Universidad Politécnica
de Madrid

Escuela Técnica Superior de Ingenieros Informáticos

Master in Data Science

Master Thesis

Interpretable Hybrid Models for Tabular Data: Integrating Kolmogorov–Arnold and Convolutional Networks via Synthetic Images

Author: VICTOR GIOVANNY MONDRAGON RUIZ

Madrid, July - 2025

This Master Thesis has been deposited in ETSI Informáticos de la Universidad Politécnica de Madrid.

Master Thesis
Master in Data Science

Title: Interpretable Hybrid Models for Tabular Data: Integrating Kolmogorov–Arnold and Convolutional Networks via Synthetic Images
July - 2025

Author: VICTOR GIOVANNY MONDRAGON RUIZ
Supervisor: RAUL GARCIA CASTRO and MANUEL CASTILLO CARA

ETSI Informáticos
Universidad Politécnica de Madrid

Resumen

Los enfoques en modelos híbridos que integran datos tabulares con representaciones en forma de imágenes sintéticas han despertado un interés considerable en los últimos años, impulsados por la ambición de aprovechar tanto el razonamiento espacial como el simbólico para mejorar el rendimiento y la interpretabilidad. Los trabajos iniciales exploraron pipelines basados puramente en Redes Neuronales Convolucionales (CNNs), utilizando imágenes sintéticas generadas mediante métodos como IGTD [1], REFINED [2] o TINTO [3], bajo la premisa de que las CNNs pueden explotar correlaciones espaciales que no son fácilmente evidentes en los formatos tabulares en bruto. Sin embargo, estas soluciones a menudo sacrificaban información crucial sobre los procesos de toma de decisiones del modelo. Investigaciones más recientes han evolucionado hacia arquitecturas híbridas que combinan representaciones espaciales derivadas de CNNs con entradas tabulares procesadas mediante Perceptrones Multicapa (MLPs) tradicionales o redes alternativas, buscando no solo mejoras predictivas, sino también vías hacia la explicabilidad [4]–[6]. En paralelo, las Kolmogorov–Arnold Networks (KANs) han surgido como un reemplazo innovador de los MLPs, ofreciendo un rendimiento competitivo con menos hiperparámetros y una interpretabilidad inherente gracias a sus mecanismos basados en splines [7], [8].

En este contexto, esta tesis propone una nueva arquitectura híbrida que combina CNNs entrenadas sobre imágenes sintéticas con componentes KAN para modelar datos tabulares, formando así un sistema híbrido CNN–KAN flexible e interpretable. Para evaluar la viabilidad de esta arquitectura, se realizaron experimentos sobre cuatro conjuntos de datos de tamaño pequeño a mediano, abarcando tanto tareas de regresión como de clasificación. Los conjuntos de datos se seleccionaron a partir de estudios previos para asegurar la disponibilidad de particiones establecidas y resultados base, permitiendo así una comparación centrada en las innovaciones de modelado en lugar de en la preparación de datos. Los pasos de preprocesamiento se limitaron al tratamiento de valores faltantes y a la normalización, acompañados de breves análisis de correlación para investigar posibles relaciones entre las dependencias de las variables y los resultados del modelo híbrido. Se probaron tres métodos de generación de imágenes sintéticas —IGTD, REFINED y TINTO— en cada conjunto de datos, aumentando progresivamente en complejidad y ofreciendo representaciones espaciales diversas de las características tabulares.

La arquitectura propuesta se estructuró en torno a un diseño de doble rama, donde una rama procesaba los datos tabulares en bruto mediante un componente KAN denominado `kan_branch`, mientras que la otra utilizaba una serie de bloques CNN para procesar las imágenes sintéticas, denominada `cnn_branch`. Las salidas de ambas ramas se concatenaban y alimentan a una capa `final_kan`, diseñada para ar-

monizar los espacios de características y capturar patrones complementarios. Para refinar aún más el equilibrio entre ambas ramas, se exploraron cuatro estrategias de concatenación, orientadas a controlar la relevancia de cada rama y a mejorar el rendimiento predictivo. Inspirado en el pipeline de entrenamiento de la librería `pykan`, esta se adaptó para dar soporte a la estructura híbrida, empleando el optimizador LBFGS en modo full-batch y utilizando el Error Cuadrático Medio (MSE) como función de pérdida. Esta configuración, aunque inicialmente diseñada para tareas de regresión, mostró resultados sólidos incluso en tareas de clasificación, validando así su versatilidad. Teniendo así una sola neurona para el resultado final del modelo híbrido CNN-KAN.

Más allá del rendimiento, esta tesis otorga un énfasis significativo a la interpretabilidad y explicabilidad. Los componentes KAN proporcionaron puntuaciones simbólicas de las características, lo que permitió una inspección directa de las contribuciones de cada variable dentro del sistema híbrido. Paralelamente, se empleó Grad-CAM [9] para visualizar atribuciones espaciales en la rama CNN, revelando aspectos como la influencia de "píxeles extra" adyacentes a regiones críticas de las características, fenómeno especialmente evidente en las imágenes generadas con TINTO. Sin embargo, se observó que arquitecturas CNN más simples, en ocasiones, tenían dificultades para centrarse en las características relevantes cuando se enfrentaban a imágenes sintéticas de alta resolución que contenían numerosos píxeles no informativos. Una contribución innovadora de este trabajo fue la introducción del Global Feature Score, una métrica unificada que combina las puntuaciones normalizadas de las características provenientes de la `kan_branch` con las salidas de Grad-CAM de la `cnn_branch`, ponderadas según la relevancia de cada rama. Esta métrica busca generar un ranking integrado de importancia de características, proporcionando una visión holística del proceso de toma de decisiones en modelos híbridos.

Los resultados empíricos demostraron que la arquitectura híbrida CNN–KAN logró de manera constante un rendimiento competitivo o superior en comparación con los modelos CNN o KAN independientes en la mayoría de los conjuntos de datos, confirmando el potencial de combinar razonamiento simbólico y espacial. La metodología basada en Grid Searchs permitió identificar configuraciones óptimas de hiperparámetros, manteniendo el enfoque en la simplicidad del modelo, ya que el híbrido logró resultados sólidos con relativamente pocos hiperparámetros —especialmente en los componentes KAN. Aunque ciertos métodos de generación de imágenes sintéticas, como REFINED, ofrecieron ventajas en conjuntos de datos específicos, no surgió un único método que fuese superior de forma universal, lo que subraya la importancia de realizar experimentación específica para cada conjunto de datos. Cabe destacar que el análisis de relevancia reveló que, pese a los esfuerzos por equilibrar las contribuciones, la rama CNN solía dominar la influencia predictiva, aunque estrategias de concatenación específicas (particularmente las Estrategias 1 y 2) ofrecieron mejoras moderadas para distribuir mejor dicha relevancia entre las ramas.

Los análisis de interpretabilidad confirmaron el valor de combinar múltiples técnicas explicativas, revelando señales coincidentes entre la relevancia simbólica procedente de la vía KAN y las percepciones espaciales derivadas de Grad-CAM. En varios casos, las mismas características emergieron como influyentes en ambas modalidades, reforzando la confianza en las conclusiones del modelo híbrido, sin embargo los patrones detectados eran específicos para cada conjunto de datos. Además, el Global Feature Score mostró una alineación prometedora con los resultados individuales de

Grad-CAM, lo que sugiere que capta de manera precisa la importancia de las características, en lugar de introducir discrepancias arbitrarias.

En conclusión, esta tesis validó con éxito la arquitectura híbrida CNN–KAN como un enfoque eficaz para modelar datos tabulares y sus representaciones sintéticas, aportando tanto un alto rendimiento como transparencia sobre la relevancia de las características. El marco metodológico propuesto —que incluye el procedimiento de entrenamiento personalizado, las estrategias de concatenación y el Global Feature Score— constituye una base sólida para futuras investigaciones en sistemas neuronales híbridos. Las líneas de investigación futura incluyen la aplicación de este marco a conjuntos de datos más grandes y complejos, la experimentación con métodos de optimización alternativos más allá de LBFGS, la integración de otras herramientas de interpretabilidad como SHAP [10] o LIME [11], y el desarrollo de mecanismos avanzados de concatenación que permitan equilibrar mejor las contribuciones de las ramas, manteniendo o incluso mejorando el rendimiento predictivo. Este trabajo constituye así un paso significativo hacia arquitecturas neuronales que no solo sean potentes, sino también interpretables y transparentes en sus procesos de toma de decisiones.

Abstract

The integration of symbolic and spatial reasoning has emerged as a compelling direction in tabular data modeling, driven by the promise of hybrid neural architectures that combine the strengths of both domains. This thesis proposes a novel hybrid framework that merges Convolutional Neural Networks (CNNs), trained on synthetic images of tabular data, with Kolmogorov–Arnold Networks (KANs), an interpretable alternative to traditional Multi-Layer Perceptrons. The hybrid CNN–KAN model aims not only to improve predictive performance but also to provide transparent insights into how features and architectural branches contribute to its decisions. The study employs a structured methodology across multiple small- to medium-sized datasets, testing various synthetic image generation techniques—IGTD, REFINED, and TINTO—and evaluating different concatenation strategies to balance the influence of each model component.

Experimental results demonstrate that the CNN–KAN hybrid consistently outperforms standalone CNN and KAN models in most cases, achieving high performance with relatively few hyperparameters, especially in the KAN components. Beyond metrics, the thesis introduces the Global Feature Score, a unified interpretability metric that combines symbolic relevance from KANs with spatial insights from Grad-CAM. This score effectively aligns with known model behaviors, reinforcing confidence in its explanatory power. The findings confirm that hybrid architectures can deliver not only superior predictive outcomes but also meaningful interpretability, positioning CNN–KAN models as a promising avenue for future research in explainable tabular data analysis.

Contents

1 Introduction and Problem Definition **1**
1.1 Background and Motivation 1
1.2 Related Work 2
1.2.1 From Tabular Representations to Hybrid Architectures 2
1.2.2 Concatenation Strategies in Hybrid Architectures 3
1.2.3 KANs: Interpretable Alternatives to MLPs 3
1.2.4 Explainability in CNNs via Grad-CAM 5
1.2.5 Formulas for Interpretability and Explainability in Hybrid Models 5
1.3 Research Objectives and Hypotheses 6
1.4 Overview of the Proposed Approach 6
1.5 Structure of the Document 7

2 Methodology **9**
2.1 Dataset Description 9
2.2 Descriptive Analysis of Correlation Structures 10
2.3 Synthetic Image Methods 10
2.3.1 IGTD: Grayscale Grid Projection 11
2.3.2 REFINED: Color-Encoded Feature Mapping 11
2.3.3 TINTO: Sparse Point-Based Layout 12
2.4 Hybrid Neuronal Network Architecture 13
2.4.1 KAN Components 14
2.4.2 CNN Branch Design 14
2.5 Concatenation Strategies 15
2.5.1 Strategy 1: Bottleneck Projection 16
2.5.2 Strategy 2: Scaled Concatenation 17
2.5.3 Strategy 3: Gating with Dynamic Weighting 17
2.5.4 Strategy 4: Cross-Attention 17
2.6 Training Strategy 18
2.7 Evaluation Metrics 18
2.8 Feature Relevance Extraction 19
2.8.0.1 Interpretability of the KAN Branch 19
2.8.0.2 Explainability using Grad-CAM 19
2.8.0.3 Branch Relevance 19
2.9 Feature Relevance Extraction 20
2.10 Unified Global Feature Score 20
2.11 Hyperparameter Search Strategy and Execution Pipeline 21

3 Experimental Setup and Performance Evaluation **25**

3.1 Experimental Setup 25
3.1.1 Environment 26
3.1.2 Generation of Synthetic Images 26
3.1.3 Training Configuration and Metric Tracking 27
3.1.4 Hyperparameter Search 28
3.1.4.1 KAN Models 28
3.1.4.2 CNN Models 28
3.1.4.3 CNN-KAN Hybrid Models 28
3.1.4.4 Concatenation Strategies 29
3.2 Exploratory Correlation Results 30
3.3 Preprocessing 30
3.4 Detailed Results Using IGTD-Based Images 30
3.4.1 Evaluation on Treasury with IGTD 31
3.4.2 Evaluation on Puma with IGTD 32
3.4.3 Evaluation on Forex with IGTD 33
3.4.4 Evaluation on Wall Robot with IGTD 35
3.5 Baseline Performance Across Synthetic Image Methods 37
3.5.1 Baseline Results on the Treasury 37
3.5.2 Baseline Results on the Puma 38
3.5.3 Baseline Results on the Forex 39
3.5.4 Baseline Results on the Wall Robot 39
3.6 Key Insights and Observations 40

4 Explainability and Interpretability Analysis 41
4.1 Feature Relevance Analysis 41
4.1.1 Visual Examples of Grad-CAM Outputs 42
4.1.2 Comparative Visualization of Feature Relevance Scores 43
4.1.3 Summary of Top Features Across Image Methods 45
4.1.3.1 Feature Relevance in CNN Models Across Datasets 45
4.1.3.2 Feature Relevance in CNN-KAN Hybrid Models Across Datasets 46
4.2 Branch Contribution Analysis in Hybrid Models 47
4.3 Concatenation Strategy Comparison 48
4.3.1 Comparison of Concatenation Techniques for Treasury 49
4.3.2 Comparison of Concatenation Techniques for Puma 49
4.3.3 Comparison of Concatenation Techniques for Forex 51
4.3.4 Comparison of Concatenation Techniques for Wall Robot 52
4.4 Cross-Architecture Evaluation and Feature Analysis 54
4.4.1 Summary Results for Treasury 54
4.4.2 Summary Results for Puma 55
4.4.3 Summary Results for Forex 56
4.4.4 Summary Results for Wall Robot 56
4.5 Global Feature Score Analysis for Best Hybrid Models 57
4.5.1 Global Feature Analysis for Treasury 58
4.5.2 Global Feature Analysis for Puma 58
4.5.3 Global Feature Analysis for Forex 60
4.5.4 Global Feature Analysis for Wall Robot 61
4.6 Summary of Insights and Discussion 62

5 Conclusion and Future Work 65

5.1 Conclusion 65
5.2 Limitations and Future Work 67

References **69**

Annex **72**

Chapter 1

Introduction and Problem Definition

This chapter introduces the motivation and background for exploring hybrid neural architectures that combine symbolic and spatial reasoning for tabular data analysis. It first reviews relevant developments in the literature, including methods that transform tabular data into synthetic images for Convolutional Neural Network (CNN) processing, the emergence of Kolmogorov–Arnold Networks (KANs) as interpretable alternatives to traditional Multi-Layer Perceptrons (MLPs), the use of explainability techniques like Grad-CAM, and the lack of a global feature score for the proposed hybrid model. Building on these insights, the chapter then articulates the research objectives, defines the specific problem this thesis addresses, and presents the core hypotheses guiding the experimental investigations. To help orient the reader, it also includes an overview of the proposed methodology summarizing the main steps of this thesis in a high-level diagram. The chapter concludes by outlining the structure of the document to guide the reader through the forthcoming chapters.

1.1 Background and Motivation

The use of hybrid models that process synthetic representations of tabular data has gained increasing attention in recent years, driven by the search for improved performance and new avenues for interpretability [2], [5], [12]. Many works have proposed transforming tabular data into structured images—such as via IGTD (Interpretable Grid Tabular-to-Image Data), REFINED (Representation of Features as Images with Neighborhood Dependencies), or TINTO (Transforming Tidy Data into Image for Classification) methods—to enable Convolutional Neural Networks (CNNs) to exploit spatial correlations and local dependencies even in non-visual domains [13], [14]. Some research has focused exclusively on relying on these synthetic images, leveraging CNNs as end-to-end predictors, while others have sought to integrate both the synthetic image data and the original tabular features into hybrid models that aim to capture complementary patterns across modalities [4], [6].

To customize the interplay between modalities, various concatenation strategies have been introduced, offering flexible ways to merge outputs from different branches and balance their contributions [15]–[17]. However, a persistent limitation remains: de-

spite their impressive predictive capabilities, many of these hybrid architectures fall short in delivering transparent explanations of how the model reaches its predictions or which original features most directly drive its decisions [5]. Understanding not just what the model predicts but why it predicts it is critical, particularly for applications in high-stakes domains such as finance, healthcare, or engineering [5], [12].

Simultaneously, KANs have emerged as a novel solution for replacing MLPs in tabular tasks [7]. KANs offer several advantages: they not only often outperform MLPs while requiring fewer hyperparameters, but they also bring inherent interpretability through their spline-based architecture, which allows precise quantification of how each input feature influences the output [8]. This capability may prove critical for unlocking the explainability of hybrid models, enabling us to dissect the relevance of each component—and potentially leading to better performance and more transparent insights [6], [12].

Motivated by these gaps, this thesis explores whether combining CNNs trained on synthetic images with KANs can deliver hybrid models that are not only high-performing but also explainable in terms of both feature-level relevance and branch-level contributions. The work builds upon recent advances in synthetic image methods, i.e., IGTD [1], REFINED [2] and TINTO [3], [18], hybrid fusion strategies [5], and explainability tools like Grad-CAM [9] to investigate new pathways toward transparent and effective neural architectures for tabular data.

1.2 Related Work

This section reviews foundational contributions that motivate the design of the proposed hybrid architecture. It begins by examining models that convert tabular data into synthetic images to leverage the spatial learning capabilities of CNNs, followed by a discussion of more recent hybrid architectures that integrate both tabular and visual modalities [6]. Next, it explores explores concatenation strategies as a means of fusing diverse neural components within such hybrid systems. The third subsection focuses on KANs, an emerging alternative to MLPs offering competitive performance and native interpretability in symbolic domains. Finally, the section closes with a discussion of Grad-CAM, a widely used technique for explaining CNN-based decision-making, particularly relevant when dealing with spatial representations of tabular data [5].

1.2.1 From Tabular Representations to Hybrid Architectures

Transforming tabular data into synthetic images has emerged as a promising approach to enable the application of CNNs in non-visual domains. Since CNNs excel at capturing local spatial dependencies, several methods have been developed to encode feature relationships into 2D grids. One of the earliest examples is IGTD [1], which assigns features to grid positions based on pairwise similarity, producing grayscale images where semantically related attributes appear nearby. Building upon this concept, REFINED [2] employs a pre-trained interpretable model to guide feature placement and encoding, resulting in more structured and semantically aligned image layouts. TINTO [3], in turn, uses dimensionality reduction techniques such as PCA or t-SNE for layout generation and applies a Gaussian blur to simulate feature influence across spatial neighborhoods.

These tabular-to-image transformations have demonstrated good results across diverse domains. For instance, TINTO has been successfully applied in Bluetooth-based indoor localization by encoding signal variation patterns as images [18]. Other studies have explored hybrid designs that combine IGTD and REFINED within a single processing pipeline [1], highlighting the flexibility of these visual encodings. Similarly, methods like Table2Image [13] and NCTD [14] transform tabular data into structured visual formats that CNNs can process directly, often achieving high classification performance.

However, these purely image-based models may lose critical interactions found in the original tabular space. To mitigate this, recent hybrid architectures integrate symbolic and spatial reasoning by combining CNN-based image branches with dedicated tabular pathways [6]. For example, LM-IGTD [19] introduces a dual-branch design that processes both synthetic images and raw features. These hybrid models aim to capture complementary information across modalities and improve performance while enhancing interpretability [4], [5]. Nevertheless, they introduce new challenges related to architectural complexity and the coordination of contributions from each component—challenges that motivate the exploration of effective fusion mechanisms, such as concatenation strategies [6], [12].

1.2.2 Concatenation Strategies in Hybrid Architectures

As hybrid models evolve to integrate multiple modalities–such as tabular features and synthetic images–effectively merging information from distinct branches becomes essential [12]. One of the most adopted fusion mechanisms is *concatenation*, which enables the joint representation of features extracted from different sub-networks. This technique has shown strong empirical performance in diverse hybrid architectures by preserving complementary information while allowing flexible downstream integration [6].

Several studies have demonstrated the versatility of this approach. Malayeri and Khodabakhshi [15] used concatenated convolutional blocks to combine fuzzy recurrence features, improving blood pressure estimation. Oyelade et al. [16] applied a twin CNN structure with concatenation and hybrid optimization to combine modality-specific representations for breast cancer images. Similarly, Wang et al. [17] incorporated concatenated multi-scale CNN features into a transformer model via cross-attention, boosting performance on hyperspectral image classification.

These examples highlight the adaptability of concatenation techniques across architectures and data modalities. However, as hybrid systems grow in complexity, achieving both performance and interpretability becomes more challenging [6]. This motivates the exploration of neural components offering built-in interpretability, as well as methods to enhance post-hoc explainability while remaining compatible with such concatenation strategies [12].

1.2.3 KANs: Interpretable Alternatives to MLPs

KANs emerge as a solution to the challenge of balancing interpretability and performance in hybrid models [6]. These networks have recently gained attention as interpretable and efficient alternatives to traditional MLPs for tabular and symbolic tasks. Rather than relying on fixed activation functions and dense weight matri-

ces, KANs employ learnable spline-based one-dimensional functions along network edges. This design, grounded in the Kolmogorov–Arnold representation theorem, enables compact representations and facilitates direct inspection of how features influence the model's behavior [7]. Figure 1.1 visually contrasts the architectures and mathematical formulations of MLPs and KANs, highlighting how KANs shift learnable nonlinearity from nodes to edges.

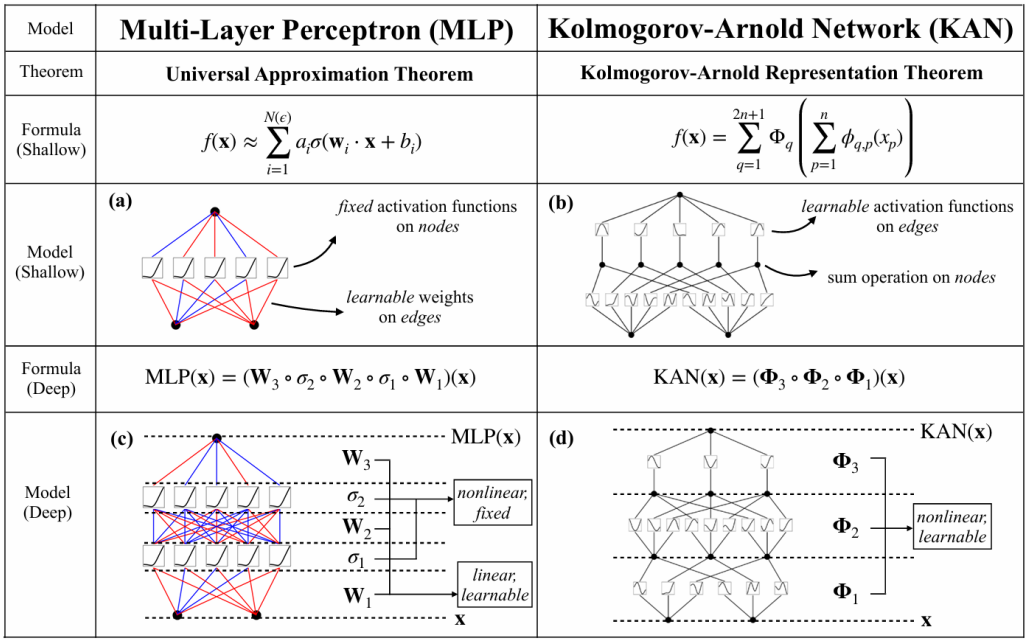


Figure 1.1: Comparison between traditional Multi-Layer MLP architectures and KANs. While MLPs employ fixed nonlinearities on nodes and learn weights on edges, KANs shift nonlinearity to learnable spline functions on edges and perform summation at nodes. Figure extracted from Liu et al. [7].

