasets can be easily located through the use of persistent identifiers like DOIs and rich metadata. This metadata, which includes descriptive tags, source information, and keywords, enhances searchability and retrieval in repositories such as Zenodo and Dryad. Furthermore, FAIR principles encourage the inclusion of machine-readable metadata to support automated data discovery.
- **Accessibility:** Accessibility ensures that datasets can be retrieved through open and standardized protocols, such as APIs or FTP. Clear documentation of licensing and usage rights helps users understand the conditions for accessing and reusing data. Moreover, accessibility extends to long-term data stewardship, ensuring datasets remain available and usable even as technologies evolve.
- **Interoperability:** Interoperability is achieved by using common data formats, ontologies, and standardized vocabularies. For example, frameworks like RDF facilitate flawless dataset linking, while formats such as JSON-LD ensure compatibility across platforms. Interoperability also involves providing detailed metadata that describes the dataset's structure and semantics, allowing it to be integrated into analytical pipelines without extensive preprocessing.
- **Reusability:** Reusability focuses on the clarity and richness of dataset documentation. Provenance metadata, which details the origins and transformations of the data, is crucial for ensuring reproducibility. Platforms like FigShare promote reusability by requiring datasets to include citations, licensing terms, and contextual metadata that describe their relevance to specific research questions.

2.6 Key Enabling Technologies

The methodological principles introduced in this chapter, ranging from multiscale physical modelling to workflow automation, structured data management and FAIR aligned practices, align with a broader technological framework recognised within European research policy. The European Commission identifies a set of Key Enabling Technologies (KETs) as foundational drivers of scientific progress and industrial innovation across multiple sectors [38]. These technologies include advanced materials, nanotechnology, micro and nano electronics, photonics, advanced manufacturing systems and information and communication technologies. Their strategic relevance has been reaffirmed in subsequent European research programmes, where KETs are positioned as essential components for addressing societal challenges and enabling next generation technological development [38, 39].

Computational materials science lies at the intersection of several of these enabling technologies. Atomistic and mesoscale modelling depend directly on advances in electronic structure theory, numerical solvers and high performance computing infrastructure. Device scale simulations draw on progress in organic electronics, thin film science and photonic energy conversion. At the same time, the transition toward data centred methodologies requires robust digital infrastructures, interoperable standards and artificial intelligence techniques capable of extracting structure and predictive insight from high dimensional scientific datasets. These dependencies illustrate that modern materials research is intrinsically embedded within a broader technological ecosystem rather than isolated from it.

The interplay between these enabling technologies becomes particularly visible in applications that involve both materials and devices. Advanced materials provide the physical basis for functionality; automated workflows and HPC environments ensure that these materials can be explored in a systematic and reproducible manner; FAIR oriented data practices support interoperability and long term reuse; and artificial intelligence contributes predictive and generative capabilities that accelerate design and optimisation. These elements do not operate independently but form an interconnected set of technologies that collectively define the contemporary scientific landscape.

Within this perspective, the methodological framework adopted in this thesis can be seen as an example of the KETs paradigm in practice. The integration of physics

based modelling, workflow automation, structured data infrastructures and machine learning exemplifies the type of technological convergence emphasised by European research initiatives. This conceptual view also sets the stage for later chapters, where these enabling technologies connect to broader technological infrastructures and higher level system considerations.

2.7 Conclusions

The aim of this chapter has been to establish the physical and methodological foundations that underpin the work presented in this thesis. The discussion introduced the scientific domains in which the subsequent chapters are situated and provided the conceptual tools required to interpret the material and device level studies developed in recent years. In the first part of the chapter, attention was devoted to the state of the art in nanoscale and mesoscale materials, with a particular focus on defective graphene and on the organic semiconductor systems that give rise to bulk heterojunction photovoltaic devices. These examples illustrate the structural complexity and physical richness that characterise modern materials research.

The second part of the chapter shifted the emphasis toward the methodological transition from a purely physics driven perspective to a data centred one. This transition involves automated simulation workflows, structured data management, FAIR aligned practices and the integration of heterogeneous scientific datasets. Together, these elements define the framework through which physical problems are reformulated in a manner suitable for data driven analysis. They also provide the theoretical basis for understanding the machine learning methodologies that appear in the following chapter.

Taken as a whole, the chapter positions computational materials and device modelling within a broader context that combines physical theory, automated data generation and structured information management.

Chapter 3

Machine Learning Foundations for Scientific Modelling

3.1 Introduction

Artificial intelligence (AI) has become a central component of contemporary scientific modelling, particularly in domains where physical systems exhibit complexity across multiple spatial or temporal scales. Traditional physics based approaches provide principled descriptions of materials and devices, yet they often struggle when confronted with high dimensional configuration spaces, heterogeneous data sources or simulation pipelines whose computational cost prohibits systematic exploration. In these settings, AI offers a complementary computational layer that can extract structure from data, approximate complex mappings and support the efficient analysis and design of physical systems.

The aim of this chapter is to introduce the methodological foundations that enable such data driven analysis within the multiscale context of this thesis. Although the atomistic and mesoscale systems studied in later chapters do not form a classical hierarchy of tightly coupled models, they share common challenges: both give rise to large and heterogeneous datasets, both require the extraction of meaningful representations and both benefit from surrogate models capable of learning physical relationships that are expensive to compute directly. Artificial intelligence provides a set of tools that address these challenges through three core components.

The first component is representation learning, which concerns the construction of compact and informative descriptions of scientific data. Atomistic configurations, three dimensional morphologies and device level outputs often inhabit high dimensional or non Euclidean spaces, and effective modelling requires representations that capture their intrinsic structure while respecting physical constraints. The second component is operator learning, which extends classical function approximation to mappings between functions, enabling the efficient emulation of computationally intensive physical models. Neural operator architectures [40] provide a principled means of learning the solution operators associated with partial differential equations and therefore act as data driven surrogates for device scale simulations. The third component is generative modelling, which allows the sampling and exploration of complex configuration spaces. Generative models support the synthesis of new atomic arrangements, the creation of artificial morphologies and the guided exploration of design spaces with specific physical objectives.

Together, these elements form the computational language used throughout this thesis. Representation learning provides the geometric and statistical structure on which the models operate, operator learning provides the predictive core required for efficient device modelling and generative modelling enables the exploration of material and morphology spaces beyond the limits of direct simulation. The remainder of this chapter develops these components in detail and establishes the methodological framework that underpins the atomistic and mesoscale analyses presented in Chapters 5 and 6.

3.1.1 Data-driven Modelling and Machine Learning Background

Machine learning (ML) has become a central component in modern computational science, providing a set of flexible tools for analysing data, modelling complex systems and accelerating large scale simulations. In materials science and device engineering, data driven methods increasingly complement conventional physics based approaches by offering efficient approximations of non linear relationships, enabling high throughput screening and supporting inverse design tasks. The techniques employed throughout this thesis range from classical supervised learning to advanced operator learning and generative modelling, forming a coherent methodological foundation for the multi scale problems addressed in the subsequent chapters.

A key motivation for introducing these methods is the growing availability of structured scientific datasets and the computational cost of solving detailed physical models such as drift–diffusion equations or atomistic electronic structure calculations. ML enables the construction of surrogate models capable of reproducing the behaviour of these solvers at a fraction of the cost, thereby opening the possibility of extensive parameter exploration and rapid what if analysis. At the same time, generative models provide a powerful framework for synthesising new candidate structures or morphologies while preserving physical plausibility, supporting a data driven approach to materials and morphology design.

Scientific machine learning further extends this perspective by integrating physical constraints with data driven representations, allowing the approximation of differential operators and the inference of high dimensional fields. Neural operator architectures, in particular, offer a principled way to learn mappings between function spaces and have emerged as an effective alternative to traditional partial differential equation (PDE) solvers in multi physics settings. Finally, explainable AI methods contribute to the interpretability of these models by identifying the most influential features, clarifying structure–property relationships and guiding generative exploration. Together, these techniques provide the conceptual and computational foundation on which the methodological framework of this thesis is built.

3.1.2 Supervised vs Unsupervised Learning

The modern field of ML emerged from a long intellectual lineage that spans statistics, cybernetics and early theories of artificial intelligence. Its foundations can be traced to work on pattern recognition and statistical inference in the mid twentieth century, most notably through the contributions of Fisher, Rosenblatt and Vapnik. Fisher introduced discriminant analysis in the 1930s, establishing the statistical principles of classification [41]. Rosenblatt’s perceptron in 1958 [42] provided the first operational model of a learning machine capable of adjusting its parameters from data. Vapnik and Chervonenkis later formalised the statistical theory of learning, giving rise to the concept of generalisation and to the framework that underpins much of contemporary supervised learning [43]. During the same period, unsupervised approaches developed from ideas in information theory and statistical physics. MacQueen introduced k means clustering in 1967, while Kohonen later developed self organising maps to model the emergence of structure from unlabelled observations.

These ideas set the stage for the probabilistic models that followed, such as mixture models and principal component analysis, which provided formal mechanisms for uncovering latent structure in high dimensional data [44]. ML today is shaped by deep neural networks, which revive and extend these earlier ideas through large scale optimisation and differentiable programming. The distinction between supervised and unsupervised learning remains central because it reflects two fundamentally different sources of information: externally provided labels and intrinsic statistical structure. In supervised learning, models are trained using pairs of inputs and targets, enabling the network to approximate a function that maps data to a known outcome. This paradigm is essential when high fidelity ground truth is available. Unsupervised learning, by contrast, seeks patterns that arise solely from the organisation of the data. Together, supervised and unsupervised learning form the methodological backbone of the data driven framework developed in this work. Their combined use allows one to construct predictive surrogates, extract intrinsic material descriptors and generate novel morphologies with controlled characteristics, thereby linking computational materials science with advanced simulation and design strategies.

3.1.3 Bias–variance tradeoff

A central concept in statistical learning theory is the bias–variance tradeoff, which describes how the generalisation error of a model evolves as a function of its complexity. Let $f(x)$ denote the true, unknown data-generating function and $\hat{f}(x)$ a model trained on a finite dataset. The expected prediction error can be decomposed as

$$\mathbb{E} [(y - \hat{f}(x))^2] = \text{bias}^2 + \text{variance} + \text{noise}. \quad (3.1)$$

Each term reflects a distinct source of error. The bias measures the systematic discrepancy introduced by an overly simple model, which fails to capture the underlying structure of the data. The variance quantifies the sensitivity of the model to fluctuations in the training set, and increases when the model is excessively flexible and adapts to noise rather than to the true signal. The noise term represents the irreducible component of the error, stemming from intrinsic variability in the data, measurement uncertainty, or unobserved factors; it is independent of model complexity. All these components can be seen in Fig. 3.1.

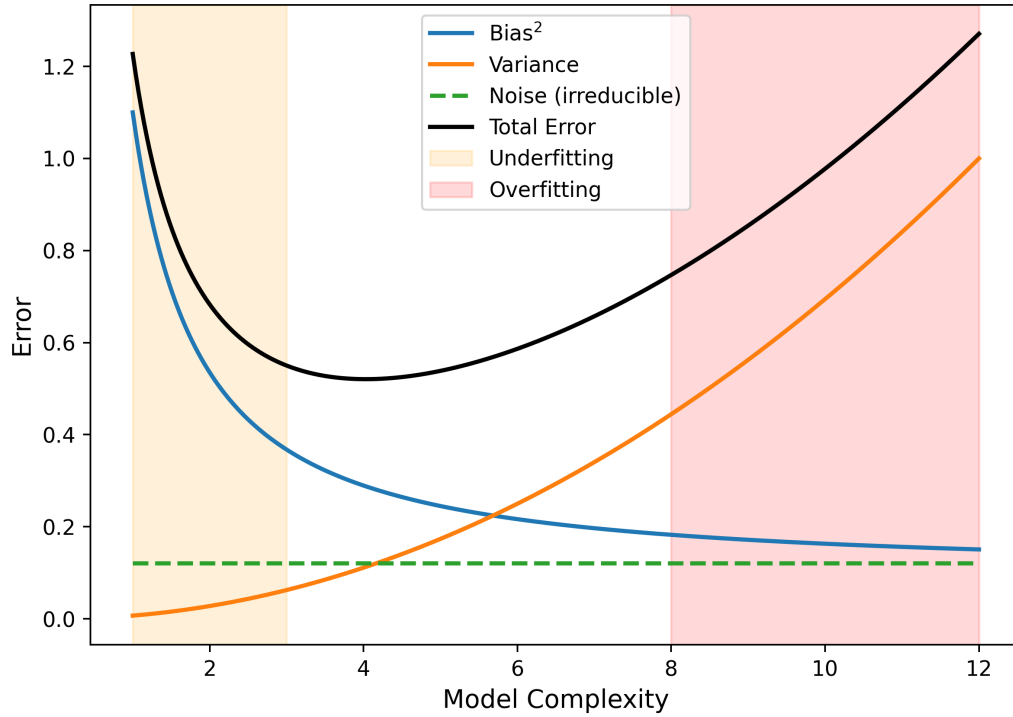


Fig. 3.1 Decomposition of the generalisation error into squared bias, variance and irreducible noise as a function of model complexity. The total error exhibits a characteristic U-shape, with an optimal region between underfitting and overfitting.

These three components exhibit characteristic behaviour as model complexity increases. In the underfitting regime, the model has high bias and low variance, leading to poor predictive performance. As complexity grows, the bias decreases and the variance increases, resulting in a characteristic U-shaped curve for the total error. The optimal complexity corresponds to the minimum of this curve, where the reduction in bias is balanced by the increase in variance. Beyond this point, the model enters the overfitting regime, in which the variance dominates and the generalisation error rises again.

This decomposition is particularly relevant in scientific machine learning, where both under-parameterised and over-parameterised models can lead to misleading physical interpretations. An insufficiently expressive model fails to capture fundamental physical relationships, while an overly complex model risks encoding numerical artefacts, discretisation noise, or experimental fluctuations. In the context of operator learning, surrogate modelling, and morphology-based predictions, the

bias–variance tradeoff provides a conceptual framework for selecting an appropriate model architecture and for understanding the sources of error that limit generalisation across unseen geometries and operating conditions.

3.2 Data Representations for Physical Systems

3.2.1 Structured and unstructured data

Data encountered in physical modelling can be broadly categorised into structured and unstructured forms. Structured data arise when measurements or simulations produce values defined on regular grids or meshes. Examples include electron density maps, potential fields, carrier distributions obtained from drift diffusion simulations and voxel based representations of bulk heterojunction morphologies. These data admit natural tensorial forms and benefit from architectures that exploit spatial locality, such as convolutional neural networks or neural operators [29].

Unstructured data, by contrast, do not lie on regular grids. Atomistic configurations represent a canonical example: atoms occupy arbitrary positions in space, form variable coordination environments and give rise to interaction graphs whose connectivity depends on geometric thresholds or bonding rules. Such data are more naturally modelled as graphs, where nodes represent atoms and edges encode local neighbourhoods. Graph based representations preserve rotational and translational invariances and form the basis for graph neural networks widely used in molecular and materials modelling [45].

A third category arises when data represent sets of high dimensional samples whose structure is not tied to a physical mesh or graph. These are often embedded in vector spaces through feature engineering or learned representations. Across all cases, a central challenge is to construct representations that are physically meaningful, computationally tractable and suitable for downstream tasks.

3.2.2 Feature Extraction and Latent Spaces

A fundamental step in data-driven modelling is the construction of informative features that summarise relevant physical information. In some settings, descriptors can

be manually engineered from domain knowledge, such as coordination numbers, radial distribution functions, tortuosity metrics or morphology-derived statistics. These structured representations enable the use of feature-based models, including gradient boosting methods such as XGBoost [46], which often perform strongly in low-data regimes or when interpretability is required. However, for many scientific datasets the underlying structure is too high-dimensional or too geometrically complex for handcrafted descriptors to capture all relevant variability. Representation learning addresses this limitation by constructing latent spaces in which samples sharing structural or physical similarity are mapped to nearby regions. Autoencoders (see Fig. 3.2 for the schematic representation of this network) [47] provide a canonical framework for this purpose. Given an input x , an encoder network computes a latent code $z = E_\theta(x)$, typically of much lower dimensionality, while a decoder reconstructs an approximation $\hat{x} = D_\phi(z)$. Training minimises a reconstruction loss

$$\mathcal{L}(x, \hat{x}) = \|x - \hat{x}\|^2 \quad (3.2)$$

or, more generally, a divergence between the input and its reconstruction. The latent space thus acts as a compressed manifold of the data distribution, encoding the most salient degrees of freedom while discarding irrelevant variability [47, 48].

Latent representations offer several advantages for physical modelling. They provide smooth and differentiable embeddings on which downstream predictive or generative models can operate; they facilitate anomaly detection by identifying samples that fall outside the typical latent manifold; and they expose geometric organisation; clusters, manifolds or continuous physical regimes, that are not visible in the raw feature space. Moreover, latent spaces can be designed to enforce invariances such as translation, rotation or permutation, which are essential for atomistic systems and for morphological fields in device-scale simulations. Variational autoencoders (VAEs) further extend this idea by placing a probabilistic prior on the latent space, enabling interpolation, sampling and principled uncertainty quantification [49].

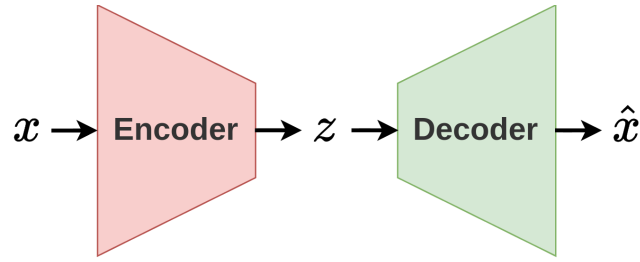


Fig. 3.2 Schematic representation of an auto-encoder network

3.2.3 Dimensionality reduction and clustering overview

Dimensionality reduction Dimensionality reduction refers to a family of techniques that transform high dimensional data into a lower dimensional representation while seeking to preserve meaningful geometric or statistical structure. In formal terms, these methods define a mapping from an original space $\mathbb{R}^N$ to a reduced space $\mathbb{R}^M$ with $M < N$. The objective is not simply compression but the construction of a representation in which latent patterns, neighbourhood relations or global trends become more accessible for analysis. Such representations are particularly valuable in scientific applications where the underlying data often lie on low dimensional manifolds embedded in a high dimensional ambient space.

In the context of materials and device modelling, dimensionality reduction provides a practical means to analyse ensembles of atomistic configurations, complex mesoscale morphologies or large sets of simulated device responses. It enables visual exploration of structural similarities, facilitates clustering of configurations with related physical properties and offers a compact foundation for subsequent machine learning models. A wide range of techniques has been developed for this purpose. Principal component analysis (PCA) [50] constructs a linear projection that maximises the variance of the transformed data and provides an interpretable orthogonal basis. Non linear methods, such as t distributed stochastic neighbour embedding (t-SNE) [51], emphasise the preservation of local neighbourhoods, while methods such as Uniform Manifold Approximation and Projection (UMAP) [52] aim to capture the structure of an underlying manifold by balancing local connectivity and global organisation. These approaches differ in their mathematical assumptions and geometric guarantees, but each provides a complementary perspective on the

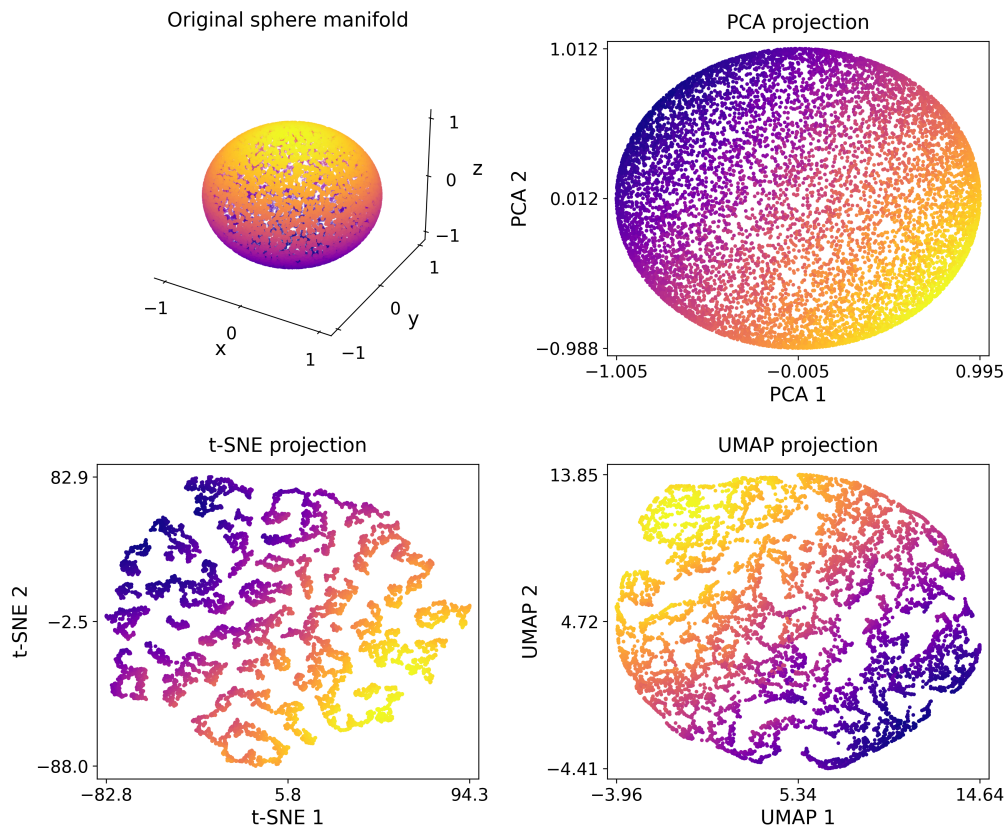


Fig. 3.3 Comparison of dimensionality reduction techniques applied to a non linear manifold. **Top left:** synthetically generated three dimensional sphere dataset illustrating an intrinsically two dimensional surface embedded in a higher dimensional ambient space. **Top right:** linear projection obtained through principal component analysis (PCA), which preserves global variance but cannot recover the curved geometry of the manifold. **Bottom left:** t distributed stochastic neighbour embedding (t-SNE), which preserves local neighbourhoods and therefore produces separated regions that reflect local structure rather than global topology. **Bottom right:** Uniform Manifold Approximation and Projection (UMAP), which balances local connectivity and global organisation to produce a continuous two dimensional embedding that more faithfully reflects the structure of the original manifold.

structure of high dimensional scientific data. A practical example of these three algorithms in action can be seen in Fig. 3.3

Clustering Clustering provides a complementary perspective to dimensionality reduction by partitioning data into groups of samples that exhibit comparable geometric or statistical characteristics. While dimensionality reduction reveals the manifold

structure on which high dimensional scientific data reside, clustering organises this structure into coherent regions that support interpretation and further analysis. This is particularly relevant in materials and device modelling, where ensembles of atomistic configurations, mesoscale morphologies or device responses often form families of related patterns that are not labelled a priori and whose organisation reflects underlying physical mechanisms.

Different clustering methods capture different aspects of data organisation. Partition based algorithms such as K-means [53] assume approximately spherical clusters and identify representative centroids, making them suitable for grouping atomic configurations or defect motifs with shared geometric features. Density based methods, including DBSCAN [54] and HDBSCAN [55], detect clusters of arbitrary shape and naturally identify outliers; spectral clustering [56], which operates on graph representations of sample similarity, is effective when the data exhibit non linear separations or manifold like structure and provides a flexible alternative when conventional distance based criteria are insufficient.

Taken together, these methods form an analytical toolkit that complements the geometric insight provided by dimensionality reduction. They will later be used to characterise the landscape of graphene defects, to organise bulk heterojunction morphologies according to structural and transport relevant descriptors and, more broadly, to identify physically meaningful regimes within large scale simulation datasets.

3.2.4 Convolutional Neural Networks and Object Detection Architectures

Convolutional neural networks (CNNs) have become a cornerstone of modern deep learning for data with spatial or spatiotemporal structure. Their fundamental operation is the convolution, a linear transformation that applies a set of local filters across the input domain. Given an input map x and a kernel $\mathcal{K}$, the discrete convolution is defined as

$$(\mathcal{K} * x)(i, j) = \sum_{m, n} \mathcal{K}(m, n) x(i - m, j - n),$$

producing feature maps that encode local correlations such as edges, textures or repeating structural motifs. The weight sharing inherent in the convolution operation

induces translation equivariance and embeds a strong prior of spatial locality, both well aligned with the structure of scientific data, including two-dimensional projections of atomistic configurations, scanning tunneling microscopy (STM) images, and spatially resolved device fields.

Stacking multiple convolutional layers yields hierarchical features, where early layers capture local geometric primitives and deeper layers synthesise global or semantic representations. Architectural innovations such as residual connections have further stabilised the training of deep models. A residual block augments a nonlinear transformation $F(x)$ with an identity mapping,

$$y = F(x) + x,$$

which alleviates vanishing-gradient issues and supports the construction of networks with substantial depth. Such architectures serve as backbones in autoencoders, variational models and diffusion-based frameworks, and constitute the feature extractors of many state-of-the-art object detection systems.

Two-stage and single-stage object detection

Object detection architectures incorporate two tasks: locating candidate regions and classifying their content. Traditional two-stage detectors, such as R-CNN [57], Fast R-CNN [58] and Faster R-CNN [59], implement these tasks sequentially. The first stage generates region proposals likely to contain objects; the second refines these proposals by performing classification and bounding-box regression. This design yields high accuracy, since the model can focus computation on informative regions, but the multi-step pipeline incurs substantial computational cost, limiting real-time applicability [60].

Single-stage detectors (SSD) adopt a different strategy. Methods such as SSD [61] and YOLO [62] eliminate the intermediate proposal phase and directly predict bounding boxes and class scores in a single forward pass. This unified formulation provides a significant efficiency gain, making single-stage detectors suitable for scenarios where latency is critical [60].

YOLO

Single-stage frameworks such as YOLO (You Only Look Once) follow a structured design composed of three main components: the backbone, the neck and the head. The backbone acts as the primary feature extractor, typically implemented as a deep CNN such as a ResNet variant, and transforms the input image into a hierarchy of progressively abstract representations [63]. The neck aggregates these representations across scales, combining fine-grained spatial information with high-level semantics. Architectures such as the Feature Pyramid Network (FPN) or the Path Aggregation Network (PAN) are commonly employed to enhance multi-scale feature fusion [63]. The detection head performs the final prediction step, outputting bounding boxes, class probabilities and confidence scores.

The most recent evolution of the YOLO family, YOLOv10 [64], introduces a series of architectural and algorithmic refinements that position it among the state-of-the-art detection models. One of its key innovations is the removal of non-maximum suppression (NMS) in post-processing, aligning YOLOv10 with end-to-end transformer-based detectors such as DETR [65]. The adoption of dual label assignment improves training stability and enhances performance across object scales. Additional design elements including lightweight classification heads, spatial channel decoupled downsampling and rank-guided blocks reduce computational overhead while preserving representational power [63, 64]. As a result, YOLOv10 delivers high accuracy and low latency, making it a strong candidate for real-time applications across scientific, industrial and embedded domains.

3.2.5 Attention Mechanisms and Representation Interaction

Attention mechanisms provide a general and flexible framework for modelling adaptive interactions within high-dimensional representations. Originally introduced in the context of sequence modelling in the groundbreaking work of Vaswani et al. [66], attention has since emerged as a unifying computational principle across a wide range of machine learning architectures, including graph neural networks, operator learning frameworks and modern generative models. From a scientific modelling perspective, attention is best understood not as a domain-specific construct, but as an operator that dynamically weights interactions between elements of a representation based on their learned relevance.

The central idea of attention is to replace fixed interaction patterns, such as local receptive fields or predefined neighbourhoods, with data-dependent couplings. Given a set of input representations, attention allows the model to determine which components should interact strongly and which can be down-weighted, thereby enabling the selective aggregation of information. This property is particularly valuable in scientific settings, where relevant dependencies may be long-ranged, non-local, or strongly context-dependent, and cannot be reliably captured through purely geometric or locality-based priors.

Formal definition. In its most common formulation, an attention mechanism operates on three sets of vectors: queries, keys and values. Given matrices $Q \in \mathbb{R}^{N \times d_k}$, $K \in \mathbb{R}^{N \times d_k}$ and $V \in \mathbb{R}^{N \times d_v}$, attention is defined as

$$\text{Att}(Q, K, V) = \text{softmax} \left(\frac{QK^\top}{\sqrt{d_k}} \right) V. \quad (3.3)$$

The dot product $QK^\top$ measures the similarity between queries and keys, while the softmax normalisation produces a set of adaptive weights that sum to unity. The resulting weighted combination of values yields an output in which each element aggregates information from all others according to learned relevance scores.

This formulation can be interpreted as a data-driven kernel operator, where the kernel is not fixed a priori but learned implicitly through the query and key embeddings. Unlike convolutional kernels, which impose translational equivariance and local support, attention kernels are global and adaptive, allowing interactions across the entire representation domain.

Attention as an adaptive aggregation operator. Beyond modelling interactions between individual elements of a representation, attention mechanisms also provide a principled framework for aggregation [67]. In many scientific learning tasks, it is necessary to map high-dimensional, spatially or structurally resolved data to low-dimensional or scalar quantities, such as global material properties, device-level performance metrics, or summary observables. Classical aggregation strategies, including global averaging, summation or max-pooling, impose fixed and uniform weighting schemes that implicitly assume all regions or components of the representation contribute equally to the final prediction.

Attention-based aggregation relaxes this assumption by introducing data-dependent weighting of contributions [68]. Formally, the attention weights define a learned measure over the elements of the representation, enabling the model to emphasise regions, features or interactions that are most relevant to the target quantity while suppressing less informative components [67]. From this perspective, attention pooling can be interpreted as a soft, learnable generalisation of global pooling operators.

This adaptive aggregation property is particularly relevant in physical systems characterised by spatial or structural heterogeneity. In atomistic systems, only a subset of atoms or bonds may dominate a given electronic or transport response. In mesoscale fields, localised regions such as interfaces, percolation pathways or high-field domains may exert a disproportionate influence on global device behaviour. Attention-based aggregation enables models to capture such effects without requiring explicit localisation rules or handcrafted importance masks.

Crucially, attention-based aggregation operates at the level of representations rather than physical equations. It therefore integrates naturally with data-driven surrogate models and operator learning frameworks, providing a flexible mechanism for summarising complex physical states into task-relevant outputs. In this role, attention acts not only as an interaction mechanism but also as an information bottleneck that preserves physically salient contributions while reducing dimensionality.

3.3 Gradient Boosting Models and Explainable AI

3.3.1 The intuition behind boosting

The conceptual foundation of boosting originates from a question posed by Kearns in his seminal work on computational learning theory [69]:

Can a collection of weak predictors be combined to form a strong predictor?

This question captures the central idea behind boosting algorithms. Rather than relying on a single highly accurate model, boosting constructs a powerful predictor by sequentially combining multiple weak learners, each of which performs only slightly better than random guessing. Through iterative reweighting of the training samples and targeted correction of previous residual errors, boosting synthesises

an ensemble whose collective predictive capability significantly exceeds that of any individual component. In this way, the weak-to-strong transition becomes a principled mechanism for achieving high accuracy and robust generalisation from models that, on their own, would be insufficient.

3.3.2 Decision Tree models

Decision trees form the core building blocks of boosting algorithms and represent one of the most widely used models in supervised learning [70]. They can be employed for both classification and regression and operate by recursively partitioning the feature space into axis-aligned regions, each associated with a locally constant prediction. As the depth of the tree increases, the resulting model is able to represent increasingly complex, non-linear decision rules by successively refining these partitions.

A major advantage of decision trees is their interpretability: the hierarchical structure makes the learned rules transparent and easy to visualise, and the model naturally accommodates both numerical and categorical features with minimal data preprocessing. Trees also handle multi-output problems and can model heterogeneous feature interactions without requiring strong parametric assumptions. These properties make them attractive as weak learners within ensemble methods such as boosting, where many shallow trees collectively achieve high predictive accuracy [70]. However, decision trees also exhibit well-known limitations. When grown to excessive depth, they tend to overfit the training data, capturing noise rather than underlying structure; pruning and depth constraints are therefore essential to maintain generalisation. Trees are also unstable with respect to small perturbations in the data, often producing different structures under minor sampling variations. Moreover, their piecewise-constant prediction surfaces limit their ability to extrapolate smoothly beyond the observed domain. Learning an optimal decision tree is an NP-complete problem, and practical algorithms rely on greedy heuristics that do not guarantee global optimality. These limitations motivate the use of ensemble techniques, such as random forests and gradient boosting, which stabilise predictions and significantly enhance performance by aggregating multiple weak learners. In Figure 3.4, a simple two-dimensional classification problem is visualised through the decision surface induced by a decision tree. The plot illustrates how the model partitions the feature

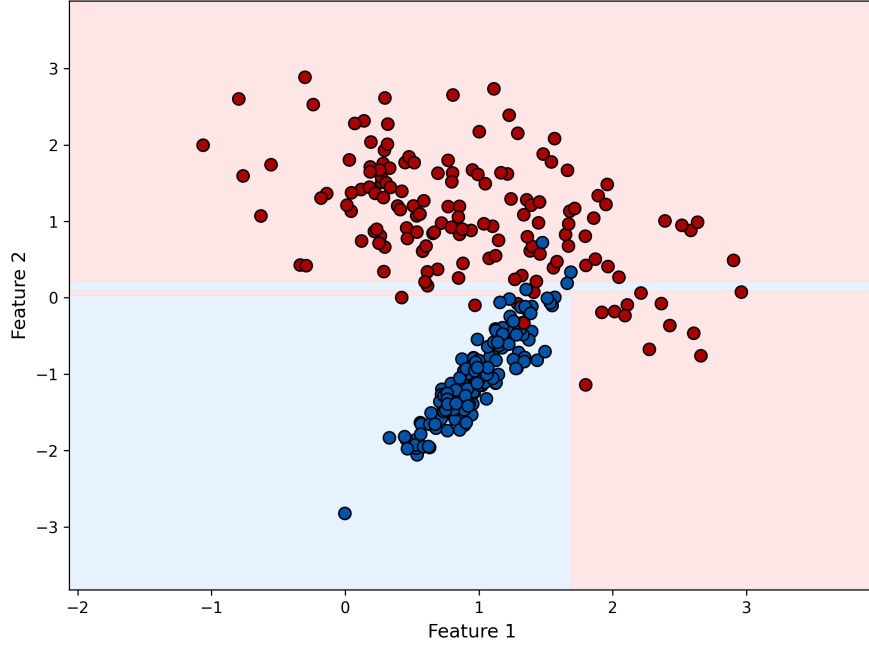


Fig. 3.4 Illustration of the decision boundaries induced by a decision tree classifier trained on a two-dimensional synthetic dataset. The model partitions the feature space into axis-aligned rectangular regions, each associated with a constant prediction. This piecewise structure reflects the hierarchical sequence of thresholding operations performed at the internal nodes of the tree. Decision trees therefore produce interpretable and locally uniform regions whose geometry provides insight into how feature values drive classification.

space into axis-aligned regions, each corresponding to a constant prediction, thereby providing an intuitive representation of the hierarchical rules learned from the data.

3.3.3 Extreme Gradient Boosting Algorithm

Extreme Gradient Boosting (XGBoost) [46] is a scalable and regularised implementation of gradient boosting that has become a standard tool for tabular predictive modelling. Like classical boosting methods, XGBoost constructs an additive model of the form

$$\hat{y}(x) = \sum_{m=1}^M f_m(x), \quad f_m \in \mathcal{F}, \quad (3.4)$$

where each f_m is a regression tree drawn from a functional space $\mathcal{F}$ of weak learners. The model is trained by iteratively adding trees that minimise a regularised

objective function

$$\mathcal{L}^{(t)} = \sum_{i=1}^n \ell(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t), \quad (3.5)$$

where ℓ is a differentiable loss function and $\Omega(f_t)$ is a structural penalty that controls the complexity of the newly added tree. XGBoost employs a second-order

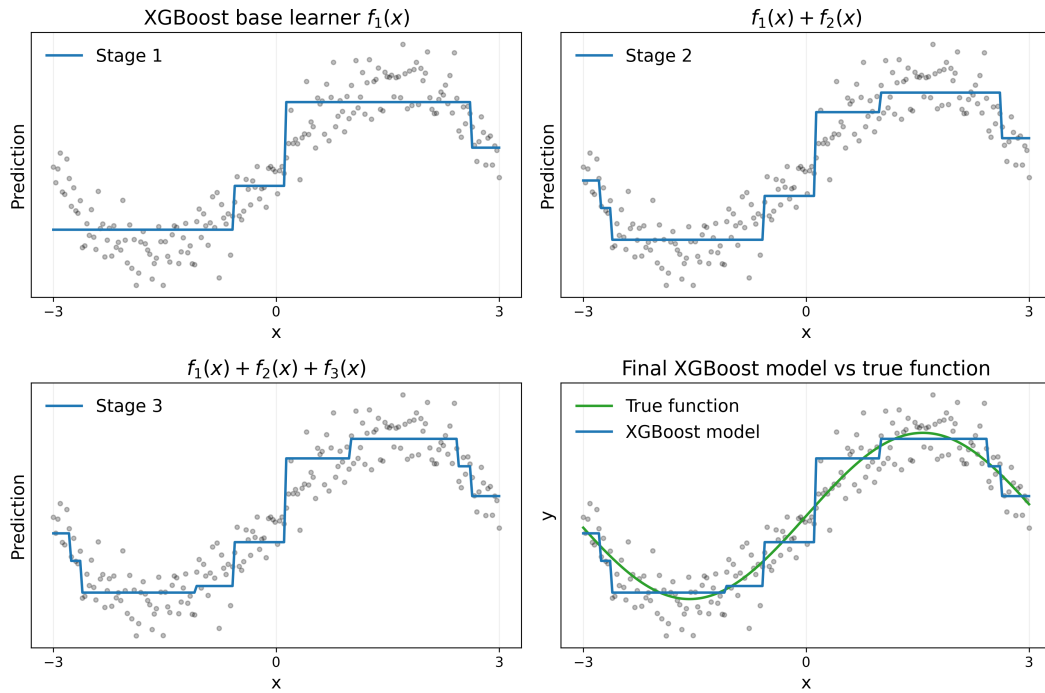


Fig. 3.5 Illustration of the additive nature of gradient boosting on a one-dimensional regression task. Each panel shows the cumulative model after successively adding weak learners f_1, f_2, f_3 , each trained to correct the residuals of the previous stage. The final boosted model approximates the underlying target function by summing a sequence of shallow regression trees, demonstrating the core weak-to-strong principle of boosting.

Taylor expansion of the loss around the current prediction, yielding a closed-form optimisation target involving the gradients g_i and Hessians h_i of the loss. This leads to an efficient node-splitting criterion and enables the algorithm to scale gracefully to large datasets.

A key strength of XGBoost lies in its regularisation mechanisms, which include penalties on the number of leaves and the magnitude of leaf weights, making the method significantly more robust to overfitting than traditional gradient boosting. Additional engineering improvements such as sparsity-aware split finding, column

subsampling and parallelised tree construction; further enhance computational efficiency. These properties make XGBoost particularly well suited for scientific modelling tasks in which heterogeneous, descriptor-based feature sets must be mapped to complex non-linear responses, as encountered in the defect–property relationships discussed in the chapter 6 of this work.

Figure 3.5 illustrates the principle introduced by Kearns [69], namely whether a collection of weak predictors can be combined to form a strong learner. The progressive refinement shown across the boosting stages demonstrates precisely this idea: each weak regressor contributes a small corrective step, and their additive combination yields a final model capable of accurately approximating the underlying target function.

3.3.4 Explainability of Tree Models and Introduction to XAI

Decision-tree models are not only attractive for their simplicity and flexibility, but also for the level of interpretability they provide. Unlike many machine-learning architectures that operate as black boxes, decision trees expose their internal decision process through a sequence of axis-aligned splits. This structure enables a natural notion of *feature importance*, offering insight into how individual features contribute to the predictive performance of the model.

In impurity-based importance, which is the standard in tree ensembles such as random forests and gradient boosting, the relevance of a feature is quantified through the reduction of node impurity it achieves when used for a split. Let $I(\cdot)$ denote an impurity measure, such as Gini, entropy or variance in classification tasks, and mean squared error (MSE) or loss reduction in regression settings [71]. Consider a node j with N_{parent} samples and impurity I_{parent} . A split on feature f produces two child nodes with impurities I_{left} and I_{right} and sample counts N_{left} and N_{right} . The reduction in impurity is given by

$$\Delta I_j(f) = I_{\text{parent}} - \left(\frac{N_{\text{left}}}{N_{\text{parent}}} I_{\text{left}} + \frac{N_{\text{right}}}{N_{\text{parent}}} I_{\text{right}} \right) \quad (3.6)$$

The overall importance of feature f in a tree is then obtained by summing this quantity over all nodes in which f is used,

$$\text{FI}(f) = \sum_{j \in \mathcal{S}(f)} \Delta I_j(f) \quad (3.7)$$

and subsequently normalising across all features. Although simple and computationally efficient, this measure is known to be biased toward features with many potential split points and does not capture interactions or non-additive effects between inputs.

These limitations motivate the use of more principled Explainable AI (XAI) [72, 73] techniques, particularly in scientific modelling where understanding structure–property relationships is essential. Among the available methods, SHAP [74, 75] (SHapley Additive exPlanations) has emerged as a theoretically grounded framework for attributing predictions to input features. SHAP values are based on cooperative game theory and assign to each feature the contribution it provides to a model’s output, averaged over all possible subsets of features. Formally, for a prediction function $f(x)$ and input feature i , the SHAP value is defined as

$$\phi_i(x) = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|! (|F| - |S| - 1)!}{|F|!} [f(S \cup \{i\})(x) - f(S)(x)], \quad (3.8)$$

where F is the full set of features and S ranges over all coalitions not containing feature i . This formulation ensures desirable axiomatic properties, including efficiency, symmetry, dummy and additivity, making SHAP one of the most rigorous frameworks for interpretability.

In the context of this thesis, SHAP values complement impurity-based feature importance by providing a local, model-agnostic and mathematically principled characterisation of how structural descriptors or morphological features influence the behaviour of tree-based predictive models. This dual perspective global; importance through impurity reduction and local contributions through SHAP offers a comprehensive interpretability layer for analysing defect–property relationships in graphene (see Section 6.4).

3.4 Operator Learning for PDE Approximation

3.4.1 Scientific Machine Learning (SciML)

Scientific Machine Learning (SciML) has emerged as a unified research area combining data-driven modelling with the principles of scientific computing, numerical analysis and dynamical systems. Traditional machine learning focuses on learning finite-dimensional mappings, typically functions of the form $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$. However, many problems in the physical sciences are not governed by finite-dimensional relationships but by operators: mappings between infinite-dimensional function spaces. Examples include differential operators arising in continuum mechanics, electromagnetism, fluid dynamics or charge transport in semiconductor and organic devices [3]. According to the work proposed by Boullé et al. [76], scientific machine learning can be organised into three broad methodological categories: *PDE solvers*, *PDE discovery*, and *operator learning*. Although these areas share common foundations in data-driven and physics-informed modelling, they address fundamentally different questions in computational science.

PDE solvers aim to approximate the solution of a given differential equation more efficiently than traditional numerical schemes. Examples include physics-informed neural networks (PINNs), neural Galerkin methods and learning augmented solvers, which incorporate differentiable models into classical numerical frameworks. These approaches typically focus on reproducing single-instance solutions under fixed discretisations or boundary conditions, making them attractive for accelerating forward simulations.

PDE discovery addresses the inverse problem of identifying the underlying physical laws from data. Methods in this class seek to recover the structure of an equation its differential operators, coefficients or constitutive relations directly from measurements or simulation outputs. Sparse regression approaches such as SINDy, symbolic neural networks and library-based discovery methods fall into this category, enabling automated extraction of interpretable governing equations [77, 78].

Operator learning, by contrast, aims to approximate the full solution operator that maps input functions (e.g., boundary conditions, forcing terms, material parameters or geometries) to output fields. Rather than learning a single solution or identifying an equation, operator-learning architectures learn the entire functional

mapping

$$\mathcal{G} : u(x) \mapsto v(x),$$

defined over infinite-dimensional spaces. Neural operators, including the Fourier Neural Operator (FNO) [79, 40], DeepONet and their variants, are specifically designed to generalise across discretisations and domains, providing fast and resolution-independent surrogates for complex multi-physics systems [80].

This tripartite classification highlights the breadth of scientific machine learning and situates operator learning as a distinct and powerful framework. Unlike PDE solvers or discovery tools, operator-learning models are inherently suited for tasks such as design exploration, uncertainty quantification and multi-query simulation, all of which are essential in materials science and device modelling. As such, operator-learning methods constitute the foundation of the surrogate modelling strategies employed in the subsequent chapters.

3.4.2 Surrogate Modelling for Physical Simulations

Surrogate models aim to approximate the behaviour of computationally expensive physical solvers with fast, data-driven predictors. In many areas of scientific computing such as atomistic simulations, drift–diffusion device modelling or multi-physics transport; high-fidelity numerical methods provide accurate but costly solutions, often requiring minutes to hours of computation per query. This prohibits their use in tasks that demand repeated evaluation, including design exploration, uncertainty quantification, optimisation and real-time prediction.

Machine learning offers an alternative by learning input–output relationships from collections of simulated or experimental data. Classical surrogate models (such as kernel regressors or gradient boosting methods) operate in finite dimensions and can be effective when the underlying physical map can be captured through handcrafted features. However, many scientific problems involve fields defined over continuous domains potentials, carrier densities, morphological configurations or force landscapes which cannot be represented adequately through low-dimensional feature vectors alone.

3.4.3 Fourier Neural Operator

Neural operators provide a framework for learning mappings between infinite-dimensional function spaces [40]. Instead of approximating a finite-dimensional map $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$, a neural operator aims to learn a solution operator of the form

$$\mathcal{G} : u(x) \mapsto v(x), \quad x \in \Omega \subset \mathbb{R}^d \quad (3.9)$$

where u and v are functions defined over a continuous spatial domain. This formulation is particularly relevant for physical systems governed by partial differential equations, where the mapping from inputs (e.g. boundary conditions, coefficients, geometries or source terms) to outputs (e.g. fields, fluxes, or potentials) is naturally operator-valued.

A generic neural operator approximates $\mathcal{G}$ by stacking local non-linear transformations with global integral operators. In its continuous formulation, a single operator layer takes the form

$$(\mathcal{K}(u))(x) = \int_{\Omega} \kappa_{\theta}(x, y) u(y) dy,$$

where κ_{θ} is a learnable kernel. The complete operator is obtained by composing several such layers with pointwise non-linearities. After discretisation on a grid $\{x_i\}_{i=1}^N$, the integral operator becomes

$$(\mathcal{K}(u))(x_i) = \sum_{j=1}^N \kappa_{\theta}(x_i, x_j) u(x_j),$$

but learning a fully parameterised kernel $\kappa_{\theta} \in \mathbb{R}^{N \times N}$ is computationally prohibitive for practical resolutions.

The *Fourier Neural Operator* (FNO) addresses this difficulty by learning the kernel κ_{θ} in Fourier space [79]. Instead of representing the kernel explicitly, the FNO parameterises its action through multiplication by a truncated set of Fourier modes. For an input feature field u , the Fourier operator layer is defined as

$$\mathcal{K}_{\theta}(u) = \mathcal{F}^{-1} \left(R_{\theta}(\mathcal{F}(u)) \right),$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the forward and inverse Fourier transforms, and R_θ is a learnable complex-valued weight matrix restricted to the lowest k Fourier modes. High-frequency components are discarded, providing a spectral regularisation that promotes smoothness and stability in the learned operator.

An FNO layer combines this global spectral operator with a local linear transformation (Visual representation of the layer update can be seen in the image ??):

$$u_{t+1}(x) = W u_t(x) + \mathcal{F}^{-1} \left(R_\theta(\mathcal{F}(u_t)) \right) \quad (3.10)$$

where W is a pointwise linear map. The architecture is completed by stacking multiple layers with non-linear activations, yielding an efficient operator approximator that is resolution-invariant and able to generalise across different discretisations of the same PDE.

The computational advantage of the FNO stems from the use of fast Fourier transforms (FFTs), which reduce the cost of the global operator from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$. This makes FNOs particularly suitable for high-dimensional fields such as those encountered in fluid mechanics, electromagnetism, or device-scale drift-diffusion modelling.

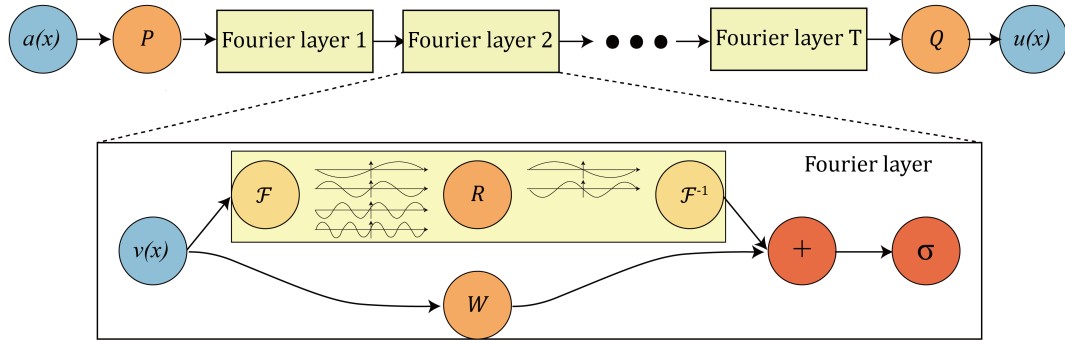


Fig. 3.6 Fourier Neural Operator architecture representation. Reprinted from [40].

In the context of this thesis, Fourier Neural Operators form the basis of the surrogate solvers developed for predicting electric fields, carrier densities and current distributions in bulk-heterojunction organic photovoltaic devices. Their ability to learn resolution-invariant mappings between morphological inputs and multi-field outputs is essential for constructing fast, accurate and generalisable surrogate models.

Zero-shot super-resolution Zero-shot super-resolution refers to the ability of a learned surrogate model to produce predictions on spatial discretisations finer than those observed during training, without requiring additional retraining or fine-tuning. From an operator-learning perspective, this capability implicitly assumes that the training data provide a consistent discretised representation of an underlying continuous mapping, such that evaluations at different resolutions correspond to different samplings of the same continuous solution.

Let $\mathcal{G}$ denote a continuous operator acting on functions defined over a domain Ω , and let $\mathcal{G}_h$ represent its numerical approximation obtained on a grid with spacing h . Zero-shot super-resolution assumes that the discrete solutions generated at a given resolution can be interpreted as samples of a resolution-independent function $u(\mathbf{x})$, and that the learned surrogate effectively approximates $\mathcal{G}$ rather than a discretisation-specific operator. Under this assumption, predictions on a finer grid $h' < h$ amount to evaluating the learned mapping on a denser sampling of the input domain.

In practice, this assumption is only valid when the numerical solutions used for training converge uniformly to the continuous solution as the grid is refined. For standard numerical schemes, the discretised operator satisfies

$$\mathcal{G}_h = \mathcal{G} + \mathcal{O}(h^p), \quad (3.11)$$

where p depends on the order of the discretisation. At finite and especially coarse resolutions, the approximation error may significantly alter the effective operator encoded in the data. As a result, solutions computed at different resolutions cannot, in general, be regarded as simple resamplings of the same continuous function, but instead correspond to different resolution-dependent operators.

Consequently, zero-shot super-resolution should be understood as a conditional property that depends on the numerical consistency of the dataset across scales. When the training data do not preserve scale-consistent physical or mathematical features, the surrogate model is required to extrapolate across discretisation levels rather than interpolate within a well-defined operator family. In such cases, the lack of explicit multi-resolution supervision limits the reliability and physical interpretability of zero-shot super-resolution, particularly with respect to the reconstruction of unresolved high-frequency components [79, 40, 81].

3.5 Generative Modelling for Materials and Morphology Design

Generative modelling has emerged as one of the most transformative developments in modern machine learning, offering the ability not only to analyse data but to synthesise new samples that follow the underlying statistical structure of a domain [82]. In materials science and device engineering, this capability is particularly significant. Physical systems such as defective graphene, molecular crystals, or bulk-heterojunction morphologies occupy vast and highly structured configuration spaces, where only a negligible fraction of all admissible configurations has ever been observed or simulated. Generative models provide a principled way to explore these spaces, construct new candidates with controlled properties and support inverse design workflows in which the model proposes structures rather than merely evaluating them [83]

In this thesis, generative modelling plays a dual conceptual role. First, it provides a compact probabilistic description of high-dimensional scientific datasets, enabling the discovery of latent variables that capture essential structural or physicochemical degrees of freedom. Second, it enables the synthesis of new atomic configurations or device morphologies that are consistent with physical constraints and fall within realistic regions of the data manifold. These capabilities are particularly relevant in settings where simulation-based data acquisition is expensive, as in atomistic electronic-structure calculations or drift–diffusion simulations of organic photovoltaic devices.

A range of architectures contribute to this modelling paradigm. Autoencoder-based models [47] provide low-dimensional embeddings that serve as a basis for controlled interpolation and morphing of configurations. Variational autoencoders extend this framework to a fully probabilistic setting, in which sampling from the latent space yields novel yet statistically coherent structures. More recently, diffusion models have become the state of the art in high-fidelity generative tasks [84]. Their iterative denoising dynamics enable the synthesis of complex, high-dimensional scientific data, including atomic geometries and functional fields, while naturally supporting conditional generation. Such models are particularly suited to the design of materials and morphologies where generative sampling must respect subtle geometric correlations or long-range physical constraints.

The methods introduced in this section form the conceptual foundation for the generative workflows developed in subsequent chapters, where they are applied to the creation of new defective graphene structures and to the exploration of morphology design spaces in bulk-heterojunction organic photovoltaics. By combining structured latent spaces with conditional generative mechanisms, these models support a data-driven approach to materials and device optimisation that complements the operator-learning and predictive frameworks presented earlier in this chapter.

3.5.1 Variational autoencoders

Variational autoencoders (VAEs) extend the classical autoencoder paradigm by replacing the deterministic latent representation with a probabilistic one. Instead of encoding an input into a single latent vector z , a VAE maps it to a probability distribution, typically a diagonal multivariate Gaussian characterised by its mean and variance [85, 86]. This formulation enables controlled sampling from the latent space and allows the model to generate new data points, making VAEs suitable for generative modelling and design-space exploration.

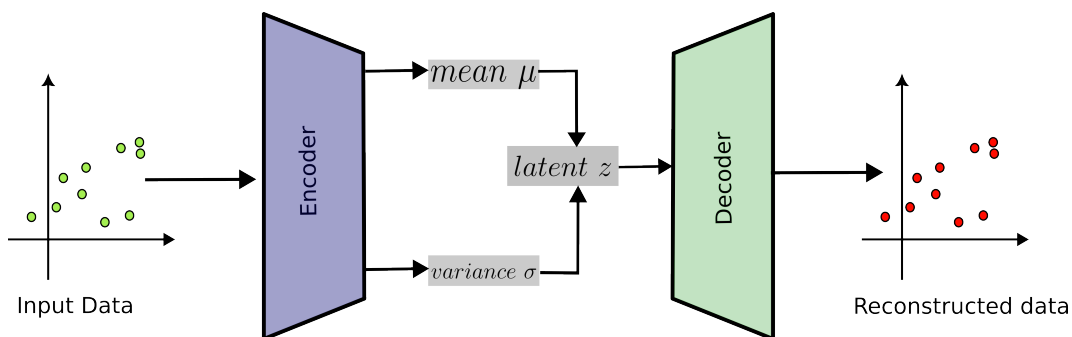


Fig. 3.7 Schematic architecture of a Variational Autoencoder (VAE). The encoder maps the input data distribution (left) into a low-dimensional latent representation, parameterised probabilistically. Latent samples drawn from this representation are then passed to the decoder, which reconstructs the data in the original space (right). The stochastic latent space enables smooth interpolation and generative sampling while enforcing a regularised representation of the input distribution.