A key advantage of KANs lies in their native interpretability: each input feature's relevance can be quantified through edge-level activations, offering a granular understanding of its contribution to predictions [8]. Unlike post-hoc explainability methods, this built-in mechanism provides immediate insights into the internal decision process, making KANs highly suitable for transparent hybrid systems. Furthermore, their spline-based architecture typically requires fewer hyperparameters than MLPs, simplifying the optimization process and reducing computational overhead.

While their initial applications focused on structured tabular data, recent studies have explored the integration of KANs into visual pipelines. For instance, KANICE [20] combines KAN modules with convolutional operations to learn from image-based inputs, while PoolKANNeXt [21] adapts KAN blocks into pooling-based CNN architectures for biomedical image classification. These efforts demonstrate the architectural flexibility of KANs and their compatibility with image-based or concatenated systems, reinforcing their potential for multimodal hybrid architectures [6], [12].

Nevertheless, even with interpretable components like KANs, models that incorporate CNN branches—especially those processing synthetic images—require dedicated explainability techniques to elucidate spatial reasoning [5]. This need motivates the incorporation of tools such as Grad-CAM to complement symbolic relevance insights

and add explainability.

1.2.4 Explainability in CNNs via Grad-CAM

Continuing within the context of hybrid models and explainability, the decision-making process of CNN components—especially when applied to synthetic tabular images—has been the focus of considerable research due to their inherently opaque nature. One widely adopted solution is Gradient-weighted Class Activation Mapping (Grad-CAM) [9], which visualizes the most influential regions in an input image by computing the gradient of the output prediction with respect to convolutional feature maps. This produces class-specific heatmaps without requiring model modifications, making it suitable for CNN-based architectures that operate on synthetic tabular images [5].

Grad-CAM has proven especially effective in hybrid pipelines that process structured data through image transformations. Studies such as LM-IGTD [19], Table2Image [13], and NCTD [14] regularly apply Grad-CAM to validate and interpret CNN behavior, often alongside symbolic components or tabular pathways. These use cases demonstrate Grad-CAM's critical role in bridging the gap between visual encoding and human-understandable explanations in hybrid learning systems. These techniques, while powerful for visualizing CNN-driven insights, still leave open the question of whether it is possible to centralize feature relevance into a unified ranking that reflects how branches in hybrid models contribute to predictions.

1.2.5 Formulas for Interpretability and Explainability in Hybrid Models

Several recent works have proposed methods for interpreting hybrid models, each leveraging different mechanisms to attribute relevance or generate explanations. Wang and Lin [22] design a collaborative hybrid framework where an interpretable model corrects or verifies black-box predictions based on confidence thresholds, but without merging feature-level contributions into a unified metric. Ferry et al. [4] take a theoretical approach, analyzing how to split feature spaces into regions governed by either interpretable rules or black-box functions. Their analysis relies on a PAC-Bayes bound to quantify how much of the prediction space can remain interpretable while maintaining acceptable accuracy. However, while these works rigorously formalize hybrid design principles, they focus primarily on model architecture and decision boundaries rather than producing a single quantitative feature relevance score.

Similarly, C-SHAP [23] attempts to improve explanation speed in hybrid contexts by clustering data to reduce the number of SHAP computations. Their method blends clustering with local SHAP analyses to approximate feature importance more efficiently, yet still remains confined to either black-box or interpretable branches rather than synthesizing insights across both. Across these examples, interpretability or explainability is typically achieved through either routing mechanisms that delegate data to different model parts, or through local surrogate explanations such as rule lists or SHAP values.

1.3 Research Objectives and Hypotheses

The overarching objective of this thesis is to design a flexible hybrid neural architecture that integrates KANs and CNNs to model tabular data and its synthetic image representations. The goal is not only to achieve strong predictive performance but also to provide interpretability and explainability, enabling insights into how each architectural component contributes to the model's predictions.

A key element of this work is the proposal of a novel formula that combines interpretability metrics from the KAN branch and explainability outputs from the CNN branch into a single global feature score, thereby offering a unified ranking of feature importance across the entire hybrid model.

To achieve these objectives, this thesis formulates the following hypotheses:

- **H1:** Replacing the traditional MLP component with a KAN module in hybrid architectures that also use CNNs improves interpretability and can enhance performance under appropriate configurations, while requiring fewer hyperparameters to tune compared to conventional MLPs.
- **H2:** Integrating CNN branches trained on synthetic images enhances predictive performance by capturing complementary spatial patterns that are not evident in the raw tabular inputs.
- **H3:** The performance of different tabular-to-image methods (e.g., IGTD, REFINED, TINTO) varies depending on dataset characteristics such as dimensionality, feature correlation, and sample size.
- **H4:** Interpretability tools, such as branch relevance and feature attribution analysis, can guide the design of more balanced and effective hybrid architectures by informing architectural choices and data fusion strategies.
- **H5:** Incorporating advanced concatenation strategies into hybrid models enhances predictive performance and contributes to a more balanced distribution of relevance between the KAN and CNN branches.
- **H6:** It is possible to define a formula that leverages interpretable and explainable outputs from the hybrid model to produce a coherent ranking of the contribution of each input feature, helping bridge the gap between high performance and model transparency.
- **H7:** The proposed hybrid architecture and the global feature score approach can scale effectively when applied to datasets of varying size and complexity.

These hypotheses collectively define the research problem addressed in this thesis, laying the groundwork for the experimental and analytical investigations presented in subsequent chapters.

1.4 Overview of the Proposed Approach

To provide the reader with a comprehensive understanding of the main steps addressed in this thesis, Figure 1.2 illustrates the overall workflow of the proposed hybrid modeling framework. The diagram summarizes the key components and their interconnections, including the transformation of tabular data into synthetic images,

the dual-branch architecture combining CNN and KAN pathways, the application of concatenation strategies, and the derivation of the Global Feature Score for interpretability. The figure explicitly indicates both the numbering and the order of these methodological steps, serving as a visual guide to the entire process. Further details and technical explanations of each of these steps are provided in Chapter 2.

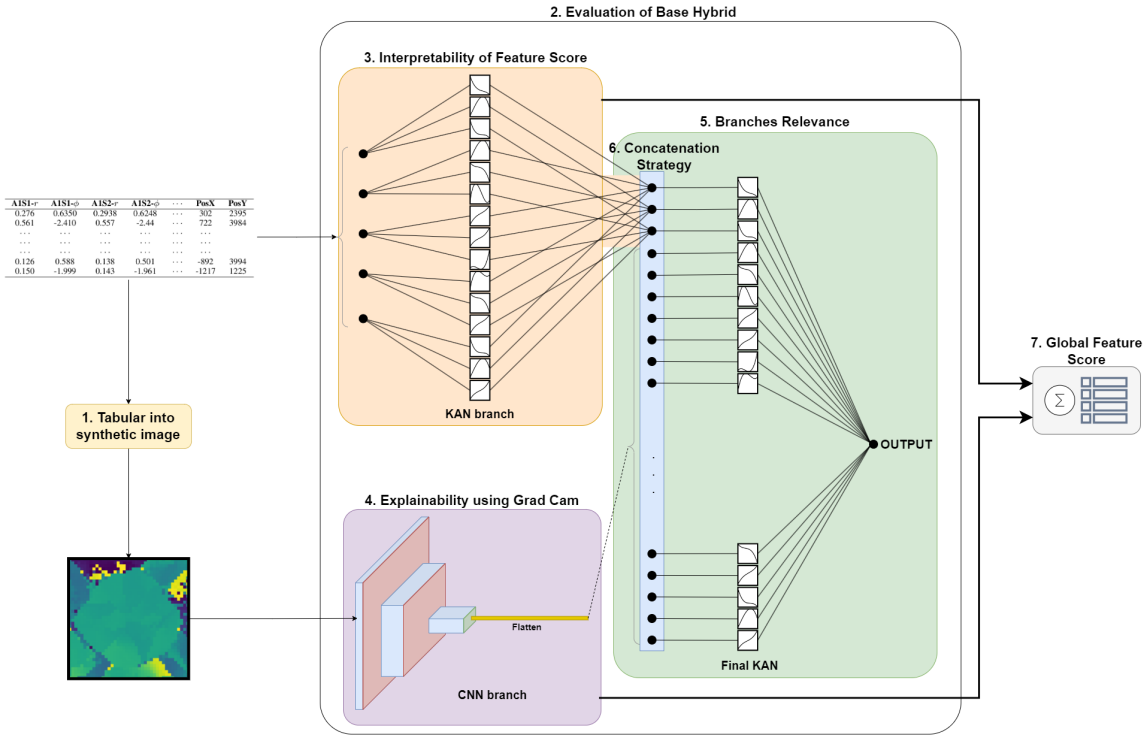


Figure 1.2: High-level overview of the methodology proposed in this thesis, summarizing the main steps, data flows, and interpretability mechanisms employed throughout this work. The steps are numbered and ordered to reflect the sequential processes detailed in Chapter 2.

1.5 Structure of the Document

The remainder of this thesis is organized as follows:

- **Chapter 1** introduces the context and motivation for this work, summarizes relevant research on hybrid architectures, defines the research objectives and hypotheses, and provides an overview of the proposed methodological approach including a high-level diagram that illustrates the main steps addressed throughout the thesis.
- **Chapter 2** presents the methodological framework, detailing the datasets, exploratory correlation analysis, synthetic image generation techniques, the proposed CNN-KAN hybrid architecture, concatenation strategies, training approaches, evaluation metrics, feature relevance extraction methods, and the formulation of the global feature score.
- **Chapter 3** describes the experimental setup and hyperparameter tuning process. It reports comprehensive performance results for standalone KAN, stan-

dalone CNN, and hybrid CNN-KAN models across different datasets and synthetic image methods, and highlights key insights regarding convergence behavior and architectural effectiveness.

- **Chapter 4** focuses on interpretability and explainability, analyzing feature and branch relevance, visualizations from Grad-CAM, comparisons across different synthetic image techniques, the effects of concatenation strategies, and presenting the global feature score analysis for the best hybrid models in each dataset.
- **Chapter 5** concludes the thesis by summarizing the main findings, discussing the limitations of the current approach, and outlining promising directions for future research in the development and explainability of hybrid neural architectures for tabular data.

Chapter 2

Methodology

This chapter describes in detail the methodological framework developed in this thesis, including the datasets and a brief exploratory correlation study, the three synthetic image generation methods, architectural design of the CNN–KAN hybrid model, concatenation strategies, training procedures, evaluation metrics, analyses of feature relevance in the hybrid components, and definition of the formula for the Global Feature Score. As introduced in Figure 1.2 in Chapter 1, the overall process consists of sequential steps ranging from synthetic image generation to the computation of the Global Feature Score. Each of these steps is explained in depth in the following sections of this chapter.

2.1 Dataset Description

To evaluate the performance and interpretability of the proposed CNN–KAN architectures, four publicly available datasets were selected from the OpenML repository.[1] These datasets were chosen to reflect a mix of regression and classification tasks, with varying levels of dimensionality and inter-feature correlation. This diversity allows for a more rigorous assessment of model generalization and interpretability under different structural and statistical conditions.

Table 2.1 provides a summary of the datasets. For ease of reference throughout the thesis, the dataset `treasury` is referred to as **Treasury**, `puma8NH` is referred to as **Puma**, `FOREX cadjpy-day-High` as **Forex**, and `wall-robot-navigation` as **Wall-Robot**. These datasets vary in number of features and samples, which makes them especially interesting for analyzing how architectural decisions—such as the inclusion of synthetic image branches—interact with data scale and complexity in hybrid models.

Each dataset represents a distinct application domain: **Puma** simulates the dynamics of a Unimation Puma 560 robot arm, **Treasury** consists of economic indicators and financial measurements, **Wall-Robot** captures ultrasound sensor readings from a robot navigating an environment, and **Forex** contains historical price data for currency markets. Notably, all datasets exclusively consist of numerical features, which ensures compatibility with both tabular and image-based representations explored in this thesis.

[1] Puma (ID 225), Treasury (ID 42367), Wall-Robot (ID 1497), Forex (ID 41839)

Dataset Name	Type	# Features	# Samples
Treasury	Regression	15	671
Puma	Regression	8	5242
Forex	Binary Classification	10	1173
Wall-Robot	Multi-class Classification	24	3491

Table 2.1: Overview of the datasets used in the experiments.

2.2 Descriptive Analysis of Correlation Structures

To characterize the structural properties of the tabular datasets before model training, a descriptive analysis was conducted, quantifying aspects such as average absolute feature correlation, silhouette scores from hierarchical clustering, local feature continuity, dimensionality, and pairwise interaction variance. These metrics, computed on standardized numerical data, were normalized and combined into a single correlation suitability score using the following formula:

$$\begin{aligned}\text{correlation\_score} = {} & 0.35 \cdot \text{norm\_corr} \\ & + 0.26 \cdot \text{norm\_silhouette} + \\ & + 0.13 \cdot \text{norm\_dim} + \\ & + 0.13 \cdot \text{norm\_continuity} + \\ & + 0.13 \cdot \text{norm\_interaction}\end{aligned}$$

Datasets were then categorized as having **Low** (score belows 0.33), **Medium** (score exceeds 0.33 but belows 0.66), or **High** (score exceeds 0.66) correlation structure, with the hypothesis that higher correlation might favor CNNs by enhancing spatial feature patterns in synthetic images, whereas lower correlation might benefit symbolic modeling in KANs. Although no strong predictive patterns ultimately emerged linking these scores to final model performance, the correlation study remains a useful reference for interpreting later feature relevance results and for contextualizing the annex tables of per-feature correlations cited in Chapters 3 and 4.

2.3 Synthetic Image Methods

To enable the application of CNNs to tabular datasets, this section describes three synthetic image generation methods: IGTD, REFINED, and TINTO, all implemented using the `TINTOlib` Python library [24], [25]. These methods transform feature vectors into structured spatial layouts or image-based representations, with the goal of preserving relationships among features in a format suitable for convolutional processing. Each method encodes specific assumptions about feature organization—such as spatial continuity, pairwise proximity, or localized density—which can significantly influence CNN learning dynamics and the interpretability of the resulting models. Further details regarding the precise parameter configurations and hyperparameter choices used for these images method in this work are provided in Section 3.1.2.

2.3.1 IGTD: Grayscale Grid Projection

The IGTD [1] method projects tabular features into a fixed-size $n \times n$ 2D square grid, producing grayscale images where each cell encodes the normalized value of a specific feature. The spatial layout is determined through task-aware placement, aiming to preserve pairwise or structural relationships among features. Among the three generation methods explored in this work, IGTD produces the most compact synthetic images—typically with a single grayscale channel and low resolution—resulting in the fastest CNN training times.

For datasets with fewer features than required grid cells, the remaining positions are filled with constant white pixels and labeled as `Ex1`, `Ex2`, etc., to preserve layout symmetry. These placeholder cells are included in relevance analyses. For example, as shown in Table 2.1, the Treasury dataset contains 15 features, requiring one additional cell to complete a 4×4 grid layout. An illustration of the resulting synthetic image for the Treasury dataset is shown in Figure 2.1.

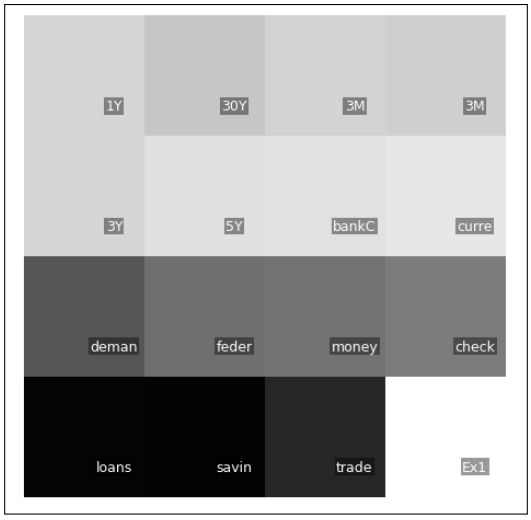


Figure 2.1: Example of a synthetic image generated via IGTD for the Treasury dataset. Labeled cells correspond to features; `Ex1` is an extra cell due to the fixed 4×4 layout.

2.3.2 REFINED: Color-Encoded Feature Mapping

REFINED [2] transforms tabular data into color-encoded 2D synthetic images by placing features in a spatial grid based on a t-SNE projection, which preserves pairwise distances and neighborhood structure. This results in a task-aware layout where related features are placed close together. Unused cells, labeled as `Ex1`, `Ex2`, etc., are automatically inserted and may appear anywhere in the grid, unlike IGTD where they are typically confined to a corner.

Figure 2.2 shows the REFINED-generated image for the Forex dataset. Notably, features like `Bid_Volume` and `Ask_Volume` are placed in prominent, separated locations, while the extra cells (`Ex1`–`Ex6`) are scattered across the grid. This layout improves semantic locality and may enhance the CNN's ability to learn spatial feature interactions.

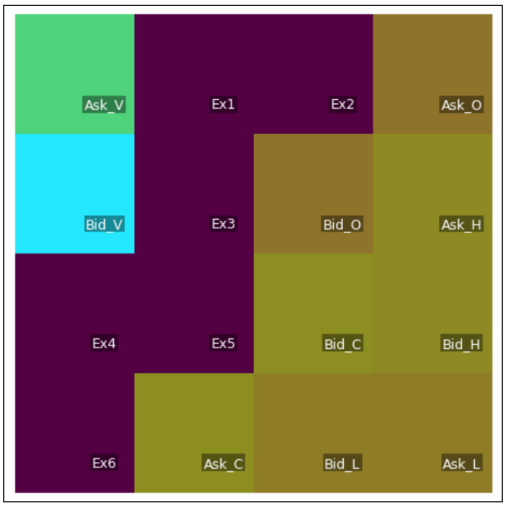


Figure 2.2: REFINED-generated synthetic image for the Forex dataset. Each cell is color-encoded; placeholders (`Ex1`–`Ex6`) appear in the middle to fill the grid.

2.3.3 TINTO: Sparse Point-Based Layout

TINTO [3], [18] generates sparse synthetic image representations by projecting features into a fixed-size grid using Principal Component Analysis (PCA) followed by a gravity-based mapping. Each feature is assigned a specific coordinate in the grid and represented as an isolated point of intensity, creating a minimal image layout that emphasizes spatial relationships over pixel density. However, due to the dimensionality reduction performed by PCA, some features may be projected onto similar or overlapping coordinates, leading to multiple features sharing the same pixel location in the synthetic image.

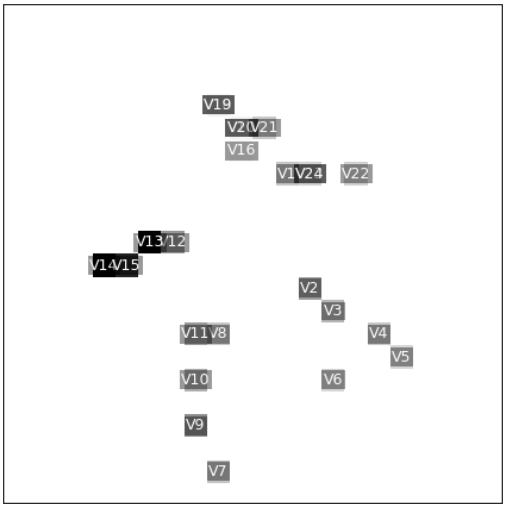


Figure 2.3: Example of a TINTO-generated image for the Wall Robot dataset. Features are represented as sparse dots in fixed positions across the image grid.

As shown in Figure 2.3, the synthetic image for the Wall Robot dataset displays scattered pixel activations in black, with each feature ideally occupying a unique and distant position within the grid. This dispersed layout enables the CNN to interpret feature interactions based on their relative spatial arrangement rather than dense

textures or tiling. However, some features inevitably share the same pixel location, for instance, variables `V14` and `V15` overlap spatially in this example, meaning that any relevance signal at that pixel aggregates information from both features.

2.4 Hybrid Neuronal Network Architecture

The proposed CNN-KAN architecture integrates symbolic and visual reasoning by combining KANs and CNNs into a unified model inspired by Manuel's hybrid architecture [6]. As illustrated in Figure 2.4, the architecture consists of three key components: a `kan_branch` that directly processes raw tabular data, a CNN branch that transforms synthetic image representations into visual feature maps, and a `final_kan` module that merges the outputs of both branches through concatenation to produce the final prediction. This design aims to leverage the expressive capacity of CNNs for capturing spatial patterns while retaining the interpretability and symbolic capabilities inherent to KANs. All variants of the architecture produce a single scalar output neuron, supporting both regression and classification tasks.

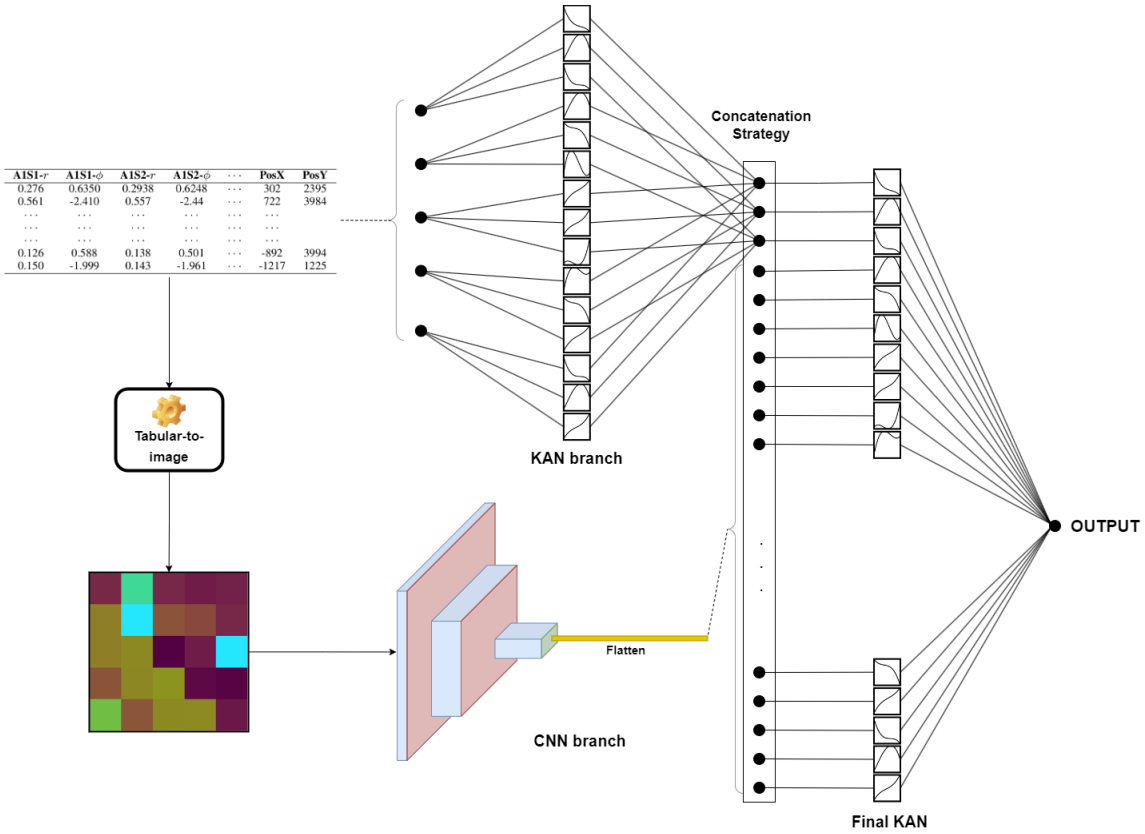


Figure 2.4: Hybrid model architecture combining a KAN branch and a CNN branch derived from a tabular-to-image transformation. The output tensors from both branches are concatenated before feeding into the final KAN component to produce the prediction. (Own elaboration.)

2.4.1 KAN Components

The KAN components of the proposed hybrid architecture are implemented using the official `pykan` library[2]. This library provides a flexible framework for constructing symbolic neural networks based on the Kolmogorov–Arnold representation theorem, where multivariate functions are expressed as compositions of univariate spline-based functions. Leveraging this framework, two distinct KAN modules are incorporated into the hybrid architecture: `kan_branch` and `final_kan` (see Figure 2.4).

The `kan_branch` operates directly on the raw tabular input, transforming it into a lower-dimensional symbolic representation through learned spline activations. Meanwhile, the CNN branch processes synthetic images to extract spatial features. The outputs of both branches are concatenated along the feature axis, forming a joint representation that serves as input to the `final_kan` module, which produces the final prediction. This dual-branch design facilitates modular learning from distinct data modalities and enables the analysis of how symbolic and visual representations interact to influence model decisions.

Both `kan_branch` and `final_kan` were implemented as single-hidden-layer KANs, aligning with the architectural recommendations of the `pykan` authors. A consistent grid resolution was applied across both modules and treated as a hyperparameter during grid search to explore the optimal degree of spline flexibility. Among configurable parameters, particular attention was given to the regularization term `lambda`, which governs the smoothness of the spline functions and is critical for avoiding overfitting while preserving the capacity to model nonlinear relationships.

Model training for the KAN components was performed using the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS) optimizer—a quasi-Newton method that is well-suited for full-batch learning and provides efficient convergence with second-order updates [26]. This choice follows the recommendations provided by the `pykan` library, where LBFGS paired with the Mean Squared Error (MSE) loss has demonstrated excellent results, particularly in structured tabular data scenarios. Notably, even for classification tasks, this approach proved effective in the experiments conducted in this thesis.

Through the integration of `pykan`, the hybrid architecture gains not only strong modeling capacity but also native interpretability, as feature relevance can be directly extracted from the spline-based representations learned within the KAN components.

2.4.2 CNN Branch Design

To develop an effective CNN branch, an initial exploratory analysis was conducted to evaluate the effect of architectural depth, activation functions, and normalization layers on the stability and informativeness of CNN outputs. The goal was to ensure that configurations not only stabilize the output—preventing excessively large activations—but also preserve enough informative content from the synthetic images to support effective integration with the KAN component, avoiding excessive magnitudes that could disrupt or break downstream processing. Insights from this analysis informed the definition of the `cnn_blocks` used in this work.

The CNN branch architecture is depicted in Figure 2.5. Each modular `cnn_block`

[2] `https://github.com/KindXiaoming/pykan`

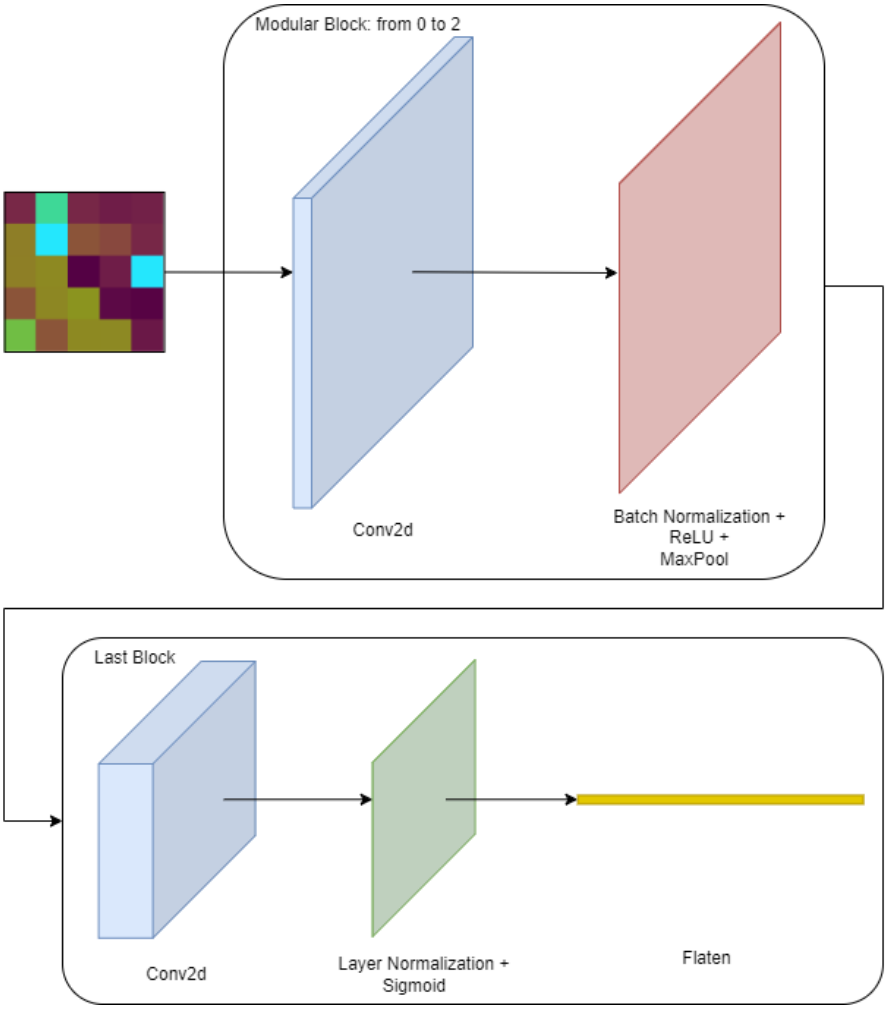


Figure 2.5: Architecture of the CNN branch composed of modular convolutional blocks. (Own elaboration.)

comprises a convolutional layer followed by Batch Normalization, a ReLU activation, and a 2D max-pooling operation—components visually represented in the figure as the red rectangular block. These blocks can be stacked to increase the representational capacity of the network. To guarantee bounded outputs suitable for concatenation with the KAN branch, the final convolutional block adopts a distinct structure: it consists of a convolutional layer, followed by Layer Normalization and a Sigmoid activation, depicted as the green rectangular block in Figure 2.5. A flattening operation then converts the resulting feature maps into a one-dimensional vector for subsequent fusion.

2.5 Concatenation Strategies

Initial experiments revealed a significant imbalance in the hybrid architecture: the output of the CNN branch frequently dominated the fused representation, overshadowing contributions from the tabular-based KAN pathway and diminishing feature relevance insights from the symbolic component. To mitigate this effect and achieve a more balanced integration of modalities, four distinct concatenation strategies were

designed and evaluated. Each strategy introduces a different mechanism for combining the visual and symbolic representations, varying both in architectural complexity and in their capacity to modulate branch relevance and enhance predictive performance.

As illustrated in Figure 2.4, the concatenation strategy sits at a critical juncture in the architecture, merging the outputs of the KAN and CNN branches before feeding them into the `final_kan` module for prediction. Table 2.2 summarizes the approximate computational cost of each concatenation strategy.

Strategy	**Approx. Computational Cost**
1. Bottleneck Projection	$\mathcal{O}(n)$
2. Scaled Concatenation	$\mathcal{O}(n)$
3. Gating with Dynamic Weighting	$\mathcal{O}(n \cdot h)$
4. Cross-Attention	$\mathcal{O}(n^2)$ or higher

Table 2.2: Approximate computational cost of each concatenation strategy. Here, n denotes the total size of the concatenated feature vectors (from the KAN and CNN branches), and h denotes the hidden layer size used in the gating MLP.

In computational terms, the strategies range from simple linear projections to more sophisticated mechanisms like gating or cross-attention, whose costs can scale quadratically depending on the dimensions involved. For clarity, is defined the key symbols used throughout this section as follows:

- n_{cnn}: dimensionality of the CNN branch output vector
- n_{kan}: dimensionality of the KAN branch output vector
- n_{input}: original dimensionality of the tabular input data
- n: dimensionality of the concatenated feature vector after fusion, typically $n = n_{\text{cnn}} + n_{\text{kan}}$

Empirically, the low-complexity strategies required notably less training time than their medium- or high-complexity counterparts. Below, each strategy is described in detail, including its theoretical computational cost and practical implications.

2.5.1 Strategy 1: Bottleneck Projection

In this baseline approach, the CNN output tensor, typically of high dimensionality n_{cnn}, is passed through a linear projection layer that reduces it to a lower dimension d. This produces a compressed embedding that better matches the scale of the KAN output tensor and prevents visual features from overwhelming the joint representation. The projected CNN tensor and the KAN output tensor are then concatenated along the feature axis before being fed into the final KAN module.

Regarding complexity, the projection involves a matrix multiplication of size $n_{\text{cnn}} \times d$, yielding a computational cost of approximately $\mathcal{O}(n_{\text{cnn}} \cdot d)$. Because d is chosen small, this remains computationally light, ultimately resulting in a total complexity of roughly $\mathcal{O}(n)$, where n represents the combined dimensionality of the fused feature vector after concatenation, as summarized in Table 2.2.

2.5.2 Strategy 2: Scaled Concatenation

This strategy introduces a scalar parameter `alpha` to rescale the CNN output tensor before fusion. The scaled CNN tensor retains its original dimensionality and is concatenated directly with the KAN output tensor along the feature axis. This mechanism allows dynamic adjustment of each branch's influence without altering feature dimensions, offering flexibility beyond the fixed projection used in Strategy 1.

From a complexity standpoint, the scalar multiplication is an element-wise operation, with a computational cost of roughly $\mathcal{O}(n_{\text{cnn}})$, making this method similarly lightweight. In practice, training times were comparable to Strategy 1, although tuning `alpha` required additional experimentation to achieve balanced performance. This keeps the method lightweight overall, with a total complexity of approximately $\mathcal{O}(n)$, in line with the low-complexity strategies listed in Table 2.2.

2.5.3 Strategy 3: Gating with Dynamic Weighting

This strategy employs a lightweight gating module implemented as a small MLP, which outputs an instance-specific scalar gate $\alpha \in [0, 1]$ based on input features. The final fused representation is computed via a convex combination:

$$\alpha \cdot \text{CNN} + (1 - \alpha) \cdot \text{KAN}$$

Unlike the global scaling in Strategy 2, this approach enables adaptive, per-sample weighting of the two modalities, potentially leading to more nuanced fusion.

Regarding computational cost, the gating MLP introduces additional operations of order $\mathcal{O}\big((n_{\text{kan}} + n_{\text{cnn}}) \cdot h\big)$, where h is the hidden layer size of the gating network. The convex combination itself adds linear cost $\mathcal{O}(n_{\text{kan}} + n_{\text{cnn}})$. Overall, the fusion step remains computationally moderate, resulting in a total complexity on the order of $\mathcal{O}(n \cdot h)$, where n denotes the combined feature size after concatenation. This places the strategy in the medium complexity category summarized in Table 2.2. Empirically, this strategy consistently increased training time relative to Strategies 1 and 2, reflecting its greater computational demands and optimization requirements.

2.5.4 Strategy 4: Cross-Attention

The most sophisticated strategy incorporates a cross-attention mechanism. Here, the KAN output tensor acts as the query Q, while the CNN output tensor serves as both key K and value V. Multi-head attention computes how each element of the structured representation selectively attends to regions of the visual feature space, producing an attention-weighted fused tensor that captures intricate feature-level interactions across modalities.

In terms of complexity, the core operation involves the matrix multiplication $QK^\top$, yielding computational costs around $\mathcal{O}(n_{\text{kan}} \cdot n_{\text{cnn}})$. For multi-head attention, this scales with the number of heads, potentially growing quadratically with feature dimensions. Altogether, this leads to a computational cost on the order of $\mathcal{O}(n^2)$ or higher, matching the high complexity classification indicated in Table 2.2. In practice, this strategy resulted in substantially longer training times than the other ap-

proaches, highlighting its expressiveness but also its significant computational demands.

2.6 Training Strategy

All architecture models were trained following the original `fit()` implementation provided by the authors of `pykan` library, using the LBFGS optimizer for full-batch training. Minor adjustments were made to the function to return the model state corresponding to the best validation metric and its associated epoch.

In hybrid models, which include both the `kan_branch` and `final_kan` modules, the training loop was extended to ensure both components were updated sequentially within each optimization step. Specifically, the logic from the original `fit()` function—responsible for managing grid updates, regularization, and gradient propagation—was applied first to `kan_branch` and then immediately to `final_kan`. This mirrored update schedule preserved consistency across both KAN modules, ensuring coherent learning dynamics throughout the hybrid architecture.

2.7 Evaluation Metrics

A consistent training objective was adopted for all models in this work, with the MSE serving as the loss function, regardless of whether the task was regression or classification. This choice stems from both architectural and empirical considerations. All network configurations in the hybrid framework produce a single continuous output neuron, which is naturally compatible with a regression-style loss:

$$\mathcal{L}(x, y) = \text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2. \tag{2.1}$$

For regression problems, the Root Mean Squared Error (RMSE) was selected as the primary evaluation metric, offering an interpretable measure of average prediction error in the original target units.

Although classification tasks are traditionally trained with cross-entropy loss and multi-class outputs, preliminary experiments in this study revealed that treating classification as a regression problem delivered markedly better results. This observation aligns with findings documented in the official `pykan` library notebook [3], where training with MSE and a single output neuron, combined with the LBFGS optimizer, yielded superior convergence and higher accuracy compared to cross-entropy methods. Consistent with those reports, attempts in this work to train classification models using cross-entropy led to unstable optimization and poorer performance, whereas adopting MSE provided robust and accurate outcomes across datasets.

For classification evaluation, accuracy was employed as the principal performance metric. Predictions were obtained by mapping the continuous outputs of the models to the nearest valid class label, effectively transforming the regression outputs into discrete classifications. This unified approach enabled the use of a consistent

[3] `https://github.com/KindXiaoming/pykan/blob/master/tutorials/Example/Example_4_classfication.ipynb`

loss function across both regression and classification tasks, simplifying the training pipeline while ensuring reliable performance assessment in diverse experimental settings.

2.8 Feature Relevance Extraction

To support interpretability and explainability, this thesis applies feature relevance extraction methods specifically tailored to the different components of the proposed hybrid architecture. Each method provides complementary insights, allowing for both symbolic and spatial understanding of the model's decision-making process.

2.8.0.1 Interpretability of the KAN Branch

For all standalone KAN models and the KAN components within the hybrid CNN–KAN architecture, interpretability is achieved via the native capabilities of the `pykan` library. Specifically, the `feature_score` attribute computes the cumulative importance of each input variable based on the learned spline coefficients distributed across the grid. This metric offers a direct estimate of each feature's contribution to the model's output, providing intrinsic interpretability without relying on post-hoc explanations.

2.8.0.2 Explainability using Grad-CAM

For standalone CNN models and the `cnn_branch` components in the hybrid architectures, explainability is provided via Gradient-weighted Class Activation Mapping (Grad-CAM). This technique operates as a post-hoc explanation method, analyzing the gradients flowing into the final convolutional layer to identify spatial regions that most strongly influence model predictions. The resulting heatmaps are then spatially mapped back onto the synthetic image grids created by IGTD, REFINED, and TINTO, enabling visual attribution of relevance to specific input features based on pixel positions.

While Grad-CAM effectively reveals spatial attention patterns, it requires careful interpretation depending on the image generation method. For IGTD and REFINED images, artificial pixel cells (e.g., `Ex1`, `Ex2`) are included in the analysis to assess whether non-informative regions erroneously attract model attention. In contrast, TINTO-generated images often contain sparse layouts with significant white areas and numerous artificial padding pixels, as shown in Figure 2.3. Including these extra pixels in Grad-CAM analyses can overwhelm the heatmaps and obscure meaningful patterns. Consequently, for TINTO images, this thesis restricts the analysis strictly to regions associated with real input features, excluding artificial or blank areas to ensure clearer and more interpretable results.

2.8.0.3 Branch Relevance

Beyond analyzing feature-level relevance within individual components, this thesis also quantifies the relative importance of each branch in the hybrid CNN–KAN architecture. Specifically, after training, the `final_kan` module enables extraction of a `feature_score` vector, which includes separate contributions from the symbolic and visual branches due to the concatenated input structure. By summing the

contributions attributable to the features from the kan_branch versus those from the cnn_branch, it becomes possible to compute branch-level relevance proportions. This branch-level analysis offers valuable insights into how the hybrid architecture distributes modeling responsibility between those branches.

2.9 Feature Relevance Extraction

To support interpretability and explainability, this work extracts feature relevance scores using techniques tailored to each model architecture.

For KAN models and the KAN components of CNN-KAN architectures, is leveraged the native interpretability capabilities provided by the pykan library. Specifically, the feature_score attribute computes the cumulative importance of each input variable based on the learned spline coefficients distributed across the grid. This metric offers a direct estimate of each feature's contribution to the model's output, enabling intrinsic interpretability without reliance on post-hoc explanations.

For CNN models and the cnn_branch components of hybrid architectures, was employed Gradient-weighted Class Activation Mapping (Grad-CAM), which is a post-hoc explainability method. Grad-CAM operates on synthetic images generated by IGTD, REFINED, and TINTO. Heatmaps are computed from the final convolutional layer and spatially mapped back onto the synthetic grid layout, attributing relevance scores to individual features based on their spatial positions. For IGTD and REFINED images, regions corresponding to artificial pixel cells (e.g., Ex1, Ex2) were also included in the analysis to assess the saliency of non-informative or empty regions.

On the other hand, TINTO-generated images often contain sparse structures with significant areas of white, pixels as shown in Figure 2.3. During Grad-CAM analysis, these white regions, especially the artificial padding cells, can visually dominate the heatmaps while conveying limited interpretive value due to their abundance and lack of meaningful signal. Therefore, for feature relevance analysis on TINTO images, it is limited our focus strictly to actual feature-related regions and excluded the numerous extra cells from detailed examination. This adjustment ensured clearer, more interpretable results while avoiding overwhelming noise in the relevance interpretation.

2.10 Unified Global Feature Score

While feature relevance was extracted separately from the symbolic and spatial branches of the CNN-KAN hybrid model—via feature_score in the PyKAN module and Grad-CAM maps in the CNN branch—these two sets of values operate on different numerical scales and thus cannot be compared directly. To address this limitation, this thesis proposes a method for computing a unified *Global Feature Score*, which represents the overall importance of each feature within the entire hybrid architecture. It is important to note that, for these calculations, only original features are considered; synthetic "Extra" pixels appearing in Grad-CAM maps (due to the spatial grid layout) are excluded to ensure interpretability remains linked to meaningful tabular variables.

The computation proceeds in several steps. First, the raw feature scores from both

branches are individually normalized to transform them into comparable proportions. Let $\mathbf{s}^{\text{kan}}$ denote the vector of feature scores from the `kan_branch`, and $\mathbf{s}^{\text{cnn}}$ the vector of average Grad-CAM scores from the `cnn_branch`. Their normalized forms are computed as:

$$\tilde{s}_i^{\text{kan}} = \frac{s_i^{\text{kan}}}{\sum_j s_j^{\text{kan}}} \quad \text{and} \quad \tilde{s}_i^{\text{cnn}} = \frac{s_i^{\text{cnn}}}{\sum_j s_j^{\text{cnn}}}.$$

Second, the overall contribution of each branch to the hybrid model's predictions is determined using the branch relevance scores obtained from the final KAN component of the architecture. Denote these branch contributions as w_{kan} and w_{cnn}, such that:

$$w_{\text{kan}} + w_{\text{cnn}} = 1.$$

Finally, for each feature i, the Global Feature Score is computed as a weighted sum of its normalized contributions from both branches:

$$\text{GlobalScore}_i = w_{\text{kan}} \cdot \tilde{s}_i^{\text{kan}} + w_{\text{cnn}} \cdot \tilde{s}_i^{\text{cnn}}.$$

This unified score quantifies the overall relevance of each feature in the CNN-KAN hybrid model, integrating both symbolic reasoning (interpretability) and spatial pattern recognition (explainability). By consolidating these perspectives into a single metric, it enables direct comparison of feature importance across branches and facilitates clearer interpretation of the hybrid architecture's decision-making process.

2.11 Hyperparameter Search Strategy and Execution Pipeline

To ensure a fair and consistent evaluation across all model families, a structured grid search procedure was established and applied uniformly. In this thesis, for clarity in the following chapters, the three primary model types will be referred to as follows:

- **KAN** — referring to KAN-only architectures that rely exclusively on symbolic processing of tabular data.
- **CNN** — referring to CNN-only architectures operating solely on synthetic image inputs.
- **CNN-KAN** — referring to the hybrid architectures that integrate both a CNN branch and KAN components.

The hyperparameter search for each of these model types was designed based on a combination of official library recommendations (e.g., `pykan`) and preliminary exploratory experiments. For example, the number of neurons in KAN models was bounded relative to the input feature dimensionality, while the number of convolutional blocks in CNN architectures was constrained by the resolution and characteristics of the synthetic images produced. For CNN-KAN architectures, the presence of a concatenation layer introduced specific hyperparameters—such as bottleneck sizes (Strategy 1), scaling factors (Strategies 2 and 3), and attention dimensions (Strategy 4)—that required separate investigation.

The entire grid search pipeline followed a hierarchical, dataset-specific procedure, with optimal configurations selected based on validation accuracy for classification tasks or RMSE for regression tasks. An overview of this process—including all interconnections—is depicted in Figure 2.6.