In a VAE, the encoder does not output a single latent code but rather the parameters ($z_{\text{mean}}, z_{\text{var}}$) describing the distribution of the latent variable. A sample is then

obtained through the reparameterisation

$$z = z_{\text{mean}} + z_{\sigma} \cdot \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, 1),$$

where $z_{\sigma} = \exp(0.5 \cdot \log(z_{\text{var}}))$ ensures a numerically stable parameterisation of the variance. The random variable ε injects stochasticity into the latent representation, while the affine transformation keeps the sampling operation differentiable. This technique, known as the *reparameterisation trick* [87], enables end-to-end training of the model using gradient-based optimisation.

Training a VAE involves two complementary objectives. The first is the reconstruction loss, which encourages the decoder to faithfully reproduce the input x from the sampled latent variable z . The second is a regularisation term that shapes the latent space by enforcing proximity between the learned distribution and a unit Gaussian prior. This penalty is expressed through the Kullback–Leibler divergence

$$L_{KL} = D_{KL}(\mathcal{N}(z_{\text{mean}}, z_{\sigma}^2) \parallel \mathcal{N}(0, 1)) = \frac{1}{2} \sum_i [z_{\sigma,i}^2 + z_{\text{mean},i}^2 - 1 - \log(z_{\sigma,i}^2)].$$

The combination of reconstruction fidelity and latent regularisation yields the total VAE loss

$$L_{\text{tot}} = L_{\text{reconstruction}} + L_{KL}.$$

This probabilistic structure endows VAEs with several desirable properties. The latent space becomes smooth and continuous, allowing meaningful interpolation between data points and robust exploration of variations that remain consistent with the data manifold. Moreover, the regularisation term prevents degenerate latent encodings, promoting global coherence and interpretability of the learned representations.

VAEs have demonstrated strong performance across diverse domains. In image modelling, they support the generation of novel and diverse samples, as well as structured manipulation of visual attributes [88]. In audio processing, VAE-based architectures produce expressive latent spaces that capture speaker identity and prosodic attributes, enabling flexible synthesis pipelines [89]. In computational chemistry and drug design, the ability to sample from a continuous latent manifold has facilitated the generation of candidate molecules with tailored physicochemical

properties, accelerating optimisation workflows and inverse-design strategies [90, 91].

3.5.2 Diffusion Models

Diffusion models constitute one of the most influential developments in modern generative modelling, providing a principled probabilistic framework capable of synthesising highly structured, high-dimensional data. Their formulation is built around two complementary Markov processes [92]: a forward diffusion process that gradually destroys structure in the data, and a learned reverse process that reconstructs samples by inverting this corruption.

Forward diffusion process. Given an input sample x_0 , the forward diffusion process defines a Markov chain $\{x_t\}_{t=1}^T$ in which small amounts of Gaussian noise are added at each step. The transition $q(x_t | x_{t-1})$ that corrupt the data, introduced in the work of Ho et al. [93]

$$q(x_t | x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{1 - \beta_t} x_{t-1}, \beta_t I\right) \quad (3.12)$$

encodes two effects. First, the mean is scaled by $\sqrt{1 - \beta_t}$, which gradually attenuates the signal contained in x_{t-1} . Second, a Gaussian perturbation with variance $\beta_t I$ is added, ensuring that at each step the sample becomes slightly more noisy. The parameters $\beta_t \in (0, 1)$ form a variance schedule, typically increasing with t , so that later diffusion steps inject larger amounts of noise.

Because the forward process is linear and Gaussian, the marginal distribution at any time step t can be computed in closed form. By iterating the transitions one obtains

$$q(x_t | x_0) = \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t) I\right), \quad (3.13)$$

where the accumulated attenuation factor is

$$\bar{\alpha}_t = \prod_{s=1}^t (1 - \beta_s). \quad (3.14)$$

This expression reveals the structure of the forward diffusion process. The mean term $\sqrt{\bar{\alpha}_t} x_0$ shows that the contribution of the original signal decreases exponentially

with t , while the variance term $(1 - \bar{\alpha}_t)I$ represents the cumulative noise injected along the chain. In the limit of large t , the product $\bar{\alpha}_t$ tends to zero and the marginal converges to

$$q(x_T | x_0) \approx \mathcal{N}(0, I), \quad (3.15)$$

meaning that x_T is almost indistinguishable from pure Gaussian noise (see Fig. 3.8 for the step by step corruption process of an image).

This property is essential for generative modelling: by pushing the data distribution into an isotropic Gaussian, the model obtains a tractable and well-defined starting point from which the learned reverse process can reconstruct meaningful samples.

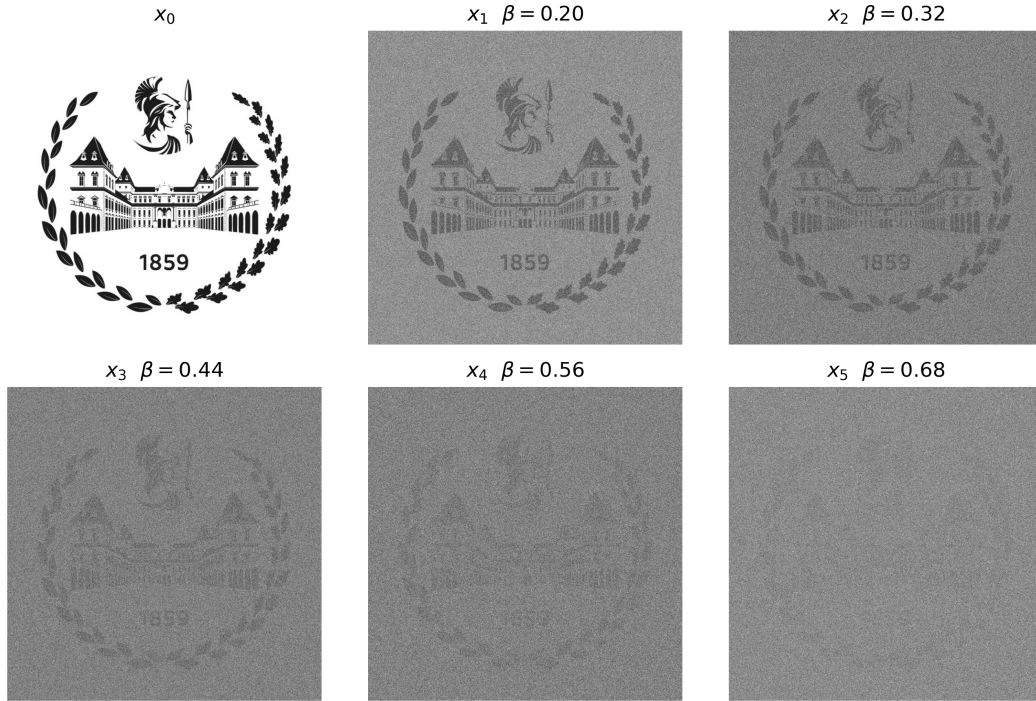


Fig. 3.8 Forward diffusion applied to an image. The sequence $x_0, x_1, \dots, x_T$ demonstrates the progressive attenuation of the signal as Gaussian noise with variance β_t ; that follow a linear schedule across the diffusion steps; is added at each step. The visible structure is rapidly lost, highlighting the tendency of the forward process to converge towards an isotropic Gaussian distribution. The diffusion procedure was implemented by the author.

Markov structure of the forward process. The forward diffusion process forms a first-order Markov chain. Although the closed-form marginal distribution $q(x_t | x_0)$

depends on all previous noise increments through the cumulative factor $\bar{\alpha}_t$, the underlying stochastic dynamics do not. Each transition is defined as

$$q(x_t | x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t} x_{t-1}, \beta_t I), \quad (3.16)$$

which shows that x_t depends only on the immediately preceding state x_{t-1} and an independent Gaussian perturbation. This implies

$$p(x_t | x_{t-1}, x_{t-2}, \dots, x_0) = p(x_t | x_{t-1}),$$

fulfilling the Markov property. The apparent dependence on the entire history in the marginal distribution arises solely from analytically integrating the sequence of Markov transitions, not from any memory in the process itself. This distinction is essential: the Markov formulation provides a tractable generative mechanism, while the closed-form marginal clarifies how signal attenuation and accumulated noise evolve over time.

Reverse denoising process. The generative mechanism of diffusion models is defined by a reverse Markov chain, introduced in the same work of forward diffusion [93]

$$p_\theta(x_{t-1} | x_t), \quad (3.17)$$

which inverts the forward corruption process. The exact reverse posterior

$$q(x_{t-1} | x_t, x_0) \quad (3.18)$$

is analytically known but depends on the unknown clean sample x_0 , and is therefore unusable during sampling.

Denoising diffusion probabilistic models (DDPMs) [93] circumvent this difficulty by exploiting the closed-form forward marginal

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I) \quad (3.19)$$

which shows that predicting the injected noise ε is equivalent to estimating the clean component of x_t . A neural network $\varepsilon_\theta(x_t, t)$ is therefore trained to approximate the

noise, leading to the stable objective

$$\mathcal{L}_{\text{DDPM}} = \mathbb{E}_{x_0, t, \varepsilon} \left[\left\| \varepsilon - \varepsilon_\theta(x_t, t) \right\|^2 \right]. \quad (3.20)$$

Given a noise estimate, the reverse transition $p_\theta(x_{t-1} | x_t)$ is a Gaussian

$$p_\theta(x_{t-1} | x_t) = \mathcal{N}(\mu_\theta(x_t, t), \sigma_t^2 I) \quad (3.21)$$

with mean

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \alpha_t}} \varepsilon_\theta(x_t, t) \right) \quad (3.22)$$

and variance $\sigma_t^2 = \tilde{\beta}_t$ determined by the noise schedule.

Sampling proceeds by iterating

$$x_{t-1} = \mu_\theta(x_t, t) + \sigma_t z, \quad z \sim \mathcal{N}(0, I), \quad (3.23)$$

starting from $x_T \sim \mathcal{N}(0, I)$. Iteratively applying these denoising transitions transforms pure Gaussian noise into a coherent image.

In this formulation, the neural network does not generate an image directly. Its sole role is to estimate the noise component present in x_t , while the generative behaviour emerges from the analytical structure of the Gaussian reverse transitions.

Connection to score-based generative modelling. Song and Ermon [94, 95] showed that diffusion models and score-based generative models are mathematically equivalent. In the continuous-time limit, the forward diffusion becomes a stochastic process governed by the Itô SDE (detailed explanation can be found in Appendix A.5)

$$dx_t = f(t)x_t dt + g(t) dW_t \quad (3.24)$$

whose marginal densities satisfy a diffusion equation. The corresponding reverse-time dynamics is given by

$$dx_t = \left(f(t)x_t - g(t)^2 \nabla_{x_t} \log q_t(x_t) \right) dt + g(t) d\bar{W}_t \quad (3.25)$$

where $\nabla_{x_t} \log q_t(x_t)$ denotes the score function of the marginal density $q_t(x_t)$ and $\bar{W}_t$ is a reverse-time Wiener process.

The neural network $s_\theta(x_t, t)$ is trained to approximate the score function,

$$s_\theta(x_t, t) \approx \nabla_{x_t} \log q_t(x_t) \quad (3.26)$$

and generative sampling corresponds to solving either the reverse SDE (stochastic generation) or its associated probability-flow ODE (deterministic generation). This viewpoint unifies DDPMs, DDIMs and score-based models, revealing them as different discretisations of the same underlying diffusion process.

Latent diffusion and modern architectures. Latent diffusion models [96], including Stable Diffusion, extend this framework by operating in a low-dimensional latent space instead of pixel space. A variational autoencoder encodes an image into a latent representation

$$z_0 = \text{Enc}(x_0),$$

and the diffusion process is applied to $(z_t)_{t=0}^T$ rather than to $(x_t)_{t=0}^T$. The reverse transitions retain the DDPM form,

$$z_{t-1} = \mu_\theta(z_t, t) + \sigma_t z, \quad z \sim \mathcal{N}(0, I),$$

while conditioning is introduced through text embeddings produced by a transformer encoder (e.g., CLIP).

The final image is reconstructed via the decoder,

$$\hat{x}_0 = \text{Dec}(z_0).$$

Operating in latent space drastically reduces computational cost and memory requirements, enabling high-resolution generation and flexible conditioning mechanisms. This latent formulation currently defines the state of the art in large-scale generative modelling.

Relevance for scientific and materials modelling. Diffusion models are particularly suitable for atomistic and mesoscale scientific data. Their iterative denoising dynamics preserve local geometric correlations, allowing them to generate physically realistic fields, molecular geometries or morphological patterns. Conditional diffusion offers a flexible mechanism for inverse design, enabling the generation

of samples that satisfy target structural, energetic or device-level constraints. This makes diffusion models an essential tool for exploring high-dimensional configuration spaces where traditional simulation is computationally prohibitive.

3.5.3 Conditioning the Generative Process

In the previous sections, generative models have been introduced and discussed in detail, with particular emphasis on Variational Autoencoders and diffusion-based approaches. Both model families provide powerful frameworks for learning complex data distributions and for synthesising novel samples that are statistically consistent with the training data. However, a fundamental limitation of unconditional generative modelling is that the generation process is inherently unconstrained: samples are drawn freely from the learned distribution, without explicit control over their semantic or physical characteristics.

In many scientific and engineering applications, this unconstrained generation is insufficient. Rather than producing arbitrary samples from the data manifold, it is often necessary to target specific regions of the latent or data space and to generate samples that satisfy predefined conditions or constraints. Typical examples include the synthesis of materials with prescribed physical properties, the generation of signals under known boundary conditions, or the reconstruction of structures compatible with given low-dimensional descriptors. To address this limitation, conditioning mechanisms are introduced into generative models, enabling controlled and directed sample generation.

Conditioned generative modelling extends the standard formulation by incorporating auxiliary information into the generation process. Instead of learning an unconditional distribution $p_{\theta}(x)$, the model is trained to approximate a conditional distribution of the form

$$p_{\theta}(x | c), \quad (3.27)$$

where x denotes the generated sample and c represents one or more conditioning variables. These conditioning variables may encode class labels, physical parameters, low-dimensional embeddings, or any set of features that characterise the desired region of the data space. Conditioning strategies have been successfully applied across a wide range of generative models, including Conditional Variational Autoen-

coders and conditional diffusion models, demonstrating improved controllability and interpretability of the generated outputs [97].

From a probabilistic perspective, conditioning modifies the generative objective by explicitly modelling the dependency between the data and the conditioning variables. In the case of a Variational Autoencoder, the Evidence Lower Bound (ELBO) becomes

$$\mathcal{L}_{\text{VAE}} = \mathbb{E}_{q_\phi(z|x,c)} [\log p_\theta(x | z, c)] - \text{KL} (q_\phi(z | x, c) \parallel p(z | c)), \quad (3.28)$$

where z denotes the latent variables, q_ϕ the approximate posterior, and p_θ the conditional decoder. The conditioning information c influences both the latent inference and the reconstruction process, thereby shaping the structure of the latent space and constraining the generated samples.

Similarly, in diffusion-based generative models, conditioning is introduced by extending the reverse process to depend explicitly on auxiliary variables. Given a forward diffusion process $q(x_t | x_0)$, the learned reverse transitions are modelled as

$$p_\theta(x_{t-1} | x_t, c), \quad (3.29)$$

where the conditioning vector c guides the denoising dynamics at each timestep. In practice, this is achieved by injecting the conditioning information into the neural network through concatenation, adaptive normalisation layers, or feature-wise modulation mechanisms. This formulation enables the model to progressively steer the generation towards samples that are not only consistent with the learned data distribution but also compatible with the imposed conditions.

Overall, conditioning transforms generative models from purely exploratory tools into controllable and goal-oriented frameworks. By explicitly incorporating auxiliary descriptors into the generative process, conditioned models provide a principled mechanism for targeted sample synthesis, significantly expanding the applicability of generative modelling in scientific and data-driven contexts.

3.6 Computational material representation

Over the years, the development of computational chemistry has led to the introduction of a wide range of methods for representing materials in silico. With the advent of artificial intelligence, these representations have been either adapted or directly adopted for data-driven learning on materials systems.

It is therefore essential to distinguish between material representations across different length and time scales. As will become clear in the following sections, this multi-scale perspective constitutes one of the central pillars of the present thesis.

3.6.1 Atomistic Representation

Coulomb Matrix The Coulomb matrix is a matrix-based molecular descriptor originally introduced by Rupp et al. [98] and represents one of the earliest systematic attempts to encode molecular structures for use in machine-learning applications. It is defined as a symmetric matrix M , whose elements M_{ij} quantify the electrostatic interaction between atoms i and j within a molecule.

$$M_{ij} = \begin{cases} 0.5Z_i^{2.4} & \forall i = j \\ \frac{Z_i Z_j}{|R_i - R_j|} & \forall i \neq j \end{cases}$$

where Z_i represents the nuclear charge of the i -th atom, while R_i represents its Cartesian coordinates. The off-diagonal elements of the matrix represent the Coulombic repulsion between atoms i and j , whereas the diagonal elements capture a polynomial approximation of the atomic energies as a function of the nuclear charges, specifically $0.5Z_i^{2.4}$ [99, 100].

One of the primary advantages of the Coulomb matrix lies in its ability to simultaneously encode both the atomic composition and the spatial arrangement of atoms within a molecule. This dual encoding makes the Coulomb matrix a powerful and conceptually simple descriptor for representing molecular systems in a form that is directly compatible with standard machine-learning algorithms.

However, this representation also exhibits a significant limitation, which is explicitly addressed in this thesis. The dimensionality of the Coulomb matrix scales quadratically with the number of atoms, growing as N^2 . This scaling rapidly becomes

computationally prohibitive for large molecular systems and extended materials, such as graphene flakes, where the number of atoms can be very large.

To mitigate this issue, several strategies have been proposed in the literature. A common approach consists in computing the eigenvalue spectrum of the Coulomb matrix and using the resulting eigenvalues as a fixed-length vector representation. This transformation reduces the scaling from quadratic to linear in the number of atoms, while preserving global structural information through the spectral properties of the matrix.

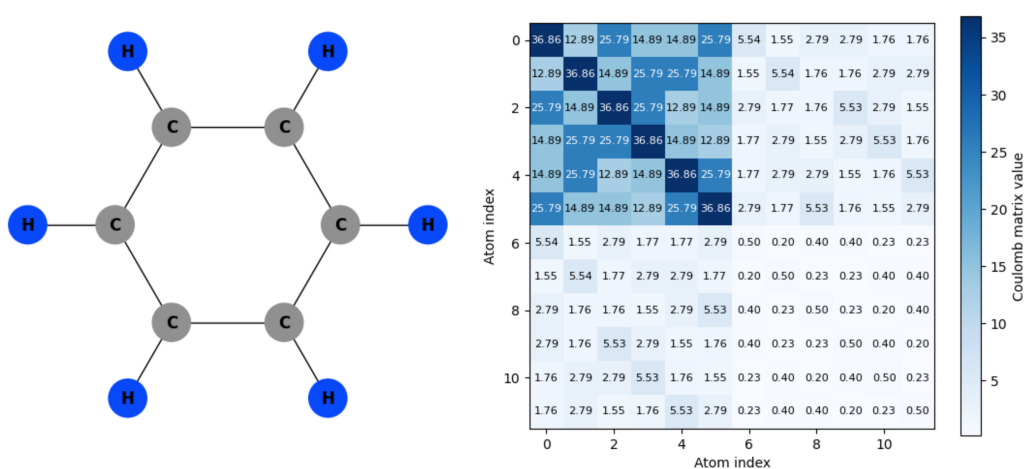


Fig. 3.9 Representation of benzene (C_6H_6) molecule (left) and sorted Coulomb matrix (right). Sorting in the coulomb matrix is essential for the invariance preservation. The calculation procedure was implemented by the author.

3.6.2 Molecular Representation

SMILES The Simplified Molecular Input Line Entry System (SMILES) [101] is a widely adopted notation for representing chemical structures as linear strings of characters, offering a compact and computationally efficient encoding of molecular systems. In recent years, SMILES has been extensively employed as an input representation for sequence-based machine-learning models, particularly transformer architectures, where molecules are treated analogously to natural language sequences. Within this framework, SMILES strings enable the direct application of tokenisation, embedding and attention mechanisms for tasks such as molecular property prediction, reaction modelling and de novo molecular generation [102, 103].

Despite its widespread use, the SMILES representation exhibits several intrinsic limitations. Most notably, the linear string does not explicitly preserve the three-dimensional geometry of the molecule, nor does it uniquely encode molecular structure, as multiple valid SMILES strings may correspond to the same molecular graph. As a result, the representation is sensitive to atom ordering and may introduce ambiguities that are not physically meaningful. Furthermore, long-range interactions and spatial relationships are only implicitly encoded through sequential dependencies, making it challenging for sequence-based models to consistently capture geometric and topological constraints. These limitations motivate the exploration of alternative representations that more directly encode molecular structure and symmetry, particularly for applications where spatial information plays a critical role.

3.7 Conclusion

This chapter has introduced the key machine learning and deep learning architectures employed throughout the thesis. Alongside a discussion of the fundamental principles underpinning modern data-driven modelling, each class of model was presented with concise insights into how it can be integrated into, or has been applied within, materials and device modelling workflows. The aim of this overview was not to provide an exhaustive mathematical treatment of every framework, but rather to establish a coherent methodological foundation. The concepts and architectures reviewed here equip the reader with the essential background required to follow the design choices, modelling strategies and computational tools developed in the subsequent chapters.

Chapter 4

The conceptual bridge

4.1 Why a Conceptual Bridge Is Needed

Modern materials and device engineering increasingly operates in regimes where physical phenomena span multiple spatial and temporal scales, from atomistic electronic interactions to mesoscale transport processes and system-level performance metrics. While multiscale modelling has long been recognised as a necessary paradigm to address such complexity, classical approaches are often built upon explicit hierarchical couplings between models operating at different resolutions. These couplings typically rely on scale separation assumptions, parameter passing strategies, or nested simulations, which become progressively fragile as system dimensionality, heterogeneity and variability increase.

In practice, many contemporary problems do not admit a clean separation of scales. Defect-driven electronic behaviour in nanomaterials, morphology-dependent transport in organic devices, and stochastic variability induced by fabrication processes all introduce non-linear feedbacks that challenge traditional multiscale schemes. As a result, the bottleneck of multiscale modelling is no longer solely the formulation of accurate physical equations, but the ability to manage, integrate and exploit the vast amount of heterogeneous data generated across modelling stages.

This observation motivates a shift in perspective. Rather than attempting to enforce direct numerical coupling between physical models, this thesis adopts the view that information, rather than equations, should act as the primary carrier across scales. In this framework, physical simulations at different resolutions are treated as

generators of structured data, which can be organised, analysed and learned from within a unified computational ecosystem.

Artificial intelligence plays a central role in enabling this transition, but not as a substitute for physics-based modelling. Instead, AI methods are employed to construct mappings between representations of physical states, allowing knowledge extracted at one scale to inform modelling decisions at another. This requires a conceptual bridge capable of linking physical structure to functional prediction through shared data abstractions and learning mechanisms.

The purpose of this chapter is to formalise this bridge. It provides the conceptual and methodological framework that connects the theoretical foundations introduced in the previous chapters with the concrete applications developed in the remainder of the thesis. By reframing multiscale modelling as a data-centred and representation-driven process, the chapter establishes the logical continuity between atomistic simulations, mesoscale surrogate models and generative design strategies.

4.2 From Physical State to Informational Representation

At the core of the proposed framework lies the transformation of physical systems into *informational representations* that can be consistently manipulated, analysed, and learned from across different scales. Any physical system can be described by a state space governed by underlying physical laws; however, direct access to this state space is typically mediated through numerical simulations, experimental measurements, or observational data, each producing outputs with distinct structures, resolutions, and semantics.

In this work, physical states are not treated as isolated numerical results, but as elements embedded in a structured data manifold. Atomic coordinates, electronic energies, current densities, morphological fields, and transport observables are interpreted as different projections of an underlying physical configuration encoded in data. This abstraction enables heterogeneous physical outputs to be integrated within a unified computational framework without requiring explicit equation-level coupling between models operating at different scales.

Formally, this process can be described as a mapping from the physical domain to an informational domain,

$$\mathcal{P} : \mathcal{S}_{\text{phys}} \rightarrow \mathcal{M}_{\text{data}}, \quad (4.1)$$

where $\mathcal{S}_{\text{phys}}$ denotes the space of admissible physical states and $\mathcal{M}_{\text{data}}$ represents a manifold defined by structured data representations. Depending on the scale and nature of the system, elements of $\mathcal{M}_{\text{data}}$ may take the form of graphs, tensors, images, scalar descriptors, or field quantities.

Once embedded in $\mathcal{M}_{\text{data}}$, physical systems can be analysed using machine learning models that operate on representations rather than on explicit governing equations. This shift does not imply a loss of physical meaning. On the contrary, the choice of representation encodes prior knowledge about the system including symmetries, conservation principles, spatial locality, and relevant degrees of freedom. For example, neural operator architectures exploit field-based representations to learn mappings between function spaces, while feature-based models rely on physically motivated descriptors extracted from atomistic or morphological configurations.

This representational viewpoint enables a unified treatment of problems that would otherwise appear fundamentally disconnected. Atomistic simulations of defective graphene and mesoscale simulations of organic photovoltaic devices differ in scale, governing equations, and numerical solvers; yet, both can be expressed as mappings between structured inputs and outputs within the same informational framework. As a result, learning strategies, data infrastructures, and validation methodologies can be transferred across domains with minimal conceptual friction.

By elevating representations to first-class objects of the modelling process, the conceptual bridge introduced here decouples physical fidelity from computational tractability. High-fidelity simulations remain essential as sources of ground-truth data, but their role shifts from direct predictors to generators of structured information. This perspective underpins the data-driven multiscale strategy developed in the subsequent chapters, where learning models operate on representations derived from physics, rather than attempting to rediscover physical laws from scratch. This representational abstraction is particularly relevant when considering the role of advanced materials within complex urban and infrastructural systems. In contemporary smart city contexts, materials are no longer passive components, but active enablers of sensing, energy conversion, communication, and adaptive functionality. The per-

formance of urban-scale technologies such as energy harvesting systems, distributed sensors, and electronic infrastructures is intrinsically linked to material properties that originate at the nanoscale and propagate through device-level behaviour.

By expressing material behaviour through informational representations, it becomes possible to connect atomistic and mesoscale descriptions to system-level objectives without requiring explicit multiscale simulation of the entire urban environment. In this perspective, materials act as the physical substrate through which urban functionalities emerge, while data-driven models provide the interface that translates material-level properties into actionable performance indicators at the city scale.

This approach enables a shift from material-specific optimisation to function-oriented design, where materials are evaluated and selected based on their contribution to broader infrastructural goals such as efficiency, robustness, and scalability. Within the proposed framework, the city is therefore not modelled directly as a physical system, but as an aggregation of functionalities informed by data generated at lower scales. The informational representation of materials becomes the key mechanism through which nanoscale physics influences urban-scale performance.

4.3 Data as the Multiscale Coupling Layer

The representational perspective introduced in Section 4.2 implies a key methodological consequence: the coupling across scales is no longer implemented primarily through direct numerical interaction between physical solvers, but through *structured data* that preserve physical meaning, provenance, and interoperability. In other words, the coupling layer is informational. This shift is not a matter of convenience; it is a practical requirement in the presence of heterogeneous models, incompatible discretisations, and high-dimensional outputs that cannot be propagated across scales through classical parameter passing alone.

Within the framework proposed in this thesis, each physical modelling stage is treated as a producer of a well-defined data product. At the atomistic scale, simulations generate structure-resolved outputs such as geometries, electronic descriptors, and transport-relevant quantities. At the mesoscale and device scale, simulations generate field-based observables (e.g., morphology-dependent potentials, current den-

sities, carrier distributions) as well as device-level performance indicators. At higher functional levels, these outputs become inputs to surrogate modelling, uncertainty assessment, generative exploration, and decision-making workflows. The coupling mechanism across these stages is therefore the ability to *store, query, transform, and reuse* data in a consistent and traceable manner.

To make this coupling reliable, the data layer must satisfy three requirements: (i) *semantic consistency*, ensuring that quantities maintain unambiguous meaning across pipelines; (ii) *technical interoperability*, enabling cross-tool exchange without lossy conversions; and (iii) *provenance and reproducibility*, ensuring that results can be traced back to the simulation conditions and computational context that generated them. Without these properties, learning-based models may achieve apparent predictive performance while silently encoding artefacts of data generation, thereby undermining scientific validity.

A robust approach to these requirements is provided by FAIR-aligned data practices, where datasets are designed to be *Findable, Accessible, Interoperable, and Reusable*. In the context of multiscale modelling, FAIR principles are not merely a set of publication-oriented guidelines, but a technical specification for enabling the informational coupling layer. Findability requires persistent identifiers and consistent naming schemes that allow models and users to locate data products reliably. Accessibility requires that data can be retrieved through stable mechanisms, without reliance on transient local environments. Interoperability demands machine-readable formats and metadata that support cross-software interpretation. Reusability requires sufficient contextual information to allow data to be used beyond the original project scope, including licensing, attribution, and detailed provenance.

In practical terms, this thesis adopts a data organisation strategy in which simulation outputs are stored in hierarchical, self-describing structures that accommodate heterogeneous modalities. A hierarchical layout supports the co-existence of atomistic descriptors, device-scale scalar observables, and high-dimensional fields within the same logical container while preserving clear semantic boundaries between data types. Equally important is the systematic inclusion of metadata. Root-level metadata describe dataset-level provenance (e.g., creation date, authorship, versioning, licensing, and persistent identifiers), while entry-level metadata capture sample-specific context, including simulation parameters, boundary conditions, solver settings, convergence thresholds, and links to the generating workflow. This dual-level metadata

strategy is essential: dataset-level metadata enable citation and governance, whereas entry-level metadata enable scientific traceability.

The coupling role of the data layer becomes explicit when considering the learning tasks addressed in the subsequent chapters. For surrogate modelling, the data layer must ensure that the mapping between inputs and outputs is well-posed and stable under dataset evolution; this requires consistent normalisation protocols, unit conventions, and unambiguous target definitions. For neural operators, the data layer must preserve spatial correspondence, grid definitions, and snapshot semantics for field quantities, so that learning models do not conflate physical evolution with representational artefacts. For generative modelling and design exploration, the data layer must support the ability to sample, condition, and validate candidate configurations while maintaining a direct link to their physical interpretation and evaluation metrics. In all cases, the quality of the coupling is constrained by the quality of the data infrastructure that mediates it.

Automation is therefore a structural component of the coupling layer rather than an implementation detail. High-throughput simulation campaigns, dataset curation, feature extraction, and model training must be orchestrated through reproducible workflows that minimise manual intervention and reduce human-induced variability. Automation also supports systematic error checking, logging, and dataset versioning, which are required to maintain integrity when datasets grow in size and complexity. In this thesis, automation is treated as a prerequisite for scientific scalability: without automated pipelines, the coupling layer becomes brittle, irreproducible, and difficult to audit.

Finally, the data layer enables a disciplined separation between *physical fidelity* and *computational tractability*. High-fidelity solvers remain the reference mechanisms for generating ground-truth information, but their outputs can be distilled into reusable representations that support learning-based acceleration and cross-domain transfer. This perspective clarifies the methodological continuity across the case studies presented later in the thesis. Although the physical domains differ substantially, both are integrated through the same data-centred coupling logic: simulations produce structured and traceable data, data enable learning, and learning enables scalable exploration and prediction.

The remainder of this chapter builds upon this foundation by framing enabling technologies as capabilities that operationalise the coupling layer (Section 4.4) and

by classifying learning models as mechanisms for transferring information across representations and scales (Section 4.5).

4.4 Key Enabling Technologies as Structural Connectors

The data-centred coupling layer described in Section 4.3 does not emerge in isolation, but is enabled by a set of technological capabilities that collectively support scalable, automated, and interoperable scientific workflows. Within the European research and innovation landscape, these capabilities are often grouped under the notion of *Key Enabling Technologies* (KETs). In this thesis, however, KETs are not treated as policy-defined sectors, but as *structural connectors* that operationalise the conceptual bridge between physical modelling and data-driven intelligence.

From a methodological perspective, the relevance of KETs lies in their functional role rather than in their categorical definition. High-performance computing, advanced data infrastructures, artificial intelligence, and workflow automation form a tightly coupled technological stack that enables the transition from isolated simulations to integrated modelling pipelines. It is this stack, rather than any individual technology, that allows information generated at one scale to be transformed, propagated, and exploited at another.

High-performance computing provides the computational backbone required to generate high-fidelity reference data. Atomistic electronic-structure simulations and mesoscale device-level solvers are computationally intensive and often prohibitively expensive when used directly for design-space exploration or uncertainty analysis. HPC resources enable the systematic generation of large, diverse datasets that capture structural variability and physical complexity, thereby supplying the ground-truth information upon which data-driven models are trained.

Advanced data infrastructures constitute the connective tissue of the framework. Hierarchical storage formats, metadata standards, persistent identifiers, and version-controlled repositories allow simulation outputs to be transformed into reusable scientific assets. These infrastructures ensure that data produced by heterogeneous solvers and software environments can be integrated within a common informational space, supporting interoperability across scales and domains. Without such infras-

tructures, the coupling layer would remain fragile, ad hoc, and difficult to extend beyond narrowly defined case studies.

Artificial intelligence acts as the mechanism through which information encoded in data is transformed into predictive and generative capability. Surrogate models, neural operators, and generative architectures enable the extraction of functional relationships from high-dimensional datasets, providing rapid approximations of expensive simulations and supporting inverse design tasks. Within the proposed framework, AI models are not viewed as standalone predictors, but as components embedded within a broader data and workflow ecosystem that preserves physical interpretability and traceability.

Automation links these components into a coherent operational pipeline. Automated workflows orchestrate simulation execution, data curation, feature extraction, model training, and evaluation with minimal manual intervention. This is essential not only for computational efficiency, but also for scientific robustness. Automation reduces human-induced variability, enables systematic error checking, and supports reproducibility when datasets and models evolve over time. In the absence of automation, the integration achieved through data infrastructures and AI models would not scale to realistic problem sizes.

Crucially, the same technological stack enables the treatment of fundamentally different physical problems within a unified methodological framework. The atomistic modelling of defective graphene and the mesoscale simulation of organic photovoltaic devices rely on distinct physical theories, numerical solvers, and data representations, yet they can be integrated within the same data-centred methodological framework.

4.5 Learning and Optimisation Across Scales

Within the data-centred framework outlined in the previous sections, learning models serve as the operational mechanism through which information is transferred across representations and scales. Rather than being treated as generic predictors, machine learning models are classified according to the specific role they play in mediating between physical descriptions, data representations, and functional objectives.

This functional classification is essential to maintain conceptual coherence across heterogeneous applications.

At the most immediate level, supervised surrogate models approximate the input–output relationships defined by high-fidelity physical simulations. These models learn mappings between structured representations of physical systems and scalar or low-dimensional targets, such as electronic properties, transport metrics, or performance indicators. Their primary function is computational acceleration: once trained, surrogates enable rapid evaluation of physical responses across large design spaces that would otherwise be inaccessible using direct simulation alone. In this context, learning acts as a compression mechanism, distilling expensive physical computations into reusable predictive representations.

Neural operator models extend this concept by shifting the learning target from finite-dimensional mappings to operators between function spaces. By operating directly on field-based representations, neural operators learn how entire physical fields transform under variations in input conditions. This is particularly relevant for mesoscale and device-level problems, where spatially resolved quantities such as potentials, carrier densities, or current distributions play a central role. Here, learning enables the approximation of governing operators without explicit discretisation-dependent coupling, providing a scale-agnostic mechanism for modelling complex physical behaviour.

Beyond prediction, learning models also enable optimisation and exploration across scales. Generative models and conditional sampling strategies support the synthesis of new physical configurations that satisfy target criteria defined in the informational domain. In this setting, optimisation is reframed as a navigation problem on the data manifold, where candidate solutions are generated, evaluated, and iteratively refined based on their predicted functional performance. This approach decouples the exploration of design spaces from direct physical simulation, while preserving a link to physical validity through surrogate evaluation and post-hoc verification.

Importantly, these learning paradigms do not operate in isolation. Surrogate models, neural operators, and generative components are embedded within a shared data and workflow infrastructure, allowing information to flow bidirectionally between them. Surrogates inform optimisation by providing rapid evaluations; neural operators supply field-level predictions that enrich performance assessment; generative

models propose new configurations that expand the domain of exploration. The coherence of this interaction depends critically on the consistency of the underlying data representations and metadata described in Section 4.3.

From a multiscale perspective, learning therefore functions as a translation layer rather than as a replacement for physics. Physical solvers remain the source of authoritative information, but learning models mediate how this information is reused, extrapolated, and propagated across scales and applications. This translation is inherently approximate, and its validity must be continuously assessed through uncertainty analysis, interpretability tools, and targeted high-fidelity validation. Such assessment is an integral part of the framework, ensuring that predictive power does not come at the expense of scientific credibility.

By positioning learning and optimisation as mechanisms for information transfer rather than as end goals, this section completes the conceptual bridge developed in this chapter. The case studies presented in the following chapters should therefore be interpreted not as isolated applications of specific models, but as concrete instantiations of this general learning-enabled multiscale methodology.

4.6 Chapter Synthesis and Transition to Case Studies

This chapter has established a conceptual and methodological bridge between the theoretical foundations introduced earlier in the thesis and the application-driven developments presented in the subsequent chapters. By reframing multiscale modelling as a data-centred process, the chapter has shown how information, rather than direct numerical coupling, can act as the primary mechanism through which physical knowledge is transferred across scales.

Section 4.1 motivated the need for a conceptual bridge by highlighting the limitations of classical multiscale approaches in the presence of heterogeneous models and complex physical interactions. Section 4.2 formalised the transformation from physical states to informational representations, establishing a common abstraction layer capable of accommodating atomistic, mesoscale, and functional descriptions within a unified framework. Section 4.3 identified structured data and metadata as the operative coupling layer that enables learning-based integration, while Section 4.4 positioned enabling technologies as the capabilities that operationalise this coupling

in practice. Finally, Section 4.5 clarified the role of learning and optimisation models as mechanisms for transferring and exploiting information across representations and scales.

Together, these elements define a general methodological pipeline in which high-fidelity physical simulations generate structured and traceable data; data infrastructures enable interoperability and reuse; learning models provide predictive, generative, and optimisation capabilities; and automation ensures scalability and reproducibility. This pipeline is intentionally agnostic to specific physical domains, focusing instead on the structure of the modelling process itself.

The remainder of the thesis demonstrates the applicability of this framework through two complementary case studies. Chapter 5 applies the methodology at the atomistic scale, focusing on the data-driven modelling and exploration of defective graphene. Chapter 6 extends the same principles to the mesoscale domain, addressing surrogate and operator learning for organic photovoltaic devices. Although the physical systems and modelling techniques differ, both chapters should be interpreted as instantiations of the same conceptual bridge, adapted to distinct data representations and learning tasks.

By making this continuity explicit, the chapter positions the following case studies not as isolated applications, but as evidence of the generality and transferability of the proposed data-driven multiscale framework.

Chapter 5

Learning the Device Physics of Organic Photovoltaics: Neural Operators, Morphology Clustering and Generative Modelling

5.1 The importance of organic devices in smart cities

In the following case study, particularly the one concerning the device scale within our multiscale framework, we addressed the challenge of exploring a data-driven solution to the drift–diffusion problem, applied specifically to bulk heterojunction (BHJ) organic photovoltaic devices. Organic semiconductors are highly promising in terms of performance, representing next-generation photovoltaic devices. Notably, they can be considered both a cost-effective and efficient alternative to conventional devices that have so far reached the market. Recent developments, particularly in polymer/non-fullerene systems, have led to significant improvements in their efficiency, reaching values of approximately 19% [104].

Recently, there has been increasing interest in BHJ architectures [105], which consist of a nanoscale blend of electron donors and acceptors sandwiched between two contacts.

The integration of these emerging technologies into the infrastructures of smart cities opens new opportunities for structural and functional innovation. Devices based on flexible and organic materials, as highlighted in [106], enable a paradigm shift in which infrastructures are no longer passive energy consumers but become active energy generators, contributing to the long-term goal of self-sufficient urban systems powered exclusively by renewable sources. Consequently, research into improving the performance of such devices has become increasingly relevant. From an engineering perspective, advancing this field requires a multiscale approach capable of linking material-level properties to system-level performance. Moreover, the integration of artificial intelligence and data-driven methodologies plays a critical role in both accelerating the discovery of novel materials and topological configurations, and in optimising device design and scalability [104]. This dual focus on material innovation and computational intelligence is central to enabling high-performance, sustainable, and scalable energy solutions for the smart cities of the future.

The main objective of this chapter is to present an automated device-level workflow capable of predicting the key targets on a 3D grid, as well as the scalar component of the J-V curve, through a hybrid architecture combining Neural Operators with attention mechanisms. Through this strongly hybrid design, the aim is to develop a data-driven drift–diffusion model that enables exploration of the morphology space and, as discussed in the final section, provides the ability to classify morphologies and, via a generative concept, to target specific regions of interest in this space in order to generate morphologies that satisfy given design requirements. Throughout the chapter it will become clear that, within the scope of this thesis, this framework represents only an initial theoretical step, paving the way for a full engineering of the process and for scaling the approach to higher-resolution and higher-dimensional morphologies.

5.2 Solving drift-diffusion equations with operator learning

Neural operators have recently emerged as a powerful class of models for learning mappings between infinite-dimensional function spaces [40]. In the context of partial

differential equations (PDEs), they aim to approximate the solution operator

$$\mathcal{G} : a(x) \mapsto u(x), \quad (5.1)$$

where $a(x)$ denotes the input function, such as material properties, source terms, or boundary conditions, and $u(x)$ represents the corresponding solution of the PDE. Unlike standard neural networks that learn finite-dimensional mappings, neural operators learn directly at the operator level, making them mesh-independent and applicable across different discretisations.

In the case of BHJ morphologies, the drift-diffusion equations describe the coupled transport of electrons and holes under the influence of internal electric fields and recombination processes. For a given three-dimensional morphology, the electron and hole densities, $n(x)$ and $p(x)$, and the electrostatic potential, $\phi(x)$, satisfy the steady-state drift-diffusion equations:

$$\nabla \cdot \mathbf{J}_n = R_n, \quad \mathbf{J}_n = -\mu_n n \nabla \phi + D_n \nabla n, \quad (5.2)$$

$$\nabla \cdot \mathbf{J}_p = -R_p, \quad \mathbf{J}_p = -\mu_p p \nabla \phi - D_p \nabla p, \quad (5.3)$$

$$-\nabla \cdot (\epsilon \nabla \phi) = q(p - n + N_{\text{dop}}), \quad (5.4)$$

where $\mathbf{J}_n$ and $\mathbf{J}_p$ are the electron and hole current densities, μ_n and μ_p denote mobilities, D_n and D_p are diffusion coefficients, R_n and R_p are recombination rates (including Shockley–Read–Hall processes), ϵ is the permittivity, q the elementary charge, and N_{dop} the doping density. The boundary conditions reflect contacts and applied voltages across the device.

Neural operators can be trained to map a given BHJ morphology and its material parameters directly to the full solution fields (n, p, ϕ) , effectively learning the solution operator $\mathcal{G}$ for the drift-diffusion system. This approach allows rapid evaluation of device behaviour for new morphologies without solving the PDEs from scratch, providing a mesh-independent surrogate for numerical simulation across the spatial domain.

This workflow is particularly justified because, as shown in Fig. 5.1, the computational cost increases sharply as the grid resolution grows, making it increasingly difficult to work with morphologies approaching realistic scales.

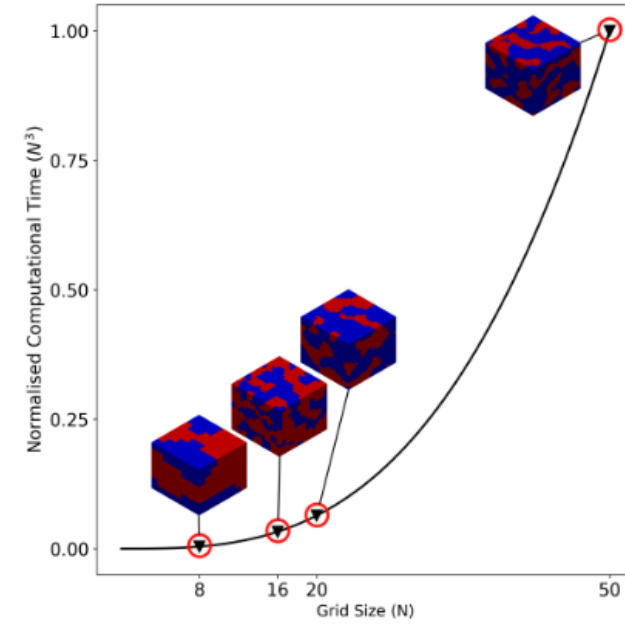


Fig. 5.1 Computational time and C constant of the drift diffusion solver for grids with size 2^N .

5.3 Automation frameworks for device simulation

The methodological paradigm followed in this chapter is aligned with that adopted across the various applications presented throughout the thesis. A recurrent challenge is the integration of heterogeneous domains and multiple levels of analysis, which necessitates robust mechanisms for interoperability and data exchange. In this context, the general principle of integration through well-defined data interfaces is once again applied. The complete workflow underpinning the construction of the dataset, including the standardisation of data flows and their subsequent processing, is illustrated in Figure 5.2. For the purposes of this work, two preparatory phases can be identified. In the first phase, the automatic generation of morphologies is carried out using the methods presented in Sections 5.3.1 and 5.3.2. Subsequently, an interface layer was constructed between the simulator and the morphology generation pipeline in order to automate the simulation process. The simulation step is essential since all the algorithms employed in this study are supervised. Accordingly, the second phase corresponds to the generation of the ground-truth data required for supervised learning.

The steps leading to the construction of the dataset and the generation of the ground-truth data can be summarised as follows:

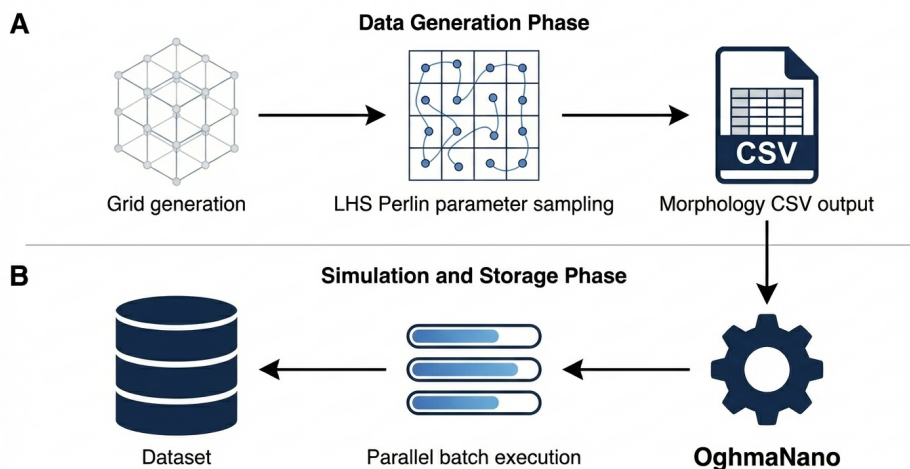


Fig. 5.2 A) Automatic workflow for morphology generation based on Perlin noise patterns. B) Simulation workflow using a parallel pool executor and systematic dataset storage.

1. Generation of the morphologies (in this work, with dimensions $20 \times 20 \times 20$);
2. Implementation of the data interface between the morphologies and the simulator;
3. Simulation of the morphologies to produce the ground-truth data.

The entire simulation workflow, as illustrated in Fig. 5.2, was implemented using parallel pooling in order to accelerate execution and assign each simulation to an individual CPU thread.

5.3.1 Morphology Generation for Bulk Heterojunction

Several approaches were explored for generating the morphology of the BHJ device, with the goal of producing realistic phase-separated structures that reflect both the correct geometric features and the appropriate final concentrations of the donor (PM6) and acceptor (Y6) materials.

The first approach was based on statistical methods, particularly noise-based generation followed by clustering. Clustering played a key role in this context, as it helped reduce the presence of isolated domains within the 3D volume. These isolated domains, if present, could lead to numerical instabilities in the subsequent simulation steps, particularly when computing charge carrier densities, energy levels,

and current density fields. However, this method presented a major drawback: the parameter calibration for clustering proved to be extremely challenging. Even after careful tuning, the resulting morphologies often exhibited unrealistic distributions of the two polymeric phases.

To address this limitation, we adopted a more structured statistical method based on volumetric Perlin noise. Perlin noise has the advantage of being controlled by a small number of parameters, with the number of octaves being the most significant. For all morphologies included in our dataset, the number of octaves was fixed to 2. This choice ensured a good balance between spatial frequency and structural smoothness. The algorithm used to generate the 3D Perlin noise fields is summarised in Algorithm 1, and was applied directly to the full device volume to produce initial binary or continuous concentration fields, later post-processed to reflect the BHJ morphology.

Algorithm 1 Generate 3D Perlin Noise Morphology

```

1: Input: number of octaves octaves
2: Create empty 3D array volume of shape  $(n_z, n_x, n_y)$ 
3: Initialize Perlin noise generator with octaves
4: for  $z = 0$  to  $n_z - 1$  do
5:   for  $x = 0$  to  $n_x - 1$  do
6:     for  $y = 0$  to  $n_y - 1$  do
7:        $u \leftarrow z/n_z$ 
8:        $v \leftarrow x/n_x$ 
9:        $w \leftarrow y/n_y$ 
10:       $\text{volume}[z][x][y] \leftarrow \text{noise}(u, v, w)$ 
11:    end for
12:  end for
13: end for
14: Normalize volume to range  $[0, 1]$ 
15: Output: binary volume where values  $> 0.5$  are set to 1, else 0

```

5.3.2 Cahn-Hilliard Modelling of Phase Separation

The second approach explored for generating realistic BHJ morphologies was to explicitly model the phase evolution over time. In particular, the two materials (Y6 and PM6) were treated as immiscible polymeric species, initially well-mixed, that undergo phase separation over time. This results in the formation of distinct domains

that characterise the final morphology of the active layer. This approach is inspired by classical phase-field models of demixing, as discussed in [107], and relies on the Cahn-Hilliard equation [108], which governs the time evolution of the concentration field.

The Cahn-Hilliard equation is a fourth-order nonlinear partial differential equation (PDE) that describes the dynamics of phase separation in binary mixtures. In its standard non-dimensionalised form, the equation is written as:

$$\frac{\partial c}{\partial t} = \nabla \cdot (M \nabla \mu), \quad \mu = \frac{\delta \mathcal{F}}{\delta c} = -\kappa \nabla^2 c + f'(c) \quad (5.5)$$

Here, the key variables and parameters are:

- $c(\mathbf{x}, t)$: the concentration field (or order parameter), representing the local volume fraction of one of the two materials (e.g., PM6); values typically range between -1 and $+1$, or 0 and 1 depending on convention.
- M : the mobility coefficient, which controls the rate of mass transport; it can be constant or concentration-dependent.
- μ : the chemical potential, derived as the variational derivative of the free energy functional $\mathcal{F}$.
- κ : the gradient energy coefficient, which penalises sharp interfaces and thus controls interface thickness.
- $f(c)$: the bulk free energy density, usually modelled as a double-well potential, such as $f(c) = \frac{1}{4}(c^2 - 1)^2$, which promotes separation into two stable phases.

The total free energy functional $\mathcal{F}[c]$ is given by:

$$\mathcal{F}[c] = \int_{\Omega} \left(\frac{\kappa}{2} |\nabla c|^2 + f(c) \right) d\mathbf{x} \quad (5.6)$$

This formulation captures the competition between two effects: the thermodynamic tendency to minimise bulk free energy (driving phase separation), and the energetic cost of creating interfaces between the two phases. The Cahn-Hilliard model naturally generates smooth, physically realistic morphologies with tunable domain sizes and interface sharpness.

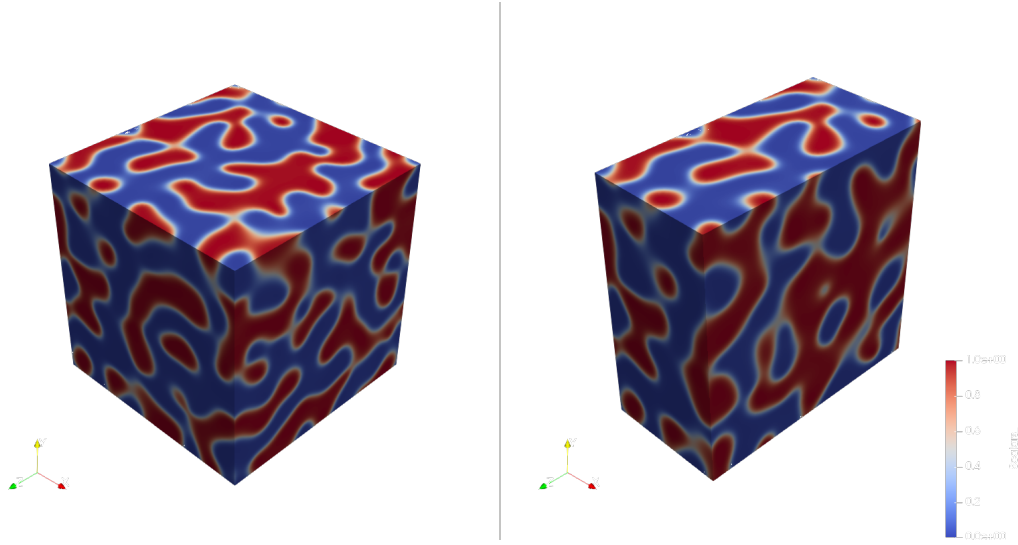


Fig. 5.3 The image on the left shows a morphology obtained using a fixed seed of 22 and the parameters listed in Table A.1

By simulating this equation over time on a 3D computational grid, we are able to produce BHJ morphologies that exhibit phase-pure domains and realistic interpenetrating networks, key features required for efficient exciton dissociation and charge transport in organic photovoltaic devices. The simulation is performed using a spectral method solution and the parameters listed in Table A.1. The implementation is fully GPU-accelerated and written in plain PyTorch, following the structure of a classic neural network (see Appendix A for detailed mathematical implementation).

5.3.3 Simulation Setup

For the generation of this dataset, the device architecture illustrated in Fig. 5.4 was adopted. Throughout the dataset construction, all principal simulation parameters were held constant to ensure reproducibility and control over external variables. The only varying component across simulations was the central active layer, corresponding to the BHJ morphologies.

The simulated device has dimensions of $110 \times 110 \times 110$ nm, a scale, particularly in terms of thickness, commonly employed in experimental studies. The device is based on the PM6:Y6 polymer blend system, and the complete stack used in the simulations comprises: ITO/ SnO_2 /PM6:Y6/ MoO_3 /Ag.

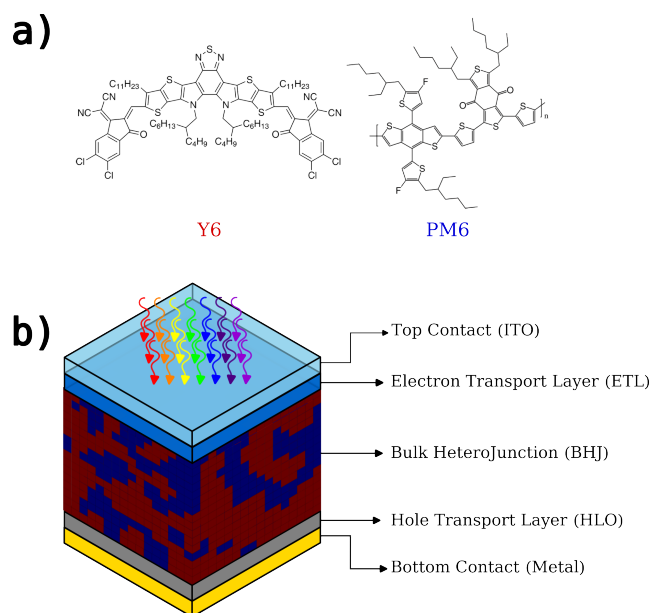


Fig. 5.4 a) Chemical structure of Y6 and PM6, materials of acceptor and donor in our simulations b) Representation of the simulation setup

All simulations were performed under strictly uniform illumination conditions, guaranteeing consistent light exposure across all devices. As a result, variations in the resulting J–V characteristics can be directly attributed to differences in the internal BHJ morphology, effectively isolating the influence of morphological factors from other potential external sources such as illumination intensity or spectral composition.

This controlled setup allows for a systematic evaluation of the effect of distinct BHJ morphologies on the electrical and physical behavior of the device, providing a robust basis for correlating structural features with performance metrics while maintaining all other conditions fixed. The entire dataset was generated using the OghmaNano software.[109–111]

Initially, the BHJ morphologies were created through the statistical algorithm detailed in Section 5.3.1. However, since we started with a limited number of samples, we observed that random sampling did not adequately cover the parameter space. To address this issue, we modified the algorithm so that the noise parameters were selected using Latin Hypercube Sampling (LHS) [112]. It is well established that LHS provides a more uniform and comprehensive exploration of the parameter space, particularly when dealing with a small sample size. Consequently, adopting LHS allowed us to sample the Perlin noise parameters more effectively, ensuring a

more thorough and representative coverage of the possible parameter combinations. The targets to be learned, which were extracted from the output files of OghmaNano, are as follows:

- **J_p**: Hole current density. Represents the current due to the drift and diffusion of positive charge carriers (holes) in the device.
- **Q_{nfree_and_ntrap}**: Total electron charge density, including both free and trapped electrons. Indicates the overall presence of electrons in the system, whether mobile or immobilised in trap states.
- **E_c**: Conduction band edge energy. Defines the minimum energy required for an electron to be in the conduction band and contribute to electrical conduction.
- **Q_{ntrap}**: Trapped electron charge density. Refers to electrons captured by localised energy states (traps) within the bandgap, reducing their contribution to conduction.
- **Q_{ptrap}**: Trapped hole charge density. Represents holes that are immobilised in trap states, limiting their participation in charge transport.
- **Q_{nfree}**: Free electron charge density. Denotes mobile electrons in the conduction band that actively contribute to electrical current.
- **Q_{pfree}**: Free hole charge density. Refers to mobile holes in the valence band that contribute to the hole current.
- **μ_n**: Electron mobility. Measures how quickly electrons move through the material under the influence of an electric field.
- **μ_p**: Hole mobility. Indicates the ease with which holes move through the material when subjected to an electric field.
- **R_{srh_nfree_to_ptap}**: SRH recombination rate from free electrons to trapped holes. Describes the rate at which free electrons recombine with trapped holes via Shockley-Read-Hall processes.
- **R_{srh_pfree_to_ntrap}**: SRH recombination rate from free holes to trapped electrons. Reflects the rate of recombination between free holes and trapped electrons, reducing carrier lifetimes.

All physical quantities are extracted from OghmaNano, which computes them at each point of the simulation grid. To enable direct comparison with the outputs of our neural network, these quantities are subsequently interpolated onto a regular cubic grid representing the device volume.

The calculations are performed for each applied voltage step, hereafter referred to as snapshots. However, not all snapshots were retained in the dataset. Instead, five representative voltage points were selected, at which all the aforementioned quantities were evaluated. Since one of our objectives is to approximate the J-V curve, the same five snapshots were also used to extract the corresponding points from the J-V characteristics.

For a given three-dimensional morphology and a fixed set of physical parameters, the simulator determines the steady-state solution of the coupled equations under different bias voltages. The resulting output is organised as a sequence of snapshots, each associated with a specific applied voltage within the bias range. At each snapshot, the solver provides spatially resolved quantities including quasi-Fermi levels, charge densities, electric fields, and current densities defined over the morphological grid. Consequently, the simulation output is inherently multi-dimensional, spanning both the spatial domain and the voltage sweep.

Formally, the dataset used to train our neural network for a generic target can be defined as:

$$\mathcal{D} = \left\{ (x_i, y_i, y_i^s) \mid x_i \in \mathbb{R}^{1 \times H \times W \times D}, y_i \in \mathbb{R}^{5 \times H \times W \times D}, y_i^s \in \mathbb{R}^5 \right\}_{i=1}^N \quad (5.7)$$

5.3.4 Selection of Representative Snapshots along the J–V Curve

In this subsection, it is important to emphasise the criteria adopted for snapshot selection, as these determine which voltage points were used for reconstructing the J–V curve.

The current–voltage (J–V) characteristic curve describes the macroscopic electrical response of a device under varying applied voltage, and is one of the most critical diagnostic tools in the characterisation of photovoltaic and optoelectronic systems. The J–V curve is typically obtained by sampling the total current density across a range of applied biases, from short-circuit conditions to forward bias. Key performance indicators such as the short-circuit current density (J_{sc}), the open-circuit

voltage (V_{oc}), the fill factor (FF), and the power conversion efficiency (PCE) can all be extracted from the shape of this curve.

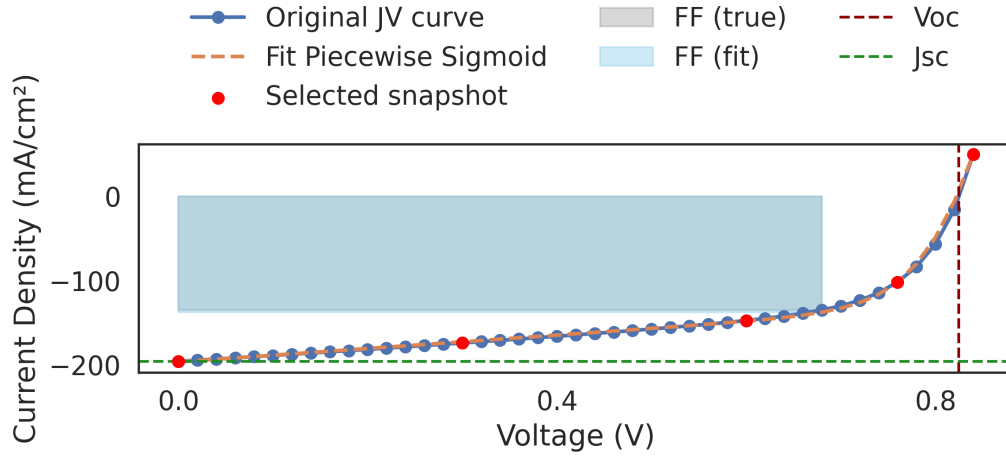


Fig. 5.5 The figure shows the full J-V curve plotted using all the snapshots provided by the simulator, alongside the approximated J-V curve obtained from the selected snapshots (red points).

After introducing the fundamental concepts of the J-V curve and the simulation outputs, we describe the methodology employed to select a reduced set of representative snapshots. The J-V curve was first trimmed to remove regions with limited relevance for photovoltaic characterization (Fig. 5.5). Specifically, all points beyond the open-circuit voltage, V_{oc} , were discarded. Beyond this voltage, the device is no longer operating within its functional regime: power output approaches zero, and the curve is dominated by parasitic effects, including series resistance and dark current contributions. Retaining these points would not contribute to the estimation of key photovoltaic figures of merit and could introduce distortions in subsequent analyses. Following trimming, an empirical procedure was applied to select representative snapshots. The approach employs a piecewise sigmoid function, designed to minimise the mean squared error (MSE) between a reconstructed J-V curve, derived from selected snapshots, and the original full curve. The number of snapshots, five in this case, was chosen based on the minimisation of the MSE. Examination of the J-V curve indicates a natural division into two regions: an exponential region at low voltages and a predominantly linear region at higher voltages.

The piecewise sigmoid function is defined as:

$$f(x) = (1 - S(x)) (a_1x + b_1) + S(x) (a_2e^{b_2x} + c_2), \quad (5.8)$$

with the sigmoid transition function

$$S(x) = \frac{1}{1 + \exp(-k(x - x_0))}. \quad (5.9)$$

Here, a_1, b_1 denote the slope and intercept of the linear region, a_2, b_2, c_2 define the exponential region, x_0 indicates the transition point, and k controls the sharpness of the transition.

To perform the fit, initial guesses and bounds were defined for each parameter (Table 5.1). Let V_{sub} denote the set of voltages corresponding to the selected snapshots along the J–V curve.

Table 5.1 Initial guesses and bounds for the piecewise sigmoid fitting parameters.

Parameter	Initial guess (p_0)	Lower bound	Upper bound
a_1	0	-10	10
b_1	0	-10	10
a_2	1	-10	10
b_2	1	-10	10
c_2	0	-10	10
x_0	$\text{median}(V_{\text{sub}})$	$\min(V_{\text{sub}})$	$\max(V_{\text{sub}})$
k	10	0	50

This methodology is essential for subsequent use with the Fourier Neural Operator (FNO). The predicted scalar outputs correspond exactly to the five selected snapshots. By this procedure, the full J–V curve can be accurately reconstructed from the limited set of FNO predictions, ensuring consistency between scalar outputs and curve representation.

5.4 Proposed Solution

Starting from the morphologies introduced in Section 5.3.1, we design a two-head architecture based on Fourier Neural Operators (FNOs) [79]. As illustrated in Figure

5.6, the network receives only the morphology as input and, through its first branch, learns a continuous mapping from the morphology to the selected target fields.

To jointly capture global properties of the morphology, the network incorporates a scalar head. This component first concatenates the input and spatial prediction along the channel dimension, then applies a masked attention pooling operator to extract a global representation. The resulting feature vector is processed by a two-layer multilayer perceptron with GELU activation, producing scalar outputs aligned with the number of spatial channels. Thus, the model is capable of simultaneously predicting spatial field-like quantities at different resolutions together with aggregated global descriptors, enabling a unified treatment of multi-scale targets.

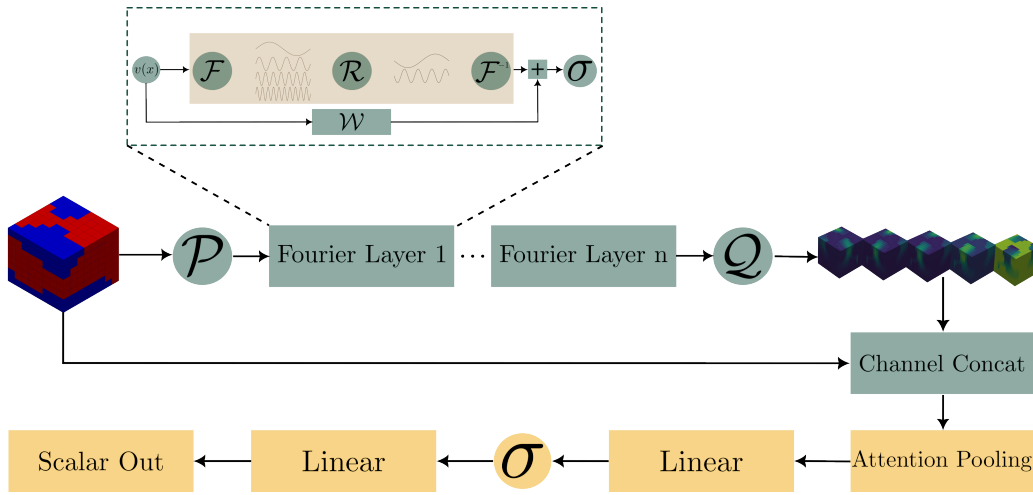


Fig. 5.6 Architecture of the network used in this study. In particular, we can observe that it is a two-head network: the green branch, starting from the generated morphologies, predicts the target fields, while the yellow branch concatenates the morphologies with the predictions and, through attention pooling, helps predict the 5 snapshots of the J-V curve, which are subsequently approximated.

The hyperparameters of the network were selected empirically based on preliminary experiments and prior experience with similar architectures. The network employs a Fourier Neural Operator (TFNO3d), which incorporates a **Tucker factorization** [113] of the spectral convolution weights (See the Appendix A.2 for detailed implementation and how this method works). In this factorization, the original weight tensor is decomposed into a smaller core tensor multiplied along each mode by factor matrices, reducing the number of learnable parameters while preserving the capacity to model long-range interactions in 3D volumetric inputs.

This decomposition allows the TFNO3d to efficiently learn 3D spectral filters while significantly reducing the number of learnable parameters. By factorising the weight tensors in this manner, the model preserves its expressive capacity, capturing complex interactions across spatial dimensions without requiring the full parameter set. Such an approach is widely employed in machine learning to achieve a more compact representation, maintain model expressivity, and reduce the risk of overfitting, particularly in high-dimensional settings where data may be limited relative to model size.

A comprehensive overview of all hyperparameters, including both the general configuration of the TFNO3d and the design of the scalar head, is provided in Table 5.2. The table summarizes the key settings used during training, such as the number of modes, layer dimensions, learning rates, and the structure of the scalar prediction head.

Hyperparameter / Component	Value
General FNO Parameters	
Batch size	32
Input channels	1
Lifting channels	64
Learning rate (initial)	0.001
Maximum learning rate	0.0005
Modes in x	10
Modes in y	10
Modes in z	10
Number of epochs	1001
Number of layers	4
Output channels	5
Projection dimension	64
Scalar Head ('scalar_head') Parameters	
MaskedAttentionPooling3D	$C_{\text{out}} + 1$
Linear 1	in: $C_{\text{out}} + 1$, out: 64
GELU	—
Linear 2	in: 64, out: C_{out}

Table 5.2 Hyperparameters of the TFNO3d network with scalar head.

With the hyperparameters selected as discussed and summarised in Table 5.2, the network was trained using an early-stopping strategy. Evidence of this procedure

is provided by the training versus validation loss curves reported in Appendix B.2, specifically Figures B.18 to B.28.

Inspection of the horizontal axis, which represents the training epochs, shows that the total number of epochs differs across these figures. This behaviour is a direct consequence of the early-stopping criterion. Training was halted whenever no improvement in the validation loss was observed for more than 100 consecutive epochs.

Furthermore, a distinct model was trained for each target quantity. In each case, the hyperparameters are shared between the models.

5.5 Results and Discussion

In the following discussion, results are presented for a single representative morphology, for which all target quantities introduced in Section 5.3.3 have been predicted. For clarity and conciseness, only the most physically and numerically significant targets are discussed in detail in this section, while the complete set of results for the remaining targets is reported in Appendix B.2, Figures B.29–B.35. In all visualisations, only intermediate temporal snapshots are shown, as these provide a representative view of the system dynamics while avoiding redundant information. It should be noted that the target $Q_{\text{nfree_and_ntrap}}$ is not included in the figures, as it can be directly obtained as a linear combination of the hole and charge densities when these quantities are considered individually. Figure 5.7 illustrates the results obtained for the hole current density J_p , showing the simulated reference field, the model prediction and the corresponding spatial distribution of the normalised squared relative error (SRE). From a global perspective, the prediction of J_p appears reasonably accurate: averaged over the full voxel domain, the SRE remains below 5%, indicating that the model successfully captures the dominant large-scale trends of the current distribution. However, a closer inspection of the local behaviour reveals a markedly different picture. In specific regions of the morphology, the local SRE increases significantly, highlighting the presence of pronounced localised discrepancies between the predicted and simulated fields. These errors are not uniformly distributed, but instead concentrate in regions where charge transport is strongly influenced by fine-scale morphological features and interface topology.

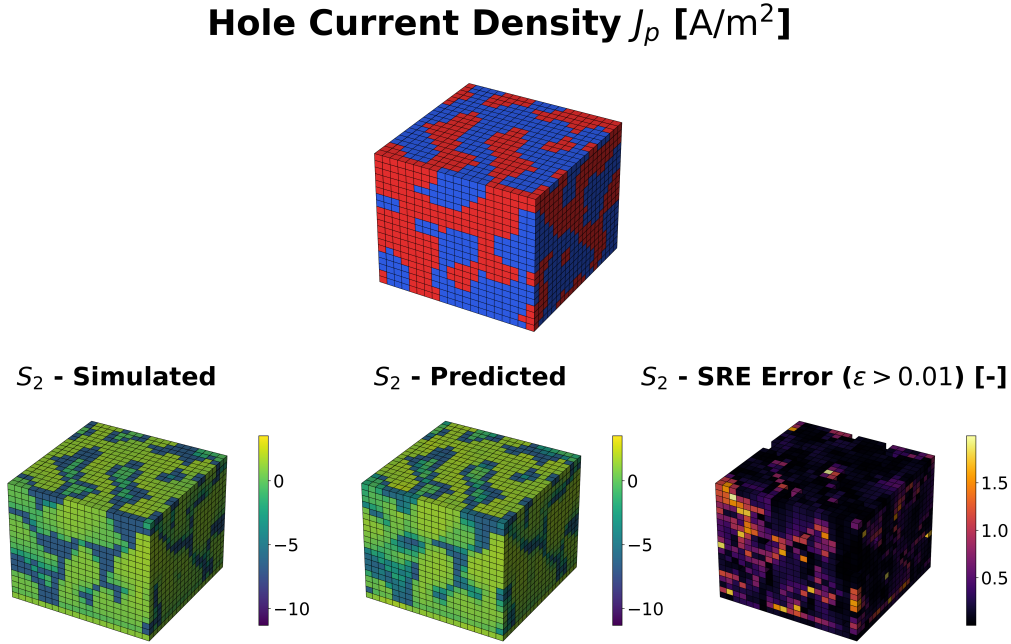


Fig. 5.7 Comparison between simulated and predicted hole current density J_p for a representative three-dimensional morphology. The top panel shows the input morphology, while the bottom panels report the simulated reference field, the neural-operator prediction and the spatial distribution of the normalised squared relative error (SRE).

A contrasting behaviour is observed for the recombination rate 5.8. For this target, both the global and local predictions are highly accurate, with SRE values remaining below 2% across the entire voxel domain. This indicates that the model is able to robustly learn and reproduce quantities that are less sensitive to fine-scale transport pathways and are instead governed by smoother, more spatially averaged physical mechanisms.

Furthermore, for the same representative morphology, Figure 5.10 illustrates the output of the scalar prediction branch. In particular, for the recombination-rate target, the model is able to accurately predict the corresponding current–voltage (J–V) characteristic. Notably, this prediction is obtained in a single forward pass: the J–V curve is inferred jointly with the associated spatial field within the same inference step, without requiring any additional post-processing or iterative coupling.

This result highlights the capability of the proposed architecture to consistently integrate spatially resolved physical information with global device-level observ-

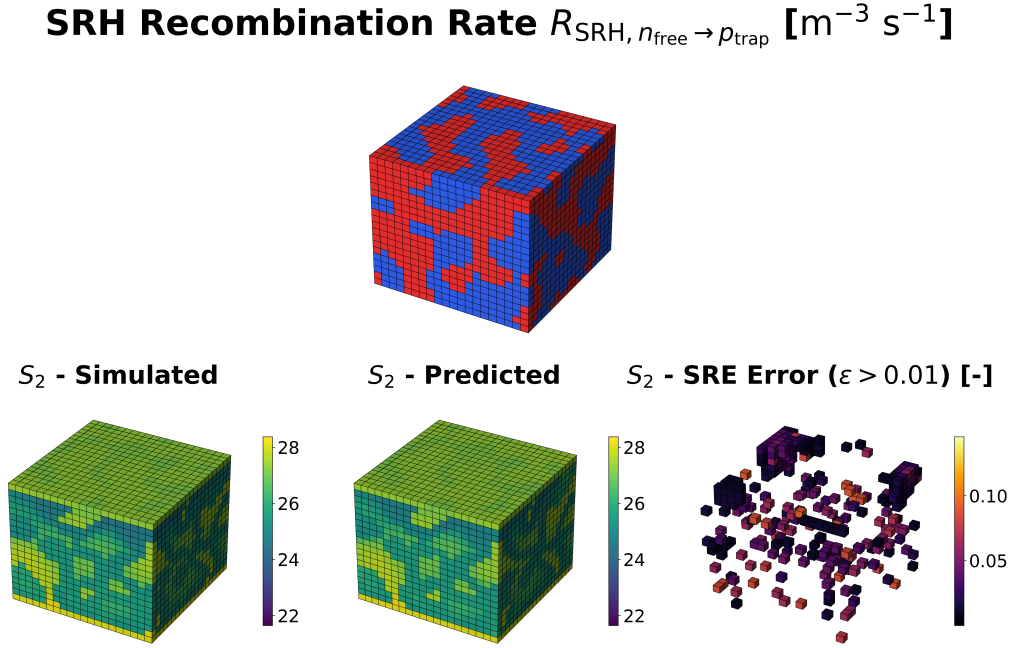


Fig. 5.8 Comparison between simulated and predicted Shockley–Read–Hall (SRH) recombination rate $R_{\text{SRH}, n_{\text{free}} \rightarrow p_{\text{trap}}}$ for a representative three-dimensional morphology. The top panel shows the input morphology, while the bottom panels report the simulated reference field, the neural-operator prediction and the spatial distribution of the normalised squared relative error (SRE).

ables, enabling simultaneous prediction of local fields and macroscopic performance metrics. The behaviour observed for the remaining targets follows trends consistent with these two limiting cases and is summarised in the appendix. Together, these results suggest that the predictive capability of the proposed framework depends strongly on the degree to which a given physical quantity is dominated by local versus global transport effects. Despite the relevance and impact of the results presented in this work, it is necessary to explicitly address the limitations encountered during the study. A major unresolved challenge concerns the direct application of the super-resolution concept introduced in Section 3.4.3. When the proposed models are trained on low-resolution morphologies, such as $20 \times 20 \times 20$ voxel grids, the resulting representations fail to intrinsically capture the physical mechanisms governing charge transport and electric-field distributions. As a consequence, transferring the learned representations to higher spatial resolutions, particularly in a zero-shot setting, proves to be highly challenging and largely ineffective. This limitation is

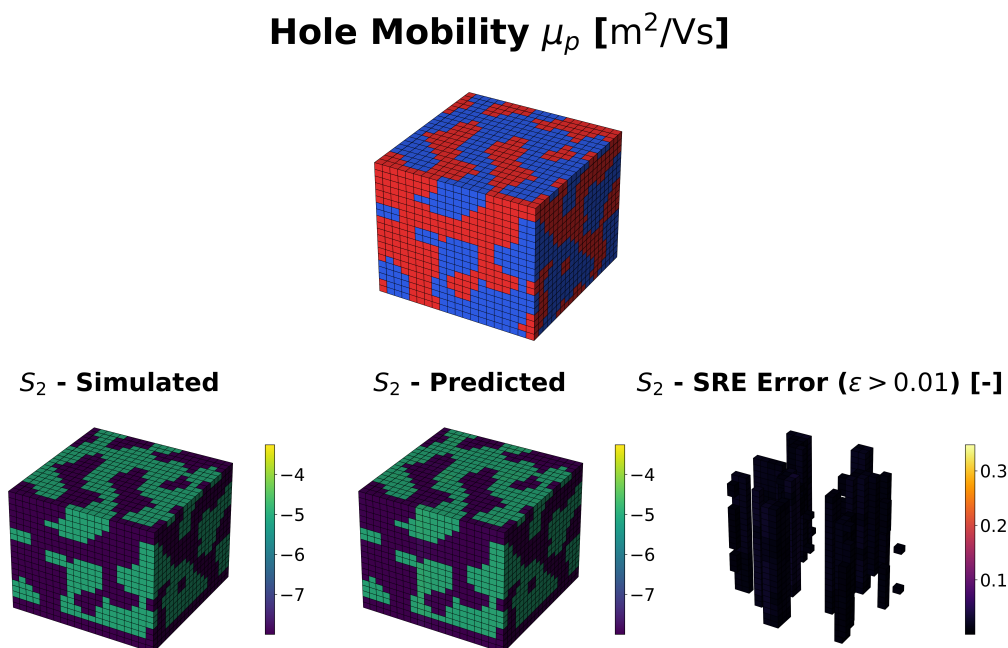


Fig. 5.9 Comparison between simulated and predicted Hole Mobility μ_p for a representative three-dimensional morphology. The top panel shows the input morphology, while the bottom panels report the simulated reference field, the neural-operator prediction and the spatial distribution of the normalised squared relative error (SRE).

reflected in the significantly degraded performance observed when the model is applied to super-resolution tasks.

A further contributing factor arises from the numerical structure of the solver employed to generate the dataset. The solver relies on finite-difference discretisation schemes, whose approximation accuracy is inherently dependent on the spatial resolution of the computational grid. At coarse resolutions, finite-difference operators provide grid-dependent approximations of the governing equations, leading to numerical solutions that do not consistently preserve scale-invariant physical features. Consequently, low-resolution simulations cannot be regarded as faithful downsampled representations of higher-resolution solutions. Instead, they correspond to distinct, resolution-specific operators, thereby preventing the construction of a physically meaningful and consistent super-resolution mapping.

An additional and critical limitation concerns the low spatial dimensionality of the dataset itself. In the class of organic device simulations considered here,

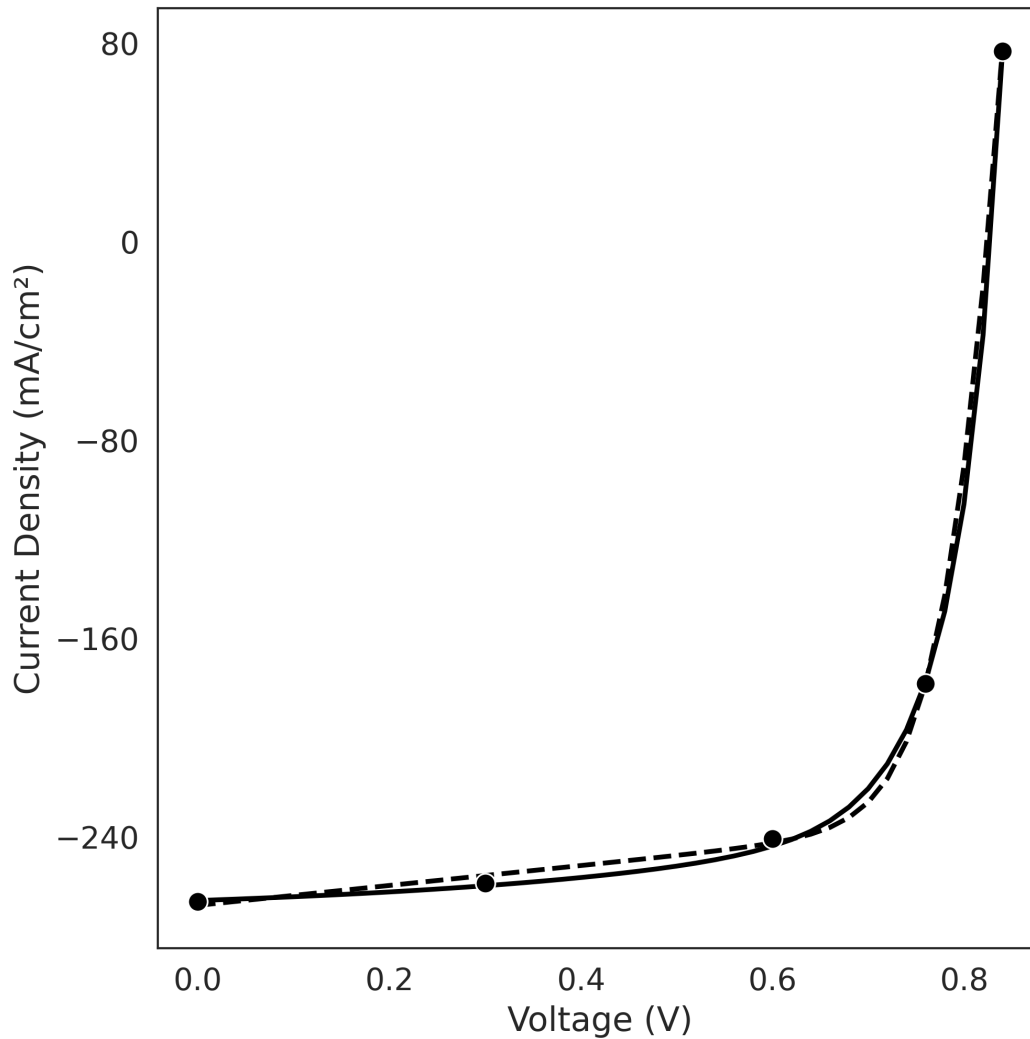


Fig. 5.10 Current–voltage (J–V) characteristic for the selected morphology. The solid line denotes the reference J–V curve obtained from the numerical simulation, while the dashed curve represents the model prediction. The predicted curve is reconstructed from the five voltage–current points inferred by the scalar prediction branch in a single forward pass, with a piecewise sigmoid interpolation applied to obtain a continuous J–V characteristic. The close agreement between the two curves demonstrates the ability of the model to accurately recover global device-level behaviour from morphology-dependent inputs.

charge transport phenomena are strongly local and sensitive to fine-scale morphological features. In particular, transport properties are heavily influenced by the local topology of donor–acceptor interfaces, percolation pathways and nanoscale connectivity patterns. A low-resolution three-dimensional voxel representation is unable to faithfully reproduce these features, as the discretisation process inherently

smooths sharp interfaces, suppresses local heterogeneities and alters connectivity. As a result, the simulated morphologies do not accurately reflect realistic donor–acceptor arrangements, leading to a loss of physically relevant information.

This limitation becomes particularly evident for quantities dominated by local transport effects, such as the hole current density J_p . As also reflected in Table 5.3, the predictive performance of the network for such targets deteriorates significantly. Importantly, the analyses conducted in this thesis indicate that this degradation does not primarily stem from insufficient expressive power of the neural architecture. Rather, it originates from the limited informativeness of the dataset, which fails to encode the fine-scale structural features required to determine locally sensitive physical observables.

A further constraint is imposed by the architectural choice of Fourier-based neural operators. These models rely on a truncated representation in Fourier space, where the number of retained modes is inherently limited by the spatial resolution of the input grid. For three-dimensional morphologies of limited size, the accessible Fourier spectrum is restricted to a small number of low-frequency modes. When such representations are extrapolated to higher resolutions in a zero-shot super-resolution setting, the learned operators remain dominated by coarse spectral components and lack access to the high-frequency information required to reconstruct fine-scale physical features. At the time of writing, this spectral limitation represents a significant bottleneck for zero-shot super-resolution within the proposed framework.

Despite these limitations, it is important to emphasise that the overall framework presented in this thesis is well validated for a range of other prediction tasks. In particular, the neural-operator-based models demonstrate stable and accurate performance in mapping morphology-dependent inputs to spatially resolved physical fields, such as electric potentials, carrier densities and current distributions. This behaviour indicates that neural operators provide a reliable approximation of the underlying drift–diffusion operator, even when evaluated on morphological configurations that were not encountered during training.

The capability of neural operators to directly learn mappings between function spaces enables robust generalisation across diverse and previously unseen morphologies, supporting their use as surrogate solvers for drift–diffusion modelling in organic semiconductor devices. From a practical perspective, this opens the possibility of

significantly accelerating device simulation workflows, enabling rapid evaluation of candidate morphologies without the need for repeated high-fidelity numerical solves.

When coupled with the generative and conditional design strategies introduced in the later chapters of this thesis, the proposed framework offers a promising pathway towards data-driven morphology optimisation. Generative models can be used to explore the space of admissible morphologies, while neural-operator surrogates provide fast and physically informed evaluation of their functional performance. Although the present dataset limits the reliable prediction of strongly local transport quantities, the framework itself remains extensible and adaptable. Future developments should therefore focus on the generation of higher-resolution, multi-scale datasets and on simulation strategies that expose the models to physically consistent fine-scale information during supervised training, thereby enabling improved treatment of local transport effects and more reliable super-resolution capabilities.

Type	Target	mean	std	min	max	median	25%	75%
Spatial	Jp	0.4271	0.0850	0.3421	0.5122	0.4271	0.3846	0.4696
	Q_nfree_and_ntrap	0.0225	0.0085	0.0140	0.0311	0.0225	0.0183	0.0268
	Ec	0.0297	0.0227	0.0070	0.0524	0.0297	0.0184	0.0411
	mu_n	0.0205	0.0154	0.0051	0.0358	0.0205	0.0128	0.0281
	mu_p	0.0157	0.0018	0.0139	0.0175	0.0157	0.0148	0.0166
	R_srh_nfree_to_ptap	0.0466	0.0102	0.0364	0.0568	0.0466	0.0415	0.0517
	R_srh_pfree_to_ntrap	0.0647	0.0254	0.0393	0.0901	0.0647	0.0520	0.0774
Scalar	Jp	0.1769	0.1456	0.0313	0.3225	0.1769	0.1041	0.2497
	Q_nfree_and_ntrap	0.0310	0.0290	0.0020	0.0599	0.0310	0.0165	0.0454
	Ec	0.1170	0.1009	0.0161	0.2180	0.1170	0.0666	0.1675
	mu_n	0.3137	0.1712	0.1425	0.4848	0.3137	0.2281	0.3993
	mu_p	0.2814	0.1635	0.1179	0.4449	0.2814	0.1997	0.3632
	R_srh_nfree_to_ptap	0.0201	0.0084	0.0116	0.0285	0.0201	0.0158	0.0243
	R_srh_pfree_to_ntrap	0.0143	0.0123	0.0020	0.0266	0.0143	0.0081	0.0204

Table 5.3 Combined Spatial and Scalar errors statistics

5.6 Introduction to the selection of promising morphology spaces

As discussed in greater depth at the end of Chapter 6, the objective here is to conclude the workflow in a similar manner. The aim is to generate candidate BHJ morphologies by organising and classifying the morphology space, identifying the most informative descriptors, and determining which regions are suitable for guided

generation. This enables a structured exploration of possible donor acceptor combinations. At this stage, the analysis remains limited to computational morphologies without considering production constraints or the actual physical feasibility of the structures.

A central descriptor in this characterisation is the tortuosity of the transport pathways. Tortuosity quantifies how strongly the routes available to charge carriers deviate from a direct path aligned with the electric field direction and is defined as follows:

$$\tau_g = \frac{L_g}{L}, \quad (5.10)$$

where L_g is the length of a given shortest path on the voxel graph and L is the direct film thickness along the transport axis.

In this work, each morphology is represented as a three dimensional arrangement of voxels associated with a conductive phase. Voxels that share a face are treated as connected, forming a discrete network that captures the actual percolation geometry. Starting from all voxels located on the injection interface, we identify the shortest available routes to the collecting interface throughout the conductive phase. This procedure produces a distribution of shortest path lengths for each morphology. Each path length is normalised by the film thickness in order to obtain the corresponding geometric tortuosity. From this distribution, we extract a set of statistical descriptors such as minimum tortuosity, median tortuosity, mean value, variance, higher order central moments, skewness and kurtosis. These descriptors provide a compact and expressive representation of the connectivity structure of each morphology.

The choice of tortuosity as a primary descriptor is supported by previous kinetic Monte Carlo studies [114]. These works show that an increase in tortuosity reduces the magnitude of the charge carrier mobility and weakens its dependence on the applied electric field. More convoluted pathways force carriers to move laterally within the domain rather than directly along the direction of the driving field. This increases the effective travel distance and lowers the net drift velocity. The strength of this effect depends on the level of energetic disorder, which indicates that tortuosity can become a dominant factor controlling the behaviour of BHJ devices. This physical relevance motivates the use of tortuosity based descriptors for the analysis of existing morphologies and for conditioning generative models that target specific transport characteristics.

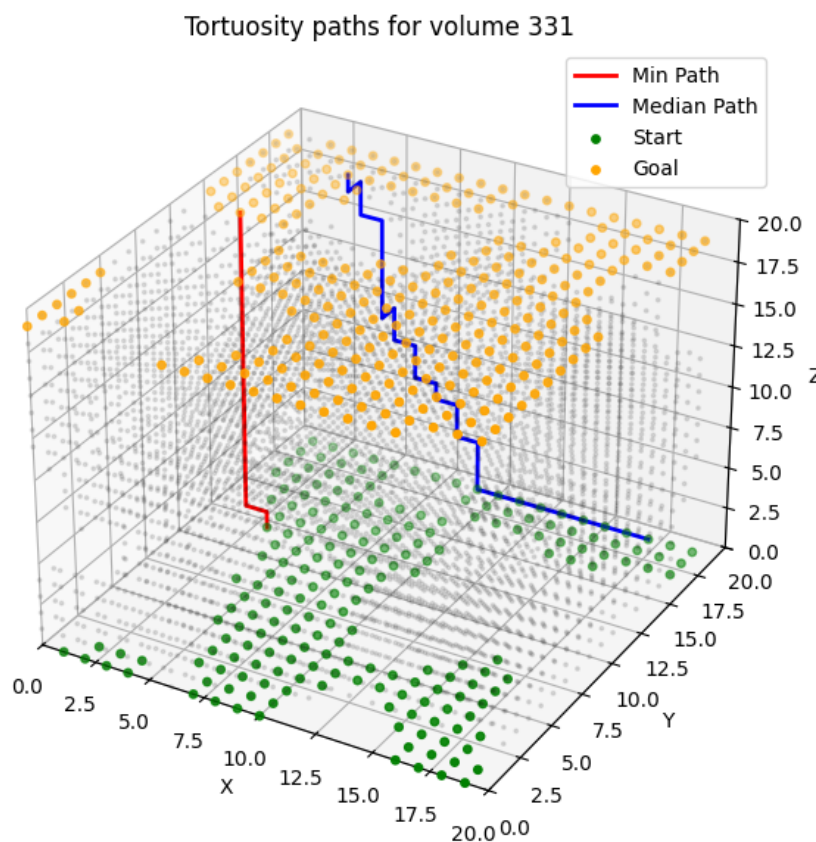


Fig. 5.11 Tortuosity extracted from a random sample in the dataset. The starting points are shown in green, the goal points in yellow, while the two extracted tortuosity paths are highlighted as the median path (blue) and the shortest path (red). The small grey dots represent the acceptor sites within the intermediate layer.

The statistical descriptors obtained from the tortuosity distribution are then used to cluster the morphology space. Each morphology is coloured according to its Fill Factor, which allows us to identify regions that appear more promising for the generation of new candidates. This combination of connectivity based descriptors and device level performance indicators provides a more informed exploration strategy.

The surrogate drift diffusion model introduced in this chapter plays a key role in this workflow. It allows rapid prediction of device quantities without dependence on the specific computational grid. This accelerates the evaluation of candidate morphologies and supports the identification of regions of interest within the morphology design space.

Cluster interpretation and limitations of the current feature space. The cluster-wise analysis reported in Fig. 5.6 provides a structured organisation of the morphology space based on tortuosity-related and connectivity-driven descriptors, jointly analysed with device-level performance indicators such as the Fill Factor. Although this representation enables the identification of regions associated with a higher prevalence of high-performance samples, it is important to note that no sharp separation between clusters is observed with respect to the Fill Factor when projected onto the reduced PCA space.

In particular, samples exhibiting low, intermediate, and high Fill Factor values coexist within the same clusters, indicating that the descriptors employed in the present analysis capture only part of the variability underlying device performance. This observation suggests that, at the current spatial resolution, the selected feature set is not sufficient to fully disentangle the structural factors governing charge extraction efficiency. Rather than indicating a failure of the clustering procedure itself, this behaviour reflects an intrinsic limitation of the dataset and of the morphological resolution at which the features are computed. BHJ transport processes are

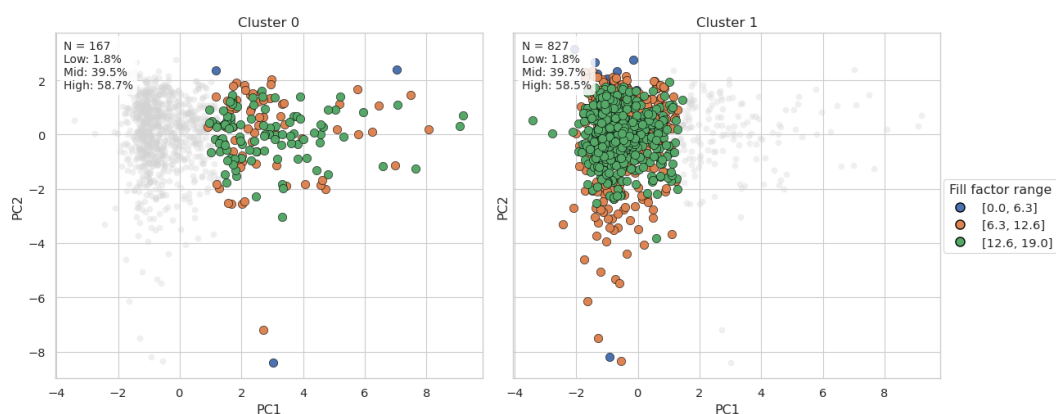


Fig. 5.12 Cluster-wise projection of morphology and tortuosity descriptors in the two-dimensional PCA space. Each panel highlights a single cluster, while samples belonging to other clusters are shown in grey. Points within the active cluster are coloured according to discretised Fill Factor ranges, normalised by cluster size. The annotations report the cluster population and the relative distribution of low-, medium-, and high-Fill-Factor samples.

known to be strongly influenced by local effects, including interfacial connectivity, nanoscale bottlenecks, and short-range percolation pathways. When morphologies are represented on relatively coarse voxel grids, such local phenomena are effectively averaged out, leading to descriptors that primarily encode global connectivity

trends while neglecting fine-scale transport-limiting features. As a consequence, the resulting clusters remain partially mixed with respect to the Fill Factor, even when physically motivated descriptors such as tortuosity are employed.

This result is consistent with the limitations discussed in Chapter 6, where it was shown that insufficient spatial resolution can hinder the ability of data-driven models to capture strongly localised physical mechanisms. In the present mesoscale setting, improving the discriminative power of the feature space therefore requires the generation of higher-resolution morphologies and corresponding ground-truth simulations, capable of resolving local transport pathways and interfacial effects more accurately.

Despite these limitations, the clustering framework remains valuable as an organisational and exploratory tool. It enables the identification of regions of the morphology space that are statistically enriched in higher Fill Factor samples and provides a principled basis for conditioning generative models. When combined with surrogate drift–diffusion solvers and future high-resolution datasets, this approach is expected to support more targeted and physically informed generation of candidate morphologies for organic photovoltaic devices.

Chapter 6

Atomistic Graphene Data to Generative Models: An AI Framework for Structure–Property Correlations

6.1 Introduction

In this chapter, which represents the final application examined in our multiscale study, we descend to the atomistic scale. Building on the automation pipeline introduced in the previous chapters, we examine how atomistic modelling, high-throughput simulation and data-driven methods can be integrated into the same unified workflow used elsewhere in this thesis.

The chapter opens with a concise discussion of graphene and the significance of its nanoscale properties for future smart-city technologies. We then present in detail the generation of a physically accurate dataset that captures a range of atomistic properties of graphene flakes. The workflow begins from the flake topology (commonly encoded in an XYZ file) and proceeds through automated calculation of structural and transport properties that depend both on global topology and on local defects. In parallel, a digital twin of a scanning tunnelling microscope (STM) is employed to produce synthetic STM images of the flakes, enriching the dataset with observable signals that mirror experimental acquisition.

Finally, we describe how the resulting dataset is coupled to an Explainable AI module to quantify how defect types and spatial distributions influence transport behaviour. This combined pipeline: atomistic simulation, synthetic experimental imagery, and explainable machine learning enables data-driven interpretation of nanoscale phenomena and opens a route towards generative design of nanoscale devices whose effects propagate to higher scales. Throughout, we emphasise the chapter's link to data integration: the same principles that enable a unified, replaceable workflow at larger scales are applied here to ensure reproducibility, interoperability and scalability of atomistic-level tools within the broader multiscale framework.

6.2 Graphene as an enabling material for the city of the future

Among emerging materials for sustainable urban technologies, graphene stands out for its multifunctional potential in energy storage, sensing, and communication infrastructures. Its unique combination of electronic, mechanical, and thermal properties provides a foundational platform for the development of smart city components that are both efficient and resilient [115–117].

A primary domain in which graphene demonstrates a particularly transformative impact is energy storage and management [118]. Graphene-based supercapacitors fundamentally redefine the energy storage-delivery paradigm by combining ultrafast charge–discharge dynamics with exceptional cyclic stability, often exceeding tens of thousands of operational cycles, significantly outperforming traditional battery systems. Such properties are particularly advantageous in urban-scale energy infrastructures, where rapid energy buffering is required for applications such as electric public transport, renewable energy integration, and demand peak mitigation. Acting as dynamic energy buffers, graphene supercapacitors effectively store surplus energy generated by intermittent renewable sources and release it on demand, thus contributing to grid stability and supporting the transition towards climate-neutral urban environments [119].

Beyond these macroscopic functionalities, the link between graphene's atomic-scale structure and its macroscopic electronic behaviour is becoming increasingly central in research. Advanced AI-assisted multiscale modelling frameworks now

allow the correlation of structural, electronic, and transport properties across scales. At the microscopic level, first-principles simulations of electron transport, defect dynamics, and surface chemistry guide the design of high-quality graphene with tunable electrical characteristics. These atomistic descriptors govern the emergent macroscopic performance of devices, determining parameters such as charge mobility, energy density, and conductive stability.

Within this context, the presence and configuration of defects including vacancies, dopants, and grain boundaries play a decisive role in modulating charge transport and overall device efficiency. Understanding and quantifying how these structural irregularities influence electronic and transport properties remains a key challenge in the rational design of graphene-based technologies.

To address this, we developed a comprehensive computational dataset of graphene nanostructures, explicitly designed to capture the interplay between morphology, electronic structure, and transport properties. This dataset enables the systematic investigation of how local atomic-scale perturbations translate into measurable variations in current flow, energy levels, and related electronic observables. By integrating data-driven modelling and machine learning techniques, the aim is to uncover predictive correlations that can inform the synthesis of defect-engineered graphene with optimised transport behaviour for next-generation smart materials and devices.

6.3 Automated dataset creation workflows

6.3.1 First steps into dataset generation

As anticipated in Section 6.1, the opening phase of this work focused on building a comprehensive dataset of nanographene samples that combines atomistic properties with synthetic scanning tunnelling microscope (STM) images. All electronic-structure and transport calculations were performed at the DFTB level, using as a starting point a publicly available repository of roughly 500 000 nanographene flakes [120]. The raw collection provides XYZ coordinate files and a limited set of DFT-computed properties (VASP) [121] for flakes containing between approximately 206 and 447 atoms.

Because the aromatic pattern and local periodicity of graphene strongly affect electronic and transport behaviour [122, 123], the raw structures required careful cleaning and normalisation before high-throughput processing. Figure 6.1 summarizes the full pipeline used to transform the repository into the final dataset. In practice, the preprocessing consisted of a sequence of targeted operations. First, we randomly sampled 50k structures from the repository and applied a topological filter: only flakes whose atom positions matched at least 78% of those of the corresponding pristine (defect-free) flake were retained (the post-selection distribution is reported in Figure 6.1b). This initial cut reduces extreme outliers and preserves a representative variety of defected configurations.

Second, to avoid spurious edge effects, flakes with defects lying close to the boundary were translated so that the defect regions were positioned near the centre of the flake. Each structure was then rigidly aligned to a common reference frame by translating the atom with minimum x and y coordinates to the origin $(0,0)$.

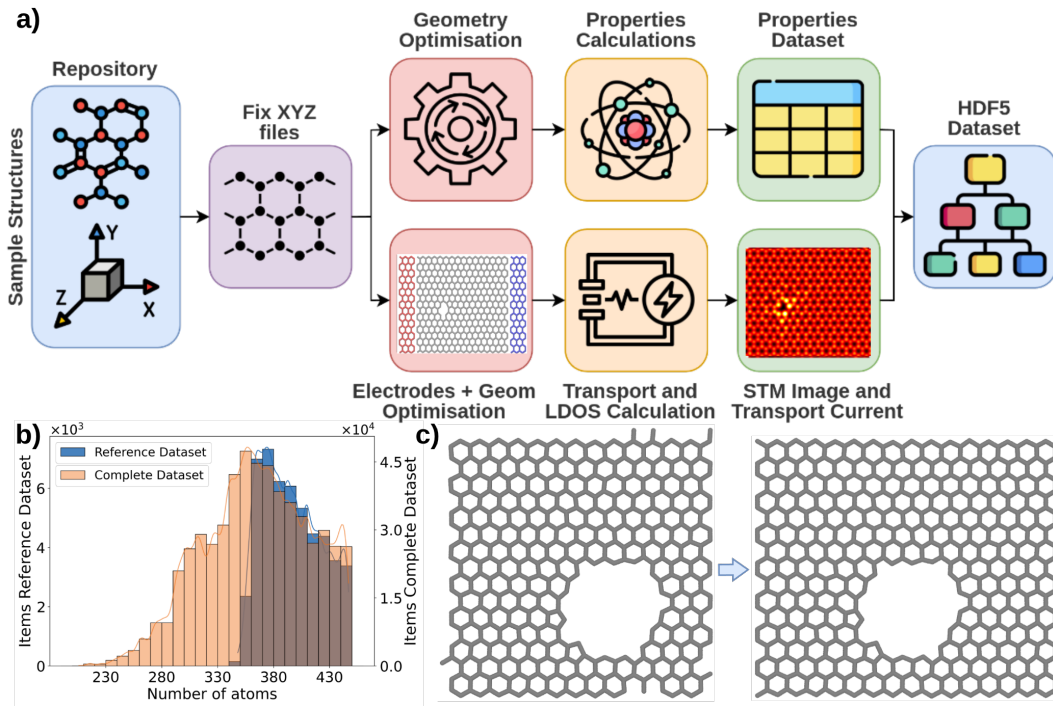


Fig. 6.1 a) Scheme of the workflow for the generation of the HDF5 dataset; b) Distribution and corresponding kernel density estimates (KDE) of the number of atoms in the reference dataset with respect to the complete starting dataset; c) Example of a procedure for the pre-processing of the original DG structures.

Finally, where the original geometry did not guarantee the local periodicity required to preserve graphene’s aromaticity, we restored the pattern by adding two rows of carbon atoms in an armchair arrangement at the flake extremities; this modification preserves the local periodic boundary conditions that strongly influence band structure and transport properties. After preprocessing, the pipeline computed DFTB-level electronic and transport observables and produced synthetic STM images through a digital twin of an STM. The resulting dataset is organised hierarchically and conforms to FAIR principles, with the explicit aim of accelerating the development of predictive models and multiscale screening workflows. In particular, generative modelling techniques such as variational autoencoders (VAEs) can be integrated with the pipeline to explore novel defect configurations and, when combined into ensembles, to efficiently sample the space of plausible nanostructures for downstream design and optimisation tasks [124].

As illustrated in the workflow shown in Fig.6.1, once the XYZ files have been preprocessed, the procedure branches into two distinct paths. The first path concerns the calculation of the electronic properties, while the second encompasses the transport calculations, ultimately leading to the generation of the STM images.

The workflow has been designed to operate fully automatically, providing a streamlined and user-friendly interface through a dedicated Python software package. It offers flexibility in execution: users can choose to run simulations in a distributed computing environment orchestrated by SLURM, or entirely on a local machine, depending on available resources. Every aspect of the process is configurable, with a series of configuration files enabling straightforward adjustment of parameters, thus allowing the workflow to be easily adapted to different computational setups or specific research needs.

6.3.2 Electronic properties calculation

All electronic properties were computed using the DFTB+ simulation package [125]. Before performing the actual calculations, we carried out a geometry optimisation step to ensure stable configurations. In particular, this procedure aimed to minimise the geometry by adjusting the atomic positions and removing the so-called spurious forces acting on the structures [126]. All optimisations were performed using a fixed simulation cell, whose dimensions were determined from a pristine graphene flake of

the same size as the defective ones. To encode both the optimised atomic positions and the information on the simulation cell, we adopted the *POSCAR* format [121] for representing the material.

From the initial dataset, only flakes without edge defects and for which the geometry optimisation successfully converged were retained. The optimisation was performed with a Fermi temperature (T_F) of 1000K and a maximum of 200 optimisation steps. Once this stage was completed, we proceeded to calculate the electronic properties. Specifically, we employed a Self-Consistent Charge (SCC) calculation at the DFTB level. The SCC approach iteratively updates the charge distribution until convergence is reached, thereby ensuring a consistent electrostatic potential across the system (more detailed explanation can be found in the work of Frauenheim et al. [127]). The properties were computed on the optimised structures only. For these simulations, we set a maximum of 200 SCC iterations, enabled collinear spin polarisation, and again used a T_F of 1000K. Collinear spin polarisation assumes that all spin moments are aligned along a single axis, allowing for spin-dependent effects to be included in the calculation while avoiding the additional complexity of non-collinear treatments.

For each optimised graphene flake, the following electronic properties were extracted:

- **Electron Affinity (EA):** energy required to add an electron to the system[128].
- **Ionization Potential (IP):** energy needed to remove an electron from the system[128].
- **Total Energy (E_{tot}):** derived from the DFTB calculations based on the optimized geometry[128].
- **Formation Energy (E_{form}):** derived from the DFTB calculations based on the optimized geometry.
- **Band Gap (E_{gap}):** calculated as the difference between the ionization potential (IP) and the electron affinity (EA)[129].
- **Fermi Energy (E_f):** determined from the electronic structure calculations on the optimized geometry.

The procedure involved performing three consecutive simulations by adjusting the *Charge* parameter, which specifies the total charge of the system in units of the electron charge. The first simulation was carried out with $Charge = 0$, providing the total energy E_{tot} and the Fermi energy E_f . From these results, the formation energy E_{form} could be derived using the total energy of the system along with the energies of individual carbon atoms.

Next, the electron affinity (EA) was determined by introducing an extra electron into the system. This was achieved by setting $Charge = -1$, and the resulting total energy (E_{tot-1}) was used in the calculation:

$$EA = E_{tot} - E_{tot-1} \quad (6.1)$$

Similarly, the ionisation potential (IP) was obtained by removing an electron from the system, i.e., by performing a simulation with $Charge = +1$. The total energy from this calculation (E_{tot+1}) allowed the evaluation of IP as follows:

$$IP = E_{tot+1} - E_{tot} \quad (6.2)$$

Finally, the electronic band gap was calculated as the difference between the ionisation potential and the electron affinity:

$$E_{gap} = IP - EA \quad (6.3)$$

6.3.3 STM Virtual Replica

The workflow for generating STM images is divided into two preparatory phases. The first phase involves setting up and performing the transport simulations, while the second focuses on constructing a digital twin to replicate the scanning tunnelling microscope. The procedure progresses step by step, allowing the transport properties to be accurately evaluated before proceeding to the generation of STM images.

Initially, defective graphene flakes are prepared and electrodes are attached at both ends to model the source and drain contacts. Each electrode comprises three rows of carbon atoms arranged in an armchair configuration, positioned according to defect-free graphene optimised at the same theoretical level (see Fig. 6.2a). Electrode

geometries are generated using the `setupgeom` utility of DFTB+, which replicates and translates atomic coordinates as required.

The atomic coordinates of the flakes are converted from XYZ to the DFTB+ GEN format. For each configuration, an input file is created specifying the source and drain regions and defining cutoff distances for Slater–Koster overlap and non-SCC interactions [130]. In particular, the `SKMaxDistance` parameter is set to 5.0 Å, and a translational offset of 4.91 Å along the transport direction ensures proper alignment of electrode atoms at the contact interfaces. Using the `setupgeom` utility, a processed GEN file is produced, representing the complete system including electrodes, along with a transport input file that defines the partitioning of atoms between the central region and the leads.

The combined system is then re-optimised, keeping the electrode atoms fixed, and convergence checks are performed to ensure stability. Subsequently, the Fermi levels of the source and drain contacts are calculated using the `ContactHamiltonian` functionality of DFTB+, applying a 0 K Fermi temperature and the `DivideAndConquer` solver. These values are then used in transport simulations, where the potential difference between the contacts is applied, and the `TransportOnly` solver computes the transmission properties.