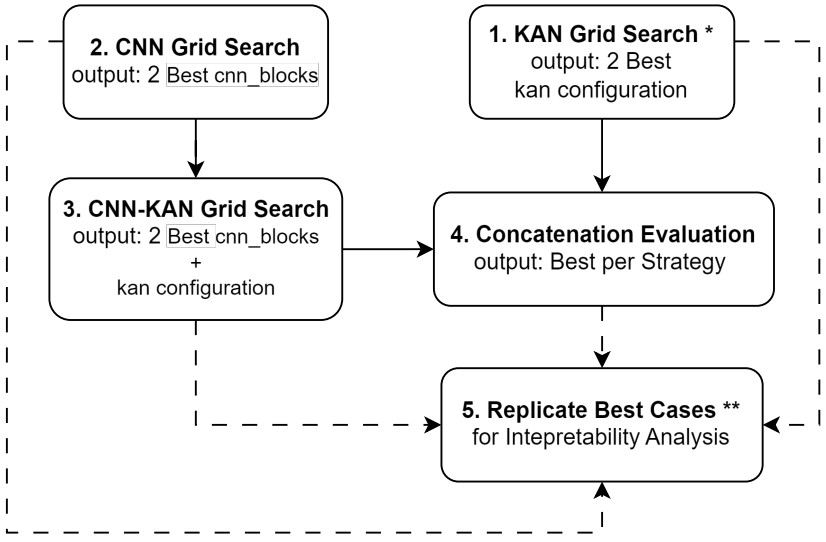


Figure 2.6: Pipeline of the hierarchical grid search methodology. The dashed lines illustrate that all final configurations are retrained for interpretability analysis. (*) Indicates that the KAN grid search is performed only once per dataset, while all other stages are repeated for each synthetic image generator. (**) Indicates Global Feature Score Formula will be applied to Best CNN-KAN case among Hybrids

Each "output" shown in the diagram represents the essential configurations passed forward to the subsequent stage. Importantly, the KAN grid search in the first step, indicated by an asterisk in the figure, was performed only once per dataset because it operates purely on tabular data, independent of synthetic images. And the two asterisks for Replicate Best Cases denotes that the Global Feature Score will be performed only for the best hybrid model.In contrast, all other stages—including the CNN, CNN-KAN, and concatenation evaluations—were repeated separately for each image method (IGTD, REFINED, and TINTO) to capture differences introduced by the visual representations.

The pipeline unfolds as follows:

1. **KAN Grid Search:** A grid search was performed on KAN models to establish a baseline using only tabular features. The top two kan configurations were retained for potential combination in later stages. This step is executed once per dataset.

2. **CNN Grid Search:** For each synthetic image generator, a separate grid search identified the two best `cnn_blocks` configurations. These serve as the foundation for designing CNN-KAN architectures.

3. **CNN-KAN Grid Search:** For each image generator and selected `cnn_blocks` configuration from step 2, a grid search was conducted on the `kan_branch` parameters, while keeping the CNN architecture fixed. In this stage, the `final_kan`

component shares the same grid settings as the `kan_branch` to ensure architectural consistency.

4. **Concatenation Strategy Evaluation:** The best two CNN-KAN configurations from step 3 were combined with the best two KAN configurations from the first step. This yielded four candidate CNN-KAN models per image generator. Each candidate was evaluated under four different concatenation strategies to balance performance and feature relevance contributions. Hence, the inputs to this step include the outputs of both step 1 (KAN-only) and step 3 (CNN-KAN), as represented by the lines converging into this stage in the diagram.
5. **Replicate Best Cases:** Finally, the best configurations across all prior steps (steps 1, 2, 3, and 4) were retrained from scratch to enable interpretability analyses, such as feature relevance scoring and branch contribution assessments. Additionally, in this stage the Global Feature Score is applied to Best Hybrid configuration from steps 3 and 4 .

Additional details on the specific hyperparameter ranges and the resulting top configurations are provided in Section 3.1.4 of Chapter 3. This staged pipeline was designed not only to ensure methodological clarity but also to manage computational resources efficiently. By isolating each search phase and reusing high-performing architectural components, it was possible to explore a tractable number of configurations while still achieving strong results. Notably, high predictive performance and interpretability were achieved with relatively few tuned hyperparameters, underscoring the practicality and scalability of the proposed approach.

Chapter 3

Experimental Setup and Performance Evaluation

This chapter outlines the experimental setup and the evaluation procedure followed throughout this work. It begins by detailing the computational resources used to train the models and generate results—both for this chapter and for Chapter 4—followed by the configuration process for generating synthetic images and the training setup adopted for conducting grid searches. A brief section on preprocessing and exploratory correlation analysis is then included to contextualize the dataset structure.

The second half of the chapter focuses on base performance evaluation. First, a detailed analysis of IGTD-based models is presented, including tables comparing performance metrics and model configurations across KAN, CNN, and CNN-KAN hybrid architectures. For each case, training and validation curves are also shown to analyze learning behavior and assess model behaviour. Finally, a summary of the top-performing base architectures using each of the three image generation methods—IGTD, REFINED, and TINTO—is provided, offering a comparative view of their convergence and effectiveness across datasets. The chapter concludes with a section highlighting key insights and overarching patterns observed across all experiments, setting the stage for the interpretability and explainability analyses presented in Chapter 4.

3.1 Experimental Setup

This section outlines the technical and methodological configuration used throughout the experiments. It begins by describing the hardware and software environment in which the models were developed and executed. Then, it presents the procedures for creating synthetic images from tabular data using three distinct methods. The training configuration adopted for all model types is described next. Then it is explained the criteria used for model selection and the tracking of task-specific metrics. The following subsections define the hyperparameter search spaces explored for KAN, CNN, and CNN-KAN hybrid models. Finally, is detailed the grid search conducted over four concatenation strategies designed to improve feature fusion and balance the contributions of the CNN and KAN branches within CNN-KAN hybrid architectures.

3.1.1 Environment

The development and initial experimentation for this work were conducted on a local workstation running Windows 11 and managed using Conda. The system was equipped with an NVIDIA GeForce RTX 4060 GPU (CUDA 12.9, Driver Version 576.52) and 32 GiB of RAM. Python 3.12.7 was used alongside PyTorch 2.5.1 for model development. The KANs were implemented using the official `pykan` library version 0.2.8[1], and synthetic image generation was performed using version 1.0.6 of the `TINTOlib` library[2], which includes the IGTD, REFINED, and TINTO modules.

Due to the greater computational demands of the REFINED and TINTO image generation methods, final training and evaluation using these methods were executed on a remote server running Ubuntu 22.04.1 LTS, hosted at the AI Supercomputing Cluster of the AI.nnovation Space IPTC-AI, Universidad Politécnica de Madrid. The system featured dual Intel® Xeon® Gold 6240R processors (24 cores @ 2.4 GHz), 192 GiB of RAM, and a dedicated NVIDIA A100-PCIE-40GB GPU (CUDA 12.4, Driver Version 550.163.01). The environment was configured to replicate the local setup, ensuring consistency in results through identical versions of Python, PyTorch, and TINTOlib.

3.1.2 Generation of Synthetic Images

Synthetic images were generated using three distinct methods—IGTD, REFINED, and TINTO—each tailored through specific parameters to align with the dataset characteristics and the learning objectives of the models. All methods leveraged a consistent random seed to ensure reproducibility. The following summarizes the main configuration settings applied for each method:

- **IGTD**:
  - `problem_type`: Set to `"regression"` for Treasury and Puma; `"supervised"` for Forex and Wall-Robot.
  - `fea_dist_method` and `image_dist_method`: Both configured as `"Euclidean"`.
  - `zoom`: Adjusted to 2 for Treasury, Puma, and Forex datasets; set to 1 for Wall-Robot-Navigation.
  - `error`: Set as `"abs"` with `max_step=30000` and `val_step=300`.
- **REFINED**:
  - `problem_type`: Consistent with IGTD (`"regression"` or `"supervised"` depending on dataset).
  - `zoom`: Set identically as IGTD, ensuring comparable spatial resolution.
- **TINTO**:
  - `problem_type`: Aligned with IGTD and REFINED methods.
  - `pixels`: Fixed at 20 for all datasets to standardize the pixel granularity of synthetic images.
  - `blur`: Disabled (`False`) to retain sharpness in feature representation.

[1] https://github.com/KindXiaoming/pykan/tree/v0.2.8
[2] https://pypi.org/project/TINTOlib/1.0.6/

3.1.3 Training Configuration and Metric Tracking

All models (KAN, CNN, and CNN-KAN) were trained using the LBFGS optimizer, following the default configuration recommended by the authors of the `pykan` library. This setup employed a learning rate of 1.0, a history size of 10, and the strong Wolfe line search condition, combined with strict tolerance thresholds to promote stable convergence during full-batch training. The number of training steps varied depending on the model family and the synthetic image generation method used.

As discussed in Section 2.7, all experiments adopted MSE as the loss function, even for classification tasks. This decision was supported by both the official `pykan` examples and preliminary experiments, which showed that treating classification as a regression problem and applying MSE—optimized with LBFGS—produced superior results compared to traditional cross-entropy losses in this context.

During training, model selection was guided by validation performance based on task-specific evaluation metrics. For regression tasks, the model achieving the lowest validation Root Mean Squared Error (RMSE) was retained. For classification tasks, the highest validation accuracy determined the chosen model state. To compute accuracy in classification tasks, continuous outputs were first rounded to the nearest integer and then clipped to fall within the valid range of class labels. This step ensures that even predictions slightly outside the expected bounds are mapped to valid classes.

For clarity, Table 3.1 illustrates how continuous model outputs are mapped to discrete class labels for a four-class classification problem. Predicted values are first rounded to the nearest integer, and then clamped to the valid range of class indices $[1, 4]$. This ensures that even predictions falling slightly outside the expected bounds are correctly assigned to a valid class.

Model Output (Continuous)	Rounded Value	Final Class Label (Clamped)
3.74	4	4
2.12	2	2
0.67	1	1
4.85	5	4
-0.44	0	1

Table 3.1: Example of converting continuous model outputs to discrete class labels for a 4-class classification task. Values are first rounded, then clamped to stay within the valid label range of 1 to 4.

Both RMSE and accuracy were logged throughout training via the built-in tracking mechanisms of the `fit()` function in `pykan`. However, the standard `fit()` method was extended in this work to implement a custom training loop tailored to hybrid architectures as mentioned in Section 2.6. This customized version retained PyKAN's optimization logic but added support for CNN inputs, concatenation strategies, and the classification rounding and clipping procedure described above. This unified training and evaluation framework ensured consistent optimization across all task types while allowing for task-specific performance assessment.

3.1.4 Hyperparameter Search

This section covers the hyperparameter ranges used during the grid search for all model families and concatenation strategies, aiming to find balanced configurations that perform well across datasets. The search strategy was intentionally kept lightweight to demonstrate that high performance and interpretability can be achieved even with a small number of tunable parameters. Rather than exhaustively exploring large hyperparameter spaces, the goal was to test whether effective and explainable models could emerge from a compact and principled search space.

3.1.4.1 KAN Models

The KAN models were trained using a grid search over key architectural and regularization parameters, following guidelines recommended by the authors of the `pykan` library **pykan_readme**. The following hyperparameter ranges were explored:

- **Hidden neurons:** `from 1 to attributes/2`
- **Grid sizes:** `from 3 to 8`
- **Regularization (lambda):** `from 1e-1 to 1e-7`

These parameters respectively control the size of the hidden layer, the resolution of the spline grid, and the smoothness regularization applied during training. Additionally, it was configured with a spline kernel degree of $k = 3$ and employed summation operations at each node. This architectural setup strikes a balance between model expressiveness and interpretability, enabling to capture complex relationships in the data while maintaining transparency in how predictions are formed.

After training, the top two configurations for each dataset were selected based on validation performance. This compact search space proved sufficient to achieve strong results across all datasets, validating both the effectiveness of the parameter guidelines and the model's adaptability.

3.1.4.2 CNN Models

The number of `cnn_blocks` was limited to a maximum of three because the synthetic images used in this work were of relatively low resolution. This limitation stems from the nature of the datasets, which contain a small number of attributes and therefore produce less detailed image representations. Applying multiple max pooling operations in deeper CNN architectures would risk excessive downsampling, leading to a loss of critical information. The exploration of `dense_layers` aimed to simulate the role of a final transformation layer, conceptually analogous to the `final_kan` module used in CNN-KAN hybrid architectures.

- **`cnn_blocks`:** {1, 2, 3}
- **`dense_layers`:** {1, 2, 3}

3.1.4.3 CNN-KAN Hybrid Models

For each dataset, the two best-performing CNN configurations—based on validation performance—were selected as the basis for constructing two hybrid models, referred to as **CNN-KAN 1** and **CNN-KAN 2**. This allowed the CNN branch to be held fixed

during hybrid evaluation, enabling a clearer assessment of the complementary contribution brought by the KAN component.

Each hybrid model was then subjected to a KAN grid search identical to that described in Section 3.1.4.1, exploring combinations of hidden neurons, grid sizes, and regularization strengths. This setup leveraged the strengths of CNNs in capturing spatial feature patterns from synthetic images and the strengths of KANs in modeling direct and interpretable feature interactions. The top two resulting KAN and CNN configurations for each CNN-KAN hybrid variant were retained, forming a candidate set of four hybrid models per dataset. These configurations were then further evaluated under multiple concatenation strategies, detailed in the following subsection, in order to improve feature fusion and enhance the balance of branch contributions.

3.1.4.4 Concatenation Strategies

To refine the integration of the CNN and KAN branches in CNN-KAN hybrid models, a secondary grid search was conducted over four concatenation strategies using the best-performing hybrid configurations identified earlier combined with the best two of KAN model configurations as detailed in Figure 2.6. This evaluation was motivated by findings in Seccion 4.2, where the `cnn_branch` often dominated in terms of relevance, potentially limiting the interpretability and contribution of the tabular branch. Each strategy was explored with its own range of hyperparameters to optimize predictive performance while promoting a more balanced and explainable representation across both modalities.

1. **Bottleneck Projection:** The CNN output tensor is projected through a linear layer (`bottleneck_dim`) with reduced dimensionality before concatenation. The tested values for the projection size were:
   - `bottleneck_dim`: values from 1 to 12
2. **Scaled Concatenation:** The CNN output tensor was scaled by a fixed scalar factor (`alpha`) before concatenation with the KAN output. The values used for the scaling factor were:
   - `alpha`: values from 0.01 to 0.9
3. **Gating with Dynamic Weighting:** A gating MLP was introduced to compute a learned per-sample mixing factor between the CNN and KAN branches tensors. The dimensionality of the hidden layer in the gate was varied as follows:
   - `hidden_dim`: values from 8 to 128
4. **Cross-Attention:** The KAN output was treated as a query and the CNN output as key and value in a multi-head attention block. The tested hyperparameters were:
   - `embed_dim`: values from 16 to 64
   - `num_heads`: values from 2 to 8

3.2 Exploratory Correlation Results

This section summarizes the outcomes of the exploratory correlation analysis described in Section 2.2. The computed correlation score reflects structural redundancy and interaction patterns among numerical features, derived from metrics such as average pairwise correlation, feature clustering, feature continuity, and interaction variance. As shown in Table 3.2, the Treasury and Forex datasets exhibit medium to high correlation, suggesting their features may be meaningfully arranged spatially to leverage local dependencies in CNN-based models. In contrast, the Puma and Wall-Robot datasets show low correlation, indicating weaker inter-feature relationships that could limit the benefits of spatial encoding. These insights are expected to provide context for interpreting subsequent modeling results and assessing the suitability of hybrid architectures for each dataset.

Dataset	Score	Correlation
Treasury	0.666	**High**
Puma	0.015	**Low**
Forex	0.645	**Medium**
Wall-Robot	0.182	**Low**

Table 3.2: Correlation scores and categories for each dataset.

Additionally, detailed tables reporting the average absolute correlation score for each feature are provided in the annex for further reference. These include Tables 1, 2, 3, and 4, which offer deeper insight into feature-level redundancy and variability across the datasets.

3.3 Preprocessing

All datasets used in this study consisted entirely of numerical features, which simplified the preprocessing pipeline by eliminating the need for categorical encoding or text-based transformations. As a result, only two basic transformations were applied before training: (1) rows with missing values were removed, and (2) feature values were scaled using the `MinMaxScaler` from `sklearn.preprocessing`. This minimal preprocessing strategy was adopted to preserve the original structure and statistical relationships in the data as much as possible.

Finally, the datasets were pre-split into training and validation sets as provided in the TALENT repository[3]. This setup was chosen to ensure comparability with the benchmark results reported by the original TALENT authors[4].

3.4 Detailed Results Using IGTD-Based Images

This section presents the first detailed results of the grid search experiments, focusing on models trained with IGTD synthetic images. It provides an overview of how the performance metrics—either RMSE for regression or accuracy for classification—fluctuate across different configurations of `cnn_blocks` and KAN parameters,

[3] https://github.com/LAMDA-Tabular/TALENT/tree/main/example_datasets
[4] https://6sy666.github.io/TALENT-Results/

highlighting the sensitivity of each architecture to specific hyperparameter choices. Similar patterns of variability are observed for the other synthetic image generation methods, albeit under different architectural and hyperparameter setups.

For each dataset, tables report the top two configurations for KAN, CNN, and CNN-KAN hybrid models, including their performance outcomes and key hyperparameters. As introduced in Section 2.11, the KAN grid search was performed only once per dataset, independent of the synthetic image methods, and its best configurations are reported here. For KAN-based models, table entries display the network widths using the `width` parameter (e.g., width=[15, 5, 1]), where the first value reflects the number of tabular features, followed by the hidden layer size, and ending with a fixed output size of 1. Additional parameters such as grid size (`grid`) and regularization (`lamb`) are also included, as detailed in Section 3.1.4.1. To help the reader quickly identify top-performing configurations, the best metric value (either RMSE or accuracy) in each table is highlighted in **bold**.

For CNN and CNN-KAN hybrid configurations, tables report the number of convolutional blocks and dense layers. Notably, the `cnn_blocks` used in hybrid models correspond to the best settings identified in the CNN grid search for each dataset, ensuring the CNN branch remains fixed so that the influence of KAN parameters can be assessed in isolation. Furthermore, plots of validation curves are provided by replicating the best cases to illustrate the variation of performance metrics across training epochs and configurations, offering insight into convergence behavior and model stability under the IGTD framework.

3.4.1 Evaluation on Treasury with IGTD

Table 3.3 summarizes the top-performing configurations for the Treasury dataset, highlighting how different architectures respond to IGTD-based synthetic images. The KAN architecture achieved the best RMSE of **0.19097**, outperforming both hybrid variants and the CNN model. Both CNN-KAN models, while competitive, trailed the pure KAN approach, whereas the standalone CNN model consistently exhibited the highest RMSEs, indicating weaker adaptation to the Treasury's structural characteristics.

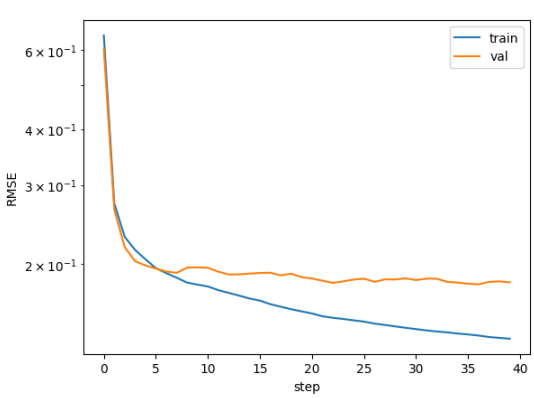
	KAN	**CNN**	**CNN-KAN 1** cnn_blocks=2	**CNN-KAN 2** cnn_blocks=3
Top1 RMSE	**0.19097**	0.24152	0.21210	0.22293
Configuration	width=[15, 5, 1] grid=4 lamb=0.0001	cnn_blocks=2 dense_layers=2	width=[15, 1] grid=5 lamb=1e-05	width=[15, 1] grid=3 lamb=1e-05
Top2 RMSE	0.19433	0.24273	0.21291	0.22405
Configuration	width=[15, 6, 1] grid=5 lamb=1e-05	cnn_blocks=3 dense_layers=1	width=[15, 6] grid=7 lamb=1e-05	width=[15, 4] grid=7 lamb=0.0001

Table 3.3: Top configurations and RMSE for the Treasury dataset. CNN and CNN-KAN hybrid model use IGTD-synthetic images. The best RMSE score is highlighted in bold.

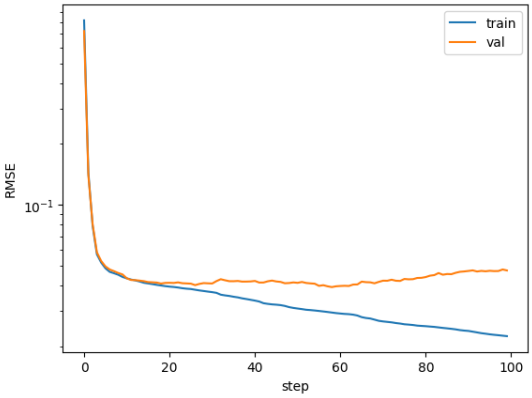
The Treasury dataset, with its relatively small sample size (671 observations) and strong inter-feature correlations, seems to favor models capable of directly capturing

numeric relationships, which likely explains the superior performance of KAN. Hybrid architectures leveraged the combined symbolic and spatial pathways, yet their dependence on CNN branches introduced slight reductions in predictive fidelity compared to pure KAN, particularly because the CNN branch might overlook fine-grained numerical dependencies embedded in tabular data. Notably, the best KAN configurations used moderate grid sizes (4 or 5) and low regularization (`lamb` between 10^{-5} and 10^{-4}), suggesting that preserving sufficient detail without excessive smoothing is advantageous for this dataset.

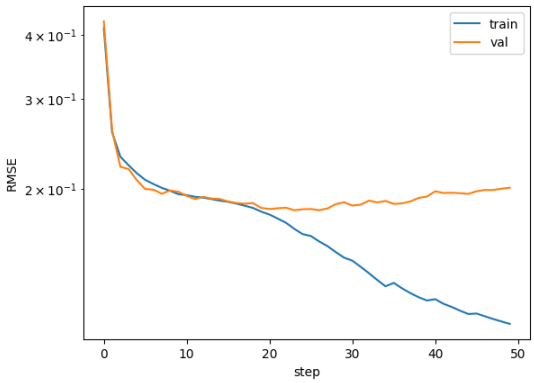
Figure 3.1 displays training and validation RMSE trajectories for the best configurations. Across all architectures, overfitting emerged after roughly 20 epochs, with validation RMSE flattening or rising while training RMSE continued to decline. This effect was especially pronounced in hybrid models, which diverged earlier due to the CNN component.



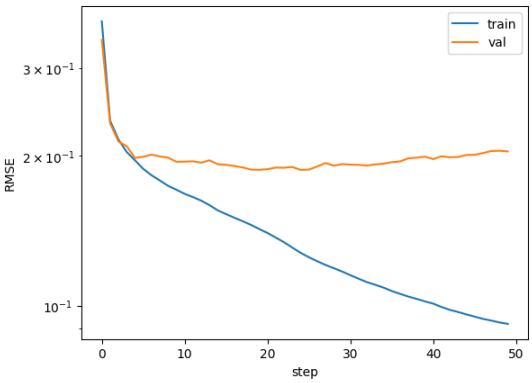
(a) KAN (Best Epoch = 36)



(b) CNN (Best Epoch = 58)



(c) CNN-KAN 1 (Best Epoch = 23)



(d) CNN-KAN 2 (Best Epoch = 24)

Figure 3.1: Training and validation RMSE curves for the best-performing configuration of each model on the Treasury dataset. CNN and CNN-KAN hybrid model curves use IGTD-synthetic images.