The energy range for evaluating the transmission function and local density of states (LDOS) is defined, along with the sampling step and projection regions. Each atom in the device is treated as a separate projection region, the energy window is set to ± 0.5 V around the average Fermi level, and 100 energy points are sampled per atom.

Transport currents are extracted for each configuration, together with LDOS data for all atoms. As expected, the current decreases with increasing defect density in the flakes (see Fig. S1 in the Supporting Information). LDOS files provide information on the spatial distribution of electronic states, with columns representing energy and density of states, and the number of rows determined by the ratio between the energy range and sampling interval. Once the transport properties are fully characterised, the workflow moves to the second phase, in which a digital twin is constructed to simulate the operation of the scanning tunneling microscope and generate the corresponding STM images.

6.3.4 Electron Transport Simulation

Electron transport simulations were conducted within the framework of the Non-Equilibrium Green's Function (NEGF) formalism [130], as implemented in DFTB+ [125]. To model transport through the defective graphene (DG) structures, electrodes were attached to both ends of each sample, representing the source and drain contacts. Each electrode was constructed from three rows of carbon atoms arranged in the armchair configuration, with atomic positions corresponding to those in defect-free graphene optimised at the same level of theory (see Fig. 6.2a).

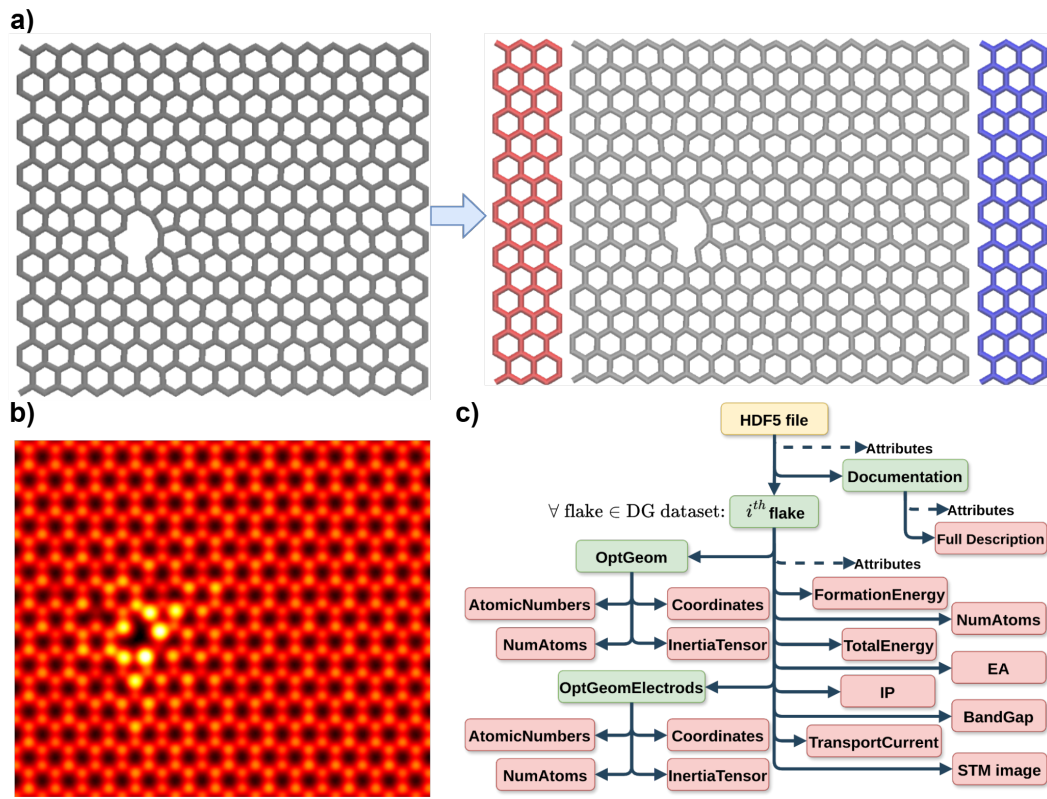


Fig. 6.2 a) Scheme of the process used to generate models for transport calculations, attaching source (red) and drain (blue) electrodes to DG samples; b) Example STM image of a DG flake, plotted with the *hot* colormap; c) Structure of the HDF5 file used to store the datasets. Data groups are represented in green, while the datasets, along with their respective attributes, are represented in red.

The electrode geometry was generated using the *setupgeom* utility of DFTB+, which allows replication and translation of atomic coordinates to extend the electrode regions. The atomic structures of the devices were first converted from XYZ format

into the GEN format required by DFTB+. For each device, an input file was prepared that defined the source and drain regions and specified the interaction parameters for the calculations. In particular, the maximum distance for Slater–Koster overlaps and non-self-consistent-charge interactions was set to 5.0 Å, while a translational offset of 4.91 Å along the transport direction (x -axis) was applied to ensure proper alignment of the electrode atoms at the interface with the DG sample.

For each sample, the *setupgeom* utility was employed to generate a processed file in GEN format, which contained the full configuration of the device, including the electrodes. In addition, an input file for the transport calculations was created, specifying the partitioning of atoms within the system into the device region, the source, and the drain. Subsequently, the Fermi levels for both the source and drain contacts were determined. To this end, calculations were performed using the ContactHamiltonian functionality of DFTB+, assigning the contact identifier to the drain and source, respectively, to obtain the corresponding Fermi level values. A Fermi temperature of 0K was applied, and the Divide-and-Conquer solver was used for these computations.

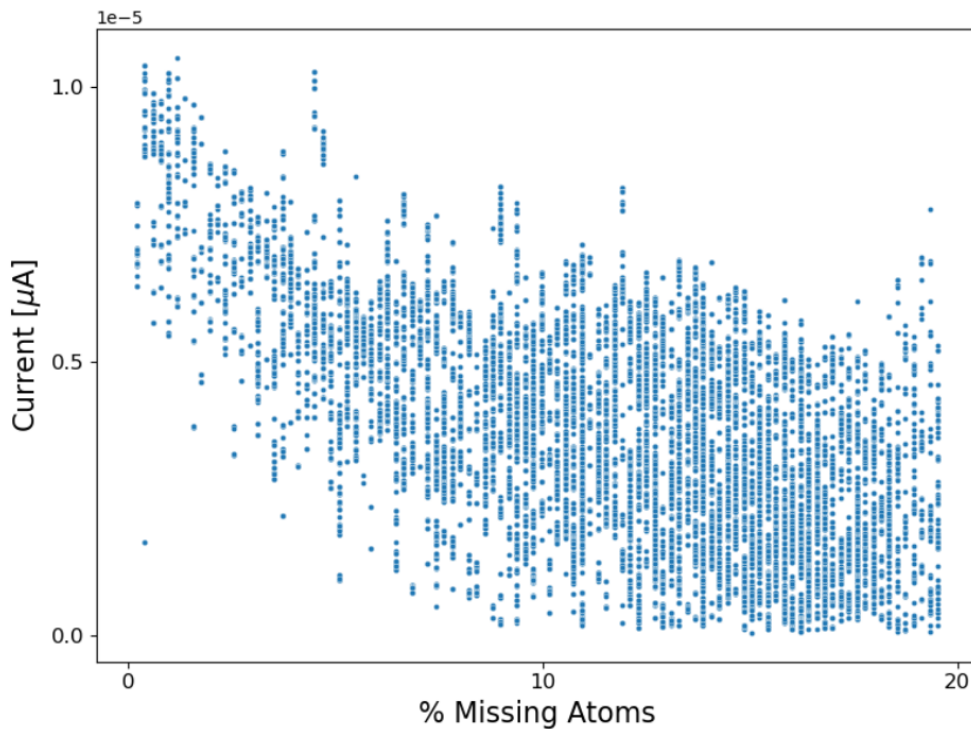


Fig. 6.3 Scatter plot of the computed transport current as a function of the percentage of missing atoms in each graphene flake

Transport simulations were then carried out using the previously computed Fermi level values, together with the potential difference applied between source and drain. The `TransportOnly` option was employed to activate the appropriate solver for the transport calculations. The `TunnelingAndDos` section of the input file defined the energy range over which both the transmission function and the Local Density of States (LDOS) were evaluated (`EnergyRange`), the energy sampling step (`EnergyStep`), and the regions onto which the LDOS was projected (`Region`). In this study, a separate region was defined for each atom in the device, with the energy range set to ± 0.5 V relative to the average Fermi level between source and drain, and the energy step chosen to sample 100 values per atomic site. The computed transport current values were collected for each simulation, along with data files for each atom in the flake. As expected, the calculated current decreased with the increasing fraction of missing atoms in the flake (see 6.3). Each data file contains the LDOS for the corresponding atomic site, providing information about the spatial distribution of electronic states. The structure of the LDOS file consists of two columns: the first column represents the energy, while the second column corresponds to the density of states. The number of rows is determined by the ratio between the defined energy range and the energy step.

6.3.5 STM images calculation

The workflow for computing Scanning Tunnelling Microscopy (STM) images through the construction of a digital twin of the microscope is based on the Tersoff–Hamann theory [131]. The first step involves calculating the Local Density of States (LDOS) for each atom of the flake. The output for a given flake therefore consists of one file per atom, where each file contains as many rows as the chosen energy step size within the selected range, and two columns: the first representing the energy values and the second the corresponding density of states for that specific atom at that energy.

The STM model implemented in this work is the constant–height mode. Once this configuration is fixed, the tunnelling current I can be calculated as the definite integral of the LDOS $\rho(E)$ over the relevant energy window. In our case, the integration limits were chosen as $\pm V_b$ around the Fermi energy, where V_b denotes the applied bias voltage. Formally:

$$I = \int_{E_{\min}}^{E_{\max}} \rho(E) dE \quad (6.4)$$

with $E_{\min}$ and $E_{\max}$ determined by the sign of the applied bias: for positive bias, $E_{\min} = E_F$ and $E_{\max} = E_F + V_b$, and conversely for negative bias. The integral was discretised and evaluated numerically using the trapezoidal rule.

The STM image corresponding to each sample was then generated by superimposing the contributions from all atoms. A 256×256 grid was defined at a fixed height z_h above the surface (as shown in Fig. 6.2b). For each grid point (x_i, y_j) , the STM signal was computed as a superposition of exponentially decaying contributions from all atoms, where the decay factor depends on the Euclidean distance between the observation point (x_i, y_j, z_h) and the atomic coordinates:

$$\text{img}(i, j) = - \sum_{a=1}^N I_a \exp[-\tau \|\mathbf{r}_{ij} - \mathbf{r}_a\|_2], \quad (6.5)$$

where $\mathbf{r}_{ij} = (x_i, y_j, z_h)$ is the position of the grid point, $\mathbf{r}_a = (x_a, y_a, z_a)$ denotes the coordinates of the a -th atom, τ is the decay constant, and I_a the tunnelling contribution associated with that atom.

All the images in the dataset were obtained using a bias voltage $V_b = 0.5$ V, a decay constant $\tau = 5$, and a scan height $z_h = 1$ Å, with the integration window chosen as ± 0.5 V. In Fig. 6.2b), an interesting effect can be observed: regions of high electronic density tend to localise in the immediate vicinity of defects.

A further remarkable feature, which deserves deeper investigation, occurs when multiple defects are present in the same flake. In such cases, the high-density regions concentrate preferentially around only one of the two defects. Within the scope of this work, the possible correlation between the type of defect and the localisation of the density was not explored in detail, nor was the physical mechanism underlying this behaviour clarified. Another consideration concerns the nature of the high-density values themselves. This effect could also be related to the initial choice of the total magnetic moment, which was set to zero in all calculations. The presence of lattice defects or vacancies breaks the symmetry of the flake, potentially inducing localised magnetic moments.

From a computational perspective, to accelerate the image generation process, we employed the numba library [132] and applied just-in-time compilation to the

nested triple for-loop responsible for constructing the STM images. Furthermore, to make the simulated STM images more realistic, a standard post-processing step, typical also of real STM instruments, was applied to enhance the visual contrast of high-intensity regions. Specifically, once the STM image was computed according to Eq. 6.5, the results were rescaled to the range $[0, 1]$. Subsequently, a contrast-limited adaptive histogram equalisation (CLAHE) [133] was applied with a kernel size $\kappa = 16 \times 16$ and a clip limit $\varepsilon = 0.01$. Finally, a gamma correction with $\gamma = 1.5$ was performed by raising the image to the power of $1/\gamma$, further refining the brightness distribution.

6.3.6 Data encoding and FAIR workflow

Taking inspiration from the QM7-X dataset [134], which has become a benchmark for molecular property prediction from structural information, we organised the data generated through the previously described workflow into a collection of HDF5 files. Each HDF5 file contains a total of 10,000 entries, with each entry corresponding to a defective graphene sample. The information is stored hierarchically, including all calculated properties together with metadata that facilitate data retrieval and reuse across different workflows, as illustrated in Figure 6.2 c.

The HDF5 structure, with its key-based organisation, enables us to encode the atomic number, atomic coordinates, and the full set of computed structural and electronic properties, as summarised in Table 6.1. A further advantage of this storage format is its ability to accommodate data of heterogeneous dimensionality. For instance, the number of atomic coordinates associated with a given entry is not fixed but depends on the number and size of defects in the flake. Accordingly, the corresponding dataset has as many rows as atoms in the structure, while the three columns represent the Cartesian coordinates (x, y, z) . To ensure full data traceability, each flake is assigned a Universally Unique Identifier (UUID). For the storage of STM images, compressed image formats were employed.

The dataset thus constructed has been deposited on the Zenodo platform [124], in line with the FAIR principles introduced in Chapter 3.1. By adopting these principles, we aim to guarantee transparency in scientific research and to maximise the reusability of the data in diverse contexts. Particular emphasis was placed on the integration of metadata, as summarised in Table 6.2. These root-level metadata

provide essential information on dataset provenance, content, and traceability of the most recent modifications, as well as the conditions of access under which the dataset can be utilised.

Symbol	Property	Unit	Dimension	Type	HDF5 keys
n_A	Number of atoms	-	1	S	OptGeom/NumAtoms
Z	Atomic numbers	-	n_A	S	OptGeom/AtomicNumbers
R	Coordinates	$\text{\AA}$	$3 \times n_A$	S,G	OptGeom/Coordinates
I	Inertia tensor	$\text{amu} \cdot \text{\AA}^2$	3×3	S,G	OptGeom/InertiaTensor
E_{tot}	Total energy	eV	1	M,G	TotalEnergy
E_{form}	Formation energy	eV	1	M,G	FormationEnergy
E_F	Fermi energy	eV	1	M,G	FermiEnergy
EA	Electron affinity	eV	1	M,G	EA
IP	Ionization potential	eV	1	M,G	IP
E_{gap}	Band gap	eV	1	M,G	BandGap
STM	STM image	-	$3 \times 256 \times 256$	S	ImageSTM
n_E	Number of atoms with electrodes	-	1	S	OptGeomElectrods/NumAtoms
Z_e	Atomic numbers with electrodes	-	n_E	S	OptGeomElectrods/AtomicNumbers
R_e	Coordinates with electrodes	$\text{\AA}$	$3 \times n_E$	S,G	OptGeomElectrods/Coordinates
I_e	Inertia tensor with electrodes	$\text{amu} \cdot \text{\AA}^2$	3×3	S,G	OptGeomElectrods/InertiaTensor
C	Transport current	μA	1	M,R	TransportCurrent

Table 6.1 Properties included in the HDF5 dataset. Each property is denoted by a symbol, which includes its units and dimensions (n_A is the number of atoms in the DG sample; n_E is the number of atoms in the sample including the electrodes), and can be located within the HDF5 files using the corresponding HDF5 keys. Properties are categorized into distinct types: structural (S), molecular (M), atom-in-a-molecule (A), ground-state (G), and response (R).

The Digital Object Identifier (DOI) was assigned by Zenodo, thereby providing a persistent and citable reference that guarantees long-term accessibility and enhances findability. To further improve usability, a uniform naming convention was adopted across datasets and groups (e.g., AtomicNumbers, Coordinates, OptGeom, TransportCurrent), enabling straightforward navigation and efficient retrieval of specific information within the hierarchical structure.

Accessibility is supported through the choice of the HDF5 format, a widely adopted open standard designed for handling large-scale numerical data and compatible with numerous scientific software packages and programming environments.

With respect to interoperability, the structured representation of properties within the HDF5 files (e.g., atomic numbers, coordinates, energies) ensures that the dataset

can be readily used with other tools adhering to common community conventions. The use of attributes attached to datasets and groups provides machine-readable metadata, which simplifies automated parsing and interpretation across different computational frameworks. This interoperability is particularly advantageous in the context of materials informatics, where data stored in HDF5 can be exported to alternative formats (e.g., JSON, SQL-based repositories) and seamlessly integrated into high-throughput pipelines for screening, property prediction, or inverse design [135]. Furthermore, adopting recognised metadata schemas strengthens compatibility with community-driven infrastructures such as the NOMAD Repository [33] and the Materials Project [136], thus reinforcing discoverability and reuse. Reusability is ensured by detailed documentation accompanying the dataset, including descriptions of the computational workflows, data acquisition protocols, and the structure of the stored information. The dataset is released under the Creative Commons Attribution 4.0 International Licence (CC-BY-4.0), which permits redistribution, modification, and reuse provided that proper attribution is given. Versioning, together with the inclusion of attributes such as *created* and *last_updated*, provides traceability of modifications, ensuring the resource remains reliable and current.

Reproducibility is supported by the explicit reporting of the data generation procedures described in the earlier sections, complemented by worked examples of geometry optimisation and property calculations (see Code Availability section). In addition, the dataset was evaluated using the F-UJI tool [137], which systematically assesses compliance with FAIR principles by analysing metadata quality and accessibility features. The assessment confirmed an *advanced* level of FAIRness, with a score of 92%, thereby validating the robustness of the metadata and the openness of the dataset.

Metadata Field	Description
authors	List of authors who contributed to the dataset.
contact	Contact email addresses for inquiries and support.
created	Date when the dataset was created.
description	General description of the dataset contents and scope.
doi	Digital Object Identifier (DOI) for the dataset.
format	File format used (e.g., HDF5).
funding	Information about funding sources and grants supporting the work.
generated_by	Software/tool used to generate the dataset.
keywords	Keywords summarizing the dataset's main topics and domains.
language	Language used in the metadata and documentation (e.g., English).
last_updated	Date of the most recent update to the dataset.
license	License under which the dataset is released (e.g., CC-BY-4.0).
mime_type	MIME type of the dataset file.
related_identifiers	References to related works or datasets via persistent identifiers.
title	Full title describing the dataset.
url	Public URL to access the dataset landing page.
version	Version number of the dataset release.

Table 6.2 Root-level metadata attributes contained in the dataset file, with their descriptions.

6.3.7 Data Validation

The entire dataset was validated to ensure the correctness of the calculated data. The dataset is deterministic and the results are reproducible. Global statistics for all properties were computed and are summarised in Table 6.3.

To validate the simulated properties, a subset of 1,000 structures was randomly extracted from the dataset, including their corresponding optimized geometries and

Properties	Mean	STD	Max	Min	Median	Skewness	Kurtosis
Number of atoms	459.78	25.91	511	412	457	0.29	1.98
Total energy per atom (eV)	-47.21	0.06	-46.99	-47.34	-47.21	0.04	2.68
Formation energy per atom (eV)	-9.16	0.06	-8.94	-9.28	-9.16	0.04	2.68
Fermi energy (eV)	-5.00	0.11	-4.68	-5.36	-5.01	0.47	3.06
EA (eV)	4.94	0.13	5.35	4.42	4.96	-0.99	5.35
IP (eV)	5.09	0.10	5.45	4.81	5.09	-0.09	2.50
Band Gap (eV)	0.14	0.05	0.52	0.08	0.13	4.61	30.95
Current (μ A)	4.17	2.30	10.54	0.05	4.14	0.33	2.49

Table 6.3 Dataset properties analysis in terms of Mean, STD, Max, Min, Median, 3rd Moment (Skewness) and 4th Moment (Kurtosis).

electrodes. This subset was subjected to new simulations using alternative Slater-Koster parameters and a more stringent convergence criterion. Specifically, key electronic properties (EA , IP , E_{tot} , E_{gap} , and E_f) were recalculated employing the *scc – cp2k* [138] Slater-Koster parameters, with the *SccTolerance* reduced from the default value of 1×10^{-5} to 1×10^{-10} . Additional transport current simulations were performed for each structure within the subset using the same *scc – cp2k* parameters.

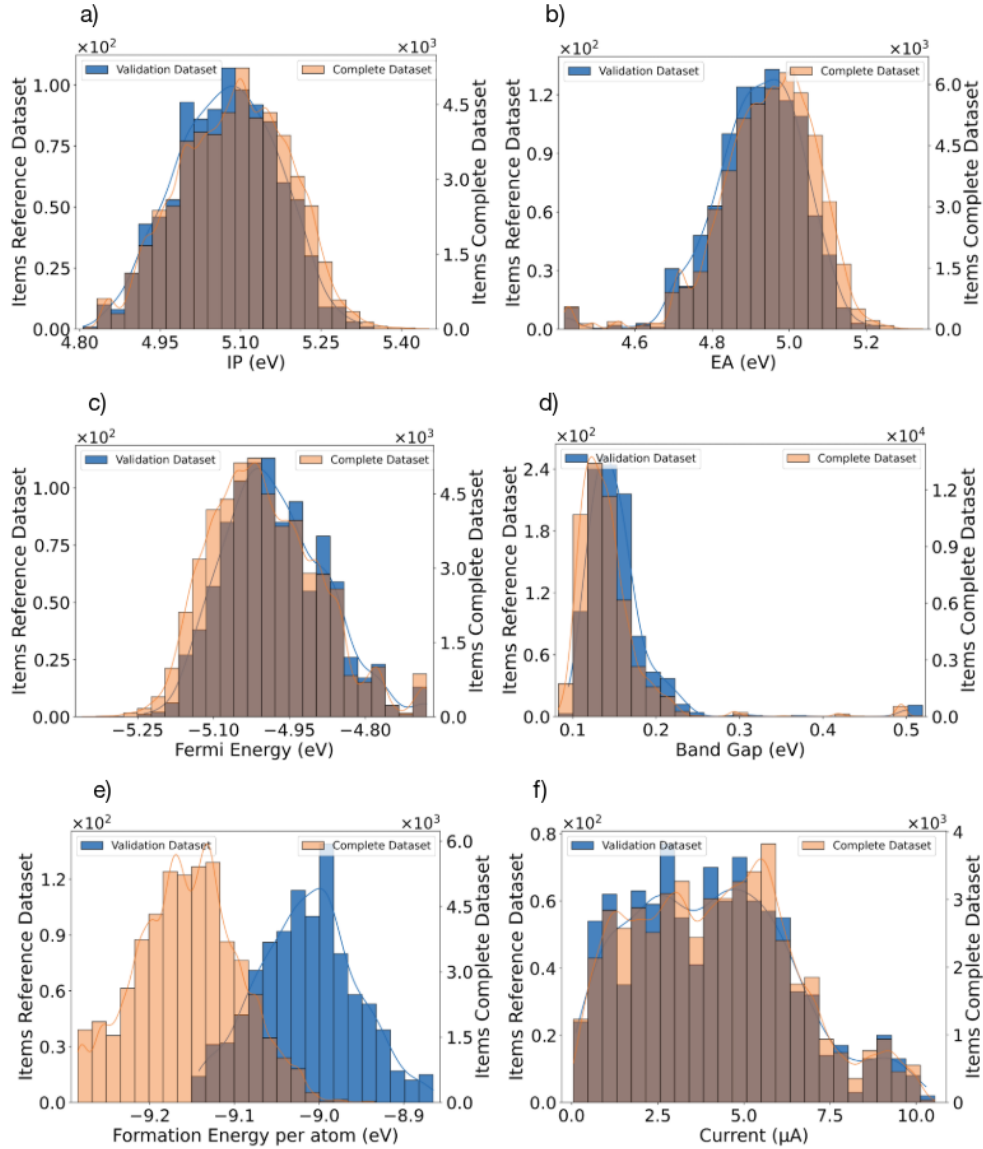


Fig. 6.4 Map of the Mulliken charges for a selected DG sample computed with the a) SCC and b) non-SCC approximations.

The resulting distributions of the validation subset were compared against those of the full dataset (see Image 6.4). The comparison revealed largely consistent distributions across all properties, with the exception of the total energy per atom, which was observed to be slightly elevated on average within the validation subset.

To further assess the reliability of the transport calculations, a single structure was randomly selected from the dataset to inspect Mulliken charge distributions under

both non-SCC and fully self-consistent SCC transport conditions, including spin polarization. The analysis showed that the charge distribution over the graphene flake and electrodes remained essentially unchanged, providing additional confirmation of the methodological robustness (see Image 6.5).

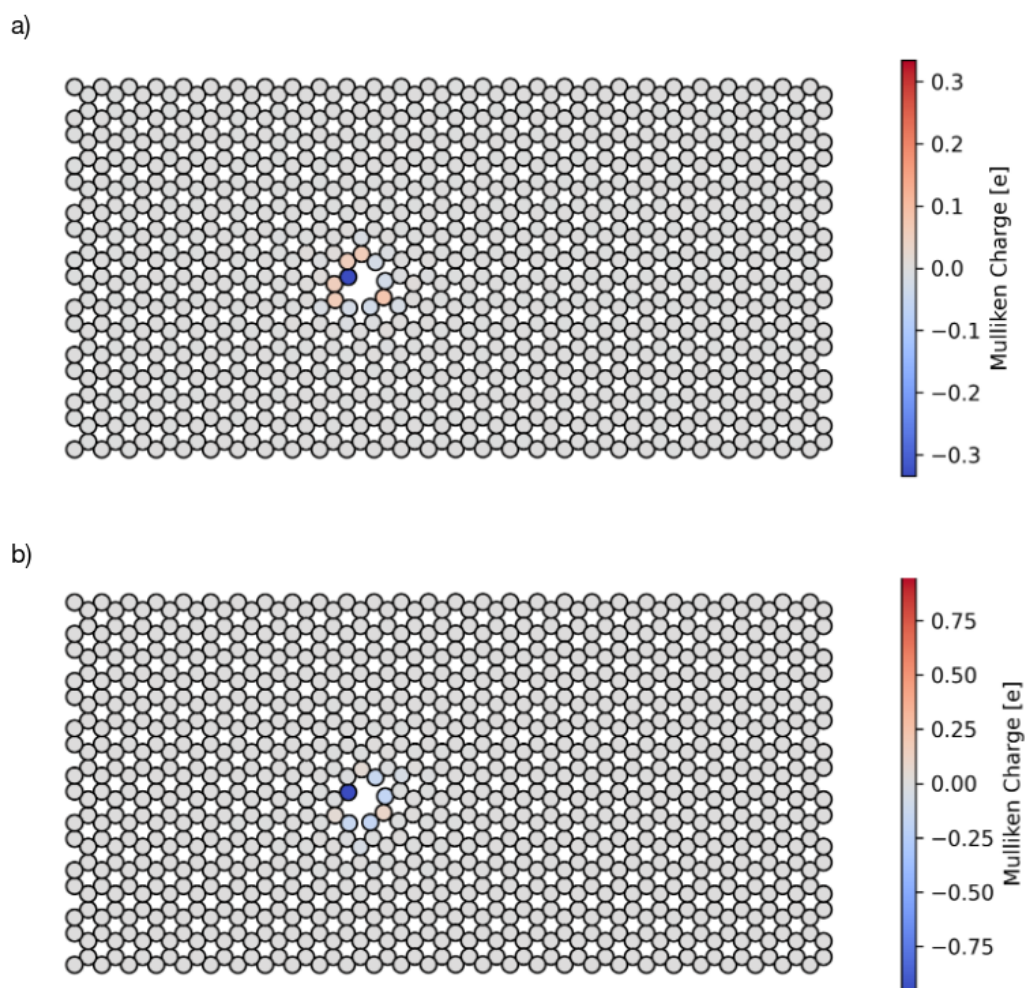


Fig. 6.5 Map of the Mulliken charges for a selected DG sample computed with the a) SCC and b) non-SCC approximations.

6.4 Defect–property correlations with XGBoost

6.4.1 General Overview

Using the dataset workflow previously described, the next step involved employing the electronic properties computed with the workflows presented in Sections 6.3.2 and 6.3.4. A new computational workflow was then introduced, integrating these electronic properties with an image-based and object-detection pipeline. This integrated approach enables the prediction of the electronic properties of graphene flakes starting solely from information that can also be accessed at the macroscale or derived from experimental microscopy images. The workflow isolates individual defects and extracts both geometric and frequency-domain features, allowing the derivation of morphological descriptors that characterise the defects and their influence on the overall structure.

All the extracted descriptors were then used to train a boosting-based tree model (XGBoost), which successfully revealed correlations between nanoscale morphology and the electronic properties of graphene. In particular, throughout this part of the work, Explainable AI (XAI) techniques were employed to highlight the relative importance of the morphological defect features within the graphene flake, and to correlate and interpret these features with respect to the prediction of the current-related target.

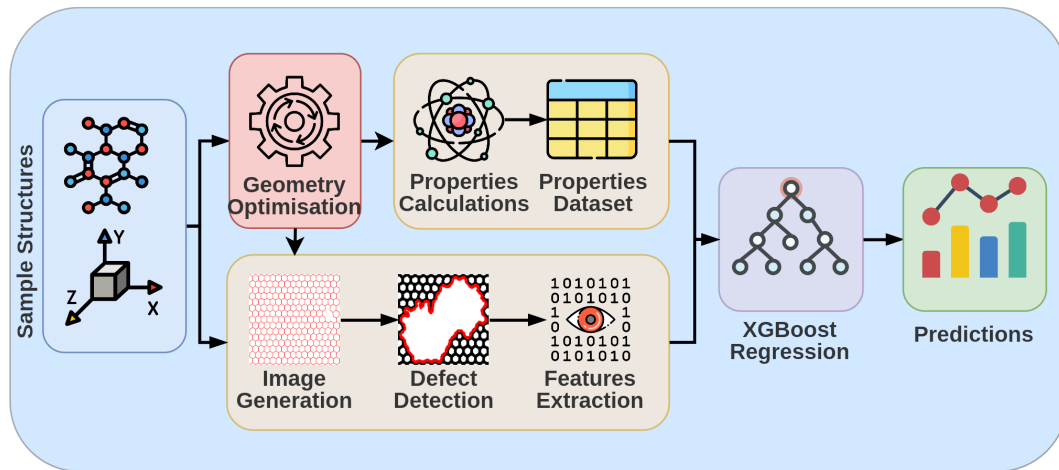


Fig. 6.6 Scheme of the prediction workflow applied to predict electronic and transport properties of DG samples from morphological features of defects.

6.4.2 The Process of Image Generation

As illustrated in Figure 6.6, two distinct workflows are involved in this study. The first, already described in Sections 6.3.2 and 6.3.4, is used to compute the electronic properties that serve as labels for model training and evaluation. The second workflow starts from the optimised geometries and generates graphical representations of the graphene flakes. From these images, structural defects are detected and morphological features are extracted. The rationale behind this approach is to convert the XYZ structural data into low-resolution images that capture the overall morphology and bonding network of the flakes. This enables a comprehensive yet computationally efficient representation of the system, allowing the processing of a large number of samples within a short computational time.

Algorithm 2 Generate Graphene Flake Images

- 1: **Input:** XYZ structural files, magnifier factor M
 - 2: Read all XYZ files and compute $X_{\max}$ and $Y_{\max}$
 - 3: Compute $res = \text{round}\left(M \times (5 + \max(|X_{\max}|, |Y_{\max}|))\right)$
 - 4: Create an empty 2D image of size $res \times res$
 - 5: Guess bonds using Chemfiles and XYZ coordinates
 - 6: **for** each bond i **do**
 - 7: $x_{\text{bond}1} \leftarrow \text{nint}(x_{\text{start}}^i \times M)$
 - 8: $y_{\text{bond}1} \leftarrow \text{nint}(y_{\text{start}}^i \times M)$
 - 9: $x_{\text{bond}2} \leftarrow \text{nint}(x_{\text{end}}^i \times M)$
 - 10: $y_{\text{bond}2} \leftarrow \text{nint}(y_{\text{end}}^i \times M)$
 - 11: Define line segment $[(x_{\text{bond}1}, y_{\text{bond}1}), (x_{\text{bond}2}, y_{\text{bond}2})]$
 - 12: Set the colour of the line based on atom types
 - 13: Draw the line on the image
 - 14: **end for**
 - 15: **Output:** 2D image representing the graphene flake
-

The image generation process is summarised in Algorithm 2. As illustrated, the first step consists of reading the XYZ files and computing $X_{\max}$ and $Y_{\max}$, which correspond to the maximum coordinates of the flake. Next, a magnifier factor (M) is defined, and the resolution is then calculated as a function of M , $X_{\max}$, and $Y_{\max}$:

$$res = f(M, X_{\max}, Y_{\max})$$

After computing the resolution, a two-dimensional array of size $res \times res$ is created. Subsequently, for each XYZ file, a bond guessing procedure is performed [139], which produces a list of tuples indicating which pairs of atoms are bonded to each other. At this stage, the coordinates of all atom pairs forming the chemical bonds are obtained. These coordinates are subsequently converted into pixel positions within the image by scaling them with the magnifier factor M and rounding to the nearest integer ($nint$). For each bond, a line segment is drawn connecting the corresponding pixel coordinates, with the line colour encoding the types of atoms involved. In the present study, since the structures under investigation consist exclusively of carbon atoms, all bonds are represented using a uniform colour; red has been chosen for this purpose.

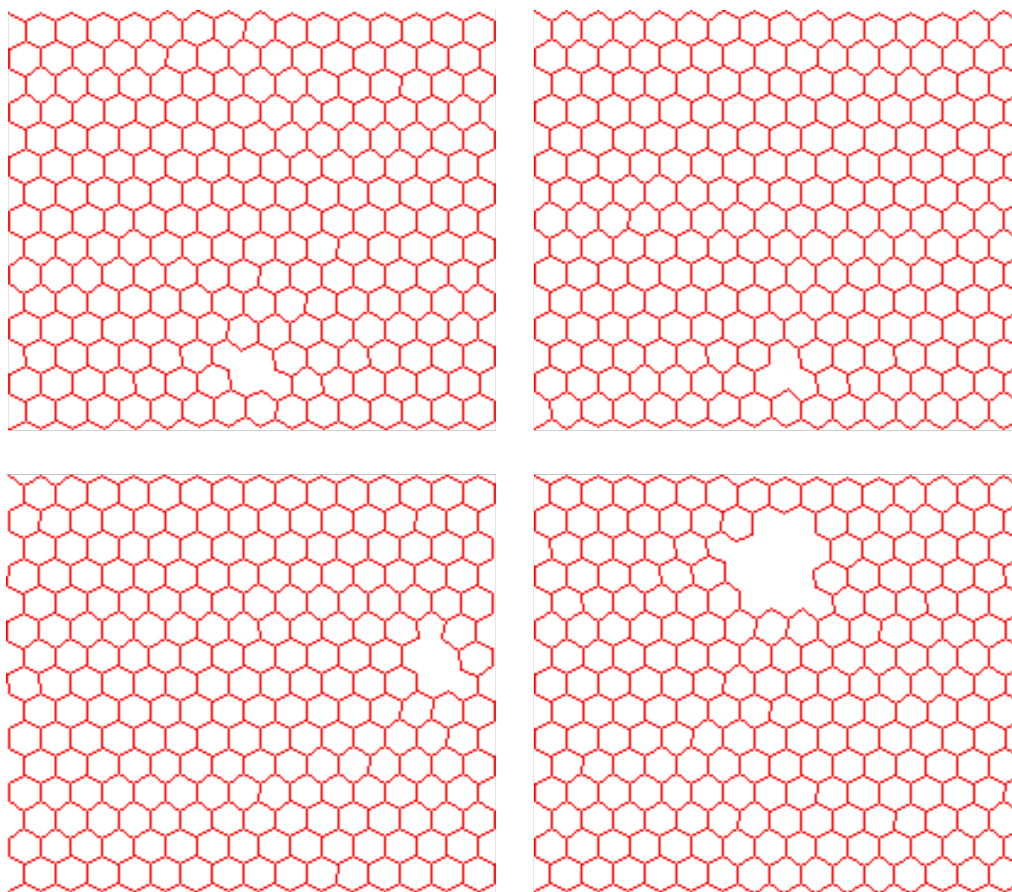


Fig. 6.7 Proposed image-based representation for defected graphene flakes

The resulting PNG images often contain substantial regions of empty space, particularly around smaller graphene flakes, which do not contribute meaningful structural information. To address this, a cropping procedure is applied to remove these extraneous areas, thereby reducing the overall image dimensions for small structures and improving the efficiency of dataset storage and subsequent processing. An example of the generated representation for a defected graphene sample is provided in Figure 6.7, illustrating both the bond network and the overall morphology captured by the workflow. In the present study, a magnifier factor of $M = 6$ was selected. Given maximum coordinates of approximately $X_{\max} \simeq 39.5$ and $Y_{\max} \simeq 34.3$, this choice results in images with a resolution of approximately 240×240 pixels after the cropping procedure has been applied.

6.4.3 Defect extraction process

To establish a reliable correlation between the defects within graphene structures and their corresponding electronic and transport properties, the first necessary step—after generating the simulated images as described in Section 6.4.2 is the identification of structural defects. This task was approached using an object detection algorithm, specifically YOLOv10 [64], applied directly to the generated images.

For training the detection model, a manually labelled dataset was constructed using the Label Studio software [140]. Approximately 300 images were randomly sampled from the entire pool of generated images and annotated with bounding boxes around the observable defect regions. Once the labelling process was completed, all annotations were exported in TXT format, following the YOLO-compatible schema.

To increase dataset variability and stabilise the training process, several data augmentation techniques were applied to the original labelled images. These included 180° rotation, horizontal, vertical, and diagonal flipping (with respect to both axes), colour channel swapping, and negative inversion. For each spatial transformation, the associated label coordinates were correspondingly transformed to preserve the consistency and alignment of the bounding boxes within the augmented images. All augmentation operations were implemented using the OpenCV library [141], resulting in the generation of six augmented variants per original image, thereby expanding the dataset size by a factor of six.

Finally, the augmented dataset was split into training, validation, and testing subsets, with 70% of the images used for training, while the remaining 30% was equally divided between validation and testing. This balanced partition ensured adequate coverage of the defect typologies during both model optimisation and performance assessment.

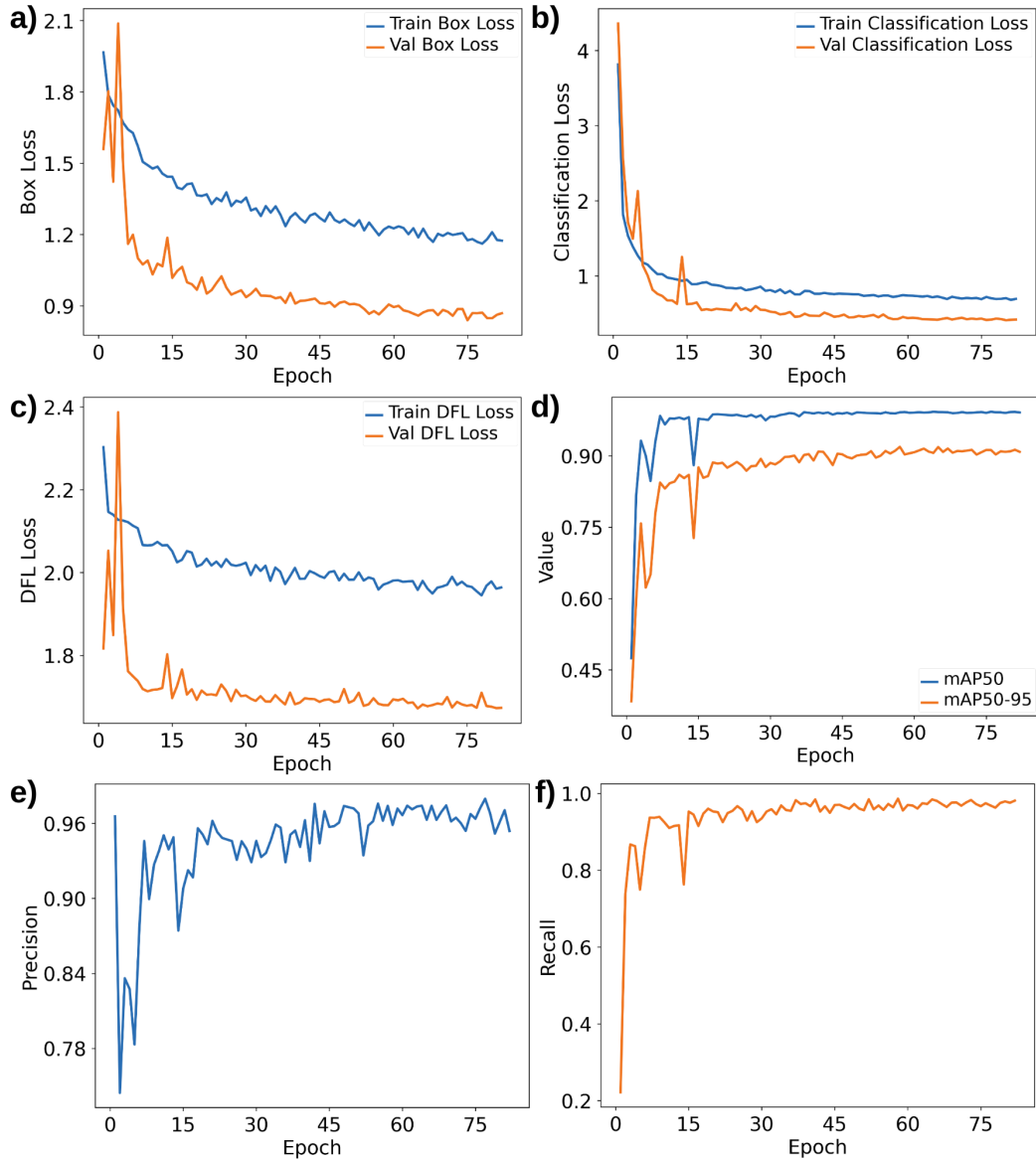


Fig. 6.8 Training and validation metrics for the YOLO model. The plots show the evolution of box loss (a), classification loss (b), and distribution focal loss (DFL) during training and validation (c), alongside the mAP50-mAP50-95 (d), precision (e), and recall (f) metrics. The consistent decline in losses and the increase in performance metrics indicate successful convergence of the model and robust detection performance.

Once the image generation and dataset preparation stages were completed, the next step involved training the detection model. The training was carried out using the original YOLOv10 implementation provided by Ultralytics, with 300 epochs, a batch size of 32, an image size of 256 pixels, and six workers for data loading. Early stopping was applied with a patience of 25 epochs, and the model was trained from scratch without using pretrained weights. All the training metrics are represented in the image 6.8.

6.4.4 Feature extraction

Once the YOLO model had been trained to identify defects within the graphene images, individual crops corresponding to each detected defect were extracted to enable quantitative feature computation. Within these cropped regions, the carbon–carbon bonds were enhanced and the RGB images were converted into binary masks, with a white background and black bonds, to facilitate subsequent feature extraction and analysis.

The extracted features were categorised into four primary groups: *Texture Features*, *Shape Features*, *Edge Features*, and *Frequency Features*. Since texture, shape, and edge features are defined only for single-channel images, each defect crop was first converted to grayscale and then subjected to histogram equalization using the OpenCV library [141]. These features served as the input for machine learning models aimed at predicting electronic and transport properties of the graphene flakes. A detailed description of the extracted features is provided in the following sections.

To obtain a comprehensive representation of each full flake image, the features extracted from individual defect crops were aggregated whenever a flake contained more than one defect. Aggregation was performed using either summation or weighted averaging, depending on the type of feature, to ensure that the contributions of multiple defects were accurately reflected in the overall characterization of the flake. Specifically, the area and perimeter of all detected defects were summed to obtain the total defect area and total defect perimeter for the flake. For the remaining geometric features, weighted averages were computed, where each defect's contribution was scaled by its area, so that larger defects had a proportionally greater impact on the aggregated values. This approach preserves the relative significance of each

defect, resulting in a balanced and representative description of the defect properties for flakes containing multiple defects.

The described feature extraction and aggregation pipeline was applied consistently across all images in the dataset, generating a set of descriptors that form the basis for the subsequent machine learning analysis of electronic and transport properties.

Texture Features

To characterise the texture of defect regions, the *Gray-Level Co-occurrence Matrix* (GLCM) was employed. This statistical method captures the spatial dependencies between pixel intensity pairs in an image, providing quantitative measures of textural patterns [142]. The implementation from the `scikit-image` library [143] was used, with analysis focused on four principal features: *contrast*, *homogeneity*, *energy*, and *correlation*.

Contrast reflects the degree of local intensity variation within the grayscale image by examining differences between neighbouring pixels. It is calculated from the GLCM for a given spatial offset, with higher values indicating more pronounced intensity changes, useful for highlighting sharp transitions or high-frequency regions.

Homogeneity describes the uniformity of the intensity distribution across the image. Computed from the GLCM, higher homogeneity values correspond to smoother and more consistent textures, making it an effective descriptor for regions with minimal intensity fluctuations.

Energy, also known as the angular second moment, quantifies the uniformity of texture by summing the squared GLCM elements. Elevated energy values indicate highly ordered or repetitive patterns, aiding the identification of structured and periodic regions.

Correlation measures the degree of linear dependence between neighbouring pixel intensities. Derived from the GLCM, higher correlation values signal a strong relationship between adjacent intensities, capturing predictable and structured variations within the texture.

Shape Features

The extraction of shape features aimed to quantify the morphological properties of each defect region, capturing information about size, geometry, and spatial distribution. The principal features considered included *area*, *perimeter*, *circularity*, *solidity*, *compactness*, *Feret diameter*, and *eccentricity* [142].

Area corresponds to the total number of pixels contained within the defect boundary, providing a direct measure of the defect’s spatial extent.

Perimeter represents the total length of the boundary surrounding the defect. This metric reflects the complexity and contour characteristics of the defect region.

Circularity assesses how closely the defect shape approximates a perfect circle, and is computed as:

$$\text{Circularity} = \frac{4\pi \cdot \text{Area}}{\text{Perimeter}^2} \quad (6.6)$$

A circularity value of 1 indicates a perfect circle, whereas lower values denote deviations, such as elongation or irregularities.

Solidity is defined as the ratio of the defect area to the area of its convex hull—the smallest convex shape enclosing the defect:

$$\text{Solidity} = \frac{\text{Area}}{\text{Convex Hull Area}} \quad (6.7)$$

Values range from 0 to 1, with 1 representing a fully solid shape and lower values indicating concavities or structural irregularities.

Compactness quantifies the roundness or density of the defect, calculated as:

$$\text{Compactness} = \frac{\text{Perimeter}^2}{\text{Area}} \quad (6.8)$$

Lower values correspond to more compact, circular shapes, while higher values suggest elongated or irregular contours.

Feret diameter, also known as the maximum caliper distance, measures the longest distance between any two points on the defect, providing an indication of its overall scale and orientation.

Eccentricity captures the elongation of the defect and is defined as:

$$\text{Eccentricity} = \sqrt{1 - \left(\frac{\text{Minor Axis}}{\text{Major Axis}} \right)^2} \quad (6.9)$$

Eccentricity values span from 0 for a perfect circle to 1 for a line, with higher values indicating greater elongation of the defect shape.

Edge Features

Edge features describe the structural characteristics and spatial distribution of edges within each defect crop, providing quantitative measures of boundary complexity and texture. In this work, the primary edge descriptors extracted were *number of edges* and *edge density*, both computed using the *Canny* edge detection algorithm [142] as implemented in the OpenCV library [141].

Number of edges quantifies the total count of pixels identified as edges within the defect crop. Unlike the *perimeter*, which measures the continuous boundary length, this metric captures all detected edge pixels, including small structural details and discontinuities along the defect boundary. Higher values correspond to more intricate or irregular defect shapes.

Edge density is defined as the proportion of edge pixels relative to the total number of pixels in the defect crop:

$$\text{Edge Density} = \frac{\text{Number of Edge Pixels}}{\text{Total Pixels in Crop}} \quad (6.10)$$

This feature provides a normalized measure of edge concentration, where higher values indicate densely packed or highly complex boundary structures, reflecting fine-scale textural patterns of the defect.

Frequency features

To capture the frequency domain characteristics of graphene flakes, both the Fourier Transform (FT) and the Haar Wavelet Transform (WT) were applied to the complete flake images [144–146]. These transforms provide complementary information to the

geometric and texture-based features, allowing for a more comprehensive analysis of defect patterns [142] .

The FT is employed to analyse the overall frequency content of the image, revealing dominant frequencies and the distribution of spectral energy. Although FT is effective in highlighting global patterns, it lacks spatial localization, which can be addressed using WT. The Haar wavelet provides a multi-resolution decomposition of the image, preserving both spatial and frequency information and enabling the detection of local features at multiple scales and orientations. This combination of FT and WT ensures both global and local aspects of the image structure are quantified.

Frequency features were extracted using the NumPy [147] and PyWavelets [148] Python libraries. The 2D FT of the input image transforms spatial variations into the frequency domain. The magnitude spectrum is computed and centered on low-frequency components as:

$$M(u, v) = \log (|F(u, v)| + \varepsilon) \quad (6.11)$$

where $F(u, v)$ represents the Fourier coefficient at frequency (u, v) , and ε is a small constant to prevent numerical instabilities. From this spectrum, statistical features were derived, including **mean**, **maximum**, **minimum**, **median**, and **standard deviation** of $M(u, v)$. The total spectral **energy** was calculated as:

$$E = \sum_{u,v} |F(u, v)|^2 \quad (6.12)$$

and the spectral complexity was quantified by **entropy**:

$$H = - \sum_i P(f_i) \log_2 (P(f_i) + \varepsilon) \quad (6.13)$$

where $P(f_i)$ denotes the normalized histogram of the magnitude spectrum. Additionally, the spectrum was divided into low- and high-frequency regions, and their respective energies were used to define the **frequency contrast**:

$$C = E_{\text{high}} - E_{\text{low}} \quad (6.14)$$

with E_{high} and E_{low} corresponding to the total energy in the high- and low-frequency quadrants, respectively.

Wavelet-based features were extracted from both the first (**H1**, **V1**, **D1**) and second (**A2**, **H2**, **V2**, **D2**) decomposition levels, capturing multi-scale information. Key metrics included **wavelet energy**, representing the total coefficient energy at each scale and orientation; **wavelet standard deviation**, measuring coefficient variability; and **wavelet entropy**, quantifying the complexity and local randomness of the coefficients. Together, these FT and WT features provide a detailed characterization of the frequency-domain properties of graphene flakes, complementing the previously computed geometric and texture descriptors.

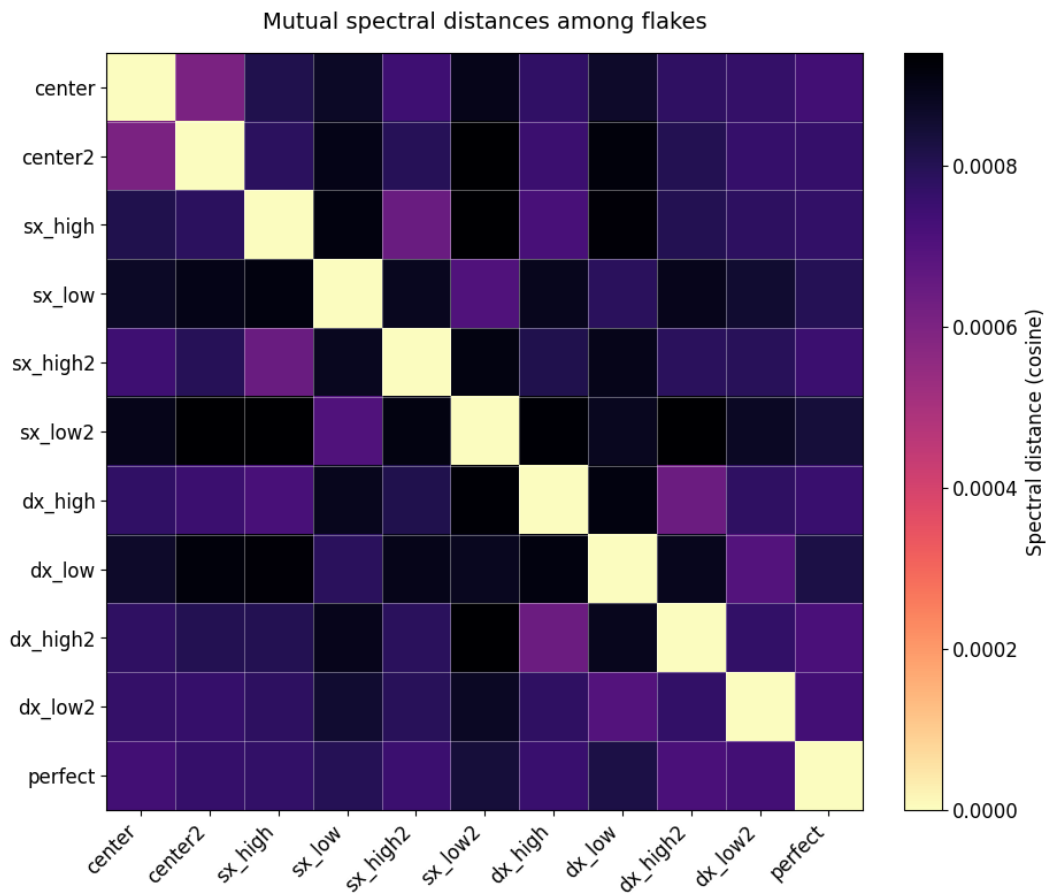


Fig. 6.9 Magnitude spectra of a graphene flake containing a single defect translated across different spatial positions. Each column corresponds to a specific translation (e.g., left, right, up, down), while the lower row shows the difference with respect to a reference configuration. The colour map highlights variations induced by defect translation, demonstrating that Fourier features are sensitive to symmetry breaking induced by defect translation, thereby indirectly encoding positional information.

In order to characterise the effect of defects positions on the spectral properties of graphene flakes, we analyse the Fourier-space representation of the flake images. For a flake containing a single defect, we systematically translate the defect across different spatial positions, both along the horizontal (left-right) and vertical (up-down) directions. The magnitude of the two-dimensional Fourier transform is then computed for each translated configuration, providing a set of spectral features that are invariant to overall image intensity but sensitive to the defect position.

Figure 6.9 illustrates the resulting spectral differences: each subplot shows the magnitude spectrum of the flake for a given defect position, while the corresponding difference map with respect to a reference configuration highlights the variations induced by the translation. As can be seen, even a single-atom shift breaks the symmetry of the Fourier spectrum, producing measurable changes in the spectral features. This confirms that the Fourier representation captures spatially localised variations, and that the extracted spectral features are representative of the defect’s position within the flake.

These observations justify the use of Fourier-based features for subsequent comparative analysis among flakes with similar single-atom defects: by encoding positional information into the spectral domain, these features allow for an effective quantification of defect-induced variations, which can be leveraged in clustering or dimensionality reduction approaches to reveal structural patterns across the dataset.

6.4.5 Model Training

Using the extracted features, we trained the XGBoost algorithm to predict the simulated target properties, employing the implementation available in the XGBoost Python package [149]. A consistent set of hyperparameters was applied across all target properties to ensure comparability and maintain uniformity within the predictive framework. The optimal hyperparameters, reported in Table 6.4, were identified using the Optuna framework [150]. The range of hyperparameters explored during the optimization process was empirically defined and is provided in Appendix A.2, Table B.1.

For hyperparameter tuning, we employed the Tree-structured Parzen Estimator (TPE) sampler [151, 152], a Bayesian optimisation method. TPE models the objective function by constructing two probability distributions: one representing

Parameters	Value
n_estimators	770
learning_rate	0.05121
max_depth	15
gamma	0.00001
min_child_weight	6.37548
colsample_bytree	0.65017
max_leavs	148
subsample	0.91375
reg_alpha	0.52680
reg_lambda	1.93097

Table 6.4 XGBoost parameters optimized using the Optuna framework [150].

hyperparameter configurations deemed promising (i.e., those yielding superior objective values) and the other representing the remaining configurations. Optimisation is then performed by maximising the ratio between these distributions, thereby dynamically balancing exploration of the parameter space with exploitation of regions known to be promising.

The optimization process was guided by a custom objective function, defined as:

$$\mathcal{F}(x) = \min [\alpha \cdot MSE_{val} + (1 - \alpha) \cdot (1 - R^2) + \beta \cdot O] \quad (6.15)$$

where:

$$O = \frac{MSE_{val} - MSE_{train}}{MSE_{train}} \quad (6.16)$$

In this formulation, the first component, scaled by the parameter α , regulates the model's fitting performance by simultaneously minimising the validation mean squared error (MSE_{val}) and maximising the coefficient of determination (R^2). The second component, denoted as O , serves to mitigate overfitting by penalising cases where MSE_{val} is substantially higher than the training mean squared error (MSE_{train}), indicating limited generalisation. The weighting parameters α and β were set to 0.5

and 0.4, respectively, providing a balanced compromise between fitting accuracy and overfitting control.

To assess the robustness of the model and its generalisation capability, a five-fold cross-validation strategy was employed [153]. In this procedure, the full dataset is partitioned into five subsets of approximately equal size; each fold, in turn, is used as the validation set while the remaining four folds serve as the training set. This approach provides more reliable estimates of performance metrics by reducing dependence on individual validation samples and mitigating variance effects due to dataset composition [154, 155]. The error and correlation measures computed for each fold are subsequently aggregated, yielding an overall assessment of the model's predictive capacity and enabling effective comparison of different hyperparameter settings or architectural configurations. The choice of five folds represents a balance between statistical reliability and computational cost, ensuring a robust evaluation without excessively increasing training time.

The final performance metric, incorporated both within the objective function and the overfitting term, was determined as the mean validation score across all cross-validation folds. To facilitate the hyperparameter optimisation procedure, tuning was conducted solely for a single target property, the transport current, due to its primary significance in assessing the applicability of DG flakes in device contexts. The optimised hyperparameters were subsequently applied to all remaining target properties, under the assumption that the model would generalise effectively across diverse prediction tasks.

Following hyperparameter selection, the dataset was divided into a training set (80%) and a test set (20%). Input features were standardised using the `RobustScaler` from the `scikit-learn` Python library [143], which relies on statistics that are insensitive to outliers. The model was then trained on all target variables using the optimised hyperparameters. Learning curves, showing the relationship between boosting rounds and L2 error for both the training and test sets, are provided in the Fig. 6.10.

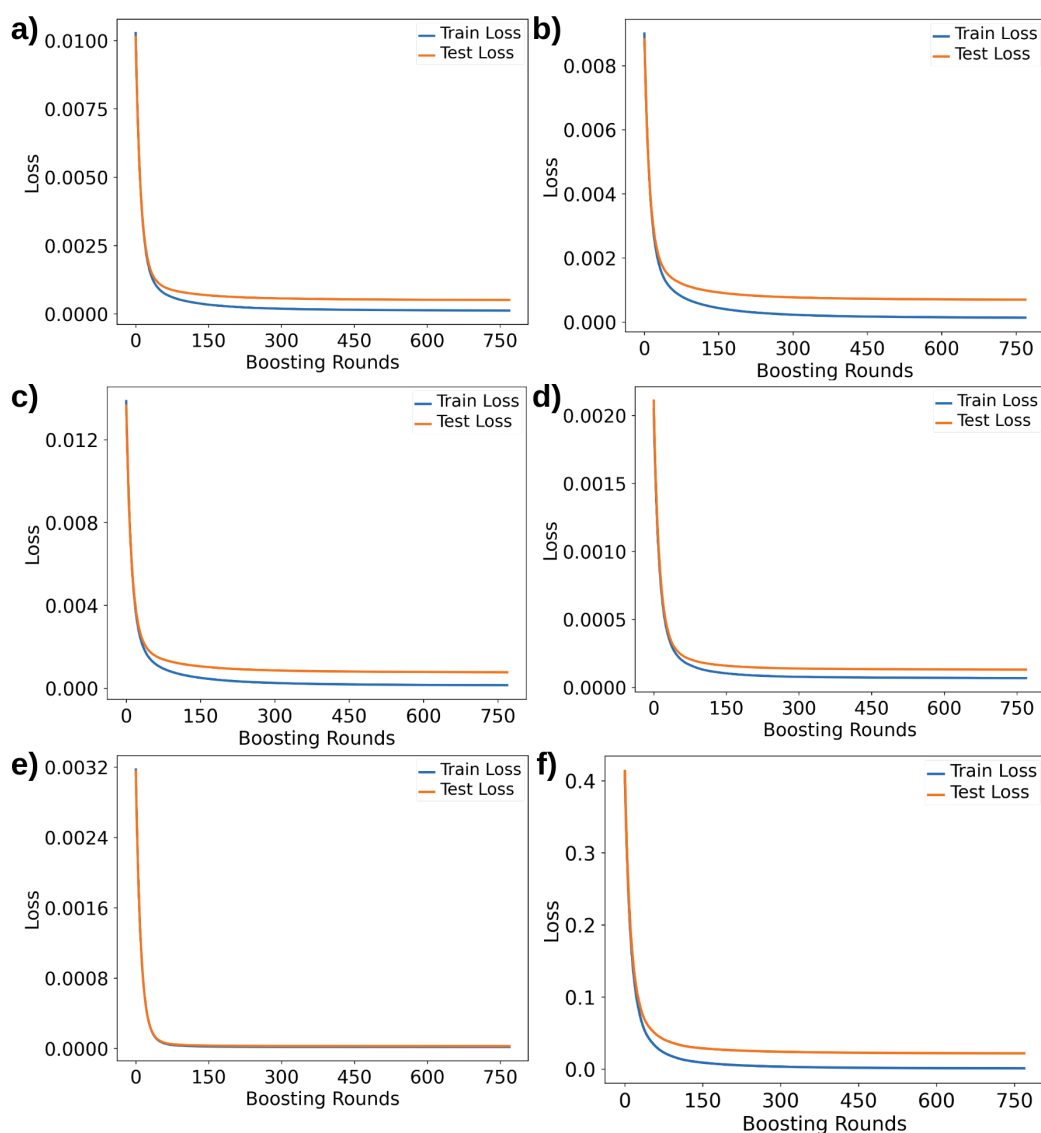


Fig. 6.10 The plots show the boosting rounds compared to RMSE errors for the train set and test set using XGBoost for Fermi level (a), ionization potential (b), electron affinity (c), band gap (d), energy per atom (e), and current (f).

All hyperparameter tuning and training were performed on a workstation equipped with an AMD Ryzen 7 5800X CPU and an Nvidia RTX 3070 Ti GPU, ensuring efficient computation and reproducibility of results.

6.4.6 Results Discussion

As a first step, we analyse the performance of our defect detection algorithm introduced in Section 6.4.3. The evaluation is supported quantitatively by Figure 6.11 and complemented with a more direct visual inspection through the cropped examples shown in Figure 6.12. The confusion matrix (Fig. 6.11a) indicates that the model correctly identifies the vast majority of defects, with only a few misclassifications, reflecting a high overall detection accuracy.

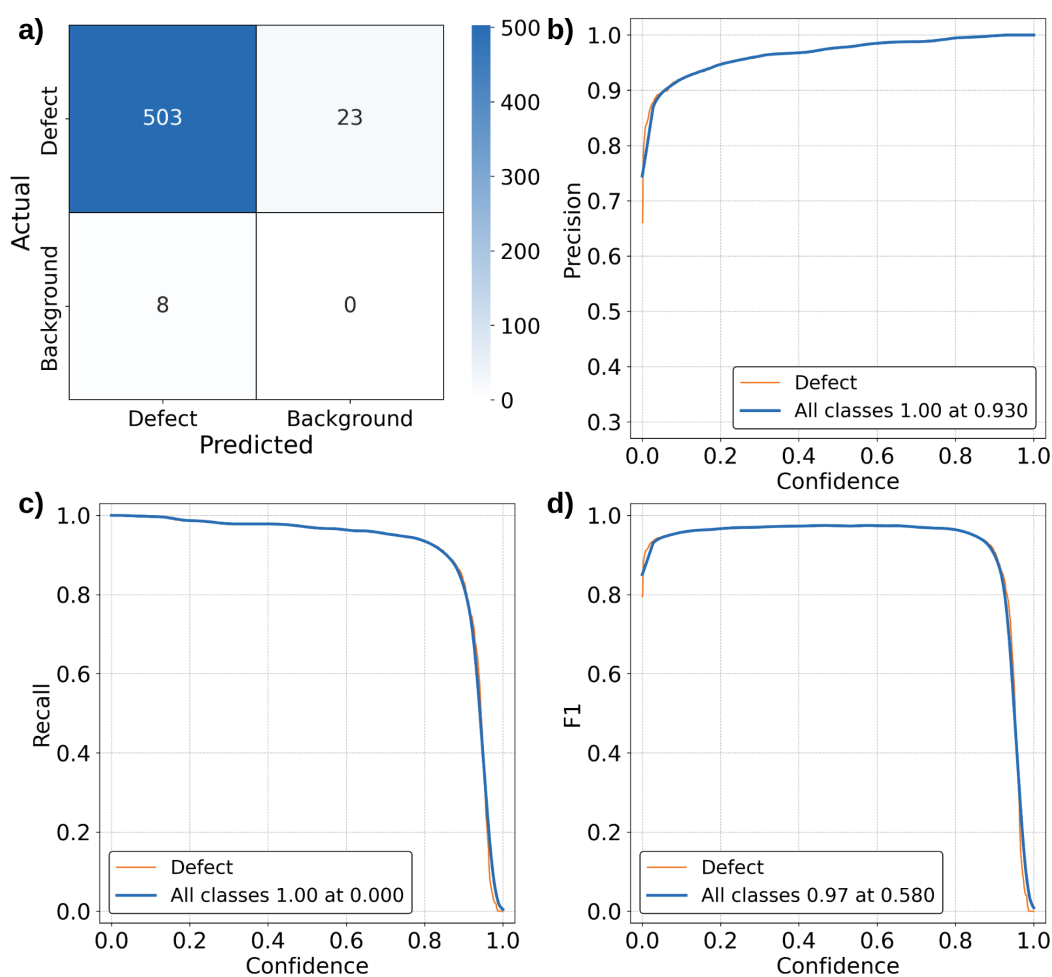


Fig. 6.11 Performance evaluation of the YOLOv10 model for defect detection in graphene structures. (a) Confusion matrix showing true positives, false positives, and false negatives for defect classification. (b) Precision curve illustrating the model's precision across varying confidence thresholds. (c) Recall curve demonstrating the model's ability to capture the majority of actual defects. (d) F1-score curve highlighting the balance between precision and recall, with the model achieving high reliability and accuracy in defect detection.

The precision curve (Fig. 6.11b) demonstrates that the model maintains a high level of precision across varying confidence thresholds, approaching unity for predictions made with higher confidence. This suggests that predictions of defect presence are highly reliable. Similarly, the recall curve (Fig. 6.11c) shows that the model successfully detects the majority of actual defects, maintaining consistently high recall values across different confidence levels. The F1-score curve (Fig. 6.11d), which balances precision and recall, reaches a maximum of approximately 0.97, indicating that the model achieves an optimal trade-off between precision and recall.

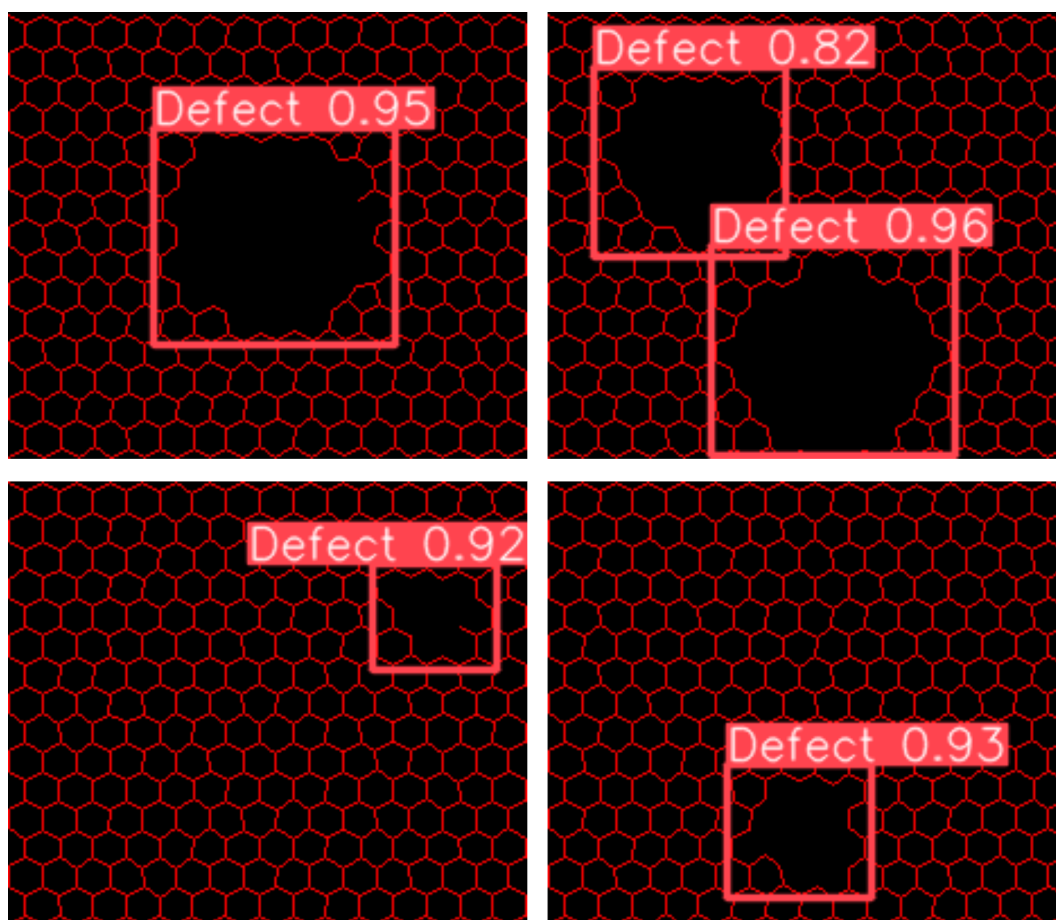


Fig. 6.12 Defects detected in 4 different nanographene samples by the trained YOLOv10 model, along with their respective confidence values.

Overall, these results demonstrate the robustness and reliability of the YOLOv10 object detector in identifying defects within DG images. Representative examples of defects detected by the trained YOLOv10 model in nanographene flakes are presented in Fig. 6.12.

Building on these findings, the next step is to investigate in greater detail the prediction of material properties using our boosting algorithm. To this end, the trained XGBoost model, described in Section 6.4.5, was employed to predict all target properties previously calculated via DFTB simulations. The model’s performance was evaluated using the MSE , MAE , and R^2 metrics, as summarised in Table 6.5.

Target	MSE	MAE	R^2
Fermi Level [eV]	0.0005	0.0146	0.9540
Ionization Potential [eV]	0.0007	0.0164	0.9269
Electron Affinity [eV]	0.0008	0.0171	0.9479
Band Gap [eV]	0.0001	0.0079	0.9427
Energy per atom [eV]	2.7e-05	0.0032	0.9922
Current [μA]	0.2506	0.2462	0.9509

Table 6.5 Performance metrics for the XGBoost model across different target properties.

The results indicate consistently strong performance across all target properties. High R^2 values demonstrate that the model effectively captures the variability in the data, while the low MSE and MAE values indicate precise predictions with minimal error. The model performs particularly well on targets with a narrower numerical range, such as energy per atom, band gap, and Fermi level, achieving high accuracy. For targets with broader numerical ranges, the model shows its robustness, maintaining good predictive accuracy.

The fits for all target properties are illustrated in Fig. 6.13, whereas the distributions of the residual errors are presented in Appendix B.1, Fig. B.1. For the majority of targets, the data points in the fit plots closely follow the identity line, indicating strong predictive accuracy and minimal variance in the errors. The residual histograms along the axes are narrow and centered, further confirming the model’s consistency and reliability. In the case of the transport current, which exhibits a wider numerical range and greater intrinsic variability, the dispersion around the identity line is slightly larger.

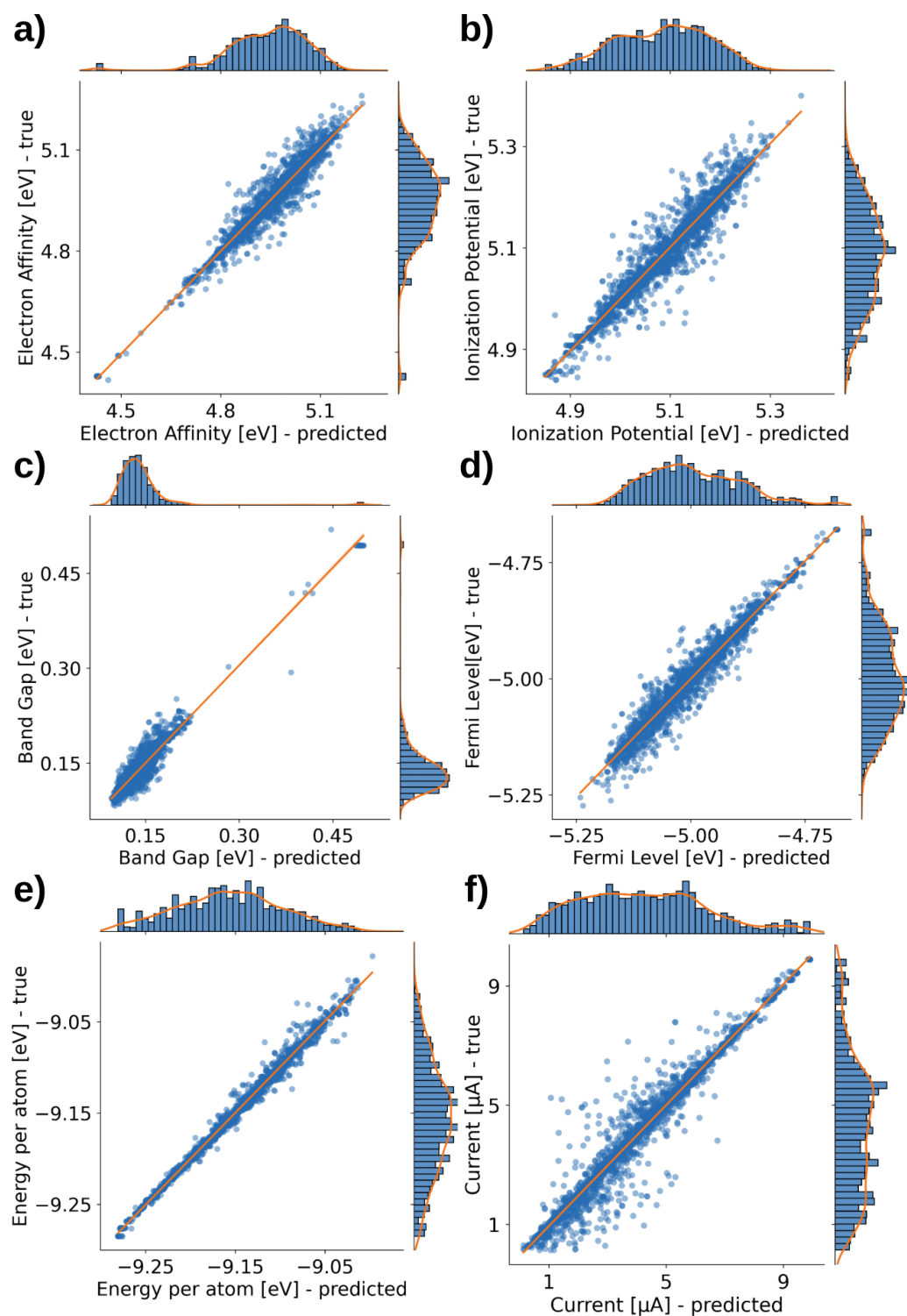


Fig. 6.13 Scatter plots show the correlation between the predicted and true values in the test set using XGBoost for electron affinity (a), ionization potential (b), band gap (c), Fermi level (d), energy per atom (e), and current (f). The lines (orange) represent the ideal $y = x$ correlation. Marginal histograms display the distribution of predicted and true values for each property, indicating the performance of the predictive model across different datasets.

Nevertheless, the general alignment remains clear, demonstrating that the model retains good predictive performance even for this more challenging target.

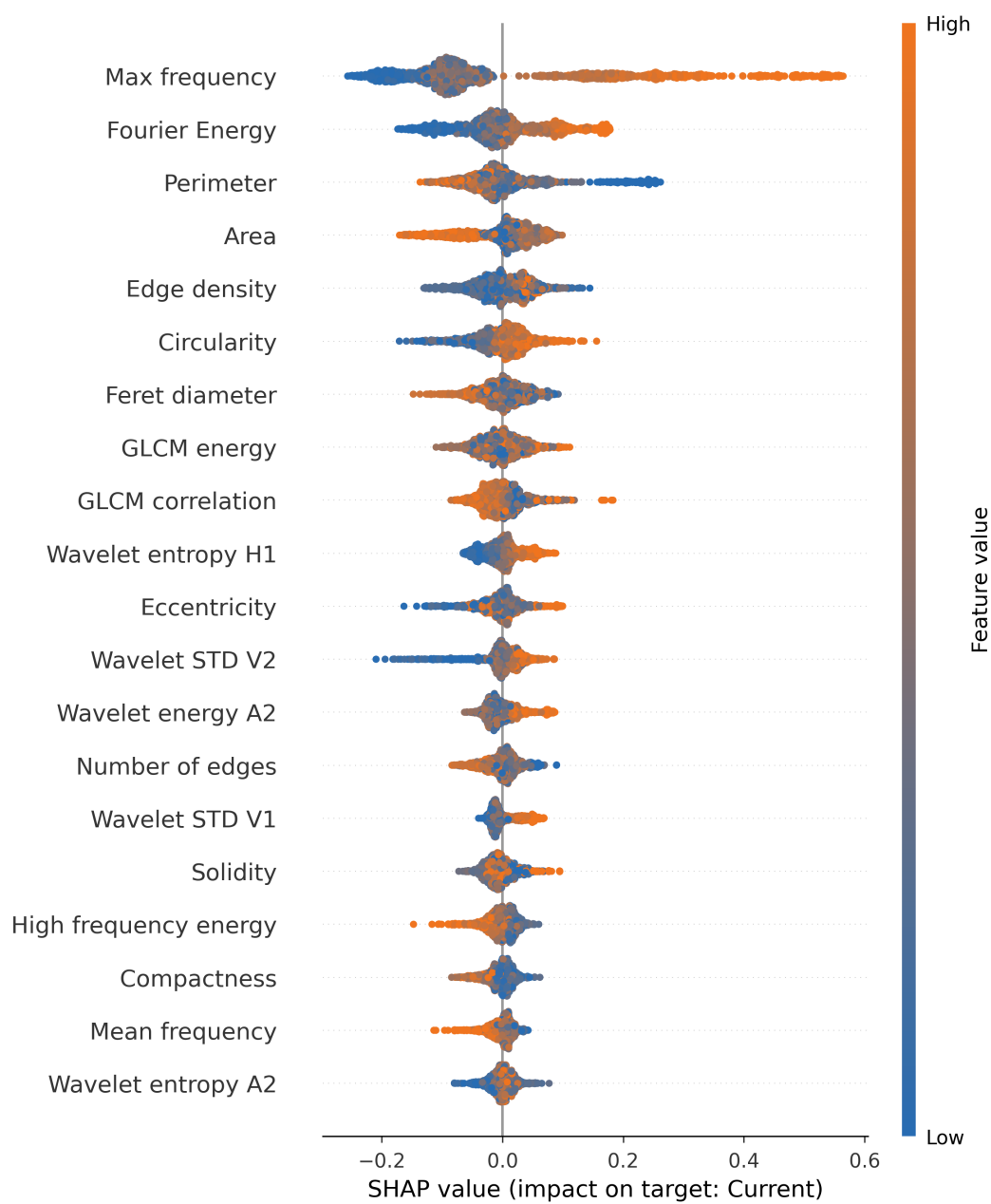


Fig. 6.14 SHAP value summary for the current target

Additionally, to complete our analysis, we trained a set of Explainable AI (XAI) models using SHAP values [74] as the reference framework. SHAP (SHapley Additive exPlanations) values provide a unified approach to interpret model predictions

by quantifying the individual contribution of each input feature to the final output. They are derived from cooperative game theory, where each feature is treated as a “player” contributing to the prediction, allowing for a consistent and mathematically grounded explanation of feature importance across all samples. The main objective of employing SHAP values in this work is to identify which features contribute most significantly to the prediction of the electronic and transport properties. As discussed in Section 6.4.4, we extracted a set of macroscopic descriptors directly related to the defects, along with frequency-domain features designed to capture the relative spatial position of the defect within the flake.

Key insights arise from this interpretability analysis (see Figure 6.14). The defect area exhibits a negative contribution to the predicted current, with larger defects corresponding to reduced electron mobility. This observation is consistent with the physical understanding that extensive defects disrupt the graphene lattice and hinder charge transport. In contrast, frequency-related features, such as maximum frequency and Fourier energy, show a positive correlation with current, indicating that higher-frequency and higher-energy spectral components enhance charge transport by promoting the formation of conductive edge states and percolation pathways.

Similarly, morphological descriptors such as the number of edges and the Feret diameter contribute negatively to the current prediction, suggesting that elongated or irregular defect geometries tend to obstruct carrier flow. Conversely, wavelet-based features, including wavelet energy (A2) and wavelet entropy (H1), exert a positive influence, possibly reflecting the role of localised vibrational modes induced by defects in facilitating electronic conduction.

Moreover, while the defect perimeter is associated with a negative SHAP contribution implying that larger perimeters increase scattering and reduce current, the circularity of the defect has a positive effect, confirming that more regular and compact defect shapes are favourable for charge transport.

Comprehensive SHAP analyses for the remaining target properties predicted by the XGBoost model are provided in Appendix B.1, Figures from B.2 to B.6.

6.4.7 Model Comparison with LightGBM

To validate the results and provide a quantitative comparison of the predictions, the final step involved benchmarking the performance of our model against another

widely used state-of-the-art algorithm, namely LightGBM [156]. The LightGBM model was provided with the same input parameters obtained through Optuna during the tuning of the XGBoost model, ensuring a fair comparison with all other settings held constant. As shown in Table 6.6, the performance scores are generally very close, and can be considered comparable. However, across most targets, XGBoost typically demonstrates a noticeably better ability to adapt to the test data. All performance metrics obtained for this model are reported in the Supporting Information, in Appendix B.1, Figure B.7, Figure B.8 and Figure B.9

Target	<i>MSE</i>	<i>MAE</i>	R^2
Fermi Level [eV]	0.0006	0.0157	0.9499
Ionization Potential [eV]	0.0008	0.0178	0.9185
Electron Affinity [eV]	0.0008	0.0185	0.9436
Band Gap [eV]	0.0001	0.0086	0.9363
Energy per atom [eV]	2.8e-05	0.0033	0.9920
Current [μA]	0.2573	0.2518	0.9496

Table 6.6 Performance metrics for selected targets using the LightGBM model.

A complementary and more in-depth analysis can be conducted by examining both SHAP values and feature importance. This is particularly evident when training the XAI model on the predictions from LightGBM, as illustrated in Appendix B.1, Figures B.10 and B.11.

LightGBM displays a wider distribution of feature importance across multiple predictors, whereas XGBoost tends to concentrate its influence on a smaller subset of highly significant features. Specifically, XGBoost assigns predominant importance to properties such as band gap and energy per atom, while also emphasising geometric descriptors, including defect area and perimeter. For targets such as electron affinity, Fermi level, ionization potential, and transport current, XGBoost places greater weight on frequency-derived features that capture global information from the entire flake image.

In contrast, LightGBM exhibits a more dispersed pattern of feature importance, which, although potentially offering a richer perspective on the feature space, can make direct physical interpretation more challenging. Despite these methodological differences, SHAP values for both models remain broadly consistent, as illustrated in Appendix B.1, Figures B.12–B.17, confirming that both approaches provide comparable levels of explainability.

Furthermore, examining the permutation feature importance [157] for the transport current provides additional insight into how each feature contributes to the predictive performance of the models. This method evaluates the decrease in model accuracy that occurs when the values of a specific feature are randomly shuffled, effectively breaking its relationship with the target variable. A larger drop in accuracy indicates that the feature plays a more critical role in the prediction process.

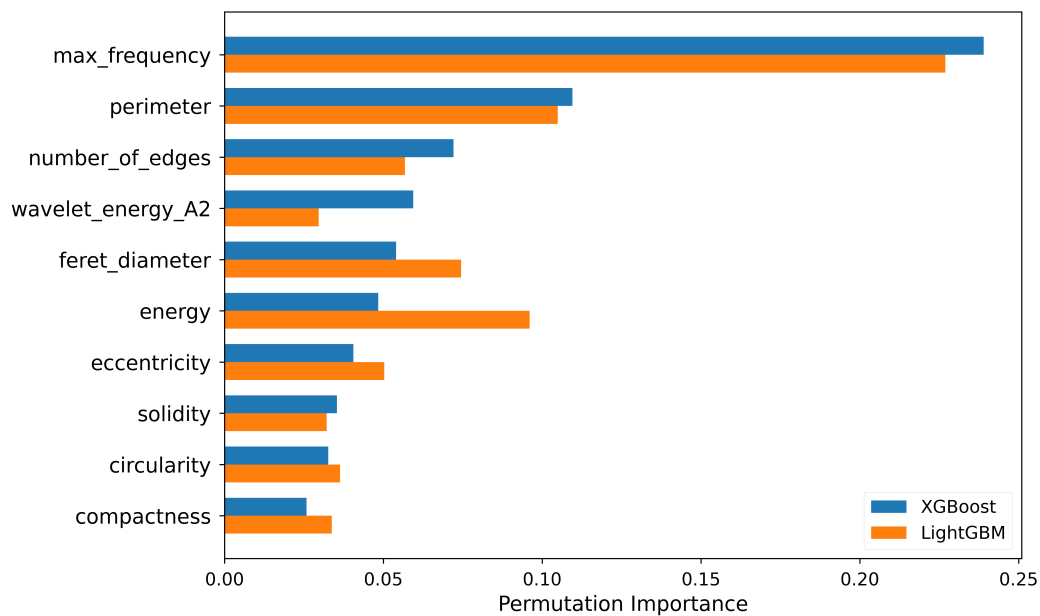


Fig. 6.15 Comparison of the top 10 features based on permutation importance. XGBoost results are shown in blue, while LightGBM results are shown in orange.

The bar chart in Figure 6.15 presents a comparison of the permutation feature importance for the top 10 features in predicting the transport current, as derived from XGBoost and LightGBM models. The values on the x-axis indicate the magnitude of the impact that each feature has on the model’s predictive performance. Notably, max frequency emerges as the most influential feature in both models,

demonstrating a dominant effect on current predictions. Other features such as perimeter, number of edges, and wavelet energy A2 show moderate importance, with some differences between XGBoost and LightGBM reflecting the distinct ways the models weight input variables. The comparison highlights that while both models identify largely overlapping key predictors, XGBoost tends to concentrate importance on fewer features, whereas LightGBM distributes importance more evenly across several inputs. This analysis provides insight into the structural and frequency-related characteristics of the data that most strongly influence transport current. A comprehensive, side-by-side evaluation of predictive performance and feature-level behaviour for XGBoost and LightGBM is reported in Appendix B.1, Table B.2.

6.5 From uncontrolled to conditional generation

6.5.1 Motivation

As discussed throughout this chapter, the optimal properties of graphene strongly depend on both the type of defect and its location within the flake. In this section, we explore the configuration space in which graphene exhibits electrical properties of interest, with the specific aim of guiding the generation of new graphene flakes exhibiting such properties. The workflow involves two Variational Autoencoders (VAEs). In the first VAE, defects are sampled starting from pristine graphene, with only noise provided as input to the latent space. This approach does not allow direct control over the defect sampling space but is particularly useful for dataset augmentation.

In the second step, we aim to condition the generation process. Specifically, the features extracted in Section 6.4.4 were projected into 2D, creating distinct feature spaces. Within each space, we computed the mean current, and then conditioned the latent space sampling on regions exhibiting the most desirable properties. This approach is novel and has not yet been fully explored, and it will be further discussed in future work.

By doing so, we pave the way for a conditional generation strategy that can be applied both for targeted dataset augmentation focusing on regions with data scarcity

and for generating new samples with specific characteristics of interest to materials science.

6.5.2 Workflow

The dataset used in this work is the same as that employed throughout this chapter. From this dataset, a total of 8,750 samples were randomly selected. In this phase, starting from the XYZ files, images were generated using an approach very similar to that described in Section 6.3.1. The main difference is that, in this case, bonds were not visualised; instead, the atomic structure was treated as a two-dimensional point cloud, representing only the spatial coordinates of the flake with white points (see Image 6.16).

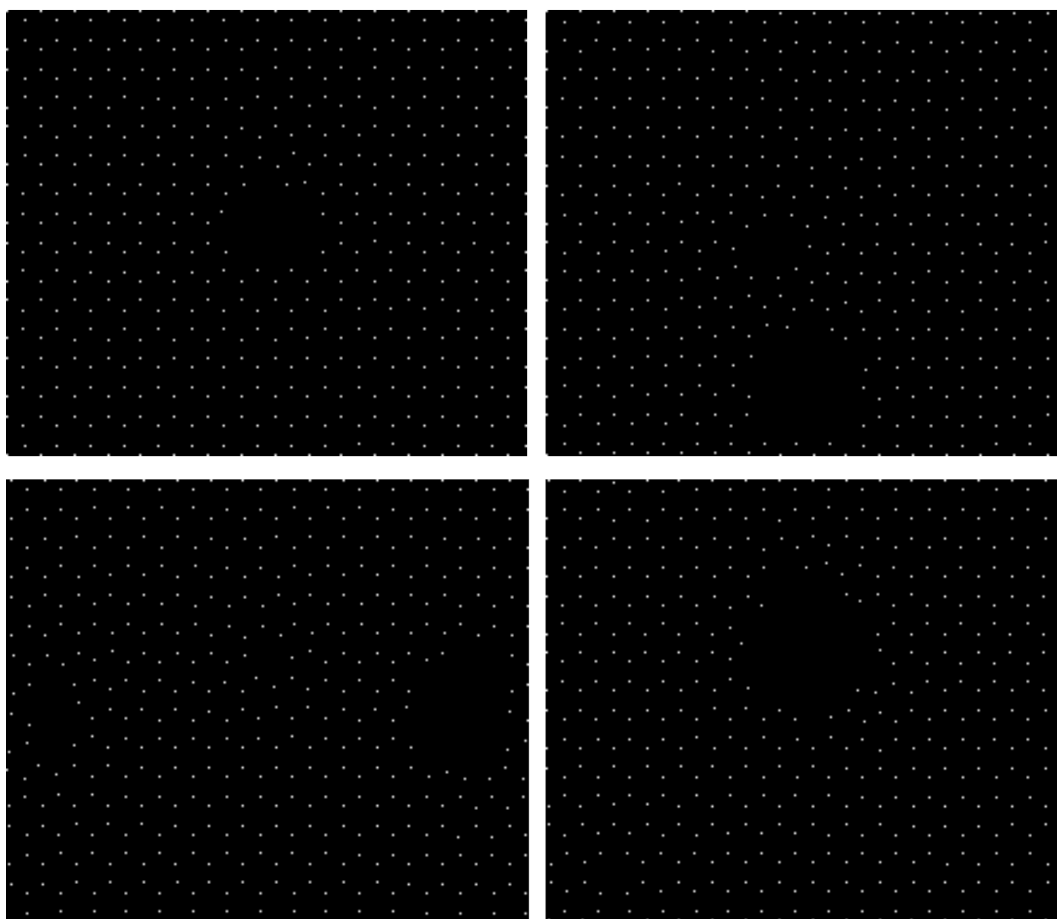


Fig. 6.16 Samples of graphene with defects image used in the training of VAEs

The decision to use atomic positions rather than the full image including bonds was based on the fact that, for an architecture such as a variational autoencoder, it is simpler to reconstruct the individual atomic positions in order to generate valid structures.

6.5.3 First Case: Model architecture and Training

In the initial, non-conditioned training approach, the architecture adopted was the Enhanced Encoder, originally proposed by Ditria et al. [158]. This model introduces a modular design inspired by Residual Networks (ResNets), where the inclusion of skip connections serves both to stabilise training and to enhance reconstruction accuracy. Such residual pathways allow the network to preserve spatial coherence and mitigate gradient degradation, ensuring that low-level structural information is not lost as the representation becomes progressively abstract. This property is particularly beneficial in physical systems where fine-scale spatial patterns exert a significant influence on global behaviour.

Structurally, the encoder begins with an initial convolutional layer that projects the input into a high-dimensional feature space, followed by a sequence of Residual Downsampling (ResDown) blocks that gradually reduce spatial resolution while increasing representational depth. Each ResDown block integrates convolutional transformations, normalisation, and residual connections, enabling efficient feature propagation across layers. The encoder concludes with two convolutional heads that parameterise the latent distribution, yielding the mean and variance required for stochastic sampling.

The decoder mirrors this organisation through a series of Residual Upsampling (ResUp) blocks, which progressively restore the spatial dimensions of the feature maps while decreasing their depth. The final convolutional layer reconstructs the output, preserving the spatial and statistical characteristics of the original input. This hierarchical structure facilitates multiscale representation learning, allowing the model to capture both local spatial details and global contextual dependencies. Overall, the Enhanced Encoder–Decoder configuration provides a robust and expressive backbone for unsupervised feature learning, well-suited to scenarios where spatial fidelity and information preservation are critical.

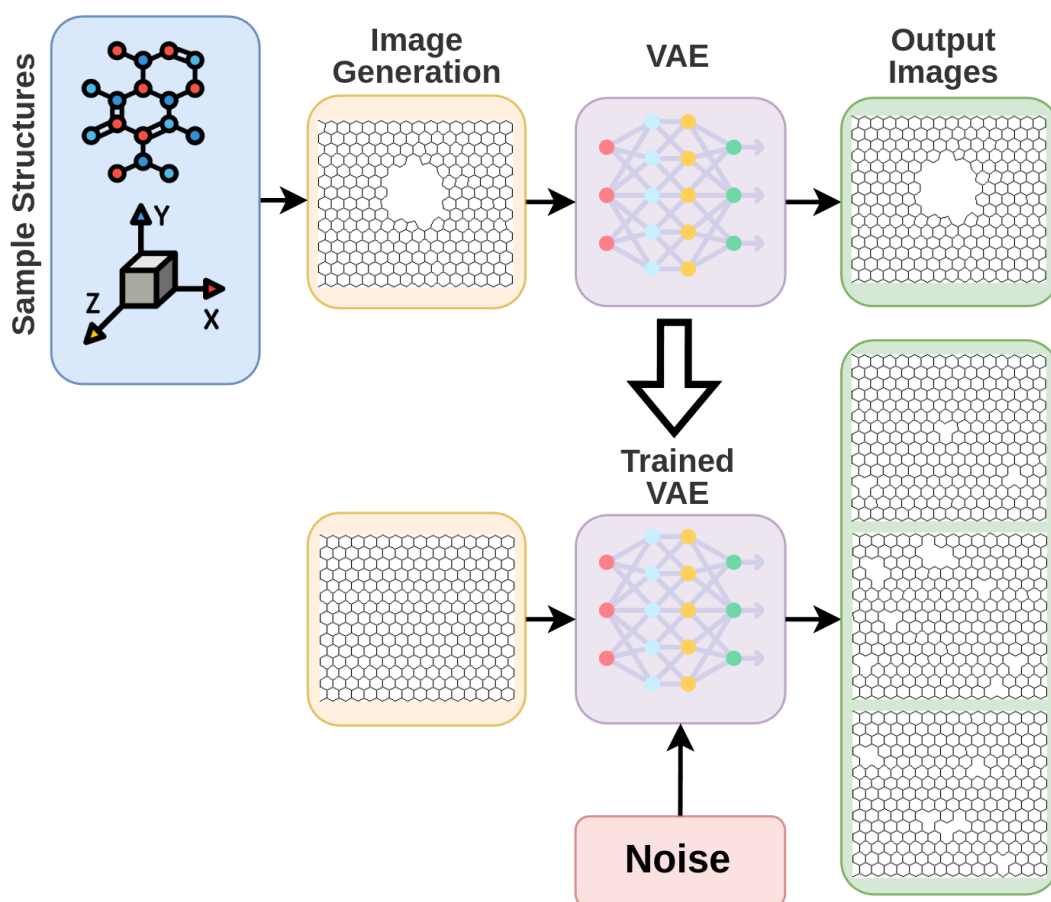


Fig. 6.17 Workflow for the generation of synthetic datasets containing graphene nanoflakes with small distributed defect

The loss function employed for training the *Enhanced Encoder–Decoder* architecture was designed as a composite, integrating multiple complementary terms to guide the network toward accurate and perceptually faithful reconstructions. The first component is the reconstruction loss, which itself combines two distinct contributions. The primary term is the *Structural Similarity Index Measure* (SSIM), originally introduced in [159], which evaluates similarity between the ground truth and reconstructed images at a structural level rather than through a purely pixel-wise comparison. SSIM effectively captures perceptual fidelity by emphasising the preservation of local luminance, contrast, and structural patterns. To complement this, a traditional pixel-wise L_1 loss was included to penalise absolute deviations at the pixel level. The combined reconstruction loss can therefore be expressed as:

$$L_{reconstruction} = 0.85 \times L_{SSIM} + 0.15 \times L_{L1} \quad (6.17)$$

The second term is the Kullback–Leibler (KL) divergence, denoted as L_{KL} , a standard component in variational autoencoder-based models with stochastic latent spaces [160, 161]. This term regularises the latent distribution, encouraging it to approximate a prior (typically a standard Gaussian), which in turn facilitates smooth and continuous latent representations suitable for generative sampling.

To further enforce the capture of global image characteristics, a Fourier-domain loss, $L_{Fourier}$, was introduced. This term computes the difference between the frequency spectra of the original and reconstructed images, obtained via Fourier transform. By doing so, $L_{Fourier}$ complements SSIM by explicitly constraining the network to preserve frequency-domain content, such as texture and periodic patterns, which might not be fully captured through spatial-domain measures alone.

Finally, a perceptual loss based on a pretrained deep network was incorporated to ensure that high-level semantic features were accurately reconstructed. In particular, the VGG19 network [162] was used to extract feature representations from both the original and reconstructed images. At each layer, the feature maps corresponding to the original and generated images were compared by computing the squared difference, which was then averaged across spatial dimensions. The mean feature loss from all layers was aggregated to form the perceptual loss, $L_{perceptual}$, and subsequently normalised by the total number of layers to ensure balanced contribution.

The total loss for the autoencoder is therefore a weighted combination of all these terms:

$$L_{total} = L_{reconstruction} + L_{KL} + L_{perceptual} + L_{Fourier} \quad (6.18)$$

This composite formulation ensures that the network simultaneously preserves fine-grained pixel information, structural coherence, high-level perceptual features, and global frequency characteristics, providing a robust training signal for learning expressive and physically meaningful latent representations.

The training procedure was conducted following standard practices and continued for a total of 800 epochs. The Adam optimizer was employed, while a ReduceLROn-Plateau scheduler dynamically adjusted the learning rate, reducing it by a factor of

0.1 if the validation loss did not improve for 50 consecutive epochs. In addition, an early stopping mechanism was implemented, which saved the model achieving the lowest validation loss and terminated training after 80 epochs without improvement.

For numerical evaluations, the dataset was split into 70% for training, 15% for validation, and 15% for testing. To facilitate effective learning of the VAE, the dataloader was configured to invert the binary masks of the images (corresponding to the samples illustrated in Fig. 6.16), ensuring that the images consistently presented a white background with the atoms depicted in black. The resulting training and validation loss curves (Fig. 6.18) indicate that the model generalises well to unseen data, with no substantial evidence of overfitting.

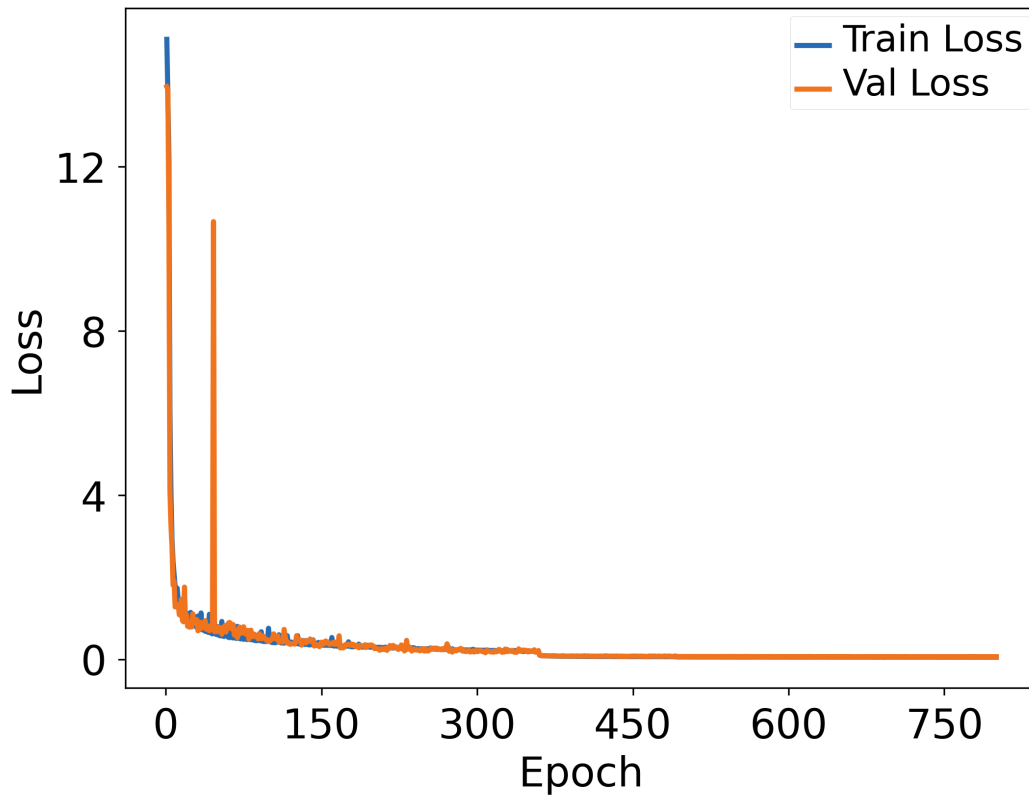


Fig. 6.18 Train and validation loss for the proposed architecture

Generation step During this generation phase, the encoder was retained rather than discarded. As illustrated in the lower part of Fig. 6.17, a perfect graphene image is fed into the encoding branch. To generate defective structures, the latent space is then perturbed, producing new atomic configurations with defects. This perturbation

is implemented by injecting scaled random noise during the reparameterization step, as expressed by:

$$z = z_{mean} + z_{\sigma} \cdot \varepsilon + 0.75 \cdot z_{noise}$$

$$\text{with } \begin{cases} z_{\sigma} = \exp(\log(z_{var}) \cdot 0.5) \\ \varepsilon \sim N(0, 1) \\ z_{noise} \sim N(0, 1) \end{cases}$$

The resulting latent vector z is then passed to the decoder of the VAE to generate the corresponding image.

Following the decoding, a post-processing step is applied to render the atomic bonds in the generated image. In brief, the algorithm resumed in Alg. 3 first identifies all atomic positions by locating black pixels. Candidate bonds are determined by computing the Euclidean distance between atoms and retaining only those within a specified minimum and maximum bond length, while resolving overlaps and eliminating invalid points. Lines are then drawn between valid atomic pairs to visualise the lattice, producing a fully connected graphene structure.

During the generation phase, there is a risk that some of the produced structures may be deemed invalid. To automatically filter out such cases, a supervised classification model was trained on a dataset of 2,000 manually labelled images, with labels indicating whether a structure is valid or not. This model enables the automatic rejection of images that, after the bond-drawing procedure, exhibit unrealistic chemical or physical configurations. Examples of both valid and invalid structures are shown in Figure 6.19. A ResNet18 [163] was adapted for this classification task by modifying the first convolutional layer to accept grayscale inputs and replacing the original multi-class output with a custom head: a fully connected layer, ReLU activation, dropout, and a final fully connected layer producing a single output. The model was trained for 300 epochs with a batch size of 32 and a learning rate of 0.001.

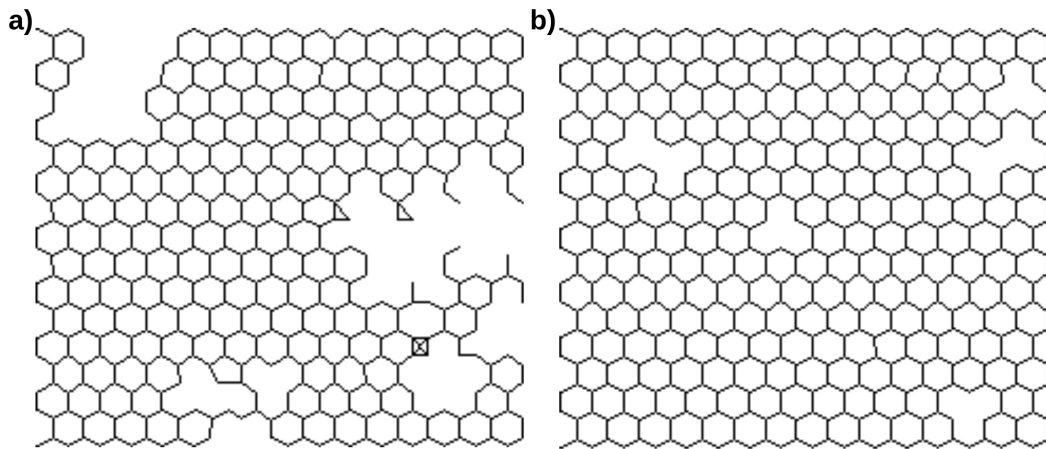


Fig. 6.19 Example of a) not valid structure and b) valid structure.

Algorithm 3 Draw Atomic Bonds from Generated Graphene Image

- 1: **Input:** Grayscale image containing atomic positions, *min_bond_length*, *max_bond_length*
 - 2: Detect all black pixels and store their coordinates as atomic positions
 - 3: Remove any invalid or overlapping positions based on local neighborhoods
 - 4: **for** each pair of atomic positions (i, j) **do**
 - 5: Compute Euclidean distance d between i and j
 - 6: **if** $min_bond_length \leq d \leq max_bond_length$ **then**
 - 7: Add (i, j) to the list of candidate bonds
 - 8: Update neighbor count for atoms i and j
 - 9: **end if**
 - 10: **end for**
 - 11: **for** each candidate bond (i, j) **do**
 - 12: **if** both i and j have at least 2 neighbors **then**
 - 13: Draw a line segment between i and j on the image
 - 14: **end if**
 - 15: **end for**
 - 16: Remove any remaining isolated black pixels
 - 17: **Output:** Image with atomic bonds visualized
-

The resulting confusion matrix (Fig. 6.20b) indicates high test accuracy, with few misclassifications. This demonstrates the model's effectiveness in distinguishing

valid from structurally broken DG nanoflakes, ensuring that the final dataset contains chemically and physically plausible structures.

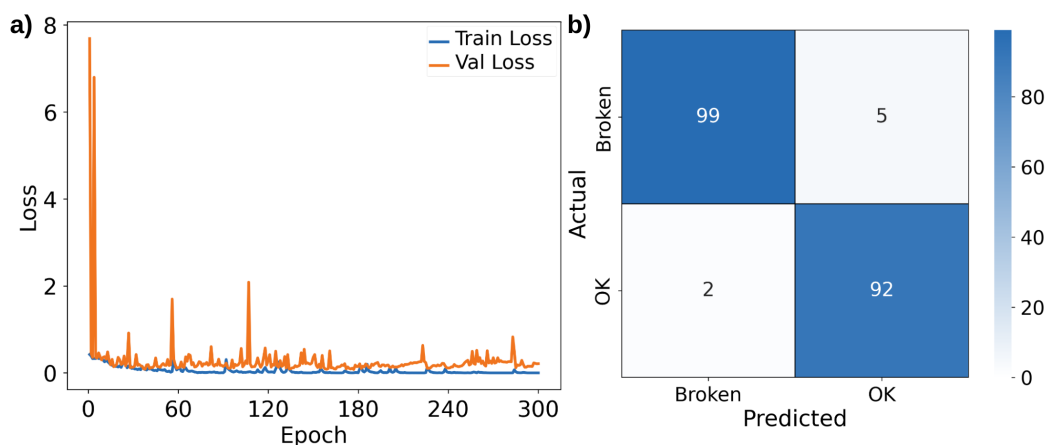


Fig. 6.20 a) Learning curves showing training and validation loss over epochs; b) Confusion matrix obtained from evaluating the trained binary classifier on the test set.

Generated dataset analysis

Building on the methodology described above, which leverages perturbations in the latent space for image generation, a synthetic dataset was created. As outlined throughout this thesis, this approach serves as a bridge between real and synthetic data, providing a framework for the subsequent study of the physical and chemical properties of the material. Specifically, the constructed dataset comprises 5,000 distinct graphene nanoflakes that passed the validation of the trained discriminator. Each flake was then processed following the same workflow described in Section 6.4. Initially, a comparison of the distributions of atom counts between the original and synthetic datasets was performed. The results are shown in Fig. 6.21a. It is evident that the synthetic dataset primarily occupies the right-hand portion of the plot. This occurs because the synthetic nanoflakes predominantly feature small, distributed defects, resulting in atom counts that are closer to those of defect-free flakes. In contrast, the original dataset exhibits a broader distribution, reflecting a wider variety of defect configurations and flake sizes.

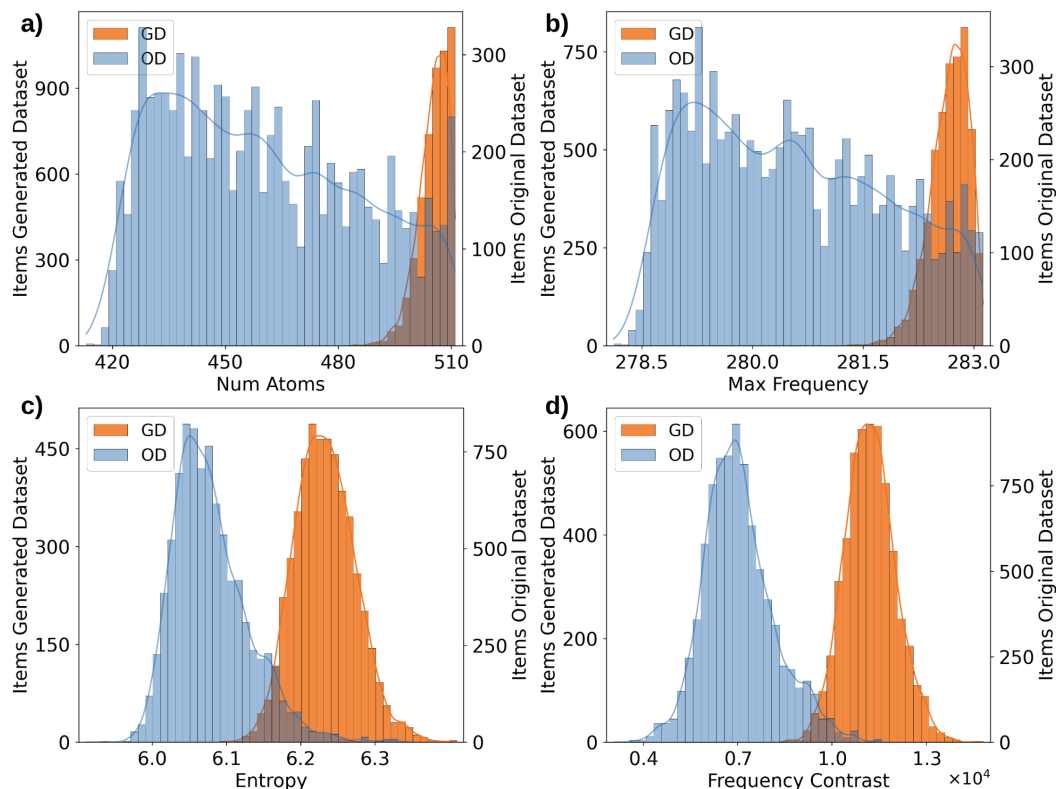


Fig. 6.21 Comparison between the original dataset (OD) and the synthetically generated dataset (GD). (a) Distribution of atom counts. Frequency-domain analysis of the nanoflakes, including (b) maximum frequency, (c) entropy, and (d) frequency contrast.