3.4.2 Evaluation on Puma with IGTD

Table 3.4 reports the top-performing configurations for the Puma dataset, characterized by only 8 features nd minimal inter-feature correlation, showcasing how each architecture handles IGTD synthetic images. The best result was achieved by the

CNN-KAN 1 model, with an RMSE of **3.21696**, slightly outperforming both the pure CNN and KAN models. Although the differences between architectures were modest, this outcome suggests that hybrid models can effectively leverage complementary insights from spatial and symbolic representations, even in datasets with low feature correlations like Puma.

	KAN	**CNN**	**CNN-KAN 1** (cnn_blocks=1)	**CNN-KAN 2** (cnn_blocks=3)
Top1 RMSE	3.22794	3.22069	**3.21696**	3.23119
Configuration	width=[8, 2, 1] grid=4 lamb=1e-05	cnn_blocks=1 dense_layers=1	width=[8, 4] grid=3 lamb=0.01	width=[8, 1] grid=4 lamb=0.01
Top2 RMSE	3.23315	3.24624	3.21744	3.23543
Configuration	width=[8, 3, 1] grid=5 lamb=0.0001	cnn_blocks=3 dense_layers=2	width=[8, 3] grid=5 lamb=0.01	width=[8, 2] grid=4 lamb=0.1

Table 3.4: Top configurations and RMSE for the Puma dataset. CNN and CNN-KAN hybrid models use IGTD-synthetic images. The best RMSE score is highlighted in bold.

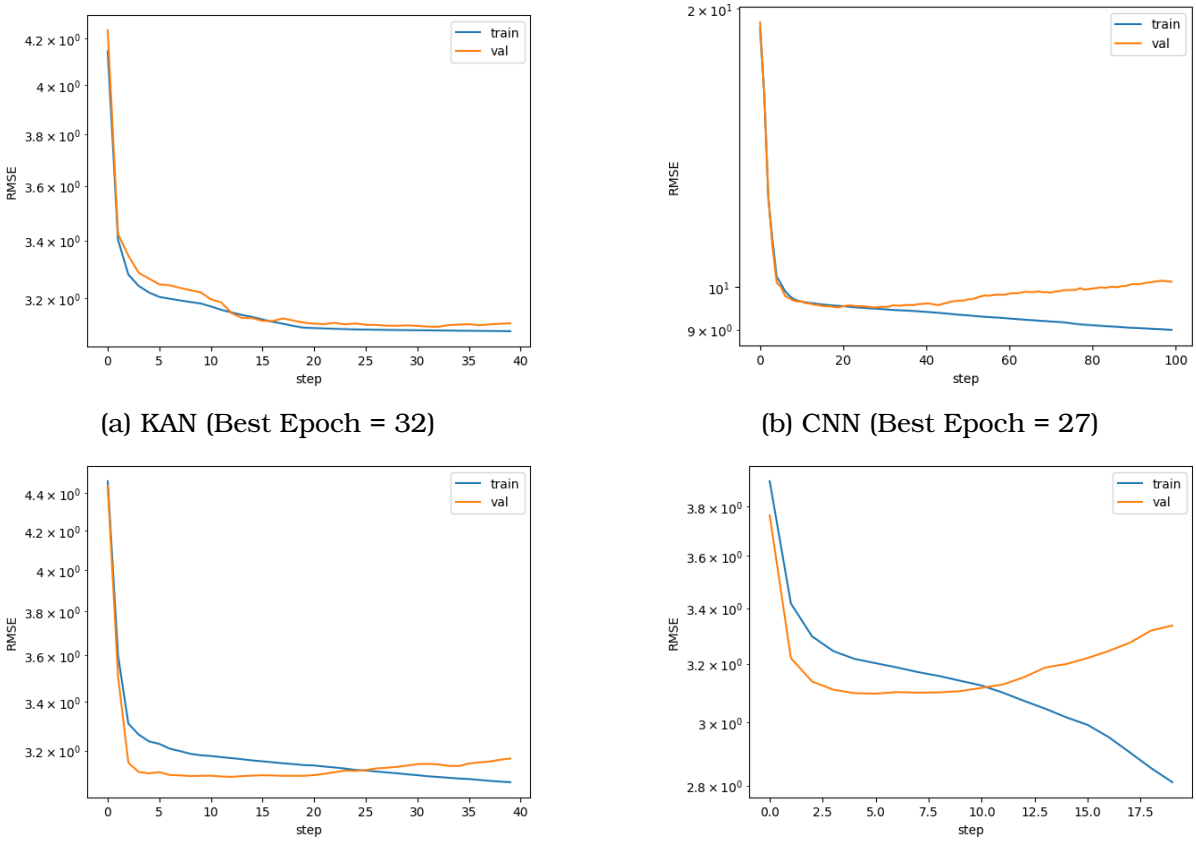
In Puma dataset, while KAN maintained solid performance, the CNN and hybrid architectures benefited from IGTD's ability to arrange features spatially, helping the CNN branch to extract local patterns not apparent in raw tabular data. Notably, the CNN-KAN 2 model performed slightly worse, likely due to its use of three `cnn_blocks`, which may have caused excessive feature compression in low-dimensional images. Interestingly, both hybrid models achieved their best results with relatively higher regularization values (0.01 or 0.1), suggesting the need for stronger constraints to prevent overfitting when merging symbolic and spatial modalities.

Figure 3.2 shows the RMSE evolution across epochs for the best configurations of each architecture. Both CNN-KAN hybrids reached their lowest validation RMSE early in training but displayed signs of overfitting thereafter—a trend also visible, though less severe, in the standalone CNN model. By contrast, the KAN model exhibited stable convergence and minimal divergence between training and validation losses, underscoring its robustness in low-correlation contexts.

3.4.3 Evaluation on Forex with IGTD

Table 3.5 presents the top-performing configurations for the Forex dataset, a binary classification task using IGTD synthetic images. Consistent with the patterns observed in Treasury, the KAN model achieved the best overall performance with an accuracy of **0.74659**, followed closely by the CNN-KAN 1 and CNN-KAN 2 hybrids, while the standalone CNN model lagged behind. These results reinforce the strength of KAN's symbolic reasoning for capturing direct feature interactions, especially in moderately correlated datasets like Forex.

The Forex dataset, with ten features and moderate inter-feature correlation, provided fertile ground for symbolic models. The best KAN and hybrid configurations often utilized small grid sizes (typically grid=3) and low regularization values for `lamb`, balancing expressiveness with generalization. Although CNN-KAN hybrids did not surpass pure KAN in absolute accuracy, they achieved competitive results, suggesting



(a) KAN (Best Epoch = 32)

(b) CNN (Best Epoch = 27)

(c) CNN-KAN 1 (Best Epoch = 12)

(d) CNN-KAN 2 (Best Epoch = 5)

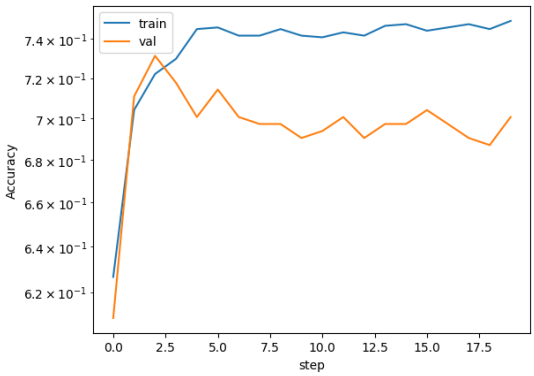
Figure 3.2: Training and validation RMSE curves for the best-performing configuration of each model on the Puma dataset. CNN and CNN-KAN hybrid model curves use IGTD-synthetic images.

	KAN	**CNN**	**CNN-KAN 1** (cnn_blocks=2)	**CNN-KAN 2** (cnn_blocks=1)
Top1 Accuracy	**0.74659**	0.72752	0.74114	0.73569
Configuration	width=[10, 3, 1] grid=3 lamb=1e-05	cnn_blocks=2 dense_layers=1	width=[10, 4] grid=3 lamb=1e-06	width=[10, 4] grid=3 lamb=0.0001
Top2 Accuracy	0.74114	0.71662	0.73025	0.73297
Configuration	width=[10, 2, 1] grid=4 lamb=1e-07	cnn_blocks=1 dense_layers=1	width=[10, 4] grid=5 lamb=0.0001	width=[10, 1] grid=3 lamb=1e-07

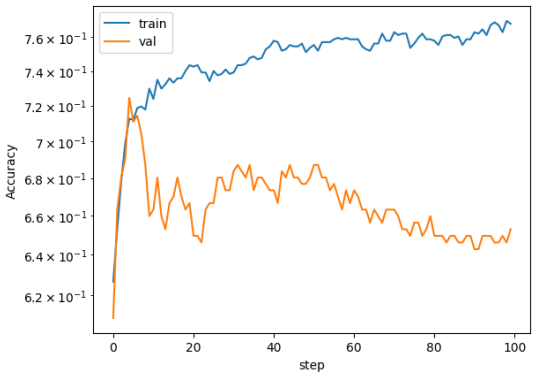
Table 3.5: Top configurations and accuracy for the Forex dataset. CNN and CNN-KAN hybrid models use IGTD-synthetic images. The best accuracy score is highlighted in bold.

that spatial encodings from synthetic images may provide complementary insights, even in classification tasks.

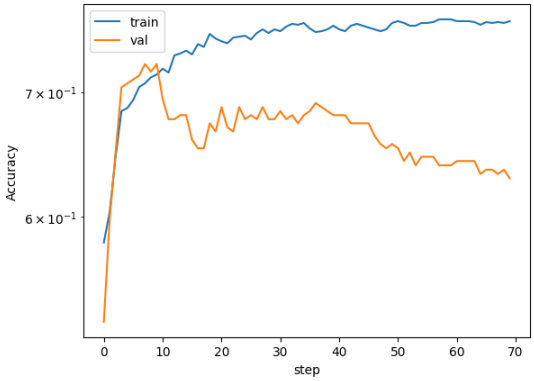
Figure 3.3 illustrates the training and validation accuracy curves for the best configurations of each architecture. Across all models, validation accuracy peaked rapidly in the first few epochs, followed by either plateauing or decline—a clear indication of early overfitting risks given the dataset's modest size (1173 samples) and binary out-



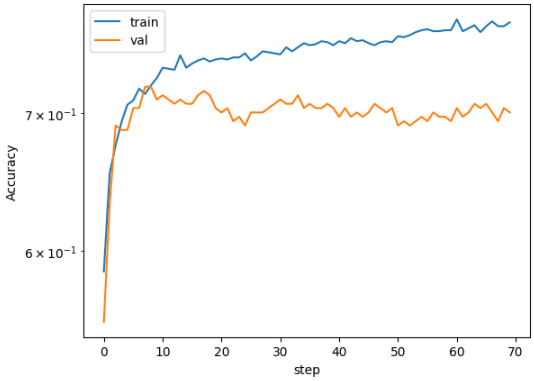
(a) KAN (Best Epoch = 2)



(b) CNN (Best Epoch = 4)



(c) CNN-KAN 1 (Best Epoch = 7)



(d) CNN-KAN 2 (Best Epoch = 7)

Figure 3.3: Training and validation RMSE curves for the best-performing configuration of each model on the Forex dataset. CNN and CNN-KAN hybrid model curves use IGTD-synthetic images.

come. This trend was most pronounced in the KAN and hybrid architectures, where the benefits of low regularization seemed exhausted after early training phases.

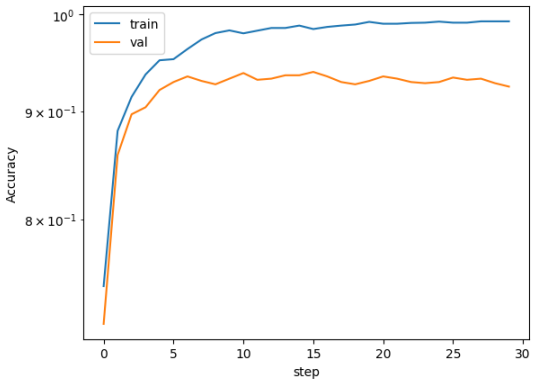
3.4.4 Evaluation on Wall Robot with IGTD

Table 3.6 summarizes the top-performing configurations for the Wall Robot dataset, which represents a multi-class classification task using IGTD-based images. As seen with Treasury and Forex, the KAN model achieved the best overall accuracy at **0.93864**, followed closely by CNN-KAN 1 and CNN-KAN 2, with the standalone CNN model trailing behind. Among KAN and hybrid models, top configurations tended to include higher hidden neuron counts (typically between 10 and 12), larger grid sizes (generally 7 or 8), and moderate regularization values, with `lamb` often set around 0.001. The competitive performance of CNN-KAN 1—despite the weaker standalone CNN results—suggests that the hybrid model leveraged the KAN branch's symbolic insights to mitigate any limitations introduced by the CNN component.

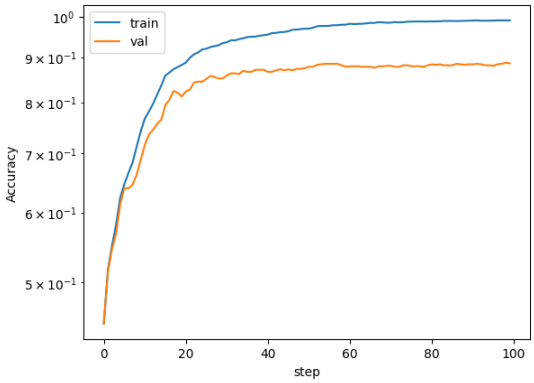
Figure 3.4 displays the training and validation accuracy curves for the best configurations of each model on the Wall Robot dataset. All models achieved high validation accuracy with minimal overfitting, signaling robust generalization capabilities. Notably, CNN-KAN 1 converged faster than the standalone CNN, likely due to guidance

	KAN	CNN	CNN-KAN 1 cnn_blocks=2	CNN-KAN 2 cnn_blocks=3
Top1 Accuracy	**0.93864**	0.86355	0.93589	0.90201
Configuration	width=[24, 12, 1] grid=8 lamb=0.001	cnn_blocks=2 dense_layers=1	width=[24, 10] grid=8 lamb=0.001	width=[24, 9] grid=6 lamb=0.01
Top2 Accuracy	0.93315	0.83791	0.92857	0.90201
Configuration	width=[24, 6, 1] grid=7 lamb=1e-05	cnn_blocks=3 dense_layers=3	width=[24, 11] grid=7 lamb=0.001	width=[24, 11] grid=5 lamb=0.01

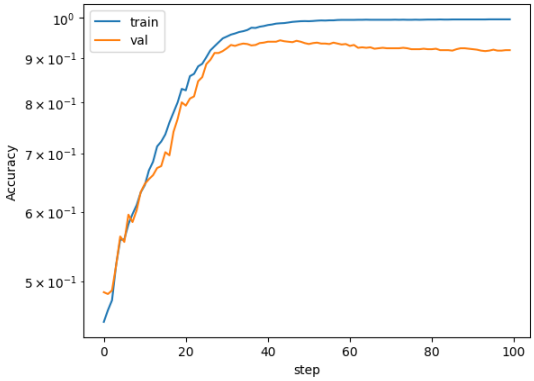
Table 3.6: Top configurations and accuracy for the Wall Robot dataset. CNN and CNN-KAN hybrid models use IGTD-synthetic images. The best accuracy score is highlighted in bold.



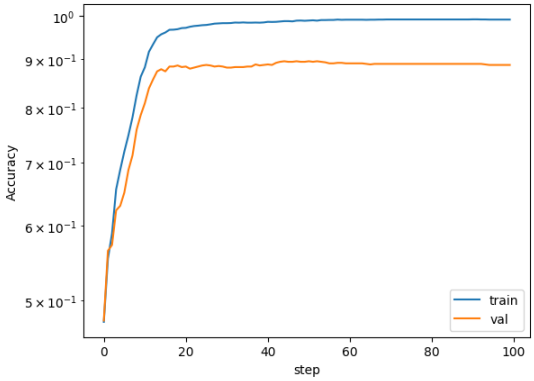
(a) KAN (Best Epoch = 15)



(b) CNN (Best Epoch = 98)



(c) CNN-KAN 1 (Best Epoch = 43)



(d) CNN-KAN 2 (Best Epoch = 44)

Figure 3.4: Training and validation RMSE curves for the best-performing configuration of each model on the Wall Robot dataset. CNN and CNN-KAN hybrid model curves use IGTD-synthetic images.

from the KAN branch, while CNN-KAN 2 exhibited mild fluctuations, perhaps stemming from increased architectural complexity and deeper CNN layers.

3.5 Baseline Performance Across Synthetic Image Methods

This section summarizes the baseline results achieved across the three synthetic image generation methods: IGTD, REFINED, and TINTO. Whereas the preceding sections focused exclusively on IGTD, highlighting its computational efficiency and rapid convergence, the broader performance trends and challenges—such as the impact of image complexity on hybrid model behavior—extend similarly to REFINED and TINTO. Notably, the experiments for REFINED and TINTO were executed on the remote server environment described in Section 3.1.1, due to their higher computational demands and longer training times. From this point forward, the comparative analysis now shifts toward systematically evaluating all the results across the three synthetic image methods, reflecting the central research question of how different visual encodings influence predictive performance and interpretability in hybrid models.

To maintain clarity and brevity, this section reports only the top-performing results for each architecture. Specifically, only the single best-performing CNN and CNN-KAN hybrid configuration is shown per synthetic image method and dataset. As observed previously for IGTD, the optimal architectural choices—such as the number of `cnn_blocks` and selected KAN hyperparameters—remained diverse and dataset-specific for REFINED and TINTO as well, underscoring the importance of tailoring model designs to exploit the distinct structural properties of each synthetic image transformation. Other pattern observed is that CNN-KAN outperform CNN architecture likely to the `cnn_blocks` for this base architecture and also the inclusion of kan components in hybrids.

For each dataset, a single consolidated table is provided, summarizing the lowest RMSE (for regression tasks) or highest accuracy (for classification tasks) achieved by both the CNN and CNN-KAN hybrid models, along with the epoch at which these best results were attained during grid search, also highlighting in **bold** the best results to support the reader to find the best cases for CNN and CNN-KAN. Notably, across all datasets, REFINED and TINTO consistently required more training epochs to reach optimal performance than IGTD. This trend is attributable to their higher image resolutions and more intricate spatial encoding strategies, which increase the learning complexity for the CNN branch and introduce greater computational demands. These insights lay the groundwork for the deeper cross-architecture comparisons and interpretability analyses undertaken in the next chapter.

3.5.1 Baseline Results on the Treasury

Table 3.7 summarizes the baseline results for the Treasury dataset, consolidating the best-performing CNN and CNN-KAN hybrid configurations across the three image generation methods. The CNN architecture achieved its lowest RMSE (**0.24152**) with IGTD, converging in just 14 epochs, suggesting that IGTD's compact grayscale layout aligns well with the highly correlated Treasury features and few samples. In contrast, REFINED and TINTO required substantially more epochs and yielded higher RMSEs, likely due to their larger images and more complex color-based spatial arrangements, which introduce additional learning complexity for CNN models.

Interestingly, incorporating the KAN branch in hybrid architectures reversed the performance trend observed in the CNN models. While IGTD remained competitive, the

	CNN Best RMSE	CNN Epoch	CNN-KAN Best RMSE	CNN-KAN Epoch
IGTD	**0.24152**	14	0.21210	30
REFINED	0.41073	109	0.19125	41
TINTO	0.31567	146	**0.17915**	56

Table 3.7: Baseline results for CNN and CNN-KAN hybrid architectures using IGTD, REFINED, and TINTO synthetic image methods on the Treasury dataset. The best RMSE score for CNN and CNN-KAN are highlighted in bold. For reference, the best standalone KAN model achieved an RMSE of 0.19097 (see Table 3.3).

CNN-KAN hybrid models reached superior RMSE scores with REFINED and specially TINTO (**0.17915**) which also surpassed the pure KAN result, highlighting how certain synthetic image methods can enhance hybrid performance beyond symbolic modeling alone.

3.5.2 Baseline Results on the Puma

Table 3.8 summarizes the baseline results for the Puma dataset, presenting the top-performing configurations for both CNN and CNN-KAN hybrid architectures across the IGTD, REFINED, and TINTO image generation methods. As observed previously with the Treasury dataset, IGTD again produced the best RMSE for CNN but also for CNN-KAN architectures, both beaten the best standalone KAN result. This suggests that IGTD's structured and compact spatial layout may be particularly well-suited to the convolutional architecture and the LBFGS optimization strategy. Such techniques are relevant for datasets like Puma, which contain relatively few attributes, allowing IGTD's dense representations to effectively capture localized patterns even with shallower CNN models.

	CNN Best RMSE	CNN Epoch	CNN-KAN Best RMSE	CNN-KAN Epoch
IGTD	**3.22069**	18	**3.21696**	10
REFINED	3.72244	37	3.24014	15
TINTO	3.71772	8	3.23807	45

Table 3.8: Baseline results for CNN and CNN-KAN hybrid architectures using IGTD, REFINED, and TINTO synthetic image methods on the Puma dataset. The best RMSE score for CNN and CNN-KAN are highlighted in bold. For reference, the best standalone KAN model achieved an RMSE of 3.22794 (see Table 3.4).

In contrast, REFINED and TINTO yielded higher RMSE values across both CNN and CNN-KAN. Although TINTO showed a low best epoch count for CNN (epoch 8), it typically required more extensive training to stabilize, reflecting a similar trend to that seen in the Treasury results. The slightly weaker performance of REFINED and TINTO may stem from the Puma dataset's low feature correlation (as reported in Table 3.2) and limited dimensionality, which reduce the benefits of complex spatial representations. These results reinforce that simpler synthetic images, like those generated by IGTD, remain more effective for datasets of this nature.

3.5.3 Baseline Results on the Forex

Table 3.9 summarizes the baseline results for both CNN and CNN-KAN architectures on the Forex dataset across the IGTD, REFINED, and TINTO image generation methods. Unlike in the previous datasets, differences in accuracy were relatively small for Forex, with REFINED emerging slightly ahead in both model types. In the CNN architecture, REFINED achieved the highest accuracy, suggesting that its spatial layout offers subtle benefits for this classification task. While the best-performing REFINED configuration required 62 epochs, overall training durations remained moderate—typically around 30 epochs—and similar behavior was observed for TINTO, whose best model converged after just 7 epochs.

	CNN Best Accuracy	**CNN Epoch**	**CNN-KAN Best Accuracy**	**CNN-KAN Epoch**
IGTD	0.72752	6	0.74114	6
REFINED	**0.73569**	62	**0.74386**	3
TINTO	0.71662	7	0.73024	39

Table 3.9: Baseline results for CNN and CNN-KAN hybrid architectures using IGTD, REFINED, and TINTO synthetic image methods on the Forex dataset. The best accuracy score for CNN and CNN-KAN are highlighted in bold. For reference, the best standalone KAN model achieved an accuracy of 0.74659 (see Table 3.5).

Regarding CNN-KAN architecture, REFINED again delivered the best accuracy with **0.74386**, close behind IGTD's. Notably, REFINED's CNN-KAN model reached its optimal performance in only 3 epochs, highlighting a rapid learning dynamic possibly linked to the Forex dataset's medium feature count and moderate correlation structure. Although TINTO remained competitive, it required a longer training time and ultimately achieved the lowest accuracy among the three methods for both CNN and CNN-KAN. Overall, the results suggest that while all three methods are viable, REFINED may offer a slight edge for modeling Forex data in CNN-KAN hybrid scenarios, nevertheless in the current architecture none of them could beat the best accuracy of pure KAN.

3.5.4 Baseline Results on the Wall Robot

Finally table 3.10 presents the baseline results for both CNN and CNN-KAN hybrid architectures on the Wall Robot dataset across IGTD, REFINED, and TINTO image generation methods. Similar to the Forex dataset, the CNN architecture achieved its best performance with REFINED-synthetic images, reaching an accuracy of 0.92582 close to the best pure KAN accuracy. This suggests that REFINED's color-coded and spatially optimized layouts effectively capture both local and global dependencies in the Wall Robot data, providing CNNs with richer spatial patterns to exploit.

Notably, this advantage also extends to the hybrid models: the CNN-KAN hybrid using REFINED attained the highest accuracy overall at **0.96153** by far, even beaten the best accuracy for pure KAN, reinforcing the method's suitability for complex, high-dimensional datasets like Wall Robot Navigation. While IGTD remained competitive and continued to require fewer epochs to converge (24), REFINED and TINTO demanded longer training times, likely due to the increased spatial complexity of their images. Overall, these findings confirm that for datasets with lower feature correla-

	CNN Best Accuracy	CNN Epoch	CNN-KAN Best Accuracy	CNN-KAN Epoch
IGTD	0.86355	56	0.93589	24
REFINED	**0.92582**	194	**0.96153**	60
TINTO	0.87728	218	0.93406	48

Table 3.10: Baseline results for CNN and CNN-KAN hybrid architectures using IGTD, REFINED, and TINTO synthetic image methods on the Wall Robot dataset. The best accuracy score for CNN and CNN-KAN are highlighted in bold. For reference, the best standalone KAN model achieved an accuracy of 0.93864 (see Table 3.5).

tions and higher dimensionality, sophisticated spatial encodings such as those provided by REFINED can significantly enhance both CNN and CNN-KAN hybrid model performance.

3.6 Key Insights and Observations

Across all experiments, several clear patterns emerged. Overall, CNN-KAN hybrid models consistently delivered the best or highly competitive results, followed closely by standalone KAN architectures. This finding supports **H1** and **H2**(see Section 1.3), confirming that replacing the traditional MLP with KAN improves both interpretability and performance, while integrating CNN branches trained on synthetic images effectively captures complementary spatial patterns that enhance predictive outcomes. Purely CNN-based models consistently lagged behind, underscoring the value of combining symbolic and spatial reasoning.

Within the CNN-KAN hybrids, additional patterns appeared regarding hyperparameter configurations, aligning with **H1** and **H5**(see Section 1.3). These models benefited from balanced regularization to avoid overfitting, while deeper CNN branches sometimes degraded performance in datasets with few features or low correlations. The variation in effectiveness of synthetic image methods across datasets also supports **H3**(see Section 1.3), as IGTD often yielded faster convergence and lower cost, while REFINED and TINTO sometimes captured more complex patterns at the expense of training time. Interestingly, in three out of the four datasets, the synthetic image method that achieved the best metric for pure CNN also achieved the best metric for the CNN-KAN hybrid models. This observation provides further evidence for **H2** and **H3**(see Section 1.3), emphasizing that the spatial encodings learned by the CNN branch are pivotal for hybrid model success, and that the performance of image methods depends on dataset-specific characteristics.

Overall, these findings confirm that hybrid architectures are a promising direction for modeling tabular data, supporting **H1** through **H3**(see Section 1.3). Yet, strong performance alone is not enough: to avoid becoming a traditional black box, the next chapter delves into explaining how these best-performing models arrive at their predictions, revealing which features and branches truly drive their outcomes.

Chapter 4

Explainability and Interpretability Analysis

Building upon the performance evaluation in the previous chapter, this chapter focuses on the interpretability and explainability of the proposed models, aiming to reveal how predictions are formed and which features drive model decisions. It begins with an exploration of feature relevance through visual examples of Grad-CAM outputs, illustrating how the CNN branch attends to synthetic image regions and thus provides explainability for spatial representations. This is followed by a comparative analysis of top-ranked features across architectures and synthetic image methods, highlighting both dataset-specific and shared relevance patterns. The chapter then delves into branch contribution analysis, quantifying how much influence the symbolic KAN pathway and the visual CNN pathway in hybrid predictions. Further, it examines how different concatenation strategies affect this balance and model performance. Importantly, a novel Global Feature Score is introduced and analyzed, integrating both symbolic and spatial relevance into a unified ranking to enhance overall interpretability and break the notion of the hybrid model as a black box. The chapter concludes with a synthesis of insights and discussion, summarizing the implications of these findings for hybrid model transparency and practical deployment across diverse datasets.

4.1 Feature Relevance Analysis

This section presents the experimental pipeline by examining how the CNN and CNN-KAN hybrid models allocate relevance to input features, thus exploring their explainability beyond pure performance metrics. It begins with visual examples from Grad-CAM, illustrating how different synthetic image methods highlight spatial regions tied to predictions. Next, a comparative visualization presents bar plots of average feature relevance scores from Grad-CAM and KAN feature scores for specific datasets. Finally, a summary is provided of the top features identified across all synthetic image methods, both for CNN models alone and for CNN-KAN hybrids, helping to reveal patterns of feature importance and how hybrid architectures distribute predictive focus between interpretable symbolic mechanisms and explainable spatial representations.

4.1.1 Visual Examples of Grad-CAM Outputs

This subsection presents illustrative examples of how the Grad-CAM algorithm highlights feature relevance within synthetic images generated by different methods, i.e., IGTD, REFINED and TINTO. The visual patterns and heatmap intensities observed in these examples are influenced not only by the synthetic image method applied but also by the trained CNN model's internal representations. Importantly, the aim of showing the following visualizations is to build intuition for how Grad-CAM functions in this context, helping to interpret the more aggregated, feature-level relevance results that will be reported in the next subsection.

Figure 4.1 depicts a Grad-CAM output for an IGTD-synthetic image from the Puma dataset. In this case, IGTD was applied with a zoom factor of 2, so each feature appears as a 2×2 block of pixels. The variation in Grad-CAM intensity across these pixels indicates that activations are influenced not only by individual feature values but also by their spatial relationships with neighboring features. Notably, the extra cell `Ex1` exhibits low but non-zero relevance, suggesting minor saliency in areas reserved for layout padding.

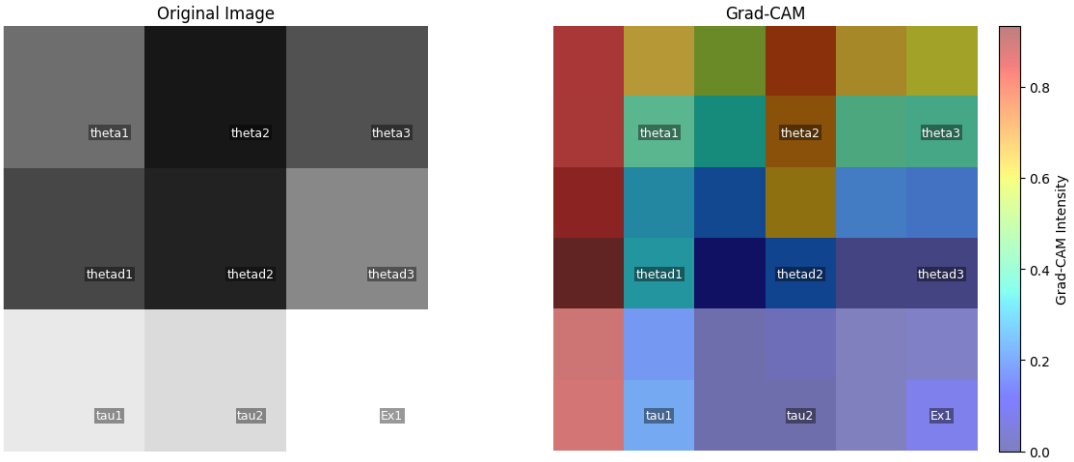


Figure 4.1: Grad-CAM visualization for a synthetic image generated via IGTD on the Puma dataset (`zoom=2`). Each feature occupies a 2×2 block of pixels.

Figure 4.2 shows a Grad-CAM heatmap for an image generated using the REFINED method on the Wall Robot dataset. Here, a zoom factor of 1 means each feature is represented by exactly one pixel. This results in more precise spatial localization of feature relevance. However, this figure also illustrates that in certain cases, Grad-CAM does not highlight a sharply defined zone, instead producing diffuse or uniformly distributed activations. This phenomenon may occur when the learned feature representations lack strong spatial differentiation or when the model distributes relevance evenly across several features due to correlations in the underlying data.

Figure 4.3 presents the Grad-CAM result for a synthetic image generated by TINTO on the Treasury dataset. In this case, the image is considerably sparser, reflecting TINTO's design, which projects features into isolated points across a large canvas. Despite features being clustered into specific regions, the Grad-CAM heatmap predominantly highlights borders or background regions rather than the actual feature areas. This behavior may arise from TINTO's default encoding, where empty space

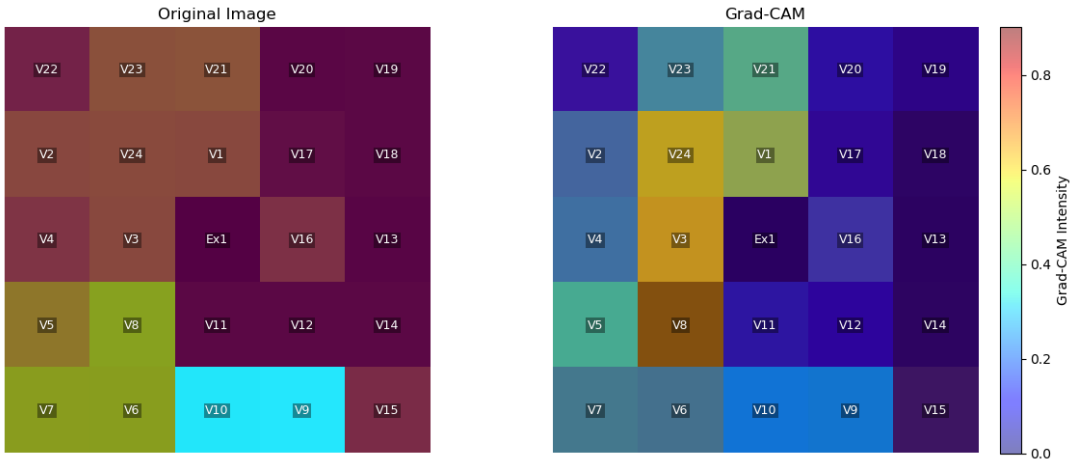


Figure 4.2: Grad-CAM visualization for a REFINED synthetic image from the Wall Robot dataset (`zoom=1`), where each feature corresponds to a single pixel.

is depicted as white and occupied pixels can vary in black's intensity, potentially confusing the CNN during the attribution process. Additionally, this phenomenon might also be influenced by the larger image size produced by TINTO compared to other methods, suggesting that alternative CNN architectures—such as employing more convolutional blocks or different structural configurations—could potentially improve feature attribution in these cases.

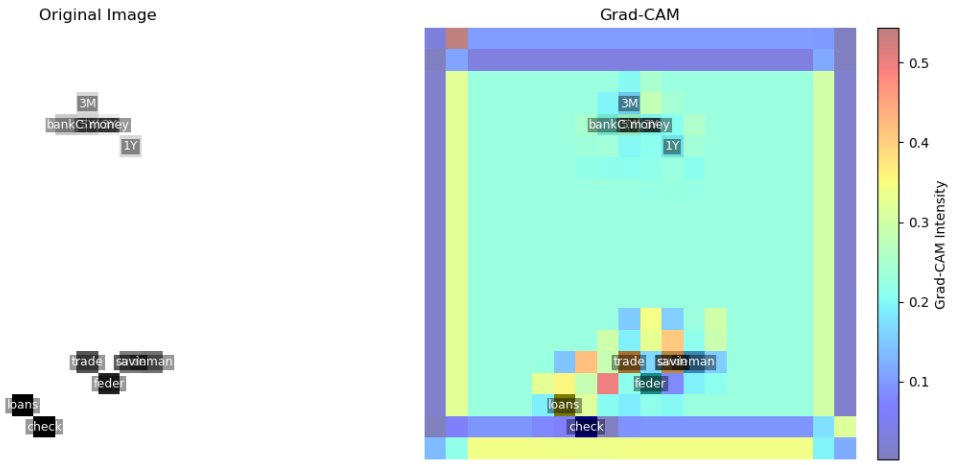
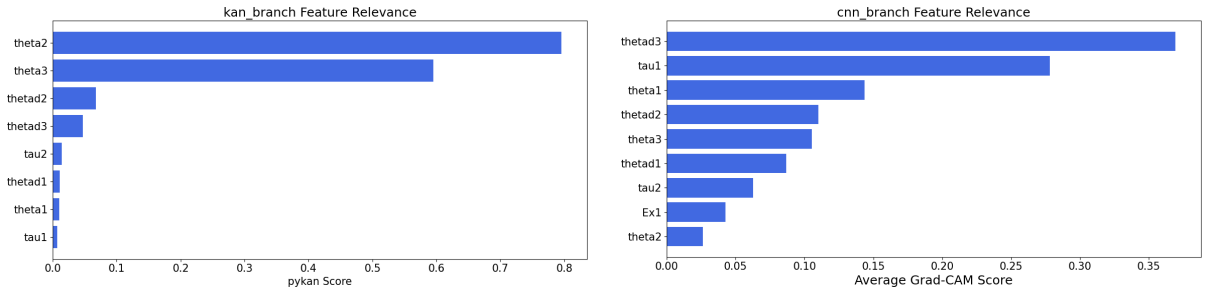


Figure 4.3: Grad-CAM visualization for a TINTO synthetic image from the Treasury dataset. The sparse layout causes Grad-CAM to emphasize borders rather than concentrated feature regions.

4.1.2 Comparative Visualization of Feature Relevance Scores

To complement the previous analyses, Figure 4.4 provides a side-by-side comparison of feature relevance scores produced by the `kan_branch` and the `cnn_branch` for the Puma dataset using REFINED synthetic images. The aim of this example is to illustrate how different relevance extraction techniques reflect the contribution of individual features under the hybrid CNN-KAN architecture.



(a) Feature relevance from `kan_branch` (`pykan` feature scores).

(b) Average Grad-CAM relevance from `cnn_branch`.

Figure 4.4: Comparison of feature relevance scores for the Puma dataset (REFINED synthetic images) obtained from symbolic (`kan_branch`) and spatial (`cnn_branch`) pathways. Note the differing value scales and the presence of extra synthetic pixels in the Grad-CAM plot.

The left panel shows the bar plot of feature importance values derived from the KAN component's `feature_score` obtained after training. These scores typically lie in a numerical range close to or below 1, reflecting the magnitude of the learned spline coefficients associated with each input feature. In contrast, the right panel displays the average Grad-CAM relevance across all REFINED synthetic images in the test dataset. Unlike KAN's scores, Grad-CAM values reflect spatial activation intensities within the CNN's convolutional layers and often present a different scale. Notably, Grad-CAM scores can highlight additional synthetic pixels—labeled as "Ex1," "Ex2," etc.—which appear due to the spatial grid structure used to encode tabular features. These extra regions sometimes capture non-informative background or neighboring influence, contributing to interpretability challenges specific to image-based methods.

A particularly important case arises with TINTO-synthetic images. As shown in Figure 4.3, TINTO often produces synthetic images containing a large number of purely white pixels representing extra regions unrelated to any original features. For this reason, when analyzing feature relevance for TINTO cases, the focus will remain strictly on the relevance scores of the original input features, excluding these extra pixels. Additionally, due to TINTO's use of dimensionality reduction techniques such as PCA, certain features may end up sharing the same spatial pixel in the synthetic image. Consequently, in the final Grad-CAM outputs, these overlapping features will exhibit identical relevance scores, which must be carefully considered when interpreting feature importance results under the TINTO method.

Overall, while both plots aim to quantify feature relevance, they capture fundamentally different aspects of the hybrid model's reasoning. KAN's scores reflect direct symbolic contributions of input features, whereas Grad-CAM emphasizes spatial influence within synthetic images. This subsection thus serves as an illustrative example of how the `kan_branch` contributes to interpretability through direct symbolic relevance, while the `cnn_branch` provides explainability via visual attributions. In the following subsection, a broader analysis will examine the aggregated feature relevance scores across different architectures and synthetic image methods, connecting these insights to the interpretability and performance of the overall models.

4.1.3 Summary of Top Features Across Image Methods

This subsection presents a comprehensive overview of the top three features identified for both the CNN and CNN-KAN hybrid architectures across the different synthetic image generation methods—IGTD, REFINED, and TINTO. The discussion is organized into one subsection for CNN models and another for CNN-KAN hybrids, each summarizing feature relevance across all datasets. As explained in Section 4.1.2, the analysis for TINTO focuses exclusively on original input features, excluding extra synthetic pixels, and acknowledges that some features may share identical Grad-CAM scores due to overlapping spatial encodings from PCA-based layouts.

4.1.3.1 Feature Relevance in CNN Models Across Datasets

Table 4.1 presents a consolidated view of the top three features, ranked by Grad-CAM scores, for the best-performing CNN architecture for each synthetic image generation method. The goal of this summary is to provide insight into the specific variables that most strongly influenced the predictions in purely visual pipelines, establishing a reference point for subsequent hybrid model analyses.

	Best CNN using IGTD	**Best CNN using REFINED**	**Best CNN using TINTO**
Treasury Top 3 Relevant	loansLeases (0.19) demandDeposits (0.02) **savingsDeposits** (0.01)	Ex1 (0.20) bankCredit (0.18) **savingsDeposits** (0.18)	3M-Rate-SecondaryMarket (0.13) 3Y-CMaturityRate (0.13) moneyStock (0.13)
Puma Top 3 Relevant	tau2 (0.34) **tau1** (0.32) **thetad1** (0.24)	**thetad3** (0.45) **tau1** (0.33) theta1 (0.16)	**tau1** (0.15) **thetad3** (0.11) **thetad1** (0.10)
Forex Top 3 Relevant	**Ask_Low** (0.43) Ex2 (0.23) **Bid_Close** (0.18)	Ask_Open (0.011) Ask_Volume (0.009) Ex1 (0.002)	Ask_Close (0.76) **Bid_Close** (0.76) **Ask_Low** (0.76)
Wall Robot Top 3 Relevant	V14 (0.63) V19 (0.61) V13 (0.52)	V9 (0.38) V10 (0.34) Ex1 (0.30)	V3 (0.239) V17 (0.238) V20 (0.238)

Table 4.1: Top 3 feature relevance scores from the replicated best CNN models using Grad-CAM across IGTD, REFINED, and TINTO synthetic image methods for all datasets.Features in **bold** appear frequently across multiple configurations and datasets, highlighting variables of potential general importance.

For the **Treasury** dataset—characterized by high inter-feature correlation—the top features identified by CNN models varied considerably across the three synthetic methods. Under IGTD, the feature `loansLeases` stands out with a substantial relevance score (0.19), whereas subsequent features have much lower impact. In contrast, for REFINED, `savingsDeposits` remains significant, but the most relevant position is occupied by `Ex1`, an extra cell in the image layout. This may be explained by REFINED's spatial design, where `Ex1` is placed in the image center, potentially blending with the high activation zones of neighboring features like `bankCredit` and `savingsDeposits`. For TINTO, completely different features appear in the top three, all sharing the same Grad-CAM score, indicating that they reside in the same pixel location within the synthetic image.

The **Puma** dataset presents an intriguing result. Here, a consistent pattern emerges: the features `tau1`, `thetad3`, and `thetad1` repeatedly appear among the top-ranked variables across all three image methods. Particularly, `tau1` is notable for its stable presence in the top three, underscoring its strong influence on the CNN models.

Interestingly, as previously highlighted in Table 2, these same features are found among those with lower correlation in the dataset's initial analysis—suggesting that CNNs may exploit less obvious interactions captured via spatial encoding.

For **Forex**, there is partial consistency across methods. The features `Ask_Low` and `Bid_Close` emerge in the top three for both IGTD and TINTO, while the remaining features differ significantly. Notably, in IGTD and REFINED, extra cells (`Ex1` or `Ex2`) appear in the top relevance list, likely because they are positioned close to genuinely informative features, thus absorbing some of their activation. In the TINTO case, the three top features share identical relevance scores and spatial coordinates due to the PCA-driven overlap.

Finally, the **Wall Robot** dataset does not exhibit a consistent feature pattern across the three synthetic methods. The top three features vary entirely between IGTD, REFINED, and TINTO, suggesting that the CNNs adaptively latch onto different signals depending on the spatial arrangement introduced by each method. This diversity may reflect the nature of the Wall Robot dataset itself, which contains many features with relatively low correlation, making it more prone to variability in feature selection across different spatial representations.

4.1.3.2 Feature Relevance in CNN-KAN Hybrid Models Across Datasets

Continuing with the analysis of feature relevance, Table 4.2 consolidates the top three most relevant features for both the `kan_branch` and the `cnn_branch` of the best-performing CNN-KAN hybrid models across the four datasets and the three synthetic image generation methods—IGTD, REFINED, and TINTO. The feature importance values shown beside each feature were derived from the `feature_score` attribute of the KAN components and the Grad-CAM relevance maps for the CNN components. In the table, features highlighted in **bold** denote attributes that appeared frequently as top contributors across different hybrid configurations, suggesting that these variables consistently carry predictive signals, regardless of the image encoding strategy or the model's internal architecture. This visual emphasis helps the reader quickly identify features of potential cross-modal and cross-dataset significance.

Beyond identifying which features appear frequently, interesting differences also arise in the magnitude and distribution of relevance scores across branches and datasets. For the `kan_branch`, the relevance scores are generally moderate and clustered closely together, indicating that several features tend to share importance rather than one single feature dominating—except in the Forex dataset with REFINED, where the top three KAN scores are notably high, suggesting that the KAN component relies heavily on a few specific variables to drive predictions. In contrast, for the `cnn_branch`, there are cases where one or two feature achieves a much higher relevance score than the others, pointing to a strong focus on a specific spatial region or feature representation. Examples include the Treasury dataset with IGTD, Puma with TINTO, and Wall Robot with TINTO—where the second and third-ranked CNN features sometimes have scores close to zero, indicating limited contribution from the rest of the feature map in those models.