6.5.4 Second Case: Conditioning with specific features

The final stage of our trip in the atomic scale, which also represents the concluding phase of the doctoral project, constitutes a natural extension of the generative methodology presented in Section 6.5.3. This development serves as an intermediate approach, bridging established methods with potential future directions, having been iteratively tested during the preparation of this work. In particular, this approach aims to represent a step forward with respect to the generation of samples with defects, as discussed in 6.5.3. Specifically, the goal is to move beyond the random sampling performed by the autoencoder through a latent space conditioning strategy, which will be presented in Section 6.5.5.

Before detailing the architectural modifications and the conditioning strategy applied to the subsequent model, it is important to recall the underlying motivation,

partially introduced in Section 6.5.1. When dealing with materials such as graphene, it is essential to guide the generative process towards specific target regions of interest. This guidance ensures that the generated structures meet desired structural or functional properties.

In the present study, the conditioning was designed to generate graphene flakes with well-defined structural characteristics, particularly focusing on properties such as transport current. The current is of significant industrial relevance, as it informs potential applications of graphene within finished products. This aspect also serves as a unifying theme across the work, connecting atomistic-level modelling with multiscale design: by understanding and controlling properties at the atomic scale, it becomes possible to engineer macroscopic devices whose performance reflects the complete modelling pipeline.

6.5.5 The Conditioning Workflow

In this section, we discuss the conditional case and present preliminary results obtained from introducing conditioning during the generative phase. When referring to a conditional model, we imply that the generative process is guided towards a specific region of the latent space according to a chosen set of conditions. In practice, this conditioning mechanism is applied both during training and inference, enabling the model to learn structured relationships between input features and the generated outputs.

In our specific implementation, the conditional training workflow was built upon the same dataset of atomic configurations in xyz format, previously introduced in Section 6.3.6. Following the data processing pipeline described in Section 6.4.4, we extracted a set of descriptors characterising each flake. As discussed earlier, this feature space is inherently high-dimensional ($\mathbb{R}^{44}$). To facilitate conditioning and improve interpretability, dimensionality reduction techniques such as UMAP and PCA [52] were employed to project the original feature space onto a two-dimensional manifold.

As illustrated in Figure 6.22, the two-dimensional PCA representation is used as the conditioning space for the generative model. A key question therefore concerns how to identify the regions of the feature space in which new samples should be generated.

To address this point, Figure 6.22 highlights three distinct regions emerging from the distribution of the training data. These regions arise from feature vectors computed from the original dataset and subsequently grouped through a clustering procedure. For each cluster, the mean value of the target quantity of interest, namely the current, is evaluated, providing a physically meaningful partition of the feature space.

The background density, visualised through kernel density estimation (KDE), represents the distribution of training samples in the reduced space and thus indicates which regions are most densely populated by real data. Cluster boundaries are approximated via convex hulls, highlighting the geometric extent of each group in the PCA plane.

The outermost cluster corresponds to the region characterised by the highest mean current. In particular, the green–cyan cluster represents samples with an average current of approximately 6.5, the blue cluster corresponds to an intermediate regime with an average current of approximately 3.5, and the violet cluster represents the lowest-current regime with an average current of approximately 2.6. The KDE contours further indicate that the high-current cluster occupies a region of comparatively lower training density, reflecting the limited availability of high-performance samples in the original dataset.

The coloured markers denote synthetically generated samples, coloured according to their predicted current values. Generation is explicitly conditioned on the high-current cluster only, corresponding to the outer green–cyan region. Within this region, predicted current values tend to be higher on the right-hand side of the PCA space and gradually decrease towards the left. Importantly, the generated samples remain consistent with the statistical and geometric properties of their parent cluster, indicating that the conditional generative process explores high-performance regions of the feature space without violating the learned data distribution.

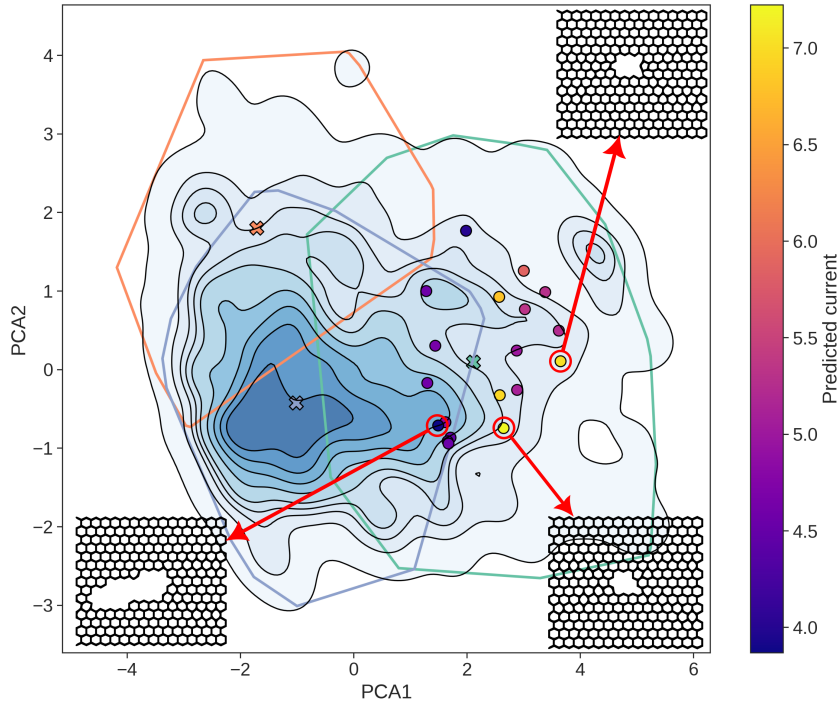


Fig. 6.22 Clusters of the training samples used in Section 6.4. Features are projected into a two-dimensional PCA space, with kernel density estimation (KDE) indicating the density of training samples. Cluster boundaries are approximated via convex hulls. Coloured points denote synthetically generated samples, conditioned on the high-current cluster and coloured according to their predicted current values.

The visible regions are the result of applying a Convex Hull algorithm separately to each cluster. Having determined the average current associated with each cluster, it was therefore possible to identify which regions were more promising for the generation of new samples. In particular, the green region, located on the far right, corresponds to the cluster exhibiting the highest average current, while overlapping areas between the red and green regions, as well as between the green and blue ones, appear to be less promising for sample generation, given the relatively low mean current values characterising those overlapping clusters. Therefore, we define $\mathcal{R}_{\text{promising}}$ as the region of the projected feature space in which it is promising to generate new samples, as follows:

$$\mathcal{R}_{\text{promising}} = \{(PCA1, PCA2) \in \mathbb{R}^2 \mid 2 \leq PCA1 \leq 5 \wedge -2 \leq PCA2 \leq 2.8\}$$

Diffusion Model The generative model employed in this work is a discrete Bernoulli diffusion process [164] conditioned on a low-dimensional feature vector extracted from the material representation (in this case, the two-dimensional PCA embedding). The diffusion dynamics operate on binary lattice images $\mathbf{x}_0 \in \{0, 1\}^{H \times W}$ representing atomistic structures. The model aims to learn the reverse-time generative distribution $p_\theta(\mathbf{x}_0 \mid \mathbf{x}_t, \mathbf{c}, t)$, where $\mathbf{c} \in \mathbb{R}^2$ denotes the conditioning features and $t \in \{0, \dots, T-1\}$ is the discrete diffusion timestep. The forward diffusion follows a strictly discrete mechanism: at each timestep t , every pixel is either kept from the original configuration or replaced by unbiased Bernoulli noise. This behaviour is governed by a decaying probability $\bar{\alpha}_t = \prod_{k=1}^t (1 - \beta_k)$, where the β_k follow a linear schedule. For each pixel (i, j) , the noising rule can be expressed as

$$x_t^{(i,j)} = m_t^{(i,j)} x_0^{(i,j)} + (1 - m_t^{(i,j)}) u^{(i,j)}, \quad (6.19)$$

where $m_t^{(i,j)} \sim \text{Bernoulli}(\bar{\alpha}_t)$ determines whether the original signal is preserved and $u^{(i,j)} \sim \text{Bernoulli}(1/2)$ represents pure random noise. This discrete formulation differs from conventional Gaussian diffusion and is appropriate for binary atomistic grids.

The conditioning mechanism plays a central role, as the generative trajectory must depend on the position of a sample in the feature space. The raw conditioning vector is first normalised,

$$\tilde{\mathbf{c}} = (\mathbf{c} - \mu) \oslash \sigma, \quad (6.20)$$

and then projected into a higher-dimensional embedding through an MLP,

$$\mathbf{e}_c = f_{\text{cond}}(\tilde{\mathbf{c}}) \in \mathbb{R}^{d_c}, \quad (6.21)$$

ensuring that the model can represent non-linear dependencies on the conditioning. In parallel, the diffusion timestep t is encoded using a sinusoidal embedding with log-spaced frequencies,

$$\mathbf{e}_t = (\sin(\omega_1 t), \dots, \sin(\omega_d t), \cos(\omega_1 t), \dots, \cos(\omega_d t)), \quad (6.22)$$

so that the network has a continuous representation of discrete diffusion steps. The two embeddings are concatenated to form a single representation

$$\mathbf{z} = [\mathbf{e}_c \parallel \mathbf{e}_t], \quad (6.23)$$

used to modulate all blocks of the UNet [165] via FiLM [166] transformations.

The UNet backbone incorporates this vector $\mathbf{z}$ at every resolution. Each convolutional block predicts, through a small linear layer, a pair of channel-wise affine modulation coefficients (γ, β) ,

$$[\gamma, \beta] = W_{\text{film}} \mathbf{z}, \quad (6.24)$$

which transform the intermediate activations $\mathbf{h}$ as

$$\text{FiLM}(\mathbf{h} \mid \mathbf{z}) = \gamma \odot \mathbf{h} + \beta \quad (6.25)$$

This mechanism allows the conditioning information to influence both coarse and fine spatial representations, making the generative process explicitly dependent on the region of the UMAP space in which the sample lies.

During training, the model receives a noised input $\mathbf{x}_t$ and learns to predict logits for the clean configuration $\mathbf{x}_0$ through binary cross-entropy. Since the target structures are expected to exhibit local atomic coordination, the loss is augmented with a neighbour-consistency regulariser. Using the predicted Bernoulli probabilities $\mathbf{p} = \sigma(\ell_\theta)$, a four-neighbour convolution estimates the expected coordination number at each pixel, and deviations from the graphene-like value of 3 are penalised:

$$\mathcal{L}_{\text{coord}} = \mathbb{E} [|\text{coord}(\mathbf{p}) - 3| \odot \mathbf{p}] \quad (6.26)$$

The total loss becomes

$$\mathcal{L} = \mathcal{L}_{\text{BCE}}(\ell_\theta, \mathbf{x}_0) + \lambda_{\text{coord}} \mathcal{L}_{\text{coord}} \quad (6.27)$$

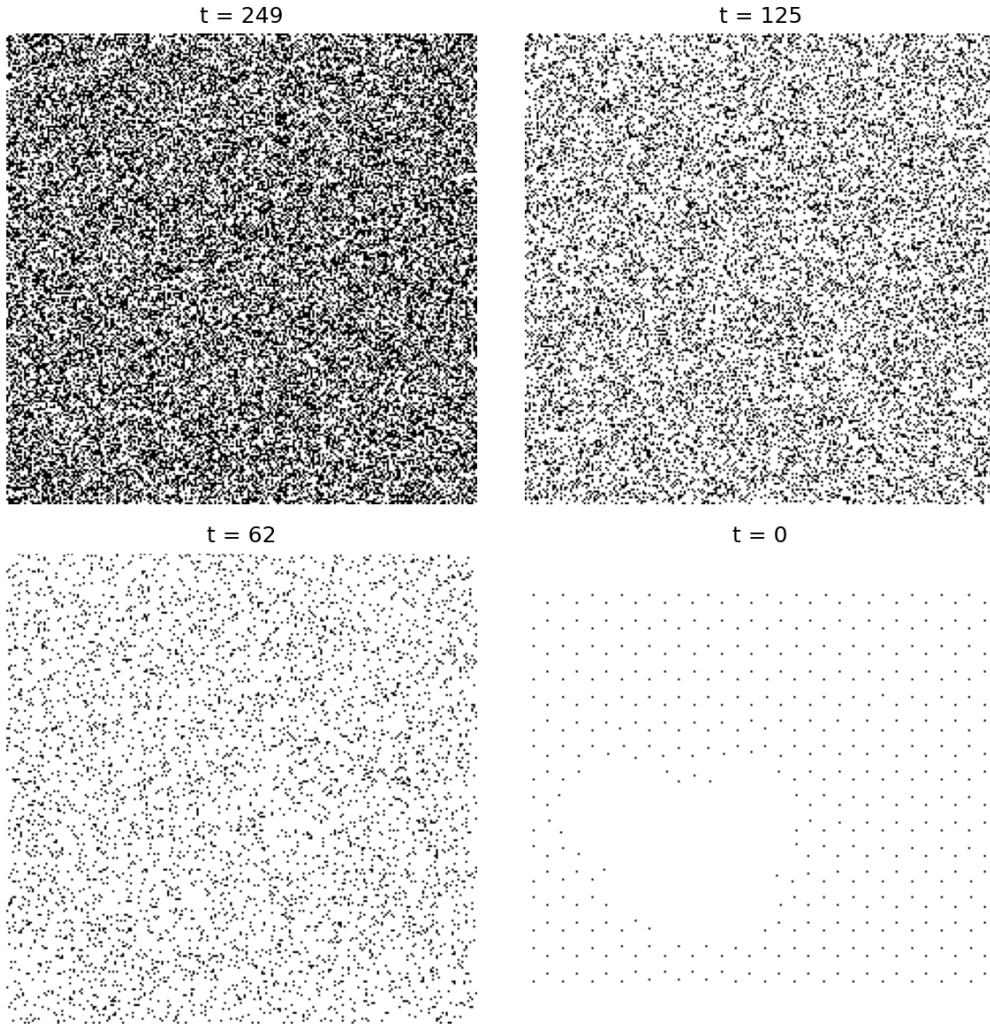


Fig. 6.23 Different diffusion steps of the conditional generation process, illustrating the reverse trajectory from the initial noise state to the final synthesised sample.

Generation proceeds by reversing the discrete chain. At each step the model predicts a distribution for $\mathbf{x}_0$, samples a candidate reconstruction, and then re-applies one step of the forward noising rule using $\bar{\alpha}_{t-1}$. This iterative mechanism eventually reaches $t = 0$, where the estimate of the clean structure is returned as the final sample. The combination of Bernoulli diffusion, FiLM-based conditioning and structural regularisation yields a controllable generative process able to produce atomistic configurations that vary smoothly with their position in the learned feature space.

The reverse sampling procedure follows a fully discrete Markov chain that reconstructs the clean configuration by progressively removing noise. Starting from

pure Bernoulli noise $\mathbf{x}_T \sim \text{Bernoulli}(1/2)$, the model iteratively estimates the clean image and reintroduces only the amount of noise consistent with an earlier diffusion timestep. At a generic step t , the UNet receives the current noisy configuration $\mathbf{x}_t$, together with the conditioning vector $\mathbf{z}$ and the timestep embedding, and outputs logits $\ell_\theta(\mathbf{x}_t, \mathbf{z}, t)$ that parametrise the Bernoulli distribution of the clean structure. The model therefore defines

$$p_\theta(x_0^{(i,j)} = 1 \mid \mathbf{x}_t, \mathbf{z}, t) = \sigma(\ell_\theta^{(i,j)}),$$

and a candidate reconstruction $\hat{\mathbf{x}}_0$ is sampled pixelwise from this distribution. All the steps described in this section are summarised schematically in Fig. 6.24.

From Figure 6.23 we observe the full sequence of the generation process, starting from the initial timestep, where the input is dominated by noise, and progressing through progressively cleaner intermediate states as the diffusion steps decrease. At the final timestep, the network produces the defected graphene sample conditioned on the PCA feature coordinates. The process of drawing bond and selection of valid samples generated is the same presented in the section 6.5.3

6.6 Conclusion

This chapter has presented the complete atomistic-scale methodology developed in this work, with a particular focus on graphene as a prototypical two-dimensional material. The workflow has been designed to convert unstructured atomic information into structured, analysable data and, ultimately, into predictive and generative models capable of informing materials design.

The first part of the chapter established the foundations of this pipeline: automated preprocessing, consistent data structuring, and the integration of electronic, transport, and synthetic STM simulations within a FAIR-compliant workflow. This step is central to the entire chapter, as it formalises the transition from raw atomistic configurations to a reproducible dataset suitable for downstream analysis.

The second part examined how macroscopic defect patterns and their spatial arrangement within the flake influence electronic and transport behaviour. By extracting geometric and frequency-domain features from STM-like representations

and employing gradient-boosting models, the chapter demonstrated that predictive correlations can be established between defect morphology and key physical observables. This provides an interpretable route to understanding how deviations from ideal crystallinity shape graphene's functional properties.

The final section explored generative modelling. Two strategies were introduced. The first relied on an unconditional autoencoder to sample new defected configurations without explicit control. The second introduced conditioning, enabling the model to generate structures within targeted regions of a reduced-dimensional feature space. This conditioning mechanism was shown to align sample generation with user-defined objectives, such as enhanced transport current or favourable energetic stability. It illustrates that generative models, when coupled with feature-space exploration, can be directed towards meaningful and physically relevant parts of the configuration space.

Across these components, the chapter emphasises two broader points. The first is methodological: effective materials-informatics pipelines require robust procedures for data generation, integration, and transformation. Without this foundation, predictive and generative phases become fragile, irreproducible, or difficult to scale. The second is scientific: bridging defect-level topology with measurable properties expands the design space beyond ideal structures, bringing modelling closer to experimentally realisable conditions.

Finally, the generative results highlight both opportunities and current limitations. While the models can target desired regions of the feature manifold, future work must address the physical validity of generated structures. In particular, assessing and enforcing energetic stability remains an essential step for turning generative exploration into actionable materials design. Extending conditioning mechanisms and incorporating physics-guided constraints represent natural evolutions of this line of research.

Taken together, the chapter shows how atomistic simulations, predictive modelling, and controlled generative mechanisms can be integrated into a coherent and automated workflow. This workflow not only supports the analysis of graphene at the nanoscale but also contributes to the broader multiscale narrative of the thesis, demonstrating how structured data and AI-driven tools can connect atomic-level insights with higher-level design objectives.

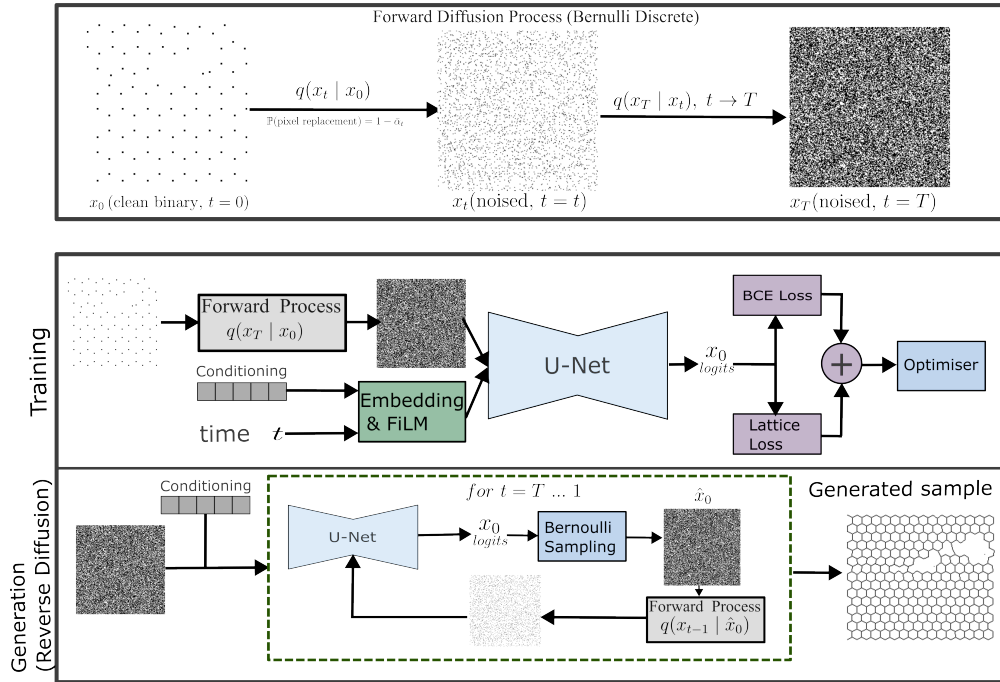


Fig. 6.24 Discrete Bernoulli diffusion framework for conditional binary structure generation. The top panel illustrates the forward diffusion process, in which a clean binary sample x_0 is progressively corrupted through discrete Bernoulli pixel replacement, producing increasingly noisy states x_t until pure Bernoulli noise x_T is reached. The central panel depicts the training phase: given a noisy sample x_t , a conditioning vector, and the corresponding timestep embedding, a FiLM-conditioned U-Net predicts the logits of the clean binary image x_0 . The model is optimised using a binary cross-entropy reconstruction loss combined with a lattice regularisation term enforcing local structural consistency. The bottom panel shows the reverse (generative) process. Starting from random Bernoulli noise x_T , the U-Net iteratively predicts logits for the clean image, which are explicitly sampled via a Bernoulli distribution to obtain a candidate clean estimate $\hat{x}_0$. This estimate is then stochastically re-corrupted according to the forward diffusion rule to produce x_{t-1} , ensuring consistency with the discrete diffusion dynamics. The procedure is repeated for $t = T \rightarrow 1$, yielding the final generated binary sample at $t = 0$.

Chapter 7

Conclusions and Perspectives

7.1 Contributions, limitations and scope

As outlined in the introductory chapter (see Section 1.3), this thesis evolved from an initially application-oriented framing towards a broader methodological investigation of automation and data-driven integration for complex physical systems. While the original motivation was grounded in applied contexts such as transportation, logistics and smart urban infrastructures, the research progressively shifted towards a more fundamental question: how complex, heterogeneous systems characterised by multiple interacting scales can be modelled, explored and reasoned about in a principled and reproducible manner.

The case studies developed throughout this work, although rooted in materials science and device-level modelling, were therefore not selected for their domain specificity, but for their ability to expose structural challenges that are common to a wide class of multiscale problems. Atomistic graphene simulations and mesoscale organic photovoltaic devices represent two conceptually distinct physical domains, governed by different physical laws and numerical solvers. Yet, both exhibit high-dimensional configuration spaces, expensive simulation pipelines and strong sensitivity to structural variability. In this sense, they serve as controlled but representative testbeds for addressing the research questions posed in Chapter 1.

The first research question concerned the representation of heterogeneous physical simulations within a unified, data-driven framework. The results presented in this thesis demonstrate that such unification does not require explicit hierarchical

coupling between physical models operating at different scales. Instead, a data-centred perspective allows information to be transferred across domains through shared representations, automated workflows and consistent data semantics. By treating data as the primary coupling layer, the framework enables interoperability across scales while preserving the specificity and integrity of the underlying physical models. This represents a departure from classical multiscale paradigms based on direct parameter passing, and instead emphasises methodological coherence over physical continuity.

A second central contribution of this work relates to the ability of machine learning models to capture the underlying physical operators governing complex material and device behaviour. In particular, the use of surrogate models and neural operators demonstrates that learning-based approaches can approximate high-dimensional physical mappings with significant accuracy when the learning task is formulated at the operator level. Rather than replacing physical modelling, machine learning reshapes how physical knowledge is abstracted, reused and queried. Expensive numerical solvers are not discarded, but encapsulated into reusable computational objects that enable rapid exploration of parameter spaces and functional responses that would otherwise be inaccessible.

The third research question addressed the role of automation and FAIR-compliant data infrastructures in enabling reproducible and scalable multiscale modelling. Throughout this thesis, automation is treated not as an auxiliary technical layer, but as a foundational methodological requirement. Automated pipelines for simulation, data generation and storage proved essential for ensuring consistency, traceability and systematic coverage of large configuration spaces. FAIR principles provided a concrete operational framework for maintaining transparency, interoperability and long-term usability of the datasets produced. Without such infrastructure, the integration of data-driven models across scales would remain fragile, difficult to validate and hard to extend.

At the same time, the thesis explicitly confronts the limitations of data-driven approaches, particularly with respect to physical interpretability. The analyses conducted show that interpretability is not an intrinsic property of a given model architecture, but an emergent property of the entire modelling pipeline, including data representation, feature construction and validation strategy. While explainable AI techniques can partially mitigate the opacity of highly expressive models, the

results clearly indicate that no post-hoc analysis can compensate for insufficient physical resolution or poorly structured data. This observation delineates an important epistemic boundary: data-driven models remain fundamentally constrained by the informational content of the data on which they are trained.

Within this broader methodological context, generative models assume a specific and non-trivial role. They are not introduced as tools for unconstrained synthesis or as substitutes for physical understanding, but as mechanisms for structured exploration of high-dimensional design spaces. In both the atomistic graphene and mesoscale photovoltaic case studies, generative models are employed to navigate configuration spaces that are prohibitively large for exhaustive simulation. By operating on reduced, physically informed representations and by conditioning on target properties, these models enable the identification and population of regions of interest within the design space rather than the optimisation of a single solution.

This generative capability represents a qualitative shift with respect to traditional optimisation-based workflows. Rather than iteratively refining a limited set of candidate designs, the methodology supports guided exploration, where families of structures or morphologies consistent with desired functional regimes can be proposed and analysed. Crucially, the thesis emphasises that such exploration remains meaningful only when generative outputs are continuously anchored to physical descriptors, surrogate evaluation and post-hoc verification. From the perspective of Key Enabling Technologies, generative models therefore act as exploration and design accelerators, extending the reach of simulation-driven research without severing the link to physical validity.

The scope of this thesis is intentionally methodological. It does not aim to deliver deployable technologies for smart-city applications, nor to optimise specific urban services. Instead, it provides a framework for organising complexity across scales, with explicit attention to reproducibility, traceability and scalability. The main limitations arise from the dependence on simulation fidelity, the approximations introduced by learning-based abstractions and the need for careful validation when extending these methods beyond their training regimes.

7.2 Future research directions

The results presented in this thesis naturally point towards future research directions that prioritise methodological integration over incremental increases in model sophistication. A first area of development concerns uncertainty quantification. As data-driven surrogates and generative models are increasingly used to inform decision-making, systematic methods for propagating and communicating uncertainty across scales become essential. This includes both epistemic uncertainty arising from limited data coverage and aleatoric uncertainty associated with intrinsic variability in physical systems.

A second important direction involves tighter coupling between data-driven models and experimental or operational data. While this thesis focuses primarily on simulation-based workflows, extending the framework to incorporate experimental measurements would significantly enhance its relevance and robustness. Such integration would require new strategies for data alignment, noise modelling and validation across heterogeneous sources.

From a broader Key Enabling Technologies perspective, future work should focus on the development of modular and interoperable digital ecosystems, in which automated data generation, learning-based abstraction and domain-specific solvers coexist within governed and auditable workflows. For smart-city contexts, this suggests a shift away from monolithic digital twins towards composable modelling infrastructures capable of supporting long-term planning, technology assessment and resilience analysis under evolving constraints.

7.2.1 Generative and integrative perspectives across scales

Beyond the specific case studies presented in this thesis, the results support a broader generative and integrative perspective on multiscale modelling and design. In particular, the combination of surrogate models (neural operators for device simulation, and XGBoost for the atomistic) and conditional generative techniques enables the construction of exploratory workflows that connect macroscopic functional requirements with device-level behaviour and atomistic structure. Within such a framework, targets originating from higher-scale considerations, such as system-level performance constraints, sensing requirements or smart-city objectives, can be propagated

downward through data-driven representations rather than through explicit physical coupling.

This process is intrinsically integrative along two complementary dimensions. Vertically, it enables information flow across scales, linking macroscopic objectives to mesoscale device properties and, ultimately, to atomistic configurations consistent with the desired functional regime. Horizontally, it integrates heterogeneous data-driven methodologies, combining high-fidelity simulations, surrogate models, explainable predictors and generative architectures within a single computational workflow. Importantly, this integration does not aim to collapse physical complexity into a single model, but to orchestrate multiple levels of abstraction in a coherent and transparent manner.

It is essential to emphasise that the approach presented here should be regarded as a proof of concept rather than a fully generalised solution. The generative capabilities demonstrated in this work are fundamentally constrained by the resolution, diversity and physical consistency of the available datasets, as well as by the assumptions embedded in the surrogate models. Nevertheless, the results indicate that meaningful structure–property relationships can be learned and exploited in a conditional setting, providing a principled foundation for more advanced multiscale generative strategies.

Taken together, this thesis positions data-driven multiscale modelling as a foundational methodology rather than a collection of isolated tools. When applied with critical awareness, it enables physically informed exploration of complex systems and provides a robust interface between atomistic materials, device-level behaviour and system-scale requirements. Artificial intelligence, within this framework, does not act as an oracle for inverse design, but as an enabling layer for guided exploration and informed decision-making, grounded in the structure, semantics and limitations of the underlying data.

References

- [1] J Tinsley Oden, Serge Prudhomme, Albert Romkes, and Paul T Bauman. Multiscale modeling of physical phenomena: Adaptive control of models. *SIAM Journal on Scientific Computing*, 28(6):2359–2389, 2006.
- [2] G. Fitzgerald, J. DeJoannis, and M. Meunier. Multiscale modeling of nanomaterials: recent developments and future prospects. *Modeling, Characterization and Production of Nanomaterials: Electronics, Photonics and Energy Applications*, pages 3–53, 1 2015.
- [3] Theodoros E Karakasidis and CA Charitidis. Multiscale modeling in nanomaterials science. *Materials Science and Engineering: C*, 27(5-8):1082–1089, 2007.
- [4] Michael J Willatt, Félix Musil, and Michele Ceriotti. Atom-density representations for machine learning. *The Journal of chemical physics*, 150(15), 2019.
- [5] Dana Bishara, Yuxi Xie, Wing Kam Liu, and Shaofan Li. A state-of-the-art review on machine learning-based multiscale modeling, simulation, homogenization and design of materials. *Archives of computational methods in engineering*, 30(1):191–222, 2023.
- [6] Hubert Rauscher, Gert Roebben, Agnieszka Mech, N Gibson, Vikram Kestens, TPJ Linsinger, and J Riego Sintes. An overview of concepts and terms used in the european commission’s definition of nanomaterial. *Publications Office of the European Union*, pages 1–44, 2019.
- [7] Jaishree Damodharan. Nanomaterials in medicine—an overview. *Materials Today: Proceedings*, 37:383–385, 2021.
- [8] Yury Gogotsi and Reginald M Penner. Energy storage in nanomaterials—capacitive, pseudocapacitive, or battery-like?, 2018.
- [9] Akanksha Gupta, Vinod Kumar, Sirajuddin Ahmed, and Siddhartha Gautam. Impact of nanotechnology in the development of smart cities. In *Smart Cities—Opportunities and Challenges: Select Proceedings of ICSC 2019*, pages 845–857. Springer, 2020.

- [10] Juh Tzeng Lue. Physical properties of nanomaterials. *Encyclopedia of nanoscience and nanotechnology*, 10(1):1–46, 2007.
- [11] JI Paredes, Silvia Villar-Rodil, Pablo Solis-Fernandez, Amelia Martinez-Alonso, and JMD Tascon. Atomic force and scanning tunneling microscopy imaging of graphene nanosheets derived from graphite oxide. *Langmuir*, 25(10):5957–5968, 2009.
- [12] Nadeem Baig, Irshad Kammakakam, and Wail Falath. Nanomaterials: A review of synthesis methods, properties, recent progress, and challenges. *Materials advances*, 2(6):1821–1871, 2021.
- [13] Nuha Al-Harbi and Nabil K Abd-Elrahman. Physical methods for preparation of nanomaterials, their characterization and applications: a review. *Journal of Umm Al-Qura University for Applied Sciences*, 11(2):356–377, 2025.
- [14] Lili Liu, Miaoqing Qing, Yibo Wang, and Shimou Chen. Defects in graphene: generation, healing, and their effects on the properties of graphene: a review. *Journal of Materials Science & Technology*, 31(6):599–606, 2015.
- [15] A. K. Geim and K. S. Novoselov. The rise of graphene. *Nature materials*, 6(3):183–191, 3 2007.
- [16] Shu Nakaharai, Tomohiko Iijima, Shinichi Ogawa, Shingo Suzuki, Song-Lin Li, Kazuhito Tsukagoshi, Shintaro Sato, and Naoki Yokoyama. Conduction tuning of graphene based on defect-induced localization. *ACS nano*, 7(7):5694–5700, 2013.
- [17] Aurélien Lherbier, Simon M-M Dubois, Xavier Declerck, Yann-Michel Niquet, Stephan Roche, and Jean-Christophe Charlier. Transport properties of graphene containing structural defects. *Physical Review B—Condensed Matter and Materials Physics*, 86(7):075402, 2012.
- [18] Christoph J Brabec. Organic photovoltaics: technology and market. *Solar energy materials and solar cells*, 83(2-3):273–292, 2004.
- [19] Jeffrey Peet, Alan J Heeger, and Guillermo C Bazan. “plastic” solar cells: self-assembly of bulk heterojunction nanomaterials by spontaneous phase separation. *Accounts of Chemical Research*, 42(11):1700–1708, 2009.
- [20] Koen Vandewal. Interfacial charge transfer states in condensed phase systems. *Annual review of physical chemistry*, 67(1):113–133, 2016.
- [21] Gang Li, Rui Zhu, and Yang Yang. Polymer solar cells. *Nature photonics*, 6(3):153–161, 2012.
- [22] Derya Baran, Thomas Kirchartz, Scot Wheeler, Stoichko Dimitrov, Maged Abdelsamie, Jeffrey Gorman, Raja S Ashraf, Sarah Holliday, Andrew Wadsworth, Nicola Gasparini, et al. Reduced voltage losses yield 10% efficient fullerene free organic solar cells with > 1 v open circuit voltages. *Energy & environmental science*, 9(12):3783–3793, 2016.

- [23] Jingnan Song, Ming Zhang, Meng Yuan, Yuhao Qian, Yanming Sun, and Feng Liu. Morphology characterization of bulk heterojunction solar cells. *Small Methods*, 2(3):1700229, 2018.
- [24] Jenny A Nelson. *The physics of solar cells*. World Scientific Publishing Company, 2003.
- [25] Paul WM Blom, Valentin D Mihailetschi, L Jan Anton Koster, and Denis E Markov. Device physics of polymer: fullerene bulk heterojunction solar cells. *Advanced Materials*, 19(12):1551–1566, 2007.
- [26] Lambert JA Koster, ECP Smits, VD Mihailetschi, and Paul WM Blom. Device model for the operation of polymer/fullerene bulk heterojunction solar cells. *Physical Review B—Condensed Matter and Materials Physics*, 72(8):085205, 2005.
- [27] Wolfgang Tress, Karl Leo, and Moritz Riede. Optimum mobility, contact properties, and open-circuit voltage of organic solar cells: A drift-diffusion simulation study. *Physical Review B—Condensed Matter and Materials Physics*, 85(15):155201, 2012.
- [28] Yuliar Firdaus and Thomas D Anthopoulos. Device physics in organic solar cells and drift-diffusion simulations. 2020.
- [29] Lauri Himanen, Amber Geurts, Adam Stuart Foster, and Patrick Rinke. Data-driven materials science: status, challenges, and perspectives. *Advanced Science*, 6(21):1900808, 2019.
- [30] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3(1):1–9, 2016.
- [31] Subashini Raghavan, Boungh Yew Lau Simon, Ying Loong Lee, Wei Lun Tan, and Keh Kim Kee. Data integration for smart cities: opportunities and challenges. *Computational Science and Technology: 6th ICCST 2019, Kota Kinabalu, Malaysia, 29-30 August 2019*, pages 393–403, 2020.
- [32] Giovanni Pizzi, Andrea Cepellotti, Riccardo Sabatini, Nicola Marzari, and Boris Kozinsky. Aiida: automated interactive infrastructure and database for computational science. *Computational Materials Science*, 111:218–230, 2016.
- [33] Claudia Draxl and Matthias Scheffler. The NOMAD laboratory: from data sharing to artificial intelligence. *Journal of Physics: Materials*, 2(3):036001, 5 2019.
- [34] Lukas Hörmann, Wojciech G Stark, and Reinhard J Maurer. Machine learning and data-driven methods in computational surface and interface science. *npj Computational Materials*, 11(1):196, 2025.

- [35] Fabio Le Piane, Mario Vozza, Matteo Baldoni, and Francesco Mercuri. Integrating high-performance computing, machine learning, data management workflows, and infrastructures for multiscale simulations and nanomaterials technologies. *Beilstein Journal of Nanotechnology*, 15(1):1498–1521, 2024.
- [36] Yuji Roh, Geon Heo, and Steven Euijong Whang. A survey on data collection for machine learning: a big data-ai integration perspective. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1328–1347, 2019.
- [37] Xin Luna Dong and Theodoros Rekatsinas. Data integration and machine learning: A natural synergy. In *Proceedings of the 2018 international conference on management of data*, pages 1645–1650, 2018.
- [38] Directorate-General for Parliamentary Research Services (European Parliament), ECORYS, Fraunhofer Fokus, IDATE, Tiana Ramahandry, Vincent Bonneau, Emarildo Bani, Nikita Vlasov, Michael Flickenschild, Olga Batura, Nikolay Tcholtchev, Philipp Lämmel, and Michell Boerger. Key enabling technologies for Europe’s technological sovereignty, 2021.
- [39] Willie Peijnenburg, AG Oomen, LG Soeteman-Hernández, M Groenewold, AJAM Sips, CW Noorlander, JAB Kettelarij, and EAJ Bleeker. Identification of emerging safety and sustainability issues of advanced materials: Proposal for a systematic approach. *NanoImpact*, 23:100342, 2021.
- [40] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.
- [41] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.
- [42] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.
- [43] Vladimir Vapnik. On the uniform convergence of relative frequencies of events to their probabilities. In *Doklady Akademii Nauk USSR*, volume 181, pages 781–787, 1968.
- [44] J MacQueen. Multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, 1967.
- [45] Patrick Reiser, Marlen Neubert, André Eberhard, Luca Torresi, Chen Zhou, Chen Shao, Houssam Metni, Clint van Hoesel, Henrik Schopmans, Timo Sommer, et al. Graph neural networks for materials science and chemistry. *Communications Materials*, 3(1):93, 2022.
- [46] Tianqi Chen. Xgboost: A scalable tree boosting system. *Cornell University*, 2016.

- [47] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.
- [48] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.
- [49] Diederik P Kingma and Max Welling. Stochastic gradient vb and the variational auto-encoder. In *Second international conference on learning representations, ICLR*, volume 19, page 121, 2014.
- [50] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.
- [51] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.
- [52] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.
- [53] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.
- [54] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. Density-based spatial clustering of applications with noise. In *Int. Conf. knowledge discovery and data mining*, volume 240, 1996.
- [55] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.
- [56] AY Ng, MI Jordan, and Y Weiss. On spectral clustering: Analysis and an algorithm. *advances 693 in neural information processing systems*, 2002.
- [57] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 580–587, 11 2013.
- [58] Ross Girshick. Fast R-CNN. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448. IEEE, 12 2015.
- [59] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 6 2015.
- [60] Zhong Qiu Zhao, Peng Zheng, Shou Tao Xu, and Xindong Wu. Object Detection with Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11):3212–3232, 11 2019.

- [61] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng Yang Fu, and Alexander C. Berg. SSD: Single Shot MultiBox Detector. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9905 LNCS:21–37, 12 2015.
- [62] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December:779–788, 6 2015.
- [63] Mujadded Al, Rabbani Alif, and Muhammad Hussain. YOLOv1 to YOLOv10: A comprehensive review of YOLO variants and their application in the agricultural domain, 6 2024.
- [64] Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. YOLOv10: Real-Time End-to-End Object Detection, 5 2024.
- [65] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12346 LNCS:213–229, 5 2020.
- [66] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [67] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pages 2127–2136. PMLR, 2018.
- [68] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [69] Michael Kearns. Thoughts on hypothesis boosting. *Unpublished manuscript*, 45:105, 1988.
- [70] Barry De Ville. Decision trees. *Wiley Interdisciplinary Reviews: Computational Statistics*, 5(6):448–455, 2013.
- [71] Camille Mondon. Classification and regression trees. 1984.
- [72] Christoph Molnar. *Model-agnostic interpretable machine learning*. PhD thesis, Imu, 2022.
- [73] Rudresh Dwivedi, Devam Dave, Het Naik, Smiti Singhal, Rana Omer, Pankesh Patel, Bin Qian, Zhenyu Wen, Tejal Shah, Graham Morgan, et al. Explainable ai (xai): Core ideas, techniques, and solutions. *ACM computing surveys*, 55(9):1–33, 2023.

- [74] Scott M. Lundberg and Su In Lee. A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*, 2017-December:4766–4775, 5 2017.
- [75] Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence*, 2(1):56–67, 2020.
- [76] Nicolas Boullé and Alex Townsend. A mathematical guide to operator learning. In *Handbook of Numerical Analysis*, volume 25, pages 83–125. Elsevier, 2024.
- [77] Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019.
- [78] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.
- [79] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.
- [80] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.
- [81] Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural pde solvers. *arXiv preprint arXiv:2202.03376*, 2022.
- [82] Kuldeep Singh Kaswan, Jagjit Singh Dhatteval, Kiran Malik, and Anupam Baliyan. Generative ai: a review on models and applications. In *2023 international conference on communication, security and artificial intelligence (ICCSAI)*, pages 699–704. IEEE, 2023.
- [83] Yue Liu, Zhengwei Yang, Zhenyao Yu, Zitu Liu, Dahui Liu, Hailong Lin, Mingqing Li, Shuchang Ma, Maxim Avdeev, and Siqi Shi. Generative artificial intelligence and its applications in materials science: Current situation and future perspectives. *Journal of Materiomics*, 9(4):798–816, 2023.
- [84] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(9):10850–10869, 2023.

- [85] Junhai Zhai, Sufang Zhang, Junfen Chen, and Qiang He. Autoencoder and its various variants. In *2018 IEEE international conference on systems, man, and cybernetics (SMC)*, pages 415–419. IEEE, 2018.
- [86] David. Foster and Karl. Friston. *Generative deep learning: teaching machines to paint, write, compose, and play*. O’Reilly Media, Incorporated, 2023.
- [87] Diederik P. Kingma, Tim Salimans, and Max Welling. Variational Dropout and the Local Reparameterization Trick. *Advances in Neural Information Processing Systems*, 2015-January:2575–2583, 6 2015.
- [88] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating Diverse High-Fidelity Images with VQ-VAE-2. *Advances in Neural Information Processing Systems*, 32, 2019.
- [89] Guangzhi Sun, Yu Zhang, Ron J. Weiss, Yuan Cao, Heiga Zen, Andrew Rosenberg, Bhuvana Ramabhadran, and Yonghui Wu. Generating Diverse and Natural Text-to-Speech Samples Using a Quantized Fine-Grained VAE and Autoregressive Prosody Prior. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2020-May:6699–6703, 5 2020.
- [90] Benjamin Sanchez-Lengeling and Alan Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, 7 2018.
- [91] Simon Axelrod, Daniel Schwalbe-Koda, Somesh Mohapatra, James Dameron, Kevin P. Greenman, and Rafael Gómez-Bombarelli. Learning Matter: Materials Design with Machine Learning and Atomistic Simulations. *Accounts of Materials Research*, 3(3):343–357, 3 2022.
- [92] Oliver Ibe. *Markov processes for stochastic modeling*. Newnes, 2013.
- [93] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [94] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- [95] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.
- [96] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

- [97] Kihyuk Sohn, Honglak Lee, and Xinchun Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.
- [98] Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O Anatole von Lilienfeld. Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning. *Phys. Rev. Lett.*, 108(5):58301, 1 2012.
- [99] Tommaso Forni, Matteo Baldoni, Fabio Le Piane, and Francesco Mercuri. GraphNet: a deep learning framework for predicting the physical and electronic properties of nanographenes using images. *Scientific Reports* 2024 14:1, 14(1):1–12, 10 2024.
- [100] Lauri Himanen, Marc O.J. Jäger, Eiaki V. Morooka, Filippo Federici Canova, Yashasvi S. Ranawat, David Z. Gao, Patrick Rinke, and Adam S. Foster. DScript: Library of descriptors for machine learning in materials science. *Computer Physics Communications*, 247:106949, 2 2020.
- [101] David Weininger. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28:31–36, 1988.
- [102] Shion Honda, Shoi Shi, and Hiroki R Ueda. Smiles transformer: Pre-trained molecular fingerprint for low data drug discovery. *arXiv preprint arXiv:1911.04738*, 2019.
- [103] Medard Edmund Mswahili and Young-Seob Jeong. Transformer-based models for chemical smiles representation: A comprehensive literature review. *Heliyon*, 10(20), 2024.
- [104] Lei Zhu, Ming Zhang, Jinqiu Xu, Chao Li, Jun Yan, Guanqing Zhou, Wenkai Zhong, Tianyu Hao, Jiali Song, Xiaonan Xue, et al. Single-junction organic solar cells with over 19% efficiency enabled by a refined double-fibril network morphology. *Nature Materials*, 21(6):656–663, 2022.
- [105] Jiaying Wang, Tianqi Chen, Wenkai Zhao, Xian Tang, Yuyang Bai, Wenying Zhou, Guankui Long, Xinyi Ji, Guanghao Lu, Wanying Feng, et al. Effective regulation of morphologies and exciton-generation process enables quasi-planar all-polymer organic solar cells exceed 18% efficiency. *Advanced Functional Materials*, 35(6):2414941, 2025.
- [106] Bhupinder Singh and Christian Kaunert. Limitless application of flexible organic photovoltaic cells revolutionizing solar energy conversion: Unveiling fuels of future cutting-edge smart cities. In *Engineering Materials for Efficient Energy Storage and Conversion*, pages 303–326. IGI Global, 2024.
- [107] Biswajit Ray, Pradeep R Nair, R Edwin García, and Muhammad A Alam. Modeling and optimization of polymer based bulk heterojunction (bh) solar cell. In *2009 IEEE International Electron Devices Meeting (IEDM)*, pages 1–4. IEEE, 2009.

- [108] Jie Shen and Xiaofeng Yang. Numerical approximations of allen-cahn and cahn-hilliard equations. *Discrete Contin. Dyn. Syst*, 28(4):1669–1691, 2010.
- [109] RCI MacKenzie, VS Balderrama, S Schmeisser, R Stoof, S Greedy, J Pallarès, LF Marsal, A Chanaewa, and E Von Hauff. Loss mechanisms in high efficiency polymer solar cells. *Advanced Energy Materials*, 6(4):1501742, 2016.
- [110] Yang Liu, Roderick CI MacKenzie, Bin Xu, Yajun Gao, Miquel Gimeno-Fabra, David Grant, Paul HM van Loosdrecht, and Wenjing Tian. Organic semiconductors with a charge carrier life time of over 2 hours at room temperature. *Journal of Materials Chemistry C*, 3(47):12260–12266, 2015.
- [111] Yajun Gao, Roderick CI MacKenzie, Yang Liu, Bin Xu, Paul HM Van Loosdrecht, and Wenjing Tian. Engineering ultra long charge carrier lifetimes in organic electronic devices at room temperature. *Adv. Mater. Interfaces*, 2(4):1–7, 2015.
- [112] Aleš Florian. An efficient sampling scheme: updated latin hypercube sampling. *Probabilistic engineering mechanics*, 7(2):123–130, 1992.
- [113] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.
- [114] Michael C Heiber, Klaus Kister, Andreas Baumann, Vladimir Dyakonov, Carsten Deibel, and Thuc-Quyen Nguyen. Impact of tortuosity on charge-carrier transport in organic bulk heterojunction blends. *Physical Review Applied*, 8(5):054043, 2017.
- [115] Caterina Soldano, Ather Mahmood, and Erik Dujardin. Production, properties and potential of graphene. *Carbon*, 48(8):2127–2150, 2010.
- [116] Antonio H Castro Neto, Francisco Guinea, Nuno MR Peres, Kostya S Novoselov, and Andre K Geim. The electronic properties of graphene. *Reviews of modern physics*, 81(1):109–162, 2009.
- [117] DSL Abergel, V Apalkov, J Berashevich, Klaus Ziegler, and Tapash Chakraborty. Properties of graphene: a theoretical perspective. *Advances in Physics*, 59(4):261–482, 2010.
- [118] Rinaldo Raccichini, Alberto Varzi, Stefano Passerini, and Bruno Scrosati. The role of graphene for electrochemical energy storage. *Nature materials*, 14(3):271–279, 2015.
- [119] Himanshu Priyadarshi, Gulzar Ahmed, Dhaneshwar Mishra, Ashish Kumar Srivastava, Randolph McGee, Ashish Shrivastava, and Kulwant Singh. Sustainable graphene-based energy storage device technology: Materials, methods, monitoring and digital twin. *Critical Reviews in Solid State and Materials Sciences*, 49(6):1086–1119, 2024.
- [120] Mills Kyle and Tamblyn Isaac. Big graphene dataset, 3 2019.

- [121] Jürgen Hafner. Ab-initio simulations of materials using VASP: Density-functional theory and beyond. *Journal of Computational Chemistry*, 29(13):2044–2078, 10 2008.
- [122] Daniele Selli and Francesco Mercuri. Correlation between atomistic morphology and electron transport properties in defect-free and defected graphene nanoribbons: An interpretation through Clar sextet theory. *Carbon*, 75:190–200, 8 2014.
- [123] Matteo Baldoni, Antonio Sgamellotti, and Francesco Mercuri. Electronic properties and stability of graphene nanoribbons: An interpretation based on Clar sextet theory. *Chemical Physics Letters*, 464:202–207, 2008.
- [124] Mario Vozza, Tommaso Forni, and Francesco Mercuri. Fair dataset of defected graphene with stm images and transport properties, May 2025.
- [125] B. Hourahine, B. Aradi, V. Blum, F. Bonafé, A. Buccheri, C. Camacho, C. Cevallos, M. Y. Deshayé, T. Dumitrică, A. Dominguez, S. Ehlert, M. Elstner, T. van der Heide, J. Hermann, S. Irle, J. J. Kranz, C. Köhler, T. Kowalczyk, T. Kubař, I. S. Lee, V. Lutsker, R. J. Maurer, S. K. Min, I. Mitchell, C. Negre, T. A. Niehaus, A. M. N. Niklasson, A. J. Page, A. Pecchia, G. Penazzi, M. P. Persson, J. Řezáč, C. G. Sánchez, M. Sternberg, M. Stöhr, F. Stuckenberg, A. Tkatchenko, V. W.-z. Yu, and T. Frauenheim. DFTB+, a software package for efficient approximate density functional theory based atomistic simulations. *The Journal of Chemical Physics*, 152(12), 3 2020.
- [126] Péter Pulay. Convergence acceleration of iterative sequences. the case of scf iteration. *Chemical Physics Letters*, 73(2):393–398, 7 1980.
- [127] Th Frauenheim, G Seifert, M Elstner, Z Hajnal, G Jungnickel, D Porezag, S Suhai, and R Scholz. A Self-Consistent Charge Density-Functional Based Tight-Binding Method for Predictive Materials Simulations in Physics, Chemistry and Biology. *phys. stat. sol. (b)*, 217, 2000.
- [128] Wolfram Koch and Max C. Holthausen. A Chemist’s Guide to Density Functional Theory. *A Chemist’s Guide to Density Functional Theory*, 7 2001.
- [129] Gustavo E. Scuseria and Viktor N. Staroverov. Progress in the development of exchange-correlation functionals. *Theory and Applications of Computational Chemistry*, pages 669–724, 1 2005.
- [130] Supriyo Datta. Electronic Transport in Mesoscopic Systems. *Electronic Transport in Mesoscopic Systems*, 9 1995.
- [131] J. Tersoff and D. R. Hamann. Theory and Application for the Scanning Tunneling Microscope. *Physical Review Letters*, 50(25):1998, 6 1983.
- [132] Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. Numba: A llvm-based python jit compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pages 1–6, 2015.

- [133] Stephen M. Pizer, R. Eugene Johnston, James P. Ericksen, Bonnie C. Yankaskas, and Keith E. Muller. Contrast-limited adaptive histogram equalization: Speed and effectiveness. *Proceedings of the First Conference on Visualization in Biomedical Computing*, pages 337–345, 1990.
- [134] Johannes Hoja, Leonardo Medrano Sandonas, Brian G. Ernst, Alvaro Vazquez-Mayagoitia, Robert A. DiStasio, and Alexandre Tkatchenko. QM7-X, a comprehensive dataset of quantum-mechanical properties spanning the chemical space of small organic molecules. *Scientific Data*, 8(1), 12 2021.
- [135] Fabio Le Piane, Mario Voza, Matteo Baldoni, and Francesco Mercuri. Integrating high-performance computing, machine learning, data management workflows, and infrastructures for multiscale simulations and nanomaterials technologies. *Beilstein J. Nanotechnol*, 15:1498–1521, 2024.
- [136] Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, and Kristin A. Persson. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL Materials*, 1(1):11002, 7 2013.
- [137] Anusuriya Devaraju and Robert Huber. An automated solution for measuring the progress toward FAIR research data. *Patterns*, 2(11):100370, 11 2021.
- [138] Thomas D. Kühne, Marcella Iannuzzi, Mauro Del Ben, Vladimir V. Rybkin, Patrick Seewald, Frederick Stein, Teodoro Laino, Rustam Z. Khaliullin, Ole Schütt, Florian Schiffmann, Dorothea Golze, Jan Wilhelm, Sergey Chulkov, Mohammad Hossein Bani-Hashemian, Valéry Weber, Urban Borštnik, Mathieu TAILLEFUMIER, Alice Shoshana Jakobovits, Alfio Lazzaro, Hans Pabst, Tiziano Müller, Robert Schade, Manuel Guidon, Samuel Andermatt, Nico Holmberg, Gregory K. Schenter, Anna Hehn, Augustin Bussy, Fabian Belleflamme, Gloria Tabacchi, Andreas Glöß, Michael Lass, Iain Bethune, Christopher J. Mundy, Christian Plessl, Matt Watkins, Joost VandeVondele, Matthias Krack, and Jürg Hutter. CP2K: An electronic structure and molecular dynamics software package -Quickstep: Efficient and accurate electronic structure calculations. *Journal of Chemical Physics*, 152(19):194103, 5 2020.
- [139] Guillaume Fraux, Jonathan Fine, ezavod, German P Barletta, Laura Scalfi, and Mykola Dimura. Chemfiles, 2 2020.
- [140] Maxim Tkachenko, Mikhail Malyuk, Andrey Holmanyuk, and Nikolai Liubimov. Label Studio: Data labeling software. <https://github.com/heartexlabs/label-studio>, 2020.
- [141] G Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [142] Rafael C Gonzalez. *Digital image processing*. Pearson education india, 2009.