Regarding the relevance of the most important characteristics, certain patterns can be identified. In the **Treasury** dataset, `loansLeases` stands out for its repeated presence across both branches and all image methods, signaling its strong predictive value in both symbolic and spatial pathways. In the **Puma** dataset, `tau1`,

	Best CNN-KAN using IGTD	**Best CNN-KAN using REFINED**	**Best CNN-KAN using TINTO**
Treasury	currency (0.37)	federalFunds (0.29)	3Y-CMaturityRate (0.40)
Top 3 Relevant	**loansLeases** (0.35)	moneyStock (0.24)	federalFunds (0.31)
kan_branch	5Y-CMaturityRate (0.31)	checkableDeposits (0.23)	30Y-CMortgageRate (0.30)
Treasury	1Y-CMaturityRate (0.21)	**loansLeases** (0.28)	currency (0.266)
Top 3 Relevant	**loansLeases** (0.08)	bankCredit (0.21)	savingsDeposits (0.266)
cnn_branch	3M-Rate-SecondaryMarket (0.04)	5Y-CMaturityRate (0.18)	**loansLeases** (0.238)
Puma	**tau1** (0.39)	**theta2** (0.79)	theta3 (0.68)
Top 3 Relevant	**thetad2** (0.34)	theta3 (0.59)	**theta2** (0.44)
kan_branch	**theta2** (0.33)	**thetad2** (0.06)	**thetad2** (0.22)
Puma	theta1 (0.30)	thetad3 (0.36)	**theta2** (0.26)
Top 3 Relevant	thetad1 (0.27)	**tau1** (0.29)	**tau1** (0.01)
cnn_branch	**tau1** (0.26)	theta1 (0.15)	**thetad2** (0.01)
Forex	**Bid_Close** (0.75)	**Bid_Close** (1.60)	Bid_Low (0.82)
Top 3 Relevant	**Bid_Open** (0.49)	Ask_Close (1.53)	**Bid_Open** (0.71)
kan_branch	Ask_Close (0.45)	**Bid_Open** (1.12)	Bid_High (0.65)
Forex	Ex2 (0.60)	Ex3 (0.57)	Ask_Volume (0.28)
Top 3 Relevant	Ex4 (0.33)	Bid_High (0.39)	Bid_Volume (0.19)
cnn_branch	Ex5 (0.33)	Ask_High (0.14)	**Bid_Close** (0.14)
			Bid_Open (0.14)
Wall Robot	**V20** (0.33)	**V15** (0.24)	**V15** (0.33)
Top 3 Relevant	V19 (0.30)	V24 (0.20)	**V20** (0.28)
kan_branch	**V15** (0.28)	V14 (0.19)	V12 (0.20)
Wall Robot	V3 (0.33)	V9 (0.27)	V7 (0.0017)
Top 3 Relevant	V2 (0.31)	V13 (0.27)	V1 (0.0)
cnn_branch	V4 (0.29)	V14 (0.25)	V2 (0.0)

Table 4.2: Top 3 feature relevance scores for the kan_branch and cnn_branch of CNN-KAN hybrid models trained on all four datasets across IGTD, REFINED, and TINTO synthetic image generation methods. Features in **bold** appear frequently across multiple configurations and datasets, highlighting variables of potential general importance.

theta2, and thetad2 are recurrently prioritized by the kan branch and, in some cases, by the CNN branch as well—highlighting their capacity to encode meaningful information despite the dataset's relatively low feature correlation. For the **Forex** dataset, Bid_Close and Bid_Open consistently rank among the top features across kan branch and cnn branch for TINTO. Interestingly, in IGTD and REFINED-based Forex models, extra synthetic pixels (e.g., Ex2, Ex3) also appear with high relevance scores in the cnn branch, suggesting possible overemphasis on synthetic grid regions rather than original features. Finally, in the **Wall Robot** dataset, the kan branch frequently identifies V15 and V20 as highly relevant across synthetic image methods, while the CNN branch shows more varied focus depending on the image method used—likely a reflection of the dataset's low inter-feature correlation and the distinct spatial layouts introduced by different encoding techniques.

Altogether, these findings underscore the hybrid model's capacity to capture and integrate complementary patterns from both symbolic and spatial representations. They also reveal how certain features can dominate interpretability and explainability insights across architectures and synthetic image techniques, offering potential targets for further optimization and domain-specific analysis.

4.2 Branch Contribution Analysis in Hybrid Models

Continuing with the replication of the best configurations for the CNN-KAN hybrids, this section presents a quantitative analysis of how much each branch contributes to

the final prediction. Table 4.3 summarizes the percentage relevance attributed to the kan_branch and cnn_branch for the optimal hybrid models across all datasets and synthetic image generation methods. These relevance values were derived using the feature_score method from the final_kan component, offering an integrated perspective on how the hybrid architectures distribute predictive responsibility between symbolic and visual pathways.

Overall, the results reveal that in most cases, the output tensor from the cnn_branch dominates the final prediction, frequently surpassing 80% relevance—particularly in datasets like Treasury and Forex. Interestingly, these two datasets also exhibit high and medium correlation scores, respectively, as shown in Table 3.2, suggesting that strong feature interrelationships may enhance the effectiveness of spatial encodings and contribute to CNN dominance. In contrast, datasets like Wall Robot and Puma, which have low correlation scores, show a partial balanced contribution from the kan_branch, indicating that symbolic reasoning may play a more significant role when spatial patterns in synthetic images are less informative. In general, the prevalent tendency for CNNs to dominate was a key motivation for exploring diverse concatenation strategies, aiming not only to improve predictive accuracy but also to achieve a more balanced distribution of relevance between branches, as discussed in previous chapters.

	Treasury	**Puma**	**Forex**	**Wall Robot**
IGTD kan_branch %	0.1	2.5	0.2	**58.2**
IGTD cnn_branch %	**99.9**	**97.5**	**99.8**	41.8
REFINED kan_branch %	27.7	**62.1**	0.4	14.1
REFINED cnn_branch %	**72.3**	37.9	**99.6**	**85.9**
TINTO kan_branch %	19.8	33.7	0.5	**94.4**
TINTO cnn_branch %	**80.2**	**66.3**	**99.5**	5.6

Table 4.3: Percentage relevance contribution of the kan_branch and cnn_branch in the best CNN-KAN hybrid configurations across all datasets and synthetic image methods. Bold values mark the dominant branch for each dataset under the synthetic method.

4.3 Concatenation Strategy Comparison

Building upon the branch contribution analysis and motivated by the observed dominance of the cnn_branch highlighted at the end of the previous section, this section reports the results of the systematic grid search introduced in Section 3.1.4.4, which explored four distinct concatenation strategies originally named *Bottleneck Projection*, *Scaled Concatenation*, *Gating with Dynamic Weighting*, and *Cross-Attention*. To simplify discussion and improve readability, these will henceforth be referred to as "Strategy 1," "Strategy 2," and so on throughout this section.

For clarity, the results are organized into dedicated subsections for each dataset, each presenting a summary table that consolidates outcomes across the three synthetic image methods. These tables detail, for every concatenation strategy, the top-1 RMSE or accuracy achieved, the epoch corresponding to the best result, the specific

hyperparameter configuration used, and the relative percentage contributions from the kan_branch and cnn_branch, offering a comprehensive view of how different fusion mechanisms influence both interpretability and performance.

4.3.1 Comparison of Concatenation Techniques for Treasury

Table 4.4 summarizes the outcomes of the concatenation strategy grid search for the Treasury dataset across all three synthetic image generation methods. A striking pattern emerges: Strategy 1 consistently achieves the lowest RMSE across the three synthetic image methods, suggesting its particular suitability for this dataset. This consistent advantage may be attributed to Treasury's relatively small sample size (671 observations), where a simpler bottleneck projection helps mitigate overfitting by constraining the CNN output's dimensionality and amplifying the role of tabular features captured in the kan_branch. Notably, in Strategy 1, the kan_branch dominates the model's contributions, frequently exceeding 90%, underscoring how limited data volume favors symbolic processing over complex spatial representations for this dataset. Within this strategy, the best overall RMSE was achieved using TINTO, where where the kan_branch accounted for 98.1% of the relevance.. This leaves a modest 1.9% contribution from the cnn_branch, suggesting that even a small infusion of additional information from synthetic images can provide a crucial performance edge over IGTD and REFINED under this configuration.

Moving to Strategy 2, this approach generally secured the second-best performance across synthetic image methods. Here, the scaling parameter alpha provides moderate balancing between the branches, allowing around one-fourth of relevance to be retained by the kan_branch in the reported cases. However, compared to Strategy 1, it appears less effective in fully shifting predictive importance toward the symbolic path, possibly because the scaling factor alone cannot impose the same strong dimensionality constraint as a bottleneck layer.

Strategies 3 and 4 reveal more pronounced trade-offs. Strategy 3, using a gating mechanism, consistently struggled to achieve balanced branch contributions, with the cnn_branch dominating nearly all cases and leaving the kan_branch with negligible influence. Meanwhile, Strategy 4—although it occasionally managed to balance relevance more evenly—produced significantly higher RMSEs, especially for REFINED and TINTO. The added complexity of cross-attention, combined with the LBFGS optimizer in full-batch mode, likely imposes challenges for Treasury's small sample size, hindering convergence and performance despite its theoretical flexibility for feature interaction modeling. Overall, these results suggest that for Treasury, simpler fusion mechanisms like Strategy 1 are better suited to leverage the strength of KANs while avoiding the computational and overfitting risks posed by more elaborate concatenation methods.

4.3.2 Comparison of Concatenation Techniques for Puma

Unlike the Treasury dataset, where Strategy 1 consistently dominated, the Puma dataset shows that Strategy 2 achieved the best RMSE across all three synthetic image generation methods, as displayed in Table 4.5. This result highlights the effectiveness of the scaling factor alpha in Strategy 2, which appears to adaptively balance the contributions between the kan_branch and the cnn_branch in this specific context. Interestingly, while this strategy, consistently emerged as the top performer, the

	Concatenation Strategy 1	Concatenation Strategy 2	Concatenation Strategy 3	Concatenation Strategy 4
IGTD Top1 RMSE	**0.19197**	0.19465	0.20925	0.20547
IGTD Epoch for Best	13	33	46	68
IGTD Configuration	width=[15, 5] bottleneck_dim=2	width=[15, 6] alpha=0.1	width=[15, 6] hidden_dim=8	width=[15, 5] embed_dim=12 num_heads=6
IGTD kan_branch %	95.3	25.7	0.3	33.7
IGTD cnn_branch %	4.7	74.3	99.7	66.3
REFINED Top1 RMSE	**0.20171**	0.20364	0.24355	0.43799
REFINED Epoch for Best	15	18	61	107
REFINED Configuration	width=[15, 5] bottleneck_dim=2	width=[15, 5] alpha=0.3	width=[15, 5] hidden_dim=32	width=[15, 5] embed_dim=16 num_heads=4
REFINED kan_branch %	92.9	27.7	0.3	33.2
REFINED cnn_branch %	7.1	72.3	99.7	66.8
TINTO Top1 RMSE	**0.18825**	0.2024	0.21459	0.30121
TINTO Epoch for Best	46	70	79	89
TINTO Configuration	width=[15, 3] bottleneck_dim=1	width=[15, 2] alpha=0.6	width=[15, 2] hidden_dim=8	width=[15, 2] embed_dim=12 num_heads=6
TINTO kan_branch %	98.1	25.6	0.3	8.5
TINTO cnn_branch %	1.9	74.4	99.7	91.5

Table 4.4: Top-1 RMSE, epoch, configuration, and branch relevance for CNN-KAN hybrid models across concatenation strategies and synthetic image methods on the Treasury dataset. Bold values indicate the best RMSE per image method.

actual branch relevance percentages varied considerably among the different synthetic methods, indicating that `alpha` tuning effectively modulates the influence of spatial versus symbolic representations.

Across all architectures, the best results were obtained using IGTD images, with Strategy 2 slightly outperforming even Strategy 1 for this method. This advantage might stem from the relatively low number of features in the Puma dataset, which allows IGTD's simpler grayscale representation to capture sufficient information without overwhelming the model's capacity. Additionally, it is notable that Strategy 2 generally required fewer training epochs to reach optimal performance than Strategy 1, suggesting that its dynamic scaling mechanism helps the model converge faster under the LBFGS optimizer, even in full-batch training.

Strategies 3 and 4, on the other hand, exhibited a strong preference toward the `cnn_branch`, resulting in much higher RMSEs, especially for REFINED and TINTO. This imbalance could again be attributed to the low feature dimensionality in Puma, where overly complex fusion mechanisms may fail to leverage the limited structured signal effectively and instead shift too much predictive weight onto the visual pathways. Overall, these results highlight that for the Puma dataset, simpler yet flexible strategies like Strategy 2 can strike an optimal balance between performance and

	Concatenation Strategy 1	Concatenation Strategy 2	Concatenation Strategy 3	Concatenation Strategy 4
IGTD Top1 RMSE	3.22672	**3.21585**	3.22992	3.229637
IGTD Epoch for Best	10	2	21	26
IGTD Configuration	width=[8, 3] bottleneck_dim=1	width=[8, 2] alpha=0.7	width=[8, 3] hidden_dim=32	width=[8, 4] embed_dim=32 num_heads=2
IGTD kan_branch %	99.1	1.5	0.1	33.7
IGTD cnn_branch %	0.9	98.5	99.9	66.3
REFINED Top1 RMSE	3.23964	**3.23636**	3.26816	3.72423
REFINED Epoch for Best	13	9	21	27
REFINED Configuration	width=[8, 3] bottleneck_dim=4	width=[8, 2] alpha=0.9	width=[8, 2] hidden_dim=16	width=[8, 2] embed_dim=32 num_heads=2
REFINED kan_branch %	93.7	49.4	26.8	5.5
REFINED cnn_branch %	6.3	50.6	73.2	94.5
TINTO Top1 RMSE	3.23629	**3.23476**	3.26247	3.71439
TINTO Epoch for Best	20	17	58	86
TINTO Configuration	width=[8, 3] bottleneck_dim=3	width=[8, 4] alpha=0.2	width=[8, 2] hidden_dim=32	width=[8, 3] embed_dim=16 num_heads=2
TINTO kan_branch %	99.6	80.4	12	13.5
TINTO cnn_branch %	0.4	19.6	88	86.5

Table 4.5: Top-1 RMSE, epoch, configuration, and branch relevance for the best CNN-KAN hybrid models across concatenation strategies and image generation methods on the Puma dataset. Bold values indicate the best RMSE per image method.

explainability.

4.3.3 Comparison of Concatenation Techniques for Forex

For the Forex dataset, a distinctive pattern emerges compared to previous experiments. As shown in Table 4.6, the best overall result was achieved with Strategy 2 using REFINED images. Apart from this, for IGTD, both Strategy 1 and Strategy 2 reached the same top accuracy of 0.73024. Interestingly, despite this parity in accuracy for IGTD, the branch relevance patterns in these two strategies are starkly opposed: Strategy 1 achieved its strong results with a dominant contribution from the `kan_branch`, while Strategy 2 relied almost entirely on the `cnn_branch`. This divergence highlights how, in Forex, good predictive performance can be obtained through either symbolic or spatial pathways, depending on how the fusion strategy modulates feature integration.

The fluctuation in branch relevance could be linked to the high number of extra pixels present in the synthetic images—particularly for IGTD and REFINED—where six extra positions add potential noise or spatial artifacts that may sway the model's attention toward either modality. Notably, across all strategies and image methods, the Forex dataset maintained solid accuracy levels, even in Strategy 4. For instance,

	Concatenation Strategy 1	Concatenation Strategy 2	Concatenation Strategy 3	Concatenation Strategy 4
IGTD Top1 Acc	**0.73024**	**0.73024**	0.71934	0.72479
IGTD Epoch for Best	3	6	8	10
IGTD Configuration	width=[10, 2] bottleneck_dim=1	width=[10, 2] alpha=0.75	width=[10, 4] hidden_dim=8	width=[10, 4] embed_dim=16 num_heads=2
IGTD kan_branch %	99.99	1.6	0.01	25.2
IGTD cnn_branch %	0.01	98.4	99.99	74.8
REFINED Top1 Acc	0.73841	**0.74659**	0.73024	0.74386
REFINED Epoch for Best	10	3	1	6
REFINED Configuration	width=[10, 3] bottleneck_dim=10	width=[10, 3] alpha=0.8	width=[10, 2] hidden_dim=64	width=[10, 5] embed_dim=12 num_heads=6
REFINED kan_branch %	15.7	1.1	0.01	49.7
REFINED cnn_branch %	84.3	98.9	99.99	50.3
TINTO Top1 Acc	**0.73569**	0.72479	0.72479	0.71934
TINTO Epoch for Best	5	44	38	13
TINTO Configuration	width=[10, 3] bottleneck_dim=4	width=[10, 1] alpha=0.2	width=[10, 5] hidden_dim=32	width=[10, 3] embed_dim=32 num_heads=2
TINTO kan_branch %	97.6	0.1	0.3	8.3
TINTO cnn_branch %	2.4	99.9	99.7	91.7

Table 4.6: Top-1 RMSE, epoch, configuration, and branch relevance for the best CNN-KAN hybrid models across concatenation strategies and image generation methods on the Forex dataset. Bold values indicate the best accuracy per image method.

REFINED under this Strategy achieved the second-best accuracy across all configurations, demonstrating that, for this dataset with REFINED synthetic images, complex fusion mechanisms can still yield competitive results.

Another consistent trend is that the best-performing configurations for Forex generally converged in very few epochs, except for models using TINTO. This rapid convergence aligns with the behavior observed in the baseline results for the other architectures, as reported in Sections 3.4.3 and 3.5.3. In contrast, the longer convergence required for TINTO persists as a pattern since the baseline results, reflecting the added complexity and sparse nature of TINTO-synthetic images, which may necessitate more training iterations to stabilize feature relevance and classification decisions.

4.3.4 Comparison of Concatenation Techniques for Wall Robot

For the Wall Robot Navigation dataset, a distinctive pattern emerges, as shown in Table 4.7. Strategy 3 consistently delivered the highest accuracy across all three synthetic image methods, although for REFINED, it tied with Strategy 2 at a remarkable accuracy of 0.95787 which is also the highest among all Strategies. Interestingly, this tie under Strategy 2 and 3 corresponded to an almost exclusive reliance on the cnn_branch, with close to 100% of the relevance attributed to the visual pathway.

By contrast, for Strategy 3 in IGTD and TINTO, approximately two-thirds of the final decision-making weight was allocated to the `cnn_branch`, indicating that this strategy can flexibly leverage both symbolic and spatial modalities depending on the image method.

	Concatenation Strategy 1	**Concatenation Strategy 2**	**Concatenation Strategy 3**	**Concatenation Strategy 4**
IGTD Top1 Acc	0.93681	0.93772	**0.94963**	0.89194
IGTD Epoch for Best	6	11	78	105
IGTD Configuration	width=[24, 12] bottleneck_dim=6	width=[24, 12] alpha=0.9	width=[24, 10] hidden_dim=8	width=[24, 11] embed_dim=64 num_heads=4
IGTD `kan_branch` %	94.2	66.3	33.9	18.9
IGTD `cnn_branch` %	5.8	33.7	66.1	81.1
REFINED Top1 Acc	0.94139	**0.95787**	**0.95787**	0.9423
REFINED Epoch for Best	4	38	73	101
REFINED Configuration	width=[24, 12] bottleneck_dim=2	width=[24, 7] alpha=0.75	width=[24, 11] hidden_dim=8	width=[24, 7] embed_dim=64 num_heads=2
REFINED `kan_branch` %	99.9	0.0001	0.0001	0.05
REFINED `cnn_branch` %	0.1	99.9999	99.9999	99.95
TINTO Top1 Acc	0.93406	0.94413	**0.94871**	0.93131
TINTO Epoch for Best	50	14	128	173
TINTO Configuration	width=[24, 8] bottleneck_dim=3	width=[24, 12] alpha=0.5	width=[24, 8] hidden_dim=16	width=[24, 6] embed_dim=32 num_heads=4
TINTO `kan_branch` %	98.3	97.2	33.1	28.3
TINTO `cnn_branch` %	1.7	2.8	66.9	71.7

Table 4.7: Top-1 RMSE, epoch, configuration, and branch relevance for the best CNN-KAN hybrid models across concatenation strategies and synthetic image methods on the Wall Robot dataset. Bold values indicate the best accuracy per image method.

Despite Strategy 1 assigning a far greater share of relevance to the `kan_branch`, it could not surpass the performance of Strategy 3. This suggests that the gating mechanism in later Strategy mentioned may excel at extracting discriminative information from the synthetic images, enabling the `cnn_branch` to capture crucial spatial patterns that are particularly beneficial for the Wall Robot task. Meanwhile, Strategies 2 and 4 exhibited more fluctuating relevance distributions between the branches, yet still achieved competitive accuracy levels, demonstrating the adaptability of hybrid architectures to different fusion strategies even in a dataset characterized by lower feature correlations.

Overall, these results highlight how, in the Wall Robot scenario, dynamic feature weighting and richer fusion mechanisms like those in Strategy 3 can effectively capitalize on the spatial cues present in synthetic images, providing a clear advantage over simpler projection-based strategies that favor symbolic representations.

4.4 Cross-Architecture Evaluation and Feature Analysis

To synthesize all the results reported in this chapter and Chapter 3—this section presents a consolidated overview of the best-performing configurations across all architectures and synthetic image methods for each dataset. In each subsection, a dedicated comparison table is provided, summarizing top models from several architectural families including the author's baseline[1], KAN, CNN, and hybrid CNN-KAN models with and without advanced concatenation strategies for the respective dataset. Each table row reports critical details including the achieved RMSE or accuracy, the synthetic image method used, the relative contribution of the `kan_branch` and `cnn_branch` (for hybrid models), and the top three most relevant features identified for each branch.

4.4.1 Summary Results for Treasury

Table 4.8 presents a consolidated view of the top-performing models for the Treasury dataset, where the best overall RMSE of **0.17915** was achieved by the CNN-KAN hybrid architecture using the TINTO image method, closely followed by the same hybrid model combined with Concatenation Strategy 1, also leveraging TINTO images. Despite relying on the same synthetic image method, these two configurations differ markedly in how they distribute predictive responsibility across branches: the standard CNN-KAN model places significant weight on the `cnn_branch`, suggesting that spatial patterns in the synthetic images primarily shape its predictions, whereas the Strategy 1 variant emphasizes the `kan_branch`, indicating stronger reliance on symbolic, tabular feature relationships. This shift is linked to the architectural choice in Strategy 1 of employing a smaller bottleneck dimension, as detailed in Table 4.4, which appears to encourage the extraction of more nuanced tabular interactions through the KAN pathway and thus influences both performance and interpretability.

	RMSE	% **Relevance** `kan_branch`	% **Relevance** `cnn_branch`	**Top 3 Feature Relevance** `kan_branch`	**Top 3 Feature Relevance** `cnn_branch`
Best Author's Model	0.21262	-	-	-	-
Best KAN	0.19097	-	-	30Y-CMortgageRate bankCredit tradeCurrencies	-
Best CNN with IGDT	0.24152	-	-	-	loansLeases demandDeposits **savingsDeposits**
Best CNN-KAN with TINTO	**0.17915**	19.8	80.2	**3Y-CMaturityRate** **federalFunds** 30Y-CMortgageRate	currency **savingsDeposits** loansLeases
Best CNN-KAN with TINTO & Strategy 1	0.18825	98.1	1.9	**3Y-CMaturityRate** checkableDeposits **federalFunds**	bankCredit 3M-Rate-SecondaryMarket **3Y-CMaturityRate**

Table 4.8: Summary of top configurations, branch relevance, and feature importance across architectures and image methods for the Treasury dataset. Bold values indicate the best RMSE overall and features repeatedly identified as highly relevant across models.