- [143] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.
- [144] Dengsheng Zhang. Wavelet Transform. In *Fundamentals of Image Data Mining: Analysis, Features, Classification and Retrieval*, pages 35–44. Springer International Publishing, 2019.
- [145] Jose A. Stuchi, Marcus A. Angeloni, Rodrigo F. Pereira, Levy Boccato, Guilherme Folego, Paulo V. S. Prado, and Romis R. F. Attux. Improving image classification with frequency domain layers for feature extraction. In *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 9 2017.
- [146] Yun Chen, Huaizhong Li, Liang Hou, and Xiangjian Bu. Feature extraction using dominant frequency bands and time-frequency image analysis for chatter detection in milling. *Precision Engineering*, 56:235–245, 3 2019.
- [147] Charles R Harris, K Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, Robert Kern, Matti Picus Hoyer, Stephan, Marten H van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 9 2020.
- [148] Gregory R. Lee, Ralf Gommers, Filip Waselewski, Kai Wohlfahrt, and Aaron O’Leary. PyWavelets: A Python package for wavelet analysis. *Journal of Open Source Software*, 4(36):1237, 4 2019.
- [149] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 13-17-August-2016:785–794, 3 2016.
- [150] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A Next-generation Hyperparameter Optimization Framework, 2019.
- [151] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for Hyper-Parameter Optimization. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, pages 2546–2554. Curran Associates Inc., 2011.
- [152] Shuhei Watanabe. Tree-Structured Parzen Estimator: A Tutorial Tree-Structured Parzen Estimator: Understanding Its Algorithm Components and Their Roles for Better Empirical Performance, 2023.

- [153] Davide Anguita, Luca Ghelardoni, Alessandro Ghio, Luca Oneto, Sandro Ridella, et al. The 'k' in k-fold cross validation. In *Esann*, volume 102, pages 441–446, 2012.
- [154] Tadayoshi Fushiki. Estimation of prediction error by using k-fold cross-validation. *Statistics and Computing*, 21(2):137–146, 2011.
- [155] Isaac Kofi Nti, Owusu Nyarko-Boateng, Justice Aning, et al. Performance of machine learning algorithms with different k values in k-fold cross-validation. *International Journal of Information Technology and Computer Science*, 13(6):61–71, 2021.
- [156] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.
- [157] André Altmann, Laura Toloşi, Oliver Sander, and Thomas Lengauer. Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347, 2010.
- [158] Luke Ditria and Tom Drummond. Long-Term Prediction of Natural Video Sequences with Robust Video Predictors, 8 2023.
- [159] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
- [160] Victor Prokhorov, Ehsan Shareghi, Yingzhen Li, Mohammad Taher Pilehvar, and Nigel Collier. On the importance of the kullback-leibler divergence term in variational autoencoders for text generation. *arXiv preprint arXiv:1909.13668*, 2019.
- [161] Andrea Asperti and Matteo Trentin. Balancing reconstruction error and kullback-leibler divergence in variational autoencoders. *Ieee Access*, 8:199440–199448, 2020.
- [162] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [163] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference CVPR*, 2016-Decem:770–778, 2016.
- [164] Julia Wolleb, Florentin Bieder, Paul Friedrich, Peter Zhang, Alicia Durrer, and Philippe C Cattin. Binary noise for binary tasks: Masked bernoulli diffusion for unsupervised anomaly detection. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 135–145. Springer, 2024.

- [165] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [166] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
- [167] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.
- [168] Pierre Hohenberg and Walter Kohn. Inhomogeneous electron gas. *Physical review*, 136(3B):B864, 1964.
- [169] Walter Kohn and Lu Jeu Sham. Self-consistent equations including exchange and correlation effects. *Physical review*, 140(4A):A1133, 1965.
- [170] G Seifert. Tight-binding density functional theory: an approximate kohn-sham dft scheme. *The Journal of Physical Chemistry A*, 111(26):5609–5613, 2007.

Appendix A

General Background Appendix

A.1 Solve the Chan-Hilliard with Spectral Methods

For the solution of the phase separation equation, we relied on spatial resolution methods that allow solving a partial differential equation (PDE) by employing the Fourier transform and subsequently its inverse to obtain the results described above. Specifically, we transformed the spatial domain problem into the frequency domain, where differential operators become algebraic multipliers, greatly simplifying the numerical computation.

$$\frac{\partial c}{\partial t} = M \nabla^2 \mu, \quad (\text{A.1})$$

where $c(\mathbf{x}, t)$ is the concentration field, M is the mobility coefficient, and μ is the chemical potential defined as the variational derivative of the free energy functional

$$\mu = \frac{\delta F}{\delta c} = -\kappa \nabla^2 c + f'(c). \quad (\text{A.2})$$

Here, κ is a positive constant related to the interfacial energy, and $f(c)$ is a double-well potential representing the bulk free energy density, typically

$$f(c) = ac^2(1 - c)^2,$$

with $a > 0$ controlling the height of the energy barrier.

Substituting Eq. (A.2) into Eq. (A.1), we obtain a fourth-order nonlinear PDE:

$$\frac{\partial c}{\partial t} = M \nabla^2 (-\kappa \nabla^2 c + f'(c)) = -M \kappa \nabla^4 c + M \nabla^2 f'(c). \quad (\text{A.3})$$

To efficiently solve Eq. (A.3) in a periodic cubic domain of size L , we employ the Fourier spectral method. The concentration field $c(\mathbf{x}, t)$ is expanded in Fourier modes:

$$c(\mathbf{x}, t) = \sum_{\mathbf{k}} \hat{c}(\mathbf{k}, t) e^{i\mathbf{k} \cdot \mathbf{x}},$$

where $\mathbf{k} = (k_x, k_y, k_z)$ are the discrete wave vectors given by

$$k_j = \frac{2\pi}{L} n_j, \quad n_j = -\frac{N}{2}, \dots, \frac{N}{2} - 1, \quad j = x, y, z,$$

and N is the number of grid points per dimension.

The Laplacian operator in Fourier space acts as multiplication by $-|\mathbf{k}|^2$:

$$\nabla^2 c(\mathbf{x}, t) \longleftrightarrow -|\mathbf{k}|^2 \hat{c}(\mathbf{k}, t),$$

with

$$|\mathbf{k}|^2 = k_x^2 + k_y^2 + k_z^2.$$

Similarly, the biharmonic operator $\nabla^4 = (\nabla^2)^2$ corresponds to multiplication by $|\mathbf{k}|^4$.

We discretise time using a semi-implicit scheme with time step Δt . Denoting by c^n the concentration at time $t_n = n\Delta t$, the update rule in Fourier space is

$$\frac{\hat{c}^{n+1} - \hat{c}^n}{\Delta t} = -M \kappa |\mathbf{k}|^4 \hat{c}^{n+1} - M |\mathbf{k}|^2 \widehat{f'(c^n)}, \quad (\text{A.4})$$

where the nonlinear term $\widehat{f'(c^n)}$ is treated explicitly.

Rearranging Eq. (A.4) to solve for $\hat{c}^{n+1}$, we get

$$\hat{c}^{n+1} = \frac{\hat{c}^n - \Delta t M |\mathbf{k}|^2 \widehat{f'(c^n)}}{1 + \Delta t M \kappa |\mathbf{k}|^4}, \quad (\text{A.5})$$

This represents the update step performed in the Fourier domain to compute the concentration at timestep $n + 1$. The final step to obtain the concentration in the physical domain is to calculate the inverse Fourier transform.

$$c^{n+1} = \mathcal{F}^{-1}[\hat{c}^{n+1}]. \quad (\text{A.6})$$

The solution is computed using mixed precision arithmetic, where calculations are performed primarily in half precision to improve computational efficiency, while key operations such as the nonlinear term evaluation and Fourier transforms are carried out in single precision to maintain numerical stability.

For clarity, see the complete algorithm 4.

All the parameters used in the simulation are summarised in the table A.1

Table A.1 Simulation parameters used for the Cahn-Hilliard equation

Parameter	Description	Value
N	Number of grid points per dimension	256
L	Domain size	100.0
Δx	Grid spacing	$\frac{L}{N} = 0.390$
M	Mobility coefficient	3.0
κ	Interfacial energy coefficient	2.5
a	Bulk free energy parameter (double-well)	3.0
Δt	Time step size	0.01
steps	Number of time steps	5000

Algorithm 4 Spectral solver for the Cahn-Hilliard equation

- 1: Initialise concentration field $c(\mathbf{x}, 0)$ with small random perturbations around mean
- 2: **for** each time step $n = 0, 1, \dots, N_{\text{steps}} - 1$ **do**
- 3: Compute nonlinear term in physical space:

$$f'(c^n) = a \cdot 2c^n(c^n - 1)(2c^n - 1)$$

- 4: Compute forward Fourier transforms:

$$\hat{c}^n = \mathcal{F}[c^n], \quad \widehat{f'(c^n)} = \mathcal{F}[f'(c^n)]$$

- 5: Update Fourier coefficients using semi-implicit scheme:

$$\hat{c}^{n+1} = \frac{\hat{c}^n - \Delta t M |\mathbf{k}|^2 \widehat{f'(c^n)}}{1 + \Delta t M \kappa |\mathbf{k}|^4}$$

- 6: Compute inverse Fourier transform to obtain c^{n+1} :

$$c^{n+1} = \mathcal{F}^{-1}[\hat{c}^{n+1}]$$

- 7: Apply clipping to maintain physical bounds:

$$c^{n+1} \leftarrow \min(\max(c^{n+1}, 0), 1)$$

- 8: **end for**

A.2 Tucker Decomposition

The Tucker decomposition is a higher-order generalisation of the singular value decomposition (SVD) for tensors, providing both dimensionality reduction and insight into the multilinear structure of multiway data. Given a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$,

the Tucker decomposition factorises it into a core tensor $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \dots \times R_N}$ and a set of factor matrices $\{\mathbf{U}^{(n)}\}_{n=1}^N$, with $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times R_n}$, such that

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \dots \times_N \mathbf{U}^{(N)}, \quad (\text{A.7})$$

where $\times_n$ denotes the mode- n tensor-matrix product defined by

$$(\mathcal{X} \times_n \mathbf{U}^{(n)})_{i_1, \dots, i_{n-1}, j_n, i_{n+1}, \dots, i_N} = \sum_{i_n=1}^{I_n} \mathcal{X}_{i_1, \dots, i_N} \mathbf{U}_{j_n, i_n}^{(n)}, \quad (\text{A.8})$$

which contracts the n -th mode of $\mathcal{X}$ with $\mathbf{U}^{(n)}$.

The standard computational method for Tucker decomposition is the Higher-Order Singular Value Decomposition (HOSVD) [167], which proceeds by computing the mode- n unfolding $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times (I_1 \dots I_{n-1} I_{n+1} \dots I_N)}$ of $\mathcal{X}$, performing an SVD on each unfolding,

$$\mathbf{X}_{(n)} = \mathbf{U}^{(n)} \mathbf{\Sigma}^{(n)} (\mathbf{V}^{(n)})^\top, \quad (\text{A.9})$$

retaining the first R_n columns of $\mathbf{U}^{(n)}$ corresponding to the largest singular values, and finally computing the core tensor as

$$\mathcal{G} = \mathcal{X} \times_1 (\mathbf{U}^{(1)})^\top \times_2 (\mathbf{U}^{(2)})^\top \dots \times_N (\mathbf{U}^{(N)})^\top, \quad (\text{A.10})$$

so that the approximation

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{U}^{(1)} \dots \times_N \mathbf{U}^{(N)} \quad (\text{A.11})$$

holds. For a third-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, this reduces to

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)}, \quad (\text{A.12})$$

with $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$ and $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times R_n}$, and the reconstruction error is measured by the Frobenius norm

$$\text{Error} = \|\mathcal{X} - \mathcal{G} \times_1 \mathbf{U}^{(1)} \dots \times_N \mathbf{U}^{(N)}\|_F. \quad (\text{A.13})$$

The Tucker decomposition possesses several important properties: the factor matrices $\mathbf{U}^{(n)}$ can be chosen orthogonal, $(\mathbf{U}^{(n)})^\top \mathbf{U}^{(n)} = \mathbf{I}$, ensuring that the core tensor preserves multilinear singular values; the tuple $(R_1, \dots, R_N)$ defines the multilinear

rank of $\mathcal{X}$; and when $R_n < I_n$ for some n , the decomposition achieves a compressed representation suitable for storage and computational efficiency. Overall, the Tucker decomposition offers a rigorous and flexible framework to analyse multiway data, providing low-rank approximations while retaining the structural relationships across tensor modes.

A.3 Density Functional Theory

Density Functional Theory (DFT) is a quantum mechanical framework for modelling the electronic structure of many-body systems, particularly atoms, molecules, and solids. In DFT, the ground-state properties of an interacting electron system are expressed as functionals of the electron density $\rho(\mathbf{r})$ rather than the many-electron wavefunction.

The formal basis of DFT is provided by the Hohenberg–Kohn theorems [168], which state that the ground-state electron density uniquely determines the external potential $v_{\text{ext}}(\mathbf{r})$, and hence all properties of the system, and that there exists a universal functional $F[\rho]$ such that the total energy can be written as

$$E[\rho] = F[\rho] + \int \rho(\mathbf{r}) v_{\text{ext}}(\mathbf{r}) d\mathbf{r}, \quad (\text{A.14})$$

where $F[\rho]$ encompasses the kinetic energy of the electrons and their mutual Coulomb interactions.

In practical applications, the Kohn–Sham formulation of DFT [169] is employed. This approach replaces the interacting many-electron problem with an auxiliary system of non-interacting electrons moving in an effective potential $v_{\text{eff}}(\mathbf{r})$, such that the electron density of the auxiliary system reproduces that of the interacting system. The Kohn–Sham equations read

$$\left[-\frac{\hbar^2}{2m} \nabla^2 + v_{\text{eff}}(\mathbf{r}) \right] \psi_i(\mathbf{r}) = \epsilon_i \psi_i(\mathbf{r}), \quad (\text{A.15})$$

with the effective potential defined as

$$v_{\text{eff}}(\mathbf{r}) = v_{\text{ext}}(\mathbf{r}) + \int \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}' + v_{\text{xc}}[\rho](\mathbf{r}), \quad (\text{A.16})$$

where $v_{xc}[\rho](\mathbf{r})$ is the exchange-correlation potential, accounting for all many-body effects beyond the classical Hartree term.

The total electron density is obtained by summing over the occupied Kohn–Sham orbitals:

$$\rho(\mathbf{r}) = \sum_i^{N_{\text{occ}}} |\psi_i(\mathbf{r})|^2. \quad (\text{A.17})$$

The choice of exchange-correlation functional is central to the accuracy of DFT calculations. Common approximations include the Local Density Approximation (LDA) and the Generalised Gradient Approximation (GGA), each offering a trade-off between computational cost and fidelity to experimental properties.

DFT provides the foundation for a wide range of computational methods in electronic structure theory, including density-functional tight-binding (DFTB) approaches, ab initio molecular dynamics, and transport simulations using non-equilibrium Green's functions.

A.4 Density Functional Tight-Binding

Density Functional Tight-Binding (DFTB) is an approximate quantum mechanical method derived from Density Functional Theory (DFT) [170, 125]. It is designed to reduce the computational cost of DFT calculations while retaining a reliable description of electronic structure, making it suitable for large systems such as nanostructures, biomolecules, and materials with defects.

The key idea of DFTB is to expand the total DFT energy around a reference density, typically a superposition of neutral atomic densities, up to second order in the density fluctuations. The total energy is then expressed as

$$E_{\text{DFTB}} = E_0 + E_1 + E_2, \quad (\text{A.18})$$

where:

- E_0 is the sum of the energies of the neutral atoms (reference energy),
- E_1 is the electronic contribution obtained from a tight-binding Hamiltonian built using Slater-Koster parametrizations of two-center integrals,

- E_2 is a second-order correction accounting for charge redistribution through a self-consistent treatment of charge fluctuations.

The electronic term E_1 is computed by solving a simplified eigenvalue problem:

$$\sum_{\nu} H_{\mu\nu}^{\text{TB}} c_{\nu}^i = \varepsilon_i \sum_{\nu} S_{\mu\nu} c_{\nu}^i, \quad (\text{A.19})$$

where $H_{\mu\nu}^{\text{TB}}$ and $S_{\mu\nu}$ are the tight-binding Hamiltonian and overlap matrices between atomic orbitals μ and ν , and c_{ν}^i are the coefficients of the i -th molecular orbital.

The charge fluctuation term E_2 is typically approximated using a Hubbard-like interaction:

$$E_2 = \frac{1}{2} \sum_{A,B} \gamma_{AB} \Delta q_A \Delta q_B, \quad (\text{A.20})$$

where Δq_A and Δq_B are the charge deviations on atoms A and B , and γ_{AB} is a distance-dependent Coulomb interaction.

DFTB can also include corrections for dispersion interactions and spin polarization. Its accuracy strongly depends on the parametrization of the Slater–Koster integrals for the element pairs in the system, and it is widely used as a computationally efficient alternative to full DFT for simulations of large-scale materials, electron transport, and molecular dynamics.

A.5 Itô Stochastic Differential Equations

A stochastic differential equation (SDE) in the sense of Itô provides a mathematical framework for modelling the evolution of a system subject to both deterministic dynamics and random fluctuations. An Itô SDE is written in the form

$$dx_t = f(x_t, t) dt + g(x_t, t) dW_t, \quad (\text{A.21})$$

where:

- x_t is the state of the system at time t ;
- $f(x_t, t)$ is the *drift* term, describing deterministic evolution;
- $g(x_t, t)$ is the *diffusion* coefficient, controlling the amplitude of the noise;

- W_t is a Wiener process (standard Brownian motion), satisfying

$$W_0 = 0, \quad W_t - W_s \sim \mathcal{N}(0, t - s), \quad \text{independent increments.}$$

Equation (A.21) must be interpreted using the rules of Itô calculus. Unlike classical calculus, increments of Brownian motion satisfy the quadratic variation identity

$$(dW_t)^2 = dt, \quad dW_t dt = 0, \quad (dt)^2 = 0,$$

which underpins the Itô integral

$$\int_0^t g(x_s, s) dW_s.$$

Itô's Lemma

A central result in stochastic calculus is Itô's lemma, the analogue of the chain rule for stochastic processes. If x_t satisfies (A.21) and $\phi(x, t)$ is a twice differentiable function, then the differential of ϕ is

$$d\phi(x_t, t) = \frac{\partial \phi}{\partial t}(x_t, t) dt + \frac{\partial \phi}{\partial x}(x_t, t) dx_t + \frac{1}{2} \frac{\partial^2 \phi}{\partial x^2}(x_t, t) g(x_t, t)^2 dt. \quad (\text{A.22})$$

The presence of the second-order term distinguishes Itô calculus from classical differentiation and reflects the non-smooth nature of Brownian paths.

Relevance to Diffusion Models

In the context of diffusion-based generative modelling, the discrete forward corruption process used in DDPMs admits a continuous-time limit that is governed by an Itô SDE of the form

$$dx_t = f(t)x_t dt + g(t) dW_t.$$

Song and Ermon demonstrated that the corresponding reverse-time generative dynamics is described by the reverse SDE

$$dx_t = [f(t)x_t - g(t)^2 \nabla_{x_t} \log q_t(x_t)] dt + g(t) d\bar{W}_t,$$

where $\nabla_{x_t} \log q_t(x_t)$ is the *score function* of the marginal density. Neural networks trained to approximate the score therefore enable sampling by numerical integration of the reverse SDE, providing a continuous-time interpretation of denoising diffusion models.

The Itô formulation thus plays a foundational role in establishing the theoretical connection between discrete DDPMs, score-based generative models and probability-flow ODE samplers.

Appendix B

Supporting information Appendix

B.1 Atomistic Graphene Data to Generative Models: An AI Framework for Structure Properties Correlation

Additional table

Parameters	Range
n_estimators	[100, 1000]
learning_rate	[0.001, 0.3] (log)
max_depth	[3, 15]
gamma	[0.00001, 5] (log)
min_child_weight	[1, 20] (log)
colsample_bytree	[0.5, 1] (log)
max_leaves	[20, 150]
subsample	[0.5, 1] (log)
reg_alpha	[0.5, 10.0] (log)
reg_lambda	[0.5, 10.0] (log)

Table B.1 Ranges of XGBoost parameters used in Optuna optimization.

Target	MSE	MAE	R^2
Fermi Level [%]	8.1161	6.4963	-0.4285
Ionization Potential [%]	10.2995	8.0458	-0.9135
Electron Affinity [%]	7.6198	7.8669	-0.4557
Band Gap [%]	9.9243	8.5206	-0.6750
Energy per atom [%]	2.4111	5.2436	-0.0202
Current [%]	2.6113	2.2004	-0.1390

Table B.2 Percentage difference between the error metrics of XGBoost and LightGBM. Positive values indicate that XGBoost yields lower error metrics (e.g., MSE and MAE) compared to LightGBM, whereas negative values indicate that XGBoost achieves higher values in metrics such as R^2 , relative to LightGBM.

Additional Figures

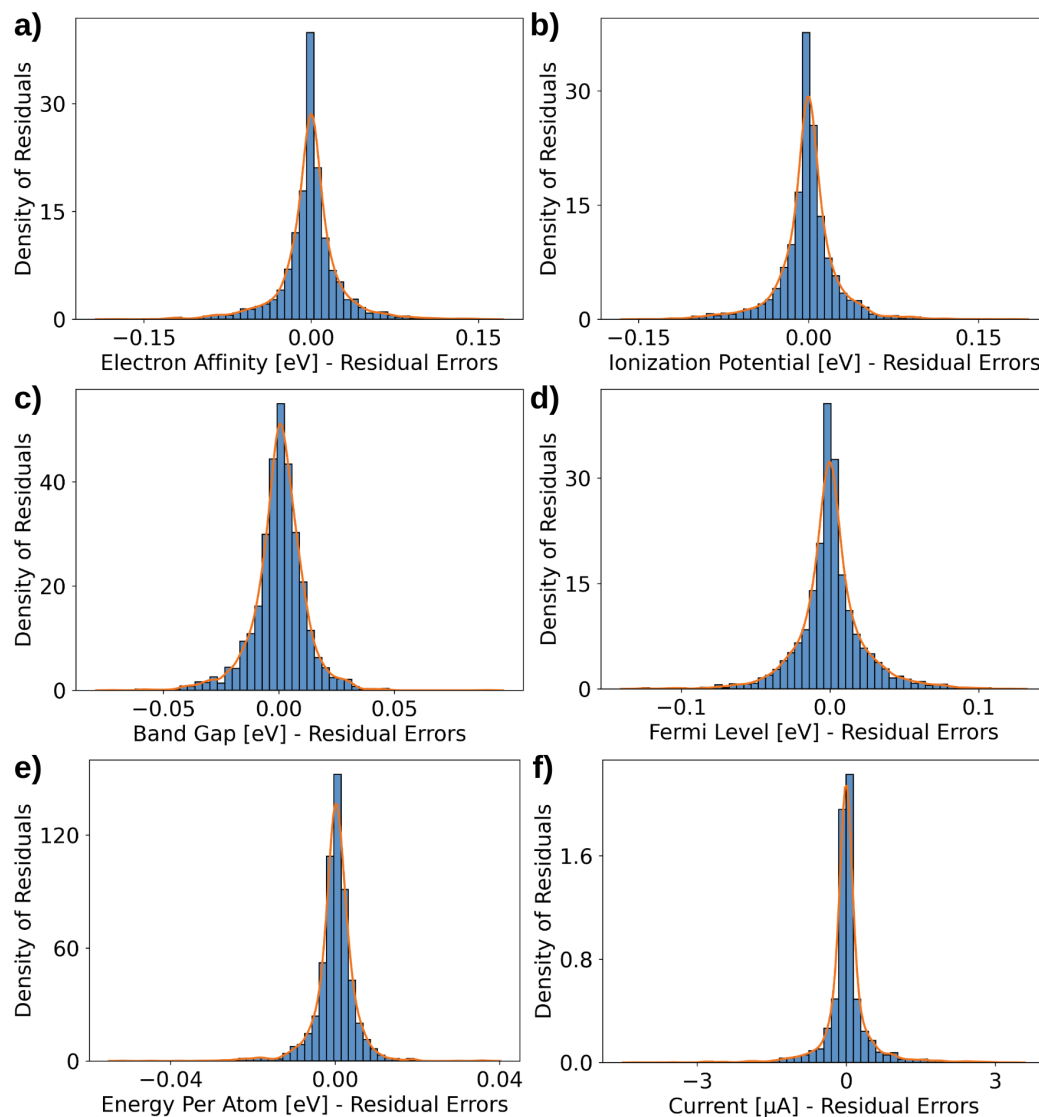


Fig. B.1 Residual errors for the predictions of: a) electron affinity, b) ionization potential, c) band gap, d) Fermi level, e) energy per atom, and f) current using the XGBoost model.

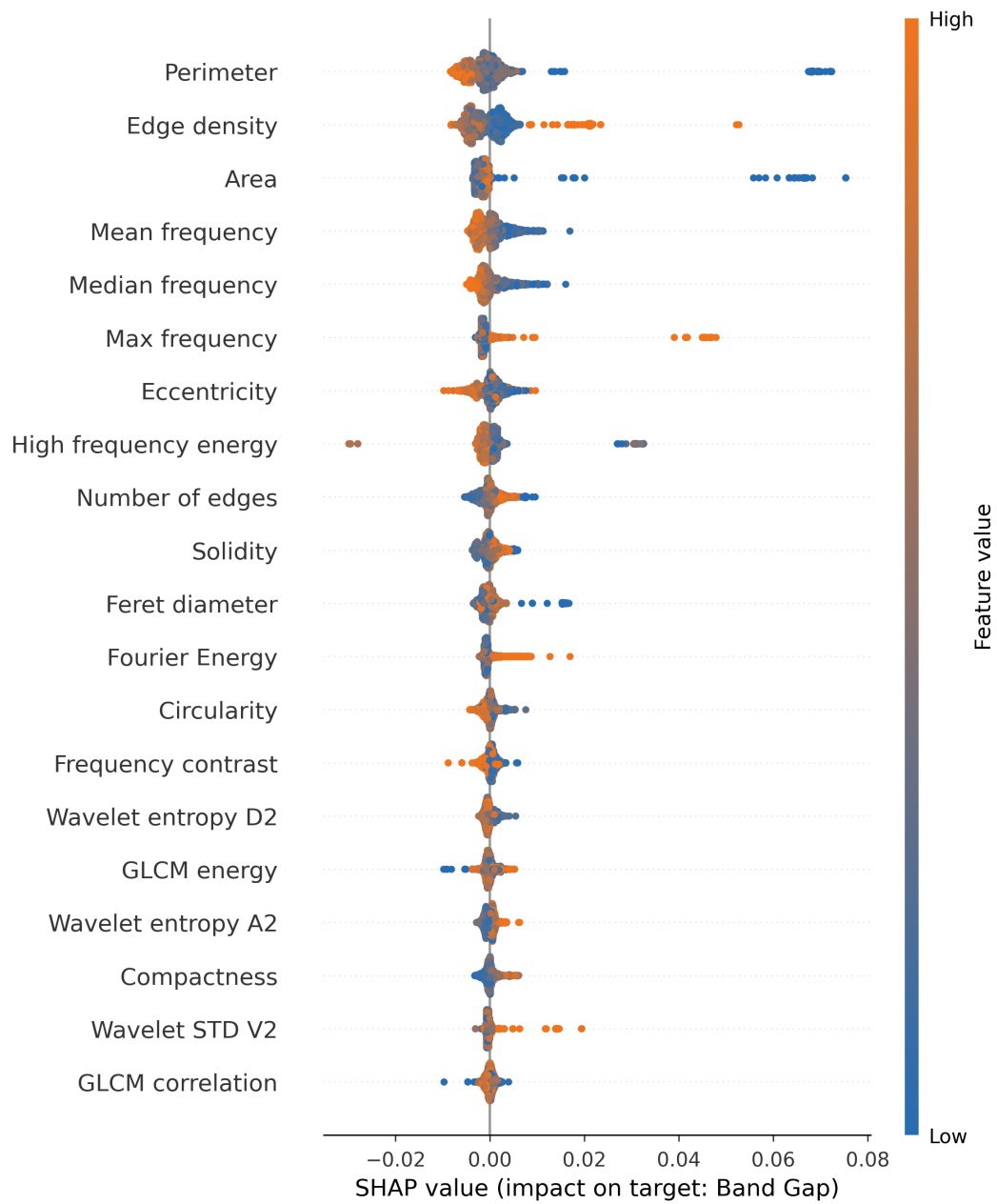


Fig. B.2 SHAP value summary trained on XGBoost for the bandgap target

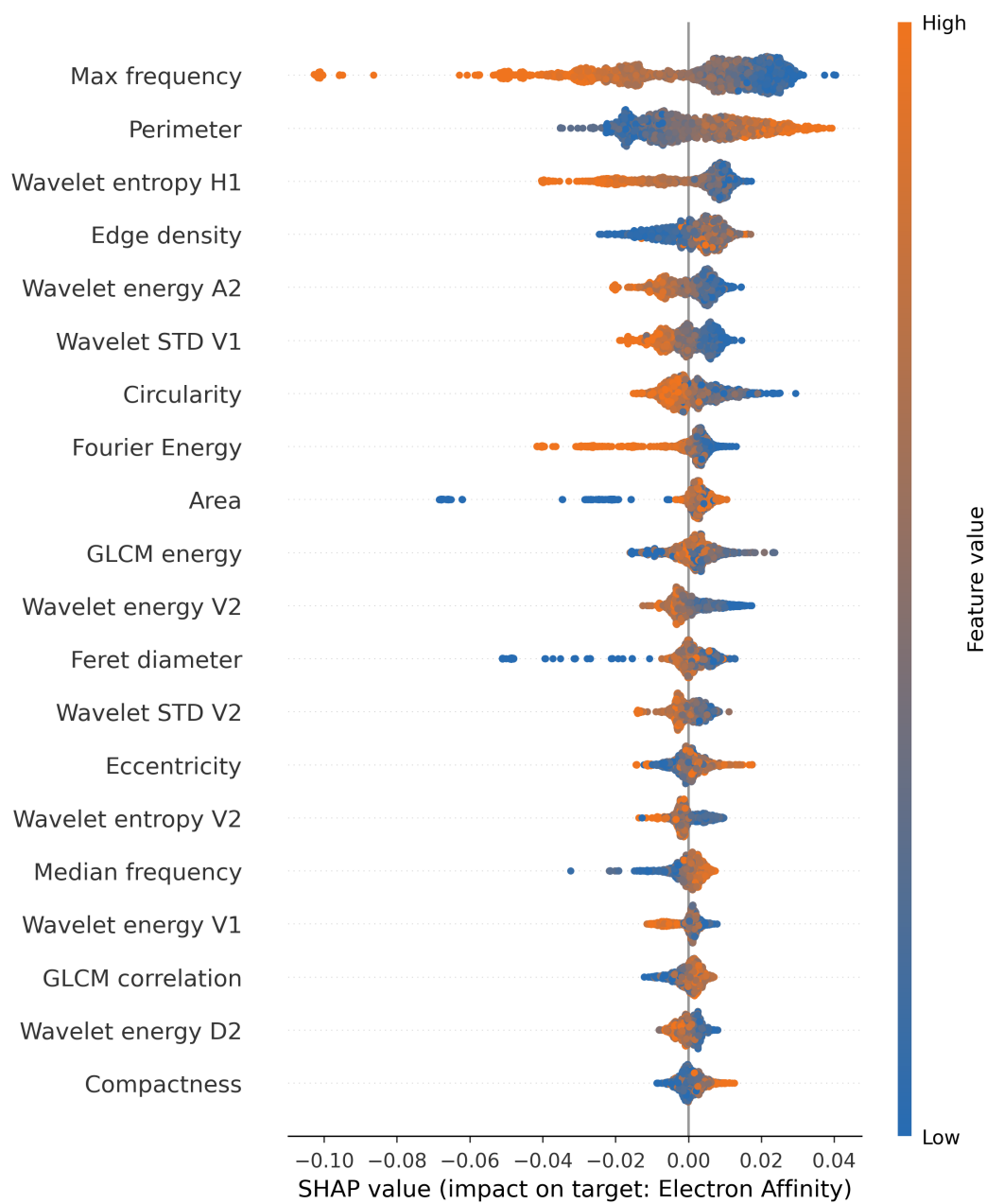


Fig. B.3 SHAP value summary trained on XGBoost for the electron affinity target

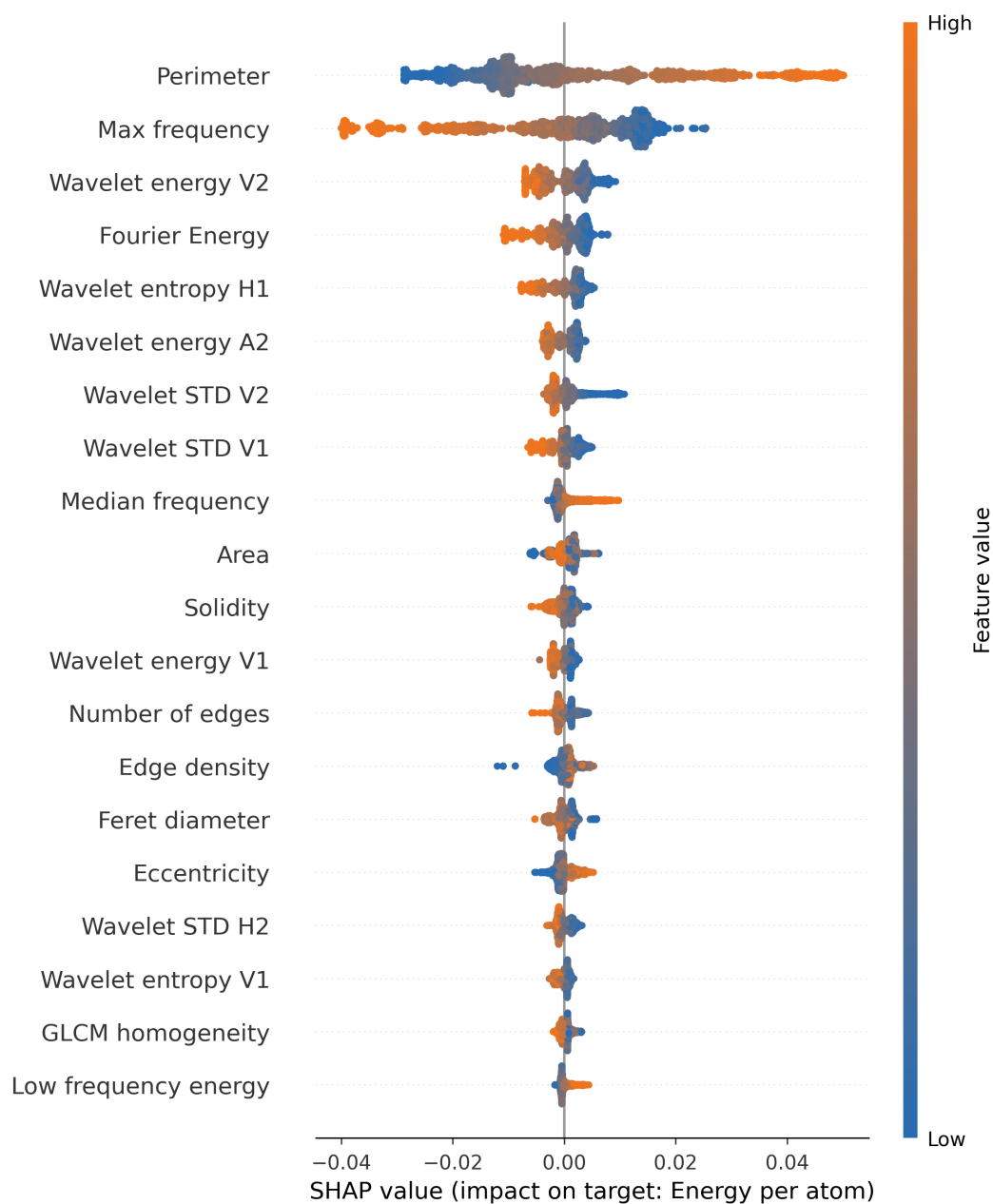


Fig. B.4 SHAP value summary trained on XGBoost for the energy per atom target

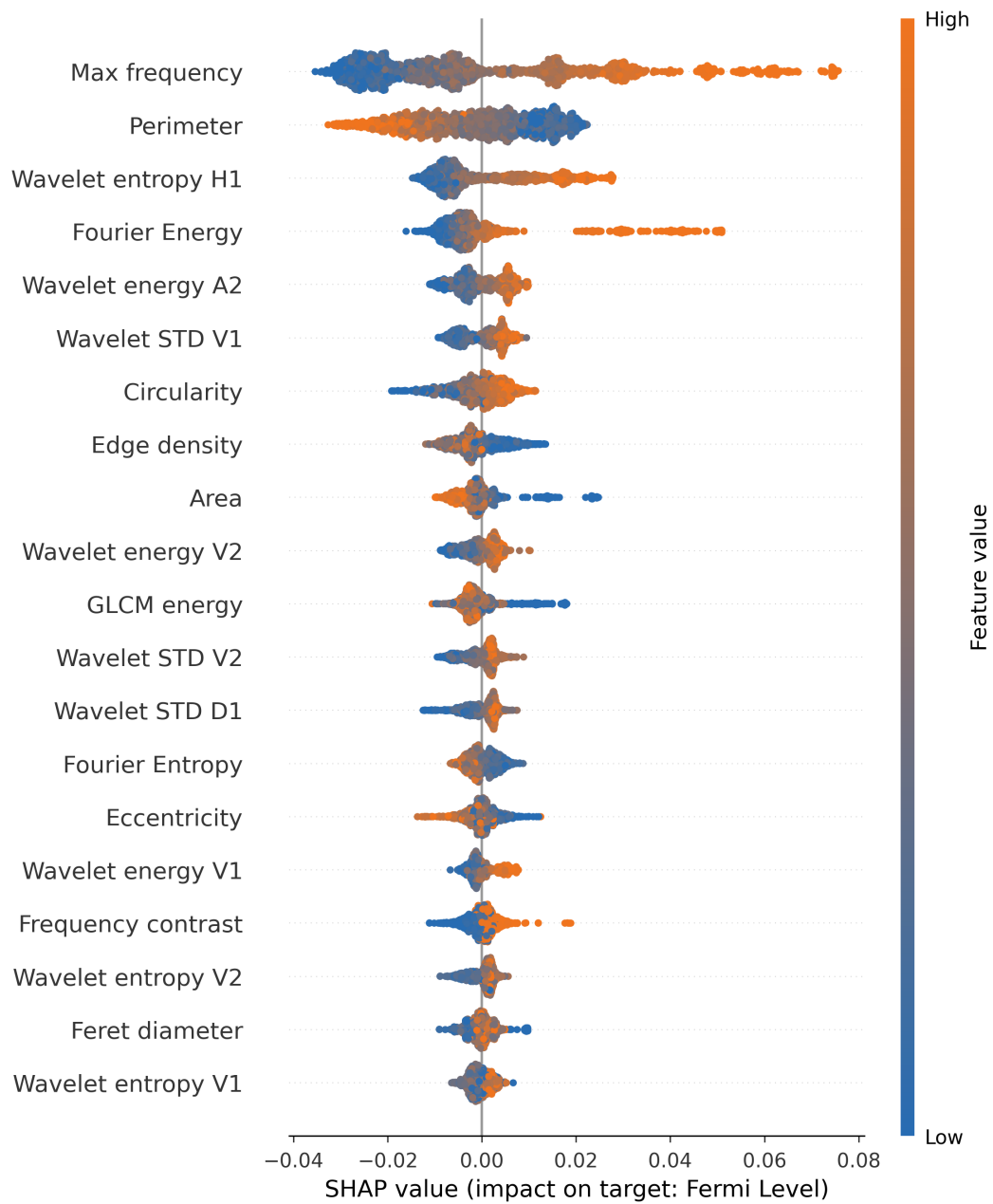


Fig. B.5 SHAP value summary trained on XGBoost for the Fermi level target

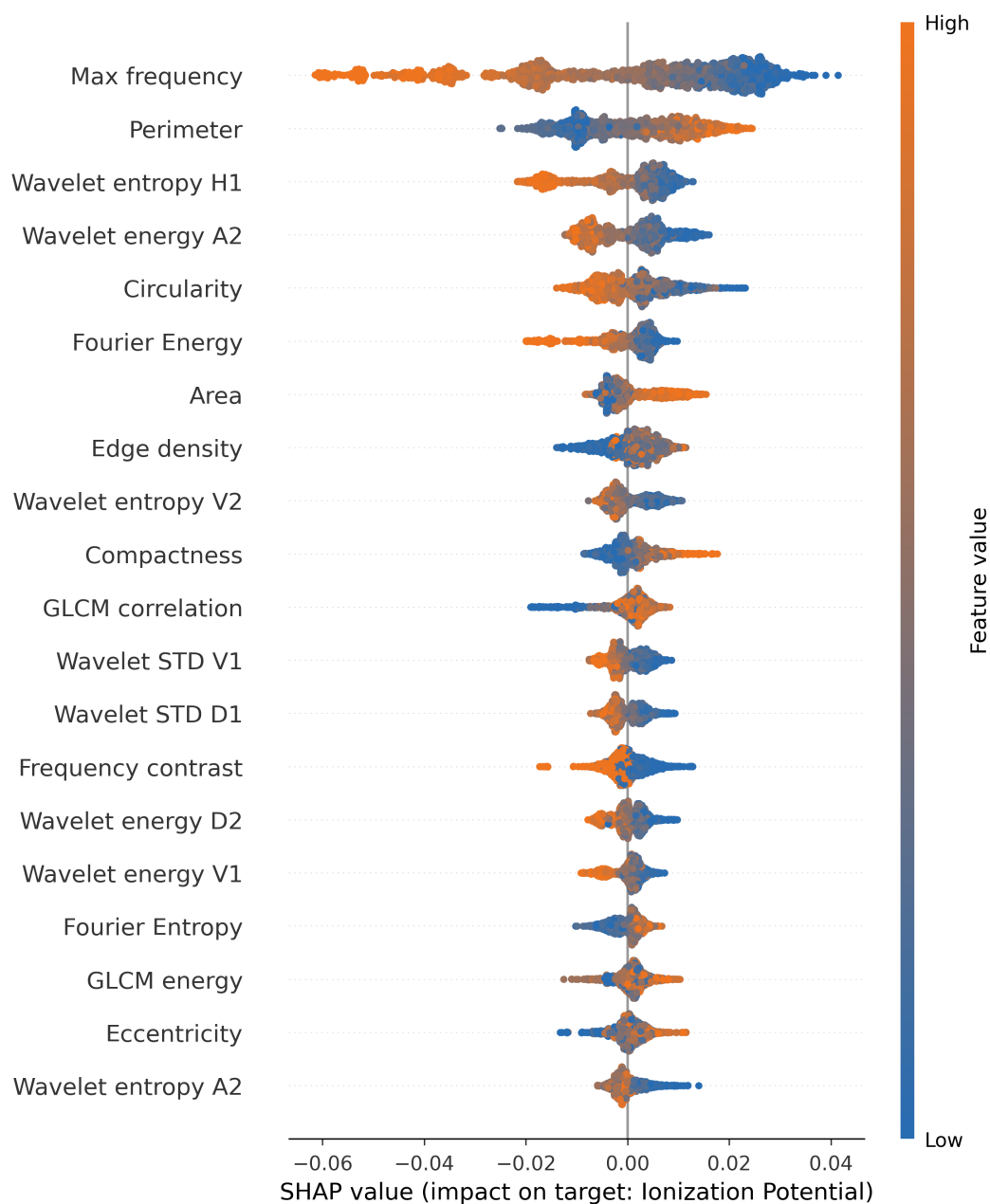


Fig. B.6 SHAP value summary trained on XGBoost for the Ionisation potential level target

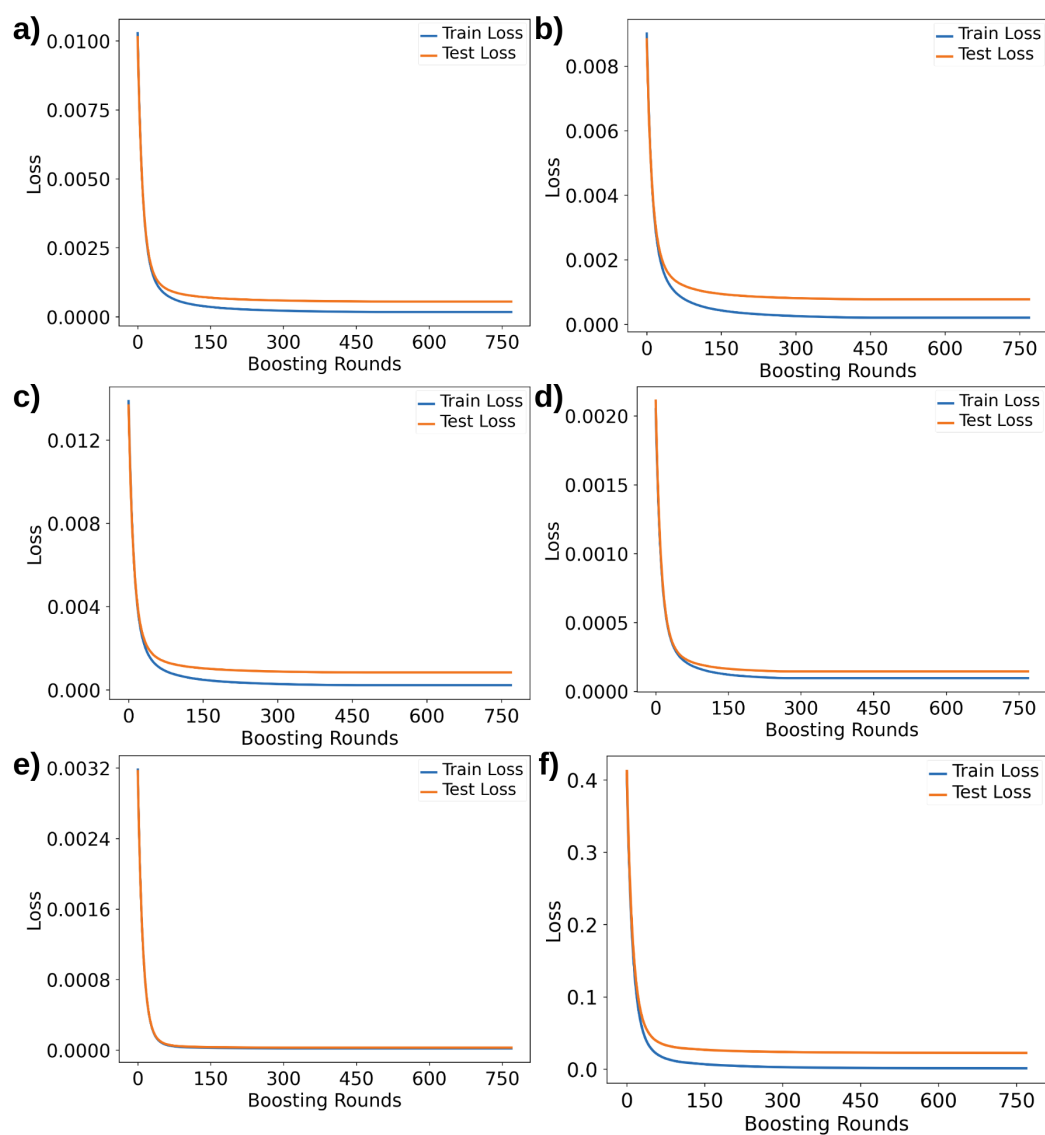


Fig. B.7 The plots show the boosting rounds compared to RMSE errors for the train set and test set using LightGBM for Fermi level (a), ionization potential (b), electron affinity (c), band gap (d), energy per atom (e), and current (f).

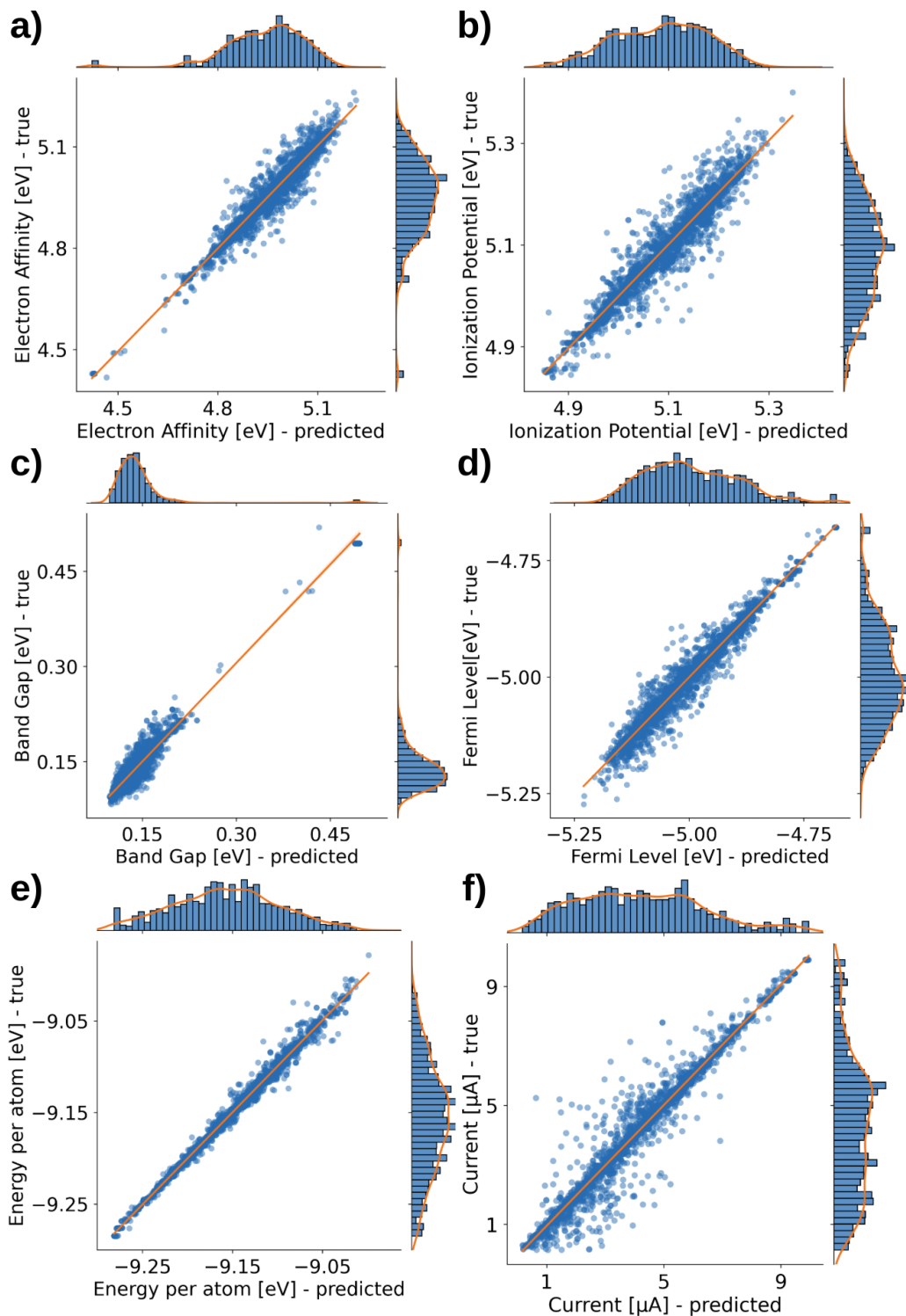


Fig. B.8 Scatter plots show the correlation between the predicted and true values in the test set using LightGBM for electron affinity (a), ionization potential (b), band gap (c), Fermi level (d), energy per atom (e), and current (f). The lines (orange) represent the ideal $y = x$ correlation. Marginal histograms display the distribution of predicted and true values for each property, indicating the performance of the predictive model across different datasets

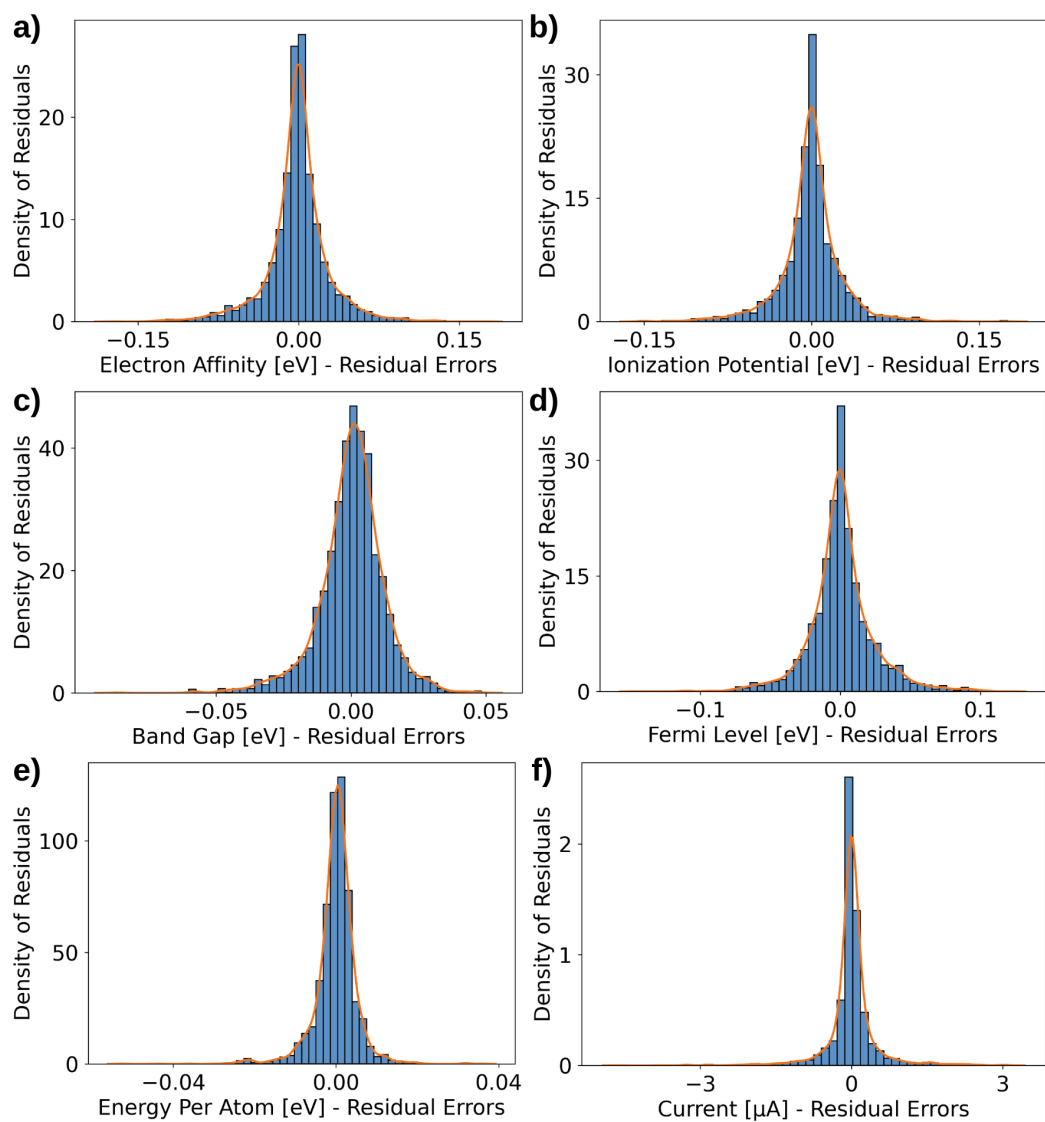


Fig. B.9 Residual errors for the predictions of: a) electron affinity, b) ionization potential, c) band gap, d) Fermi level, e) energy per atom, and f) current using the LightGBM model.

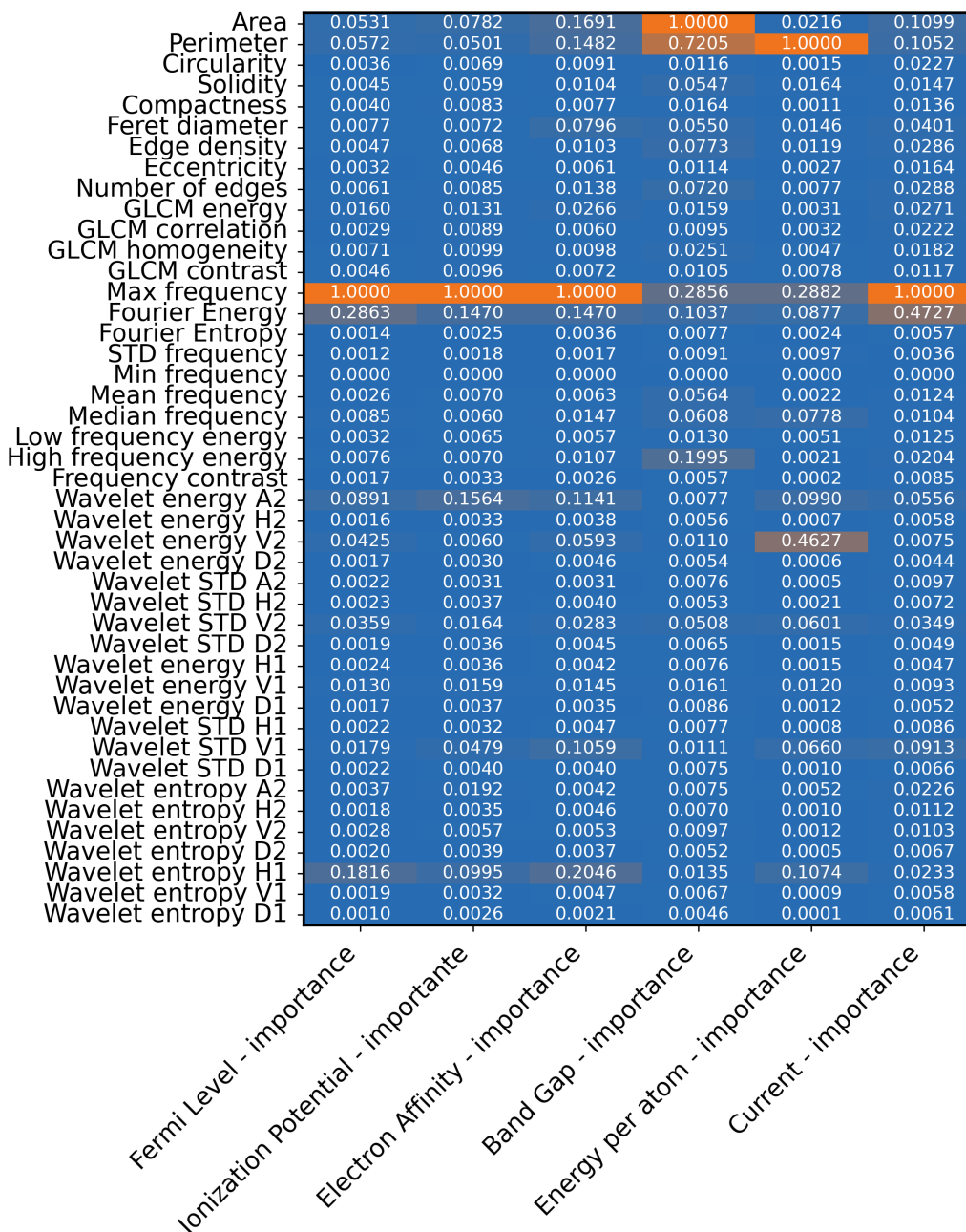


Fig. B.10 XGBoost feature importance visualised as a heatmap. Each column corresponds to a different target property, while each row represents a distinct feature.

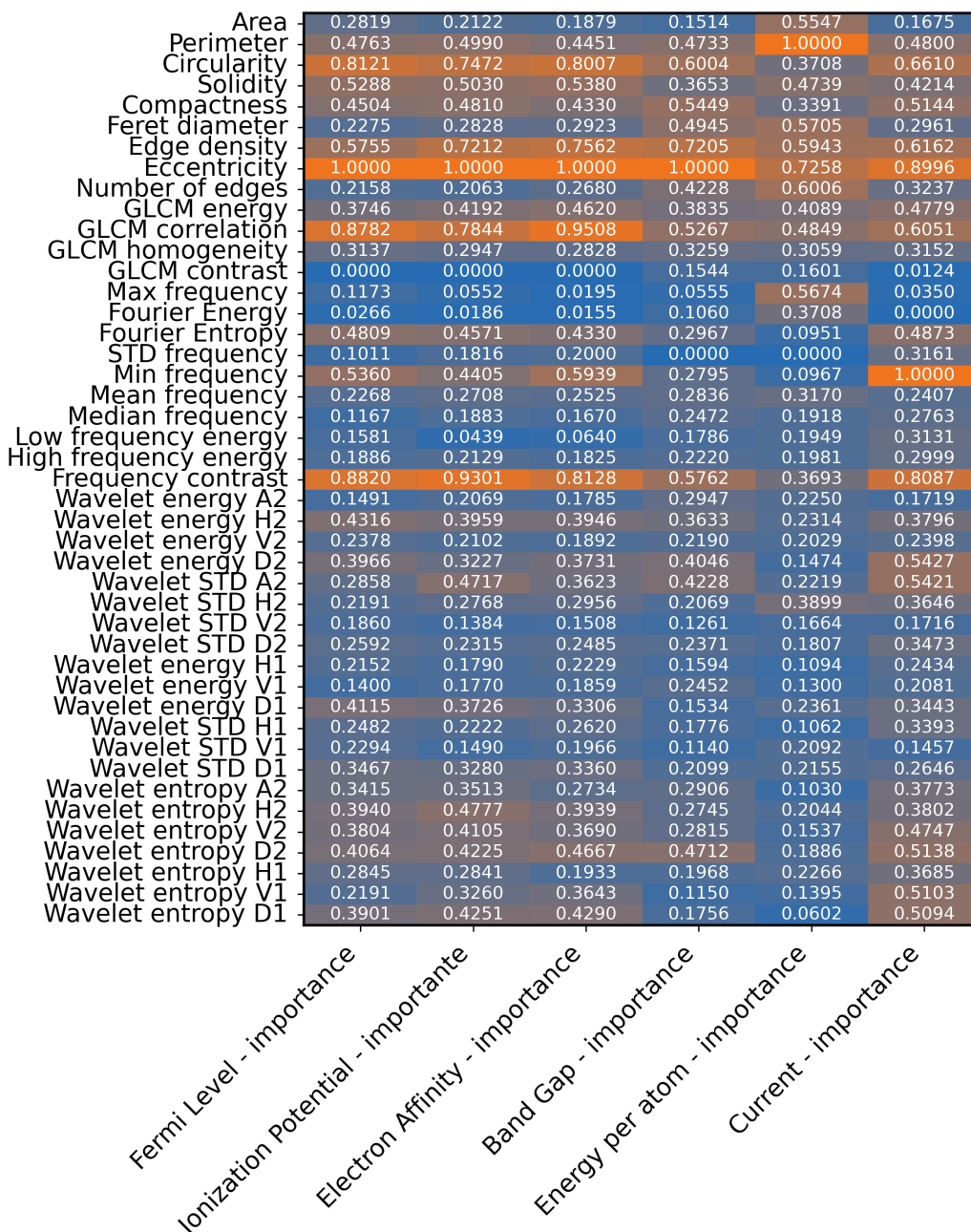


Fig. B.11 LightGBM feature importance visualised as a heatmap. Each column corresponds to a different target property, while each row represents a distinct feature.

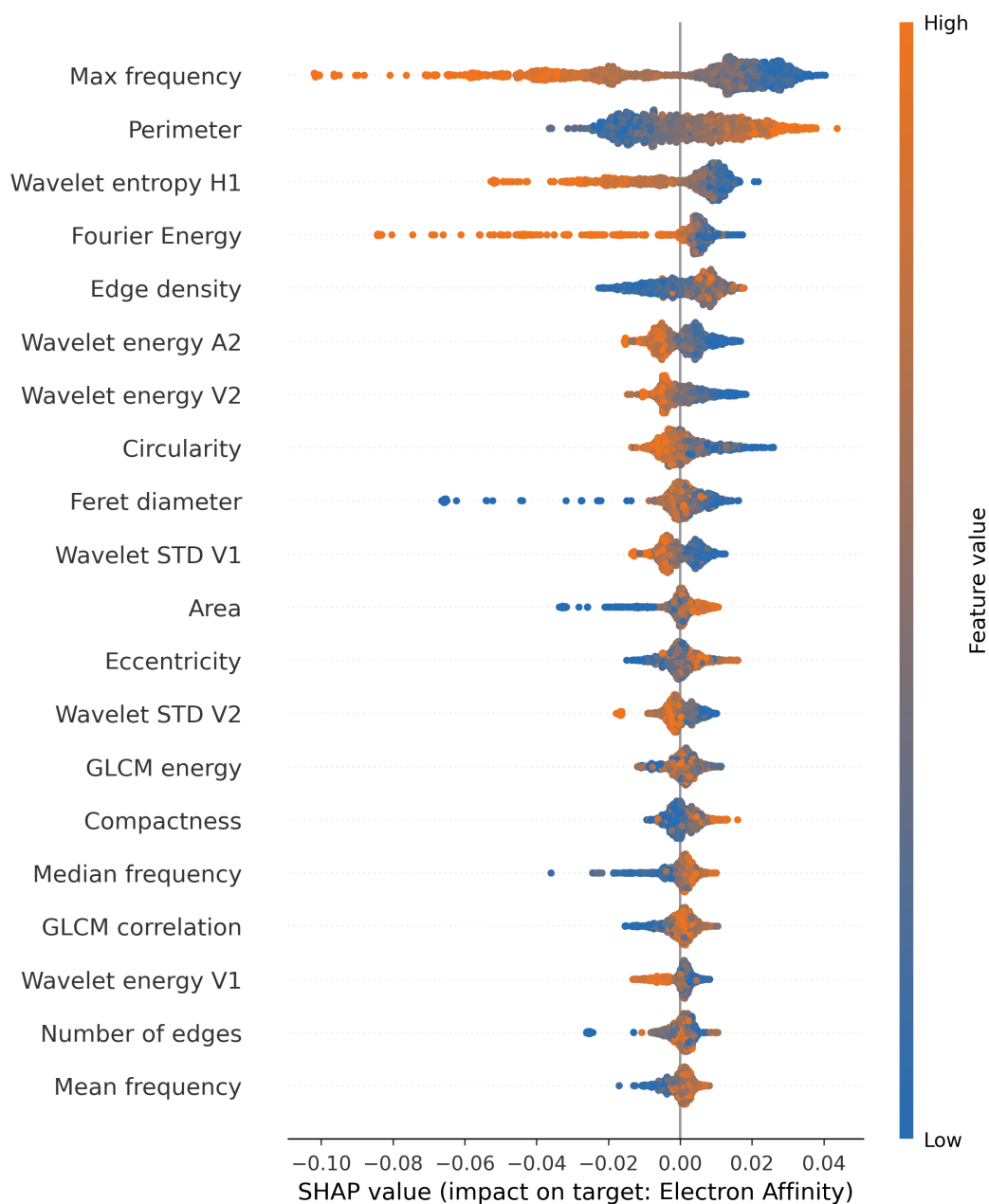


Fig. B.12 SHAP value summary trained on LightGBM for the electron affinity target

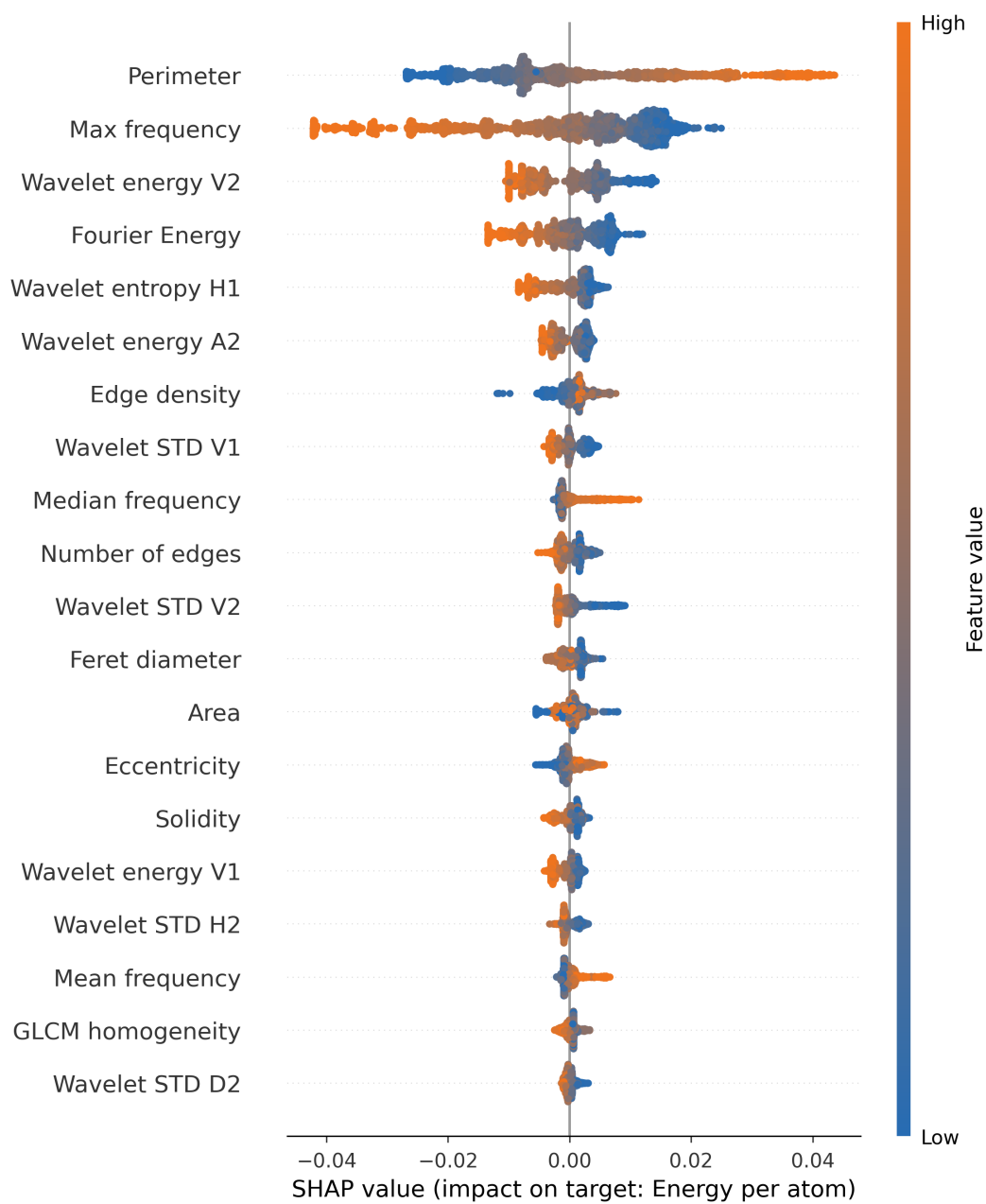


Fig. B.13 SHAP value summary trained on LightGBM for the energy per atom target

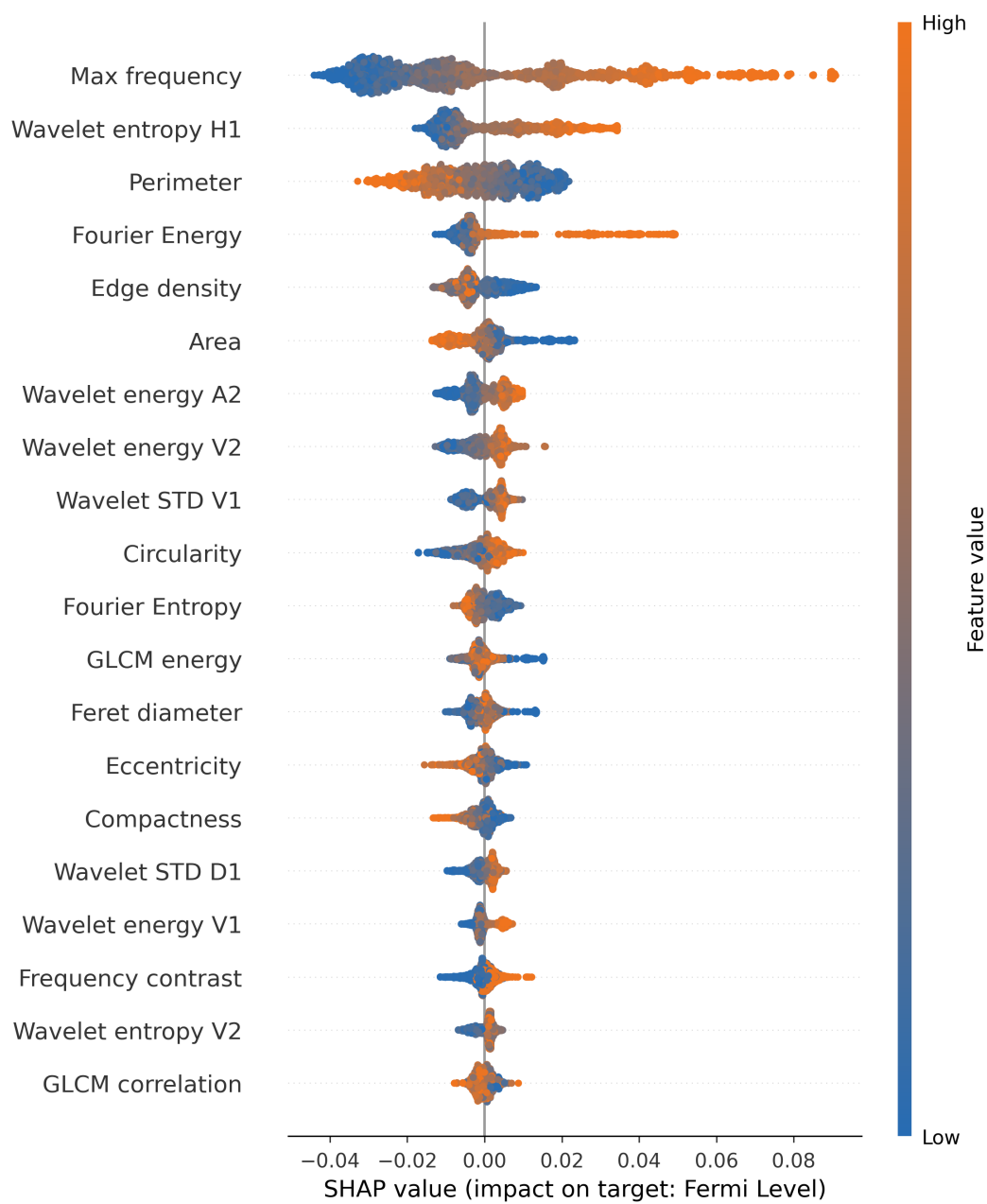


Fig. B.14 SHAP value summary trained on LightGBM for the Fermi level target

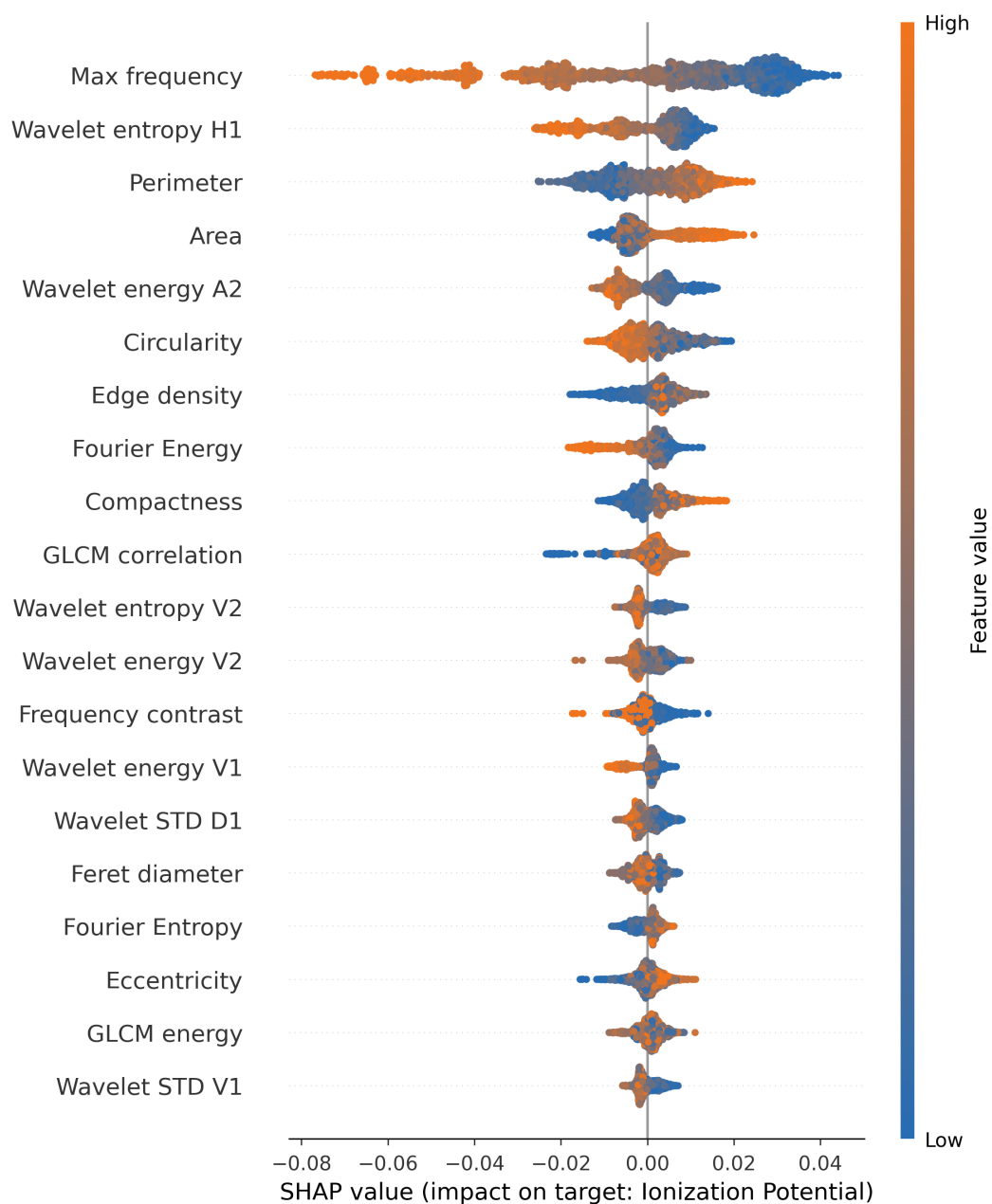


Fig. B.15 SHAP value summary trained on LightGBM for the Ionisation potential level target



Fig. B.16 SHAP value summary trained on LightGBM for the bandgap target

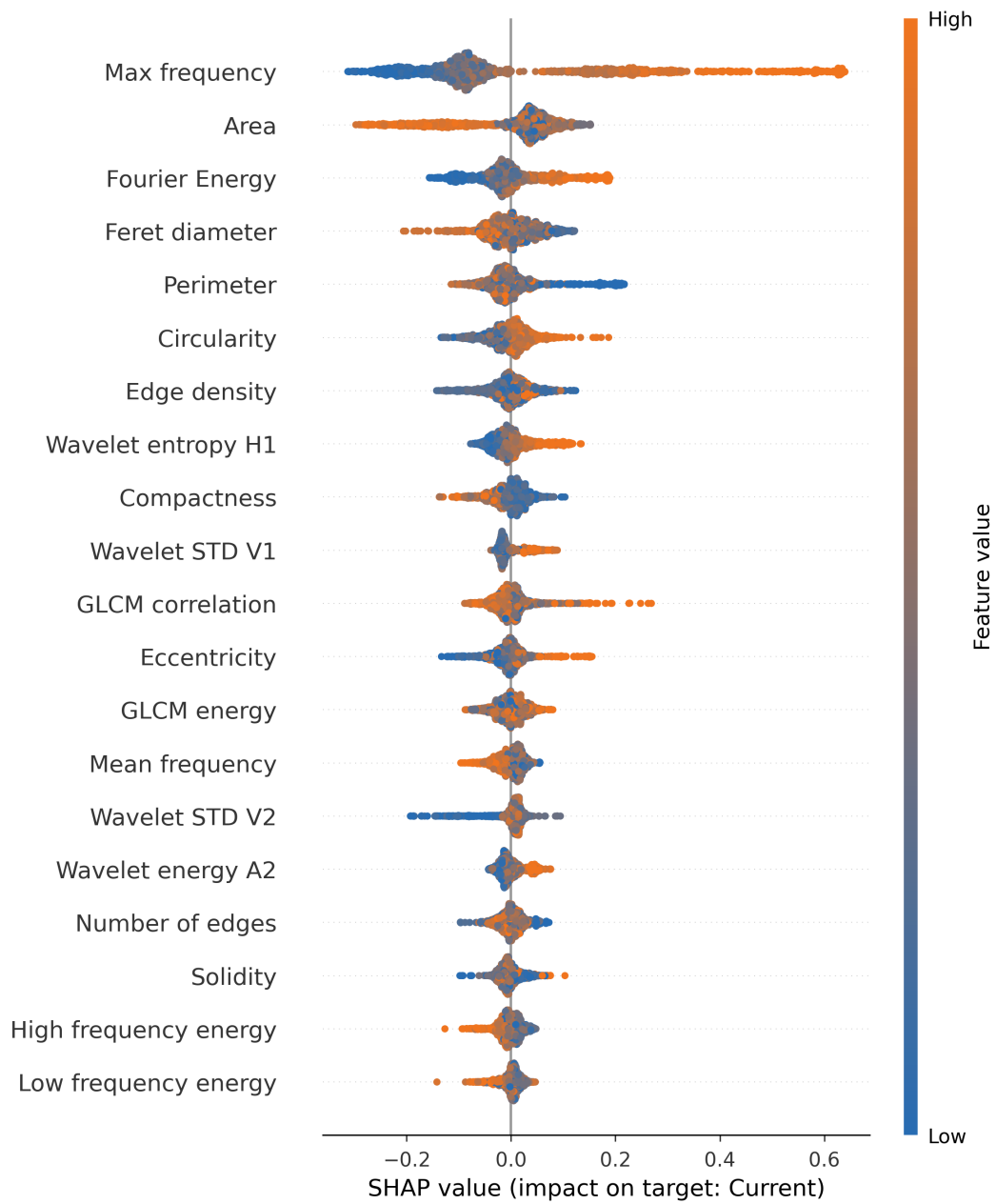


Fig. B.17 SHAP value summary trained on LightGBM for the current target

B.2 Learning the Device Physics of Organic Photovoltaics: Neural Operators, Morphology Clustering and Generative Modelling

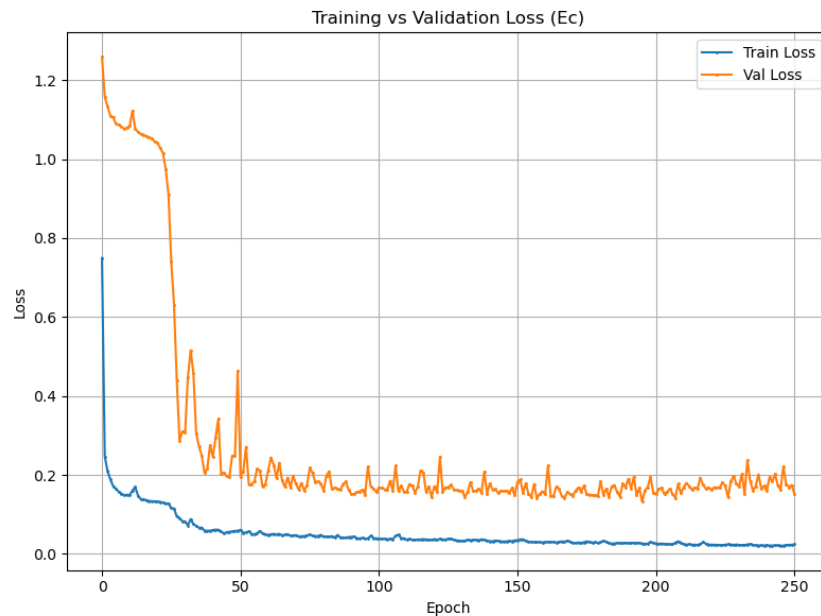


Fig. B.18 Training vs validation curve for the Ec Target

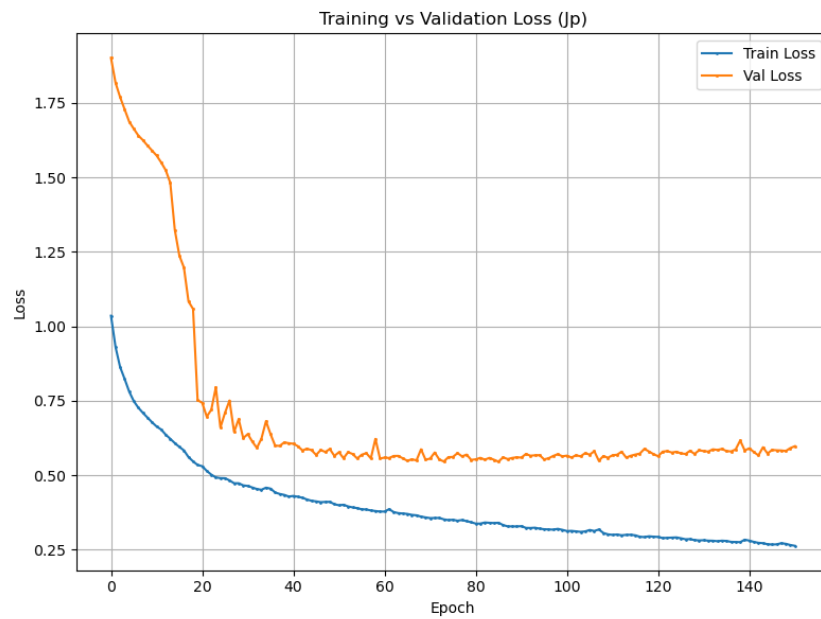


Fig. B.19 Training vs validation curve for the J_p Target

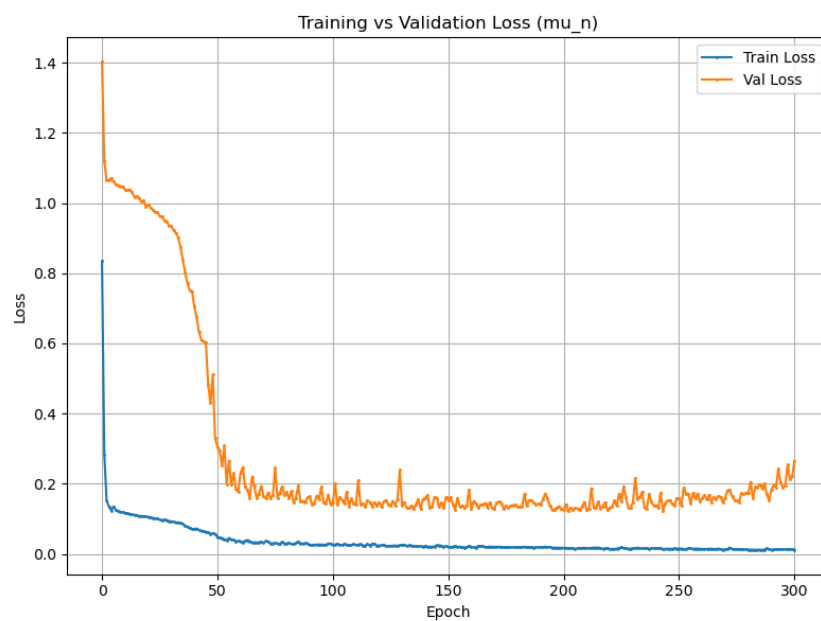


Fig. B.20 Training vs validation curve for the μ_n Target

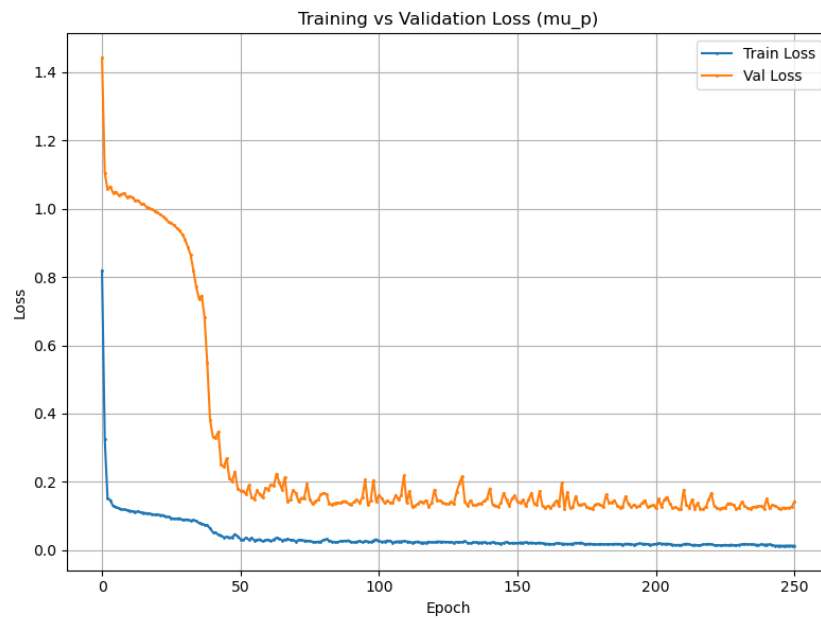


Fig. B.21 Training vs validation curve for the μ_p Target

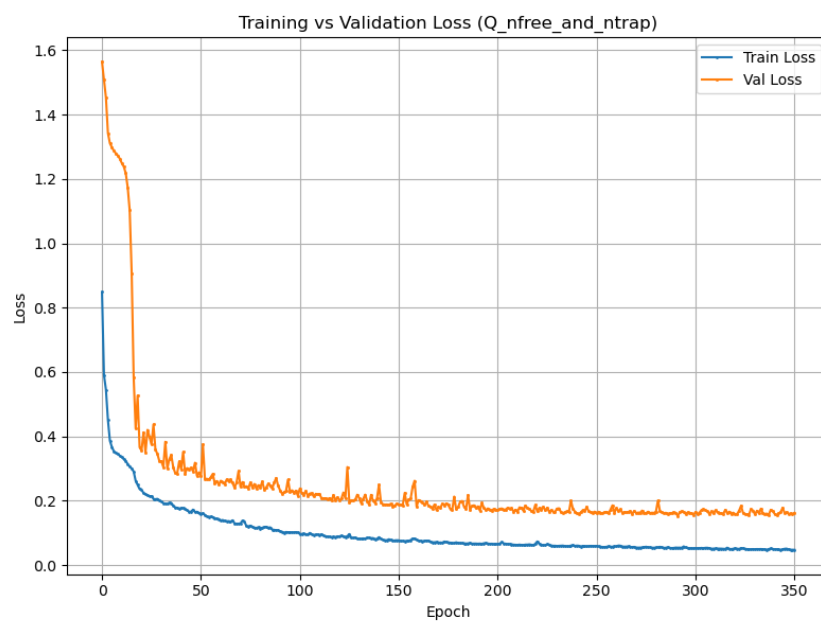


Fig. B.22 Training vs validation curve for the $Q_{nfree\&ntrap}$ Target

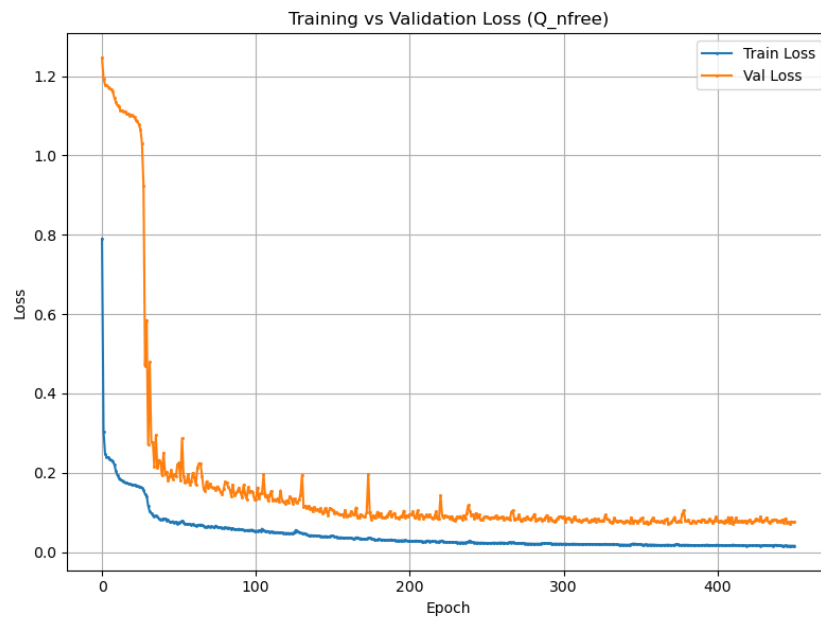


Fig. B.23 Training vs validation curve for the Q_{nfree} Target

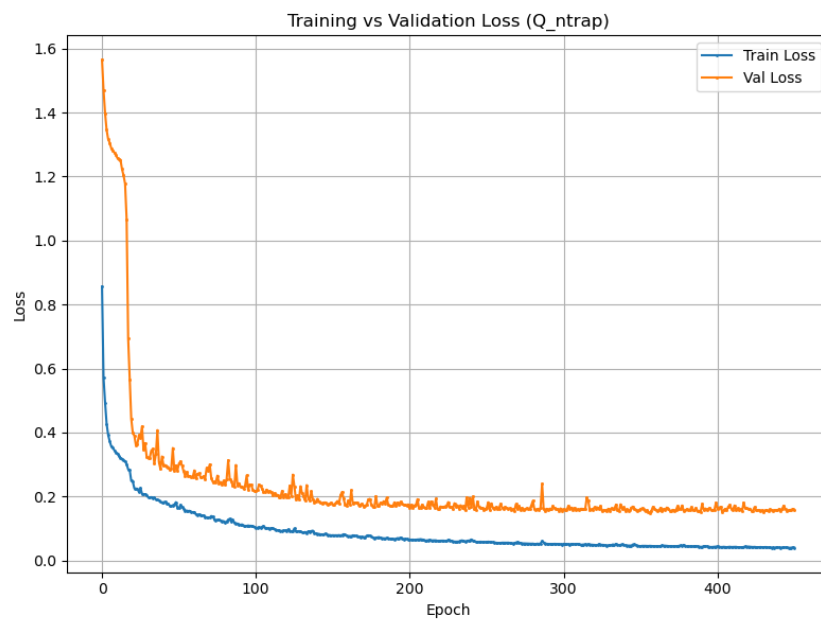


Fig. B.24 Training vs validation curve for the Q_{ntrap} Target

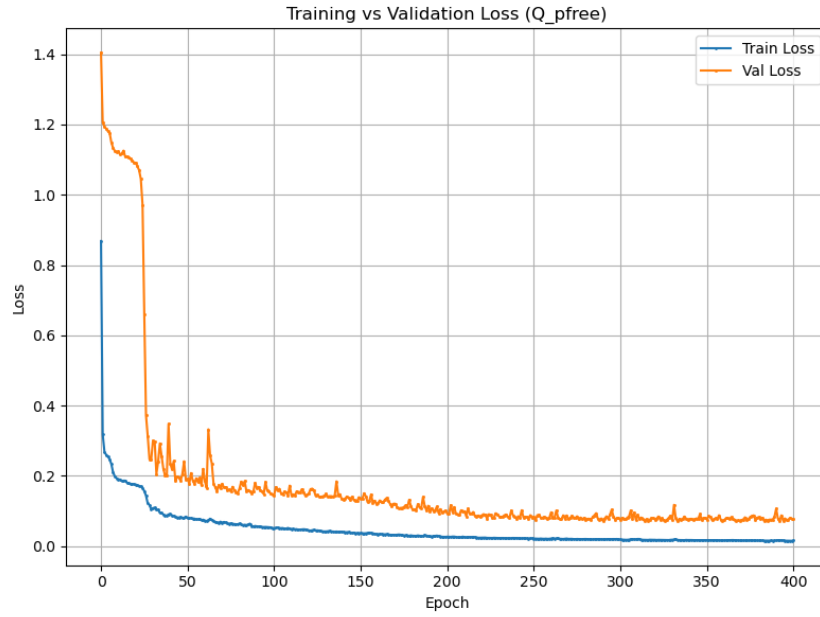


Fig. B.25 Training vs validation curve for the Q_{pfree} Target

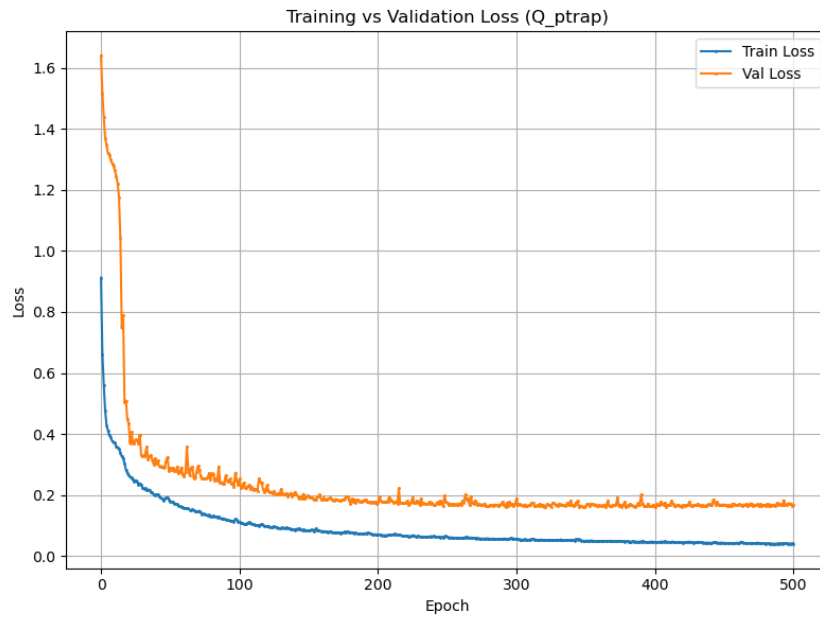


Fig. B.26 Training vs validation curve for the Q_{ptrap} Target

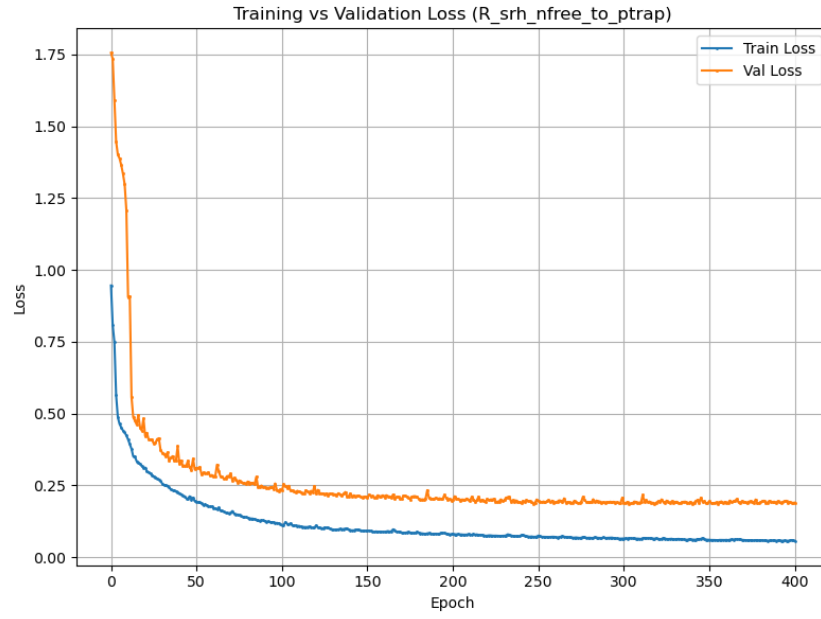


Fig. B.27 Training vs validation curve for the Recombination rate from n_{free} to p_{trap} Target

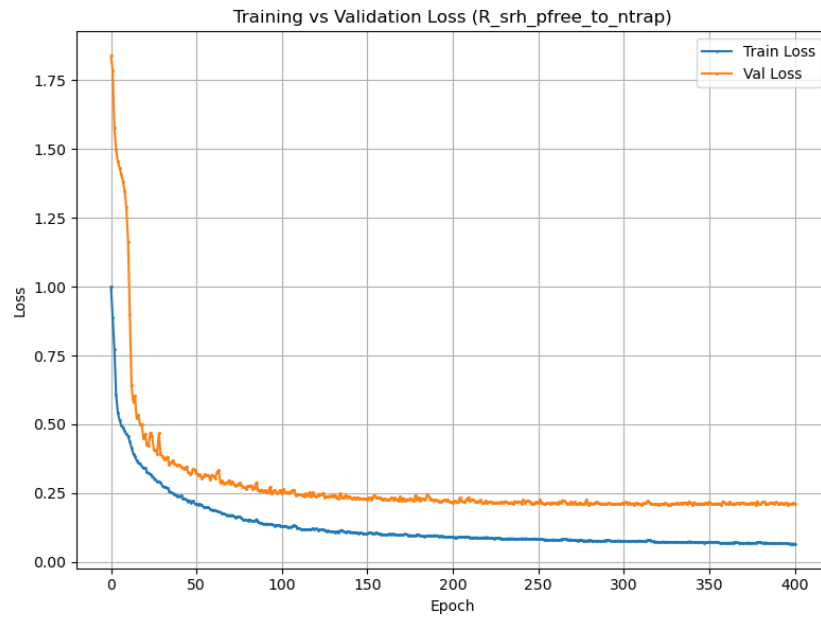


Fig. B.28 Training vs validation curve for the Recombination rate from p_{free} to n_{trap} Target

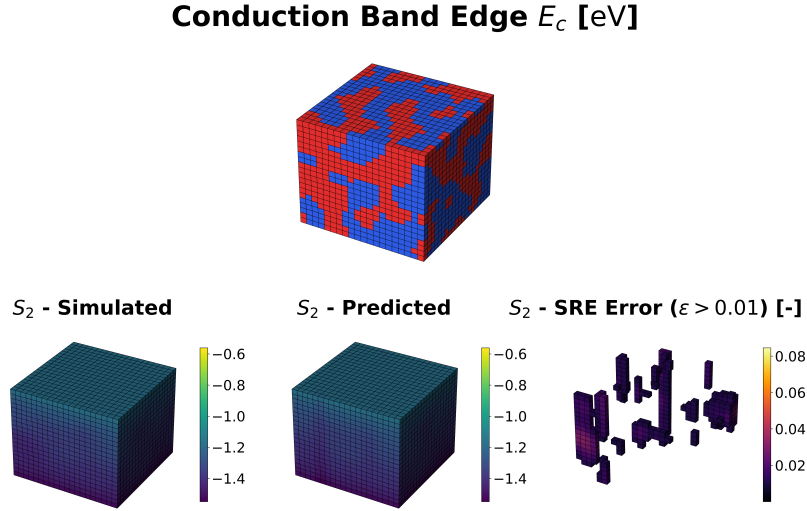


Fig. B.29 Comparison between simulated and predicted E_c for a representative three-dimensional morphology.

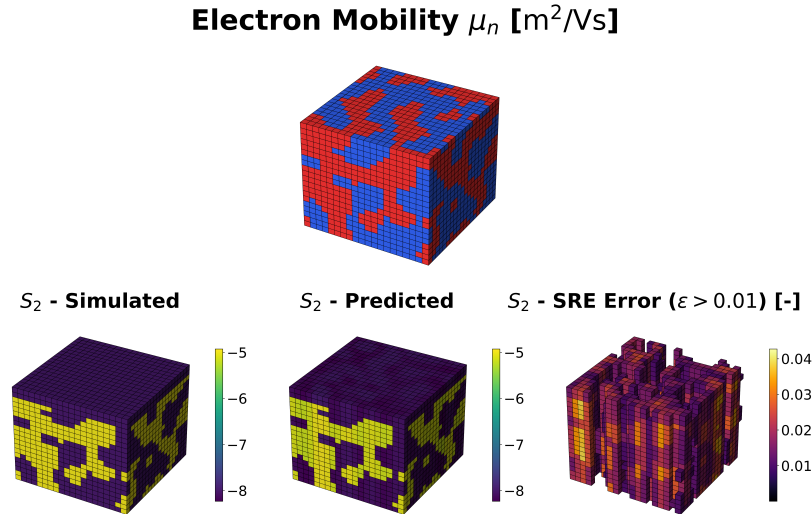


Fig. B.30 Comparison between simulated and predicted μ_n for a representative three-dimensional morphology.

Free Electron Density $Q_{n, \text{free}}$ [m^{-3}]

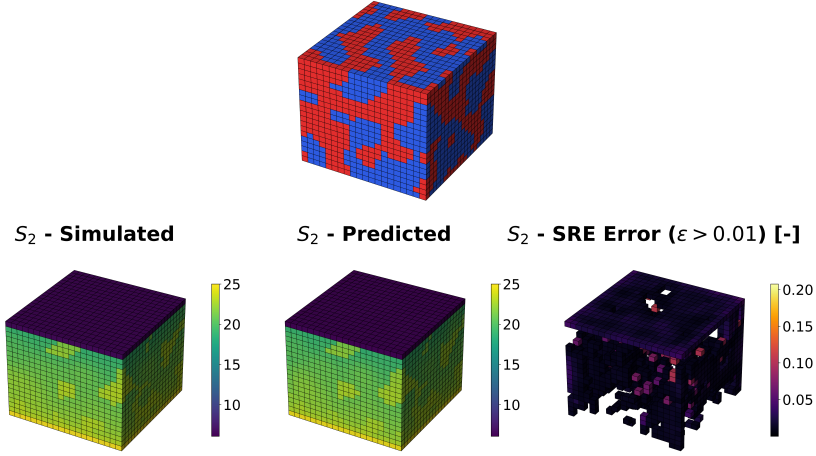


Fig. B.31 Comparison between simulated and predicted n_{free} for a representative three-dimensional morphology.

Trapped Electron Density $Q_{n, \text{trap}}$ [m^{-3}]

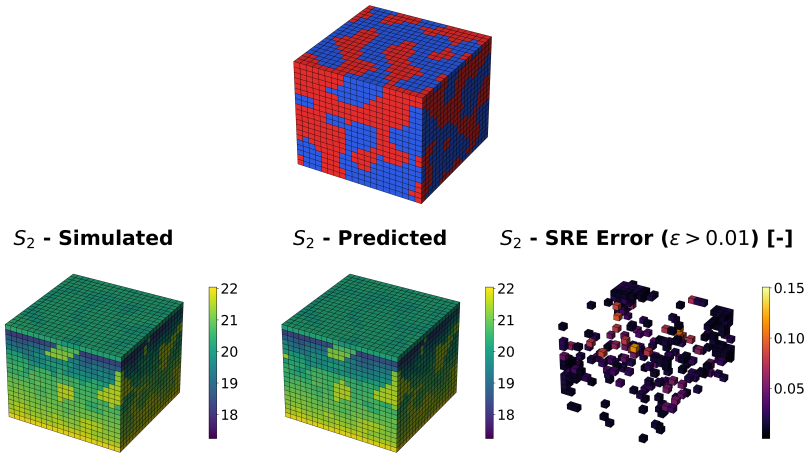


Fig. B.32 Comparison between simulated and predicted n_{trap} for a representative three-dimensional morphology.

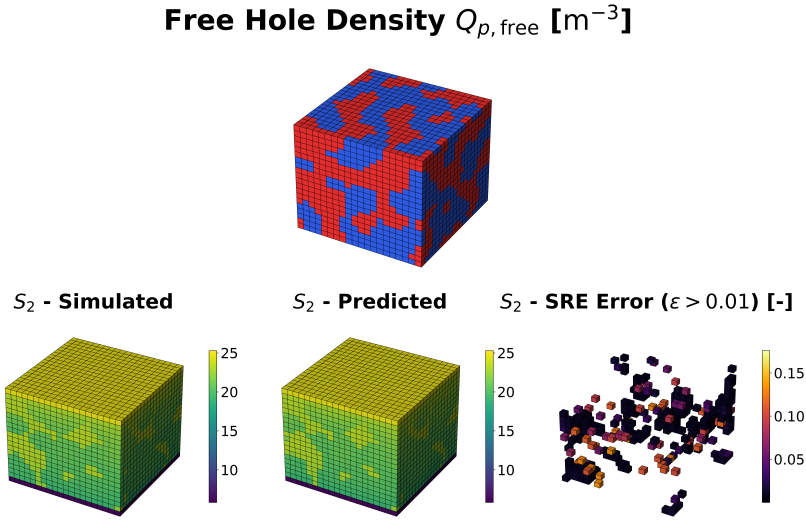


Fig. B.33 Comparison between simulated and predicted p_{free} for a representative three-dimensional morphology.

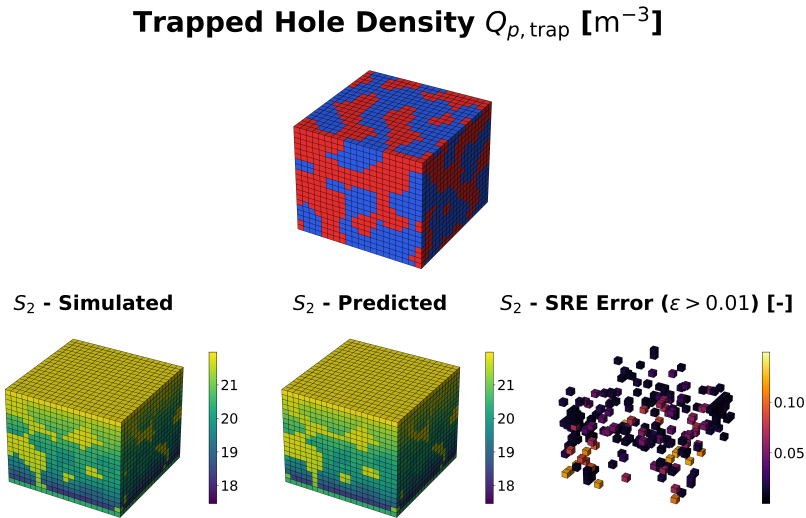


Fig. B.34 Comparison between simulated and predicted p_{trap} for a representative three-dimensional morphology.

SRH Recombination Rate $R_{\text{SRH}, p_{\text{free}} \rightarrow n_{\text{trap}}} [\text{m}^{-3} \text{s}^{-1}]$

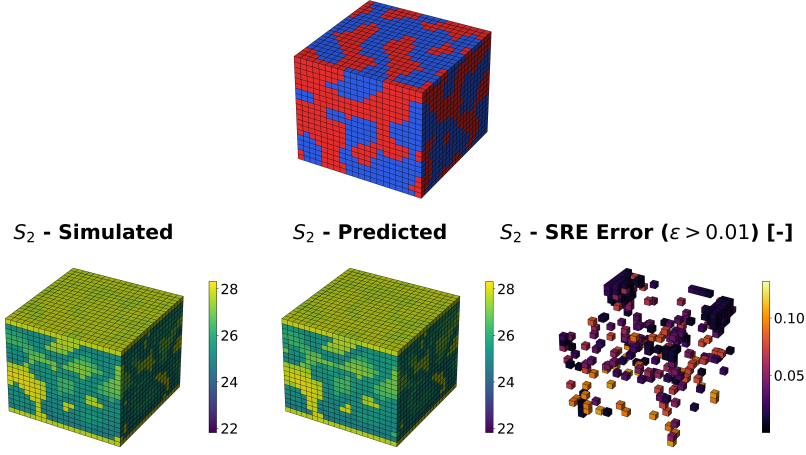


Fig. B.35 Comparison between simulated and predicted $R_{\text{srh}_p \text{free}_n \text{trap}}$ for a representative three-dimensional morphology.