In contrast, the CNN model using IGTD yielded the worst RMSE, and was notably the only model that failed to outperform the baseline model provided by the original

[1] `https://6sy666.github.io/TALENT-Results/`

dataset authors. This highlights the limited standalone effectiveness of the CNN in the absence of feature-rich spatial representations in this dataset. However, when integrated within a CNN-KAN hybrid model, the CNN features extracted from TINTO contributed positively, suggesting that spatial representations created by this method better capture complementary patterns that the CNN-KAN model can effectively leverage.

From a feature relevance perspective, certain variables consistently surfaced across top-performing configurations—and these often align with the most correlated features in the dataset (Table 1). In particular, `3Y-CMaturityRate` followed by `bankCredit`, and `federalFunds` are all among the highest-ranked features by correlation and appear repeatedly as highly relevant, especially within the KAN branches. On the other hand, `savingsDeposits` shows context-specific importance, depending on the branch and image method. Overall, this analysis reveals that for this dataset, the base CNN-KAN hybrid better exploits spatial and symbolic representations, improving the RMSE.

4.4.2 Summary Results for Puma

Table 4.9 provides a summary of the best-performing configurations for the Puma dataset. The best result was obtained by the CNN-KAN hybrid model incorporating Concatenation Strategy 2 with the IGTD image method, reaching an RMSE of **3.21585**. This slightly outperformed both the base CNN-KAN model and the CNN model, and notably surpassed the best author's baseline. Both CNN-KAN hybrid configurations showed a clear dominance of the `cnn_branch`, highlighting the pivotal role of the image-based representation for this task.

Interestingly, all top-performing models—CNN, CNN-KAN hybrid, and CNN-KAN hybrid with concatenation—used IGTD as their image method, indicating its superior ability to encode useful spatial representations for the Puma dataset. This effectiveness may stem from the dataset's characteristics: it contains only 8 input features, and some exhibit low correlations. IGTD, which preserves structural and neighborhood relations through its layout, allowing the `cnn_branch` to extract meaningful patterns from relatively low-dimensional data.

In terms of feature relevance, the Puma dataset reveals notable consistency across models, with multiple features appearing among the top contributors in both branches. Specifically, `theta2` and `thetad2` emerge as dominant features within the `kan_branch` across all three architectures, suggesting that these variables provide valuable symbolic signals for the models. Simultaneously, `thetad1` stands out as the only feature consistently prioritized by the `cnn_branch` in both pure CNN and hybrid CNN-KAN models, underscoring its significance in spatial representations derived from synthetic images. Nevertheless, the high `cnn_branch` relevance—exceeding 97% in the hybrid configurations—indicates that spatial information extracted through `thetad1` ultimately exerts a more dominant influence on the model's predictions, highlighting the poor balance between symbolic and spatial pathways in hybrid learning but curiously achieving the best RMSE.

	RMSE	% **Relevance** kan_branch	% **Relevance** cnn_branch	**Top 3 Feature Relevance** kan_branch	**Top 3 Feature Relevance** cnn_branch
Best Author's Model	3.25262	-	-	-	-
Best KAN	3.22794	-	-	theta3 **theta2** **thetad2**	-
Best CNN with IGDT	3.22069	-	-	-	tau2 tau1 **thetad1**
Best CNN-KAN with IGTD	3.21696	2.5	97.5	tau1 **thetad2** **theta2**	theta1 **thetad1** tau1
Best CNN-KAN with IGTD & Strategy 2	**3.21585**	1.5	98.5	**theta2** **thetad2** tau2	**thetad1** theta1 tau2

Table 4.9: Summary of top configurations, branch relevance, and feature importance across architectures and image methods for the Puma dataset. Bold values indicate the best RMSE overall and features repeatedly identified as highly relevant across models.

4.4.3 Summary Results for Forex

Table 4.10 summarizes the top-performing models for the Forex dataset, including classification accuracy, branch relevance, and top contributing features identified by the KAN and CNN branches. The highest accuracy (**0.74659**) was shared by the pure KAN model and the CNN-KAN hybrid model using Concatenation Strategy 1 with REFINED images. Slightly lower performances were observed in the base CNN-KAN hybrid and CNN models, yet all outperformed the author's baseline, demonstrating the value of combining synthetic images with symbolic modeling for this financial classification task.

As in earlier analyses, the hybrid models for the Forex dataset demonstrated a pronounced reliance on the `cnn_branch`, with contributions exceeding 98% in most configurations, underscoring the dominant role of spatial representations derived from REFINED synthetic images in driving predictive performance. Regarding feature relevance, variables such as `Bid_Close`, `Ask_High`, and `Bid_Open` appeared in a dispersed pattern across both the `kan_branch` and `cnn_branch`, suggesting that both symbolic and spatial pathways leverage these key financial indicators, albeit with varying emphasis depending on the architecture. Meanwhile, synthetic placeholders like `Ex3` and `Ex5` emerged as relevant in some CNN branches, likely because they are spatially adjacent to critical features like `Bid_Close` and `Bid_Open` in the REFINED image layout, as visualized in Figure 2.2, rather than due to any intrinsic predictive signal of their own.

4.4.4 Summary Results for Wall Robot

Table 4.11 presents the top-performing configurations for the Wall Robot dataset, including accuracy, branch relevance, and feature importance for each model architecture. Unlike other datasets, none of the proposed models were able to surpass the best author's model accuracy (0.998169). The closest performance was obtained by the base CNN-KAN hybrid configuration (**0.96153**), followed by a tie between hybrid models using concatenation strategies 2 and 3.

	Accuracy	% **Relevance** kan_branch	% **Relevance** cnn_branch	**Top 3 Feature Relevance** kan_branch	**Top 3 Feature Relevance** cnn_branch
Best Author's Model	0.733152	-	-	-	-
Best KAN	**0.74659**	-	-	**Bid_Close** Ask_Close **Ask_High**	-
Best CNN with REFINED	0.73569	-	-	-	**Bid_Close** **Bid_Open** Ex5
Best CNN-KAN with REFINED	0.74387	0.4	99.6	**Bid_Close** Ask_Close **Bid_Open**	Ex3 Bid_High **Ask_High**
Best CNN-KAN with REFINED & Strategy 1	**0.74659**	1.1	98.9	Bid_Volume Bid_High **Bid_Open**	Bid_High **Ask_High** Ask_Low

Table 4.10: Summary of top configurations, branch relevance, and feature importance across architectures and image methods for the Forex dataset. Bold values indicate the best accuracies overall and features repeatedly identified as highly relevant across models.

Despite not achieving the top accuracy, the CNN-KAN hybrid models demonstrate a consistent pattern in branch contribution. Across all CNN-KAN configurations, the cnn_branch overwhelmingly dominates, with relevance scores exceeding 85%—reaching up to 99.99%. This trend suggests that the REFINED image representations provide meaningful structural cues that are strongly leveraged by the CNN blocks. Notably, as in previous datasets, the same image method (REFINED) led to the best performance across both CNN and hybrid architectures, reinforcing its effectiveness in extracting spatial patterns even from tabular data.

From the feature relevance perspective, variable V15 stands out for its repeated appearance among the top KAN features across several configurations, despite being one of the least correlated variables in Table 4. In contrast, for the cnn_branch, it is difficult to identify consistent key features, as V9 and V11 appear in only two cases each, suggesting that no single spatial feature consistently dominates the CNN branch's importance rankings. This aligns with earlier observations in Table 4.2, indicating a more dispersed pattern of relevance in the CNN pathways compared to the symbolic branch. Overall, these findings reinforce that the KAN and CNN branches capture distinct, complementary aspects of the data, enabling the hybrid architecture to integrate both symbolic relationships and spatial patterns—even if the cnn_branch often exerts greater influence on final predictions.

4.5 Global Feature Score Analysis for Best Hybrid Models

Building on the results presented in Section 4.4, this section introduces an analysis of feature importance through the lens of the proposed Global Feature Score detailed in Section 2.10. For each dataset, a dedicated subsection reports the feature relevance results for the best-performing CNN-KAN hybrid architecture identified earlier that curiously are the one with the metric in bold. Thus in each case, a single figure presents three bar plots side by side: the first shows the feature scores obtained from the kan_branch using PyKAN's native feature_score; the second displays the average Grad-CAM scores from the cnn_branch, excluding contributions from Extra

	Accuracy	% **Relevance** kan_branch	% **Relevance** cnn_branch	**Top 3 Feature Relevance** kan_branch	**Top 3 Feature Relevance** cnn_branch
Best Author's Model	**0.998169**	-	-	-	-
Best KAN	0.93864	-	-	**V15** V20 V19	-
Best CNN with REFINED	0.92582	-	-	-	**V9** V10 Ex1
Best CNN-KAN with REFINED	**0.96153**	14.1	85.9	**V15** V24 V14	**V9** V13 V14
Best CNN-KAN with REFINED & Strategy 2	0.95787	0.01	99.99	V9 V12 V22	V3 **V11** V8
Best CNN-KAN with REFINED & Strategy 3	0.95787	13.1	86.9	V20 **V15** V19	V3 V8 **V11**

Table 4.11: Summary of top configurations, branch relevance, and feature importance across architectures and image methods for the Wall Robot Navigation dataset. Bold values indicate the best accuracies overall and features repeatedly identified as highly relevant across models.

pixels that do not correspond to original tabular features; and the third illustrates the computed Global Feature Scores, which integrate both branches' contributions into a unified relevance ranking.

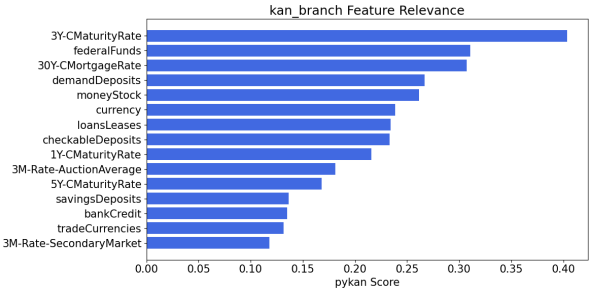
4.5.1 Global Feature Analysis for Treasury

Figure 4.5 illustrates the feature relevance for the best CNN-KAN hybrid model on the Treasury dataset, as described in Table 4.8. Since the top-performing configuration utilized the TINTO image method, the Grad-CAM scores for the `cnn_branch` show several features with identical values, a direct consequence of TINTO's dimensionality reduction, where multiple original features share the same pixel in the synthetic layout. However, the computation of the global feature score successfully resolves these ties, integrating the symbolic insights from the `kan_branch`, which contributes 20% of the model's overall relevance, as indicated in Table 4.8. This combined perspective enables a more nuanced ranking of feature importance, balancing both spatial and symbolic information.

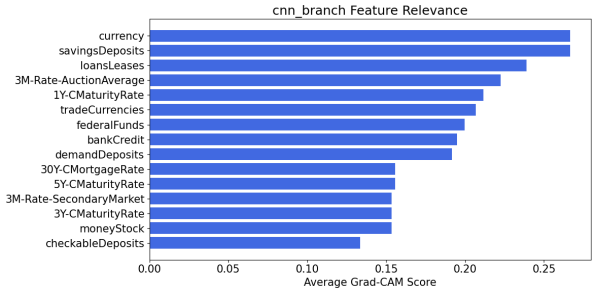
Interestingly, a comparison with Table 1 reveals that many of the top-ranked features in the global score occupy mid-to-lower positions in the correlation table, whereas features with high absolute correlations often appear further down the global ranking. This discrepancy highlights that predictive relevance in the hybrid architecture is not solely determined by raw statistical correlations. Instead, it reflects the model's ability to capture complex, possibly non-linear patterns through both symbolic reasoning and spatial representations, underscoring the added value of hybrid architectures for explainability and robust feature importance estimation.

4.5.2 Global Feature Analysis for Puma

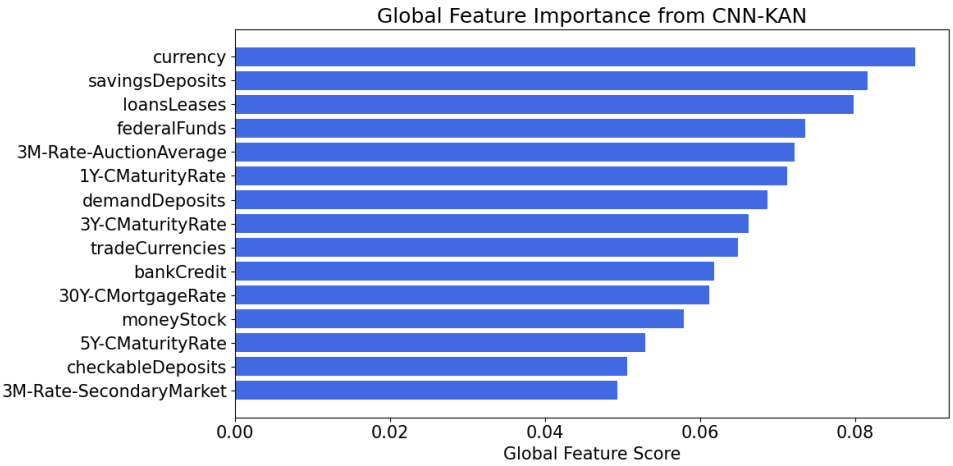
Figure 4.6 presents the feature relevance plots for the best-performing hybrid configuration on the Puma dataset, corresponding to the CNN-KAN hybrid model using



(a) Feature relevance from `kan_branch` (pyKAN scores).



(b) Feature relevance from `cnn_branch` (Grad-CAM scores).

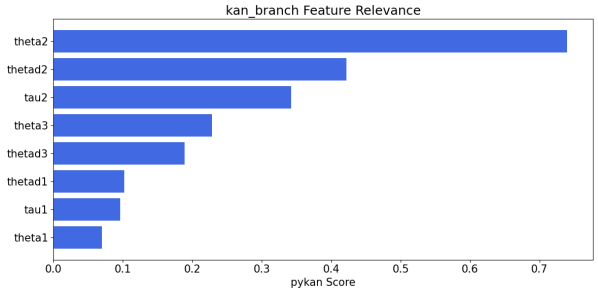


(c) Global feature scores for the CNN-KAN hybrid model.

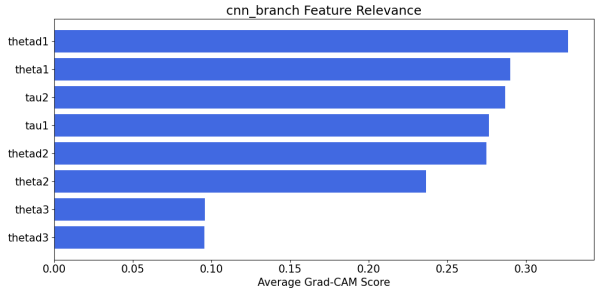
Figure 4.5: Visualization of feature relevance for the best CNN-KAN hybrid model on the Treasury dataset. The top charts show feature importance separately for the symbolic (`kan_branch`) and spatial (`cnn_branch`) pathways. The bottom chart presents the aggregated global feature scores, integrating both branches according to Global Feature Formula.

Concatenation Strategy 2 with IGTD synthetic images as indicated in Table 4.9. As shown in that table, the `cnn_branch` accounts for over 98% of the model's total relevance in this configuration. Consequently, the feature relevance plot for the CNN branch closely resembles the final global feature score plot, reflecting how spatial patterns learned through Grad-CAM dominate the integrated importance ranking for this particular hybrid setup.

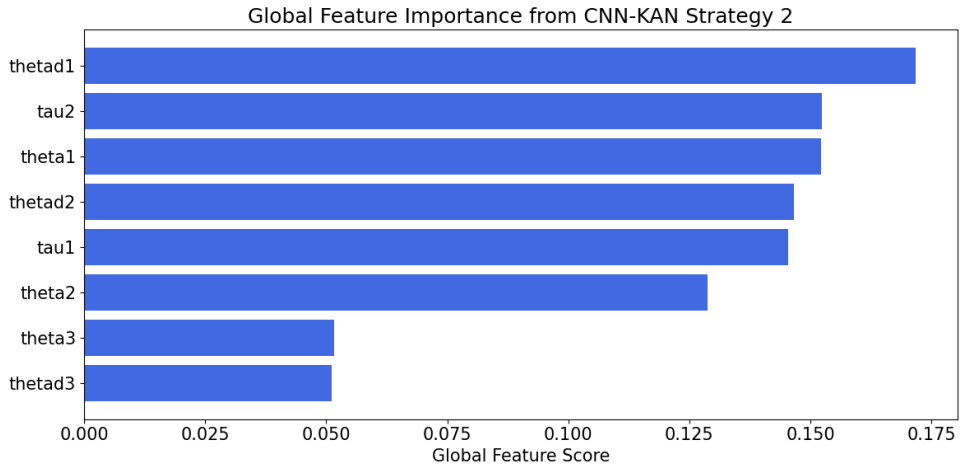
Regarding individual features, no consistent pattern emerges between correlation levels between Table 2 and the global feature rankings. Variables that sit near the top of the correlation table—such as `tau2` and `theta2`—appear intermixed with less correlated variables like `thetad1` or `thetad3` across both the highest and lowest positions in the global scores. This dispersion suggests that, for the Puma dataset, neither high nor low feature correlation guarantees prominence in the final hybrid model's reasoning. Instead, the CNN-KAN architecture captures complex and potentially non-linear relationships, allocating importance based on learned patterns rather than purely statistical associations.



(a) Feature relevance from `kan_branch` (pyKAN scores).



(b) Feature relevance from `cnn_branch` (Grad-CAM scores).



(c) Global feature scores for the CNN-KAN hybrid model.

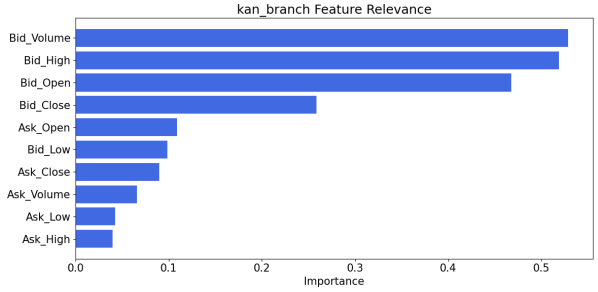
Figure 4.6: Visualization of feature relevance for the best CNN-KAN hybrid model on the Puma dataset. The top charts show feature importance separately for the symbolic (`kan_branch`) and spatial (`cnn_branch`) pathways. The bottom chart presents the aggregated global feature scores, integrating both branches according to Global Feature Formula.

4.5.3 Global Feature Analysis for Forex

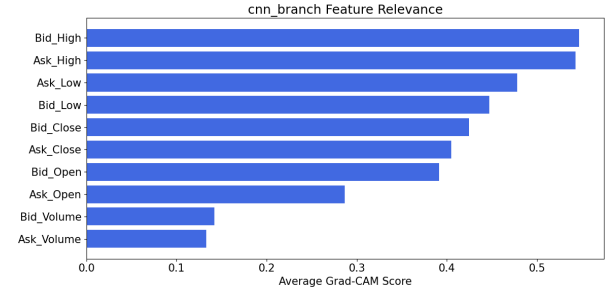
Figure 4.7 displays the feature relevance plots for the best-performing hybrid model on the Forex dataset, corresponding to the CNN-KAN hybrid configuration applying Concatenation Strategy 1 with REFINED synthetic images as shown in Table 4.10. Similar to the pattern observed in the Puma dataset, the `cnn_branch` dominates the model's reasoning in Forex, contributing over 98% of the total relevance. This strong dominance leads to a high similarity between the CNN feature relevance graph and the final global feature score plot, indicating that the global scores are heavily driven by the spatial representations learned by the convolutional layers.

Regarding feature importance, a comparison with Table 3 reveals that the two features with the lowest average correlation also appear at the bottom of the global feature scores, suggesting some consistency for those variables. However, the remaining features show a dispersed pattern, with high- and mid-correlation variables scattered across both the upper and lower ranks of the global scores. This highlights that, while certain low-correlation features consistently carry less weight, the CNN-

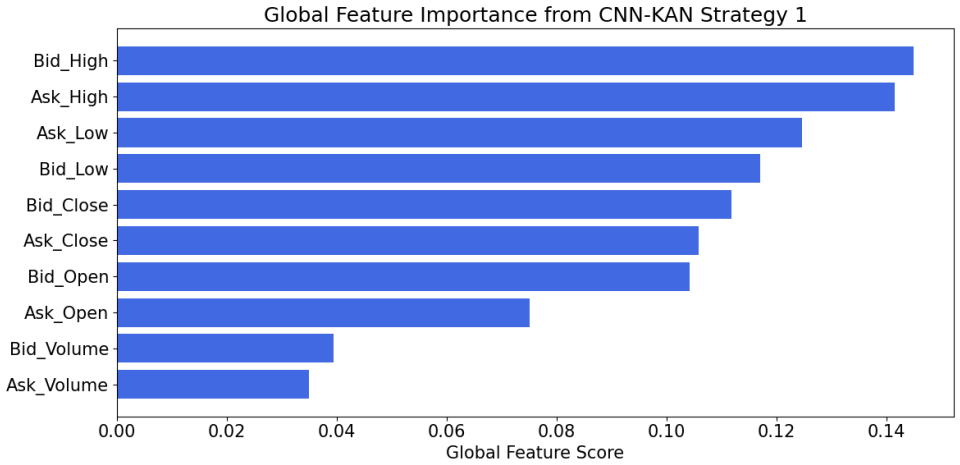
KAN model's global relevance allocation reflects more complex learned dependencies beyond simple statistical correlations.



(a) Feature relevance from `kan_branch` (pyKAN scores).



(b) Feature relevance from `cnn_branch` (Grad-CAM scores).

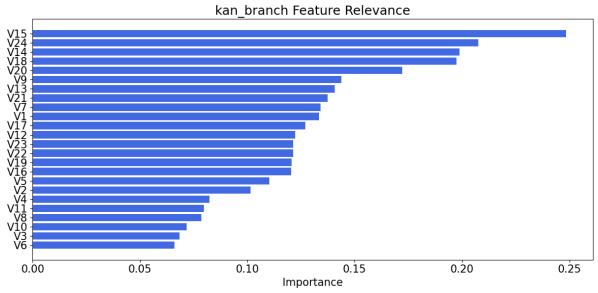


(c) Global feature scores for the CNN-KAN hybrid model.

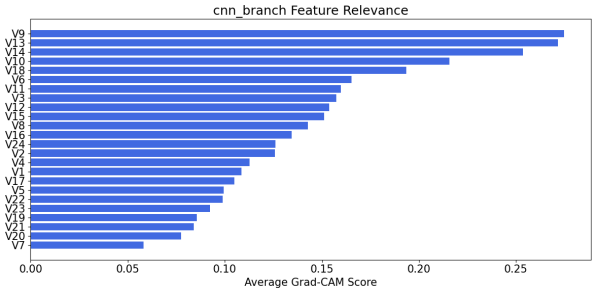
Figure 4.7: Visualization of feature relevance for the best CNN-KAN hybrid model on the Forex dataset. The top charts show feature importance separately for the symbolic (`kan_branch`) and spatial (`cnn_branch`) pathways. The bottom chart presents the aggregated global feature scores, integrating both branches according to Global Feature Formula.

4.5.4 Global Feature Analysis for Wall Robot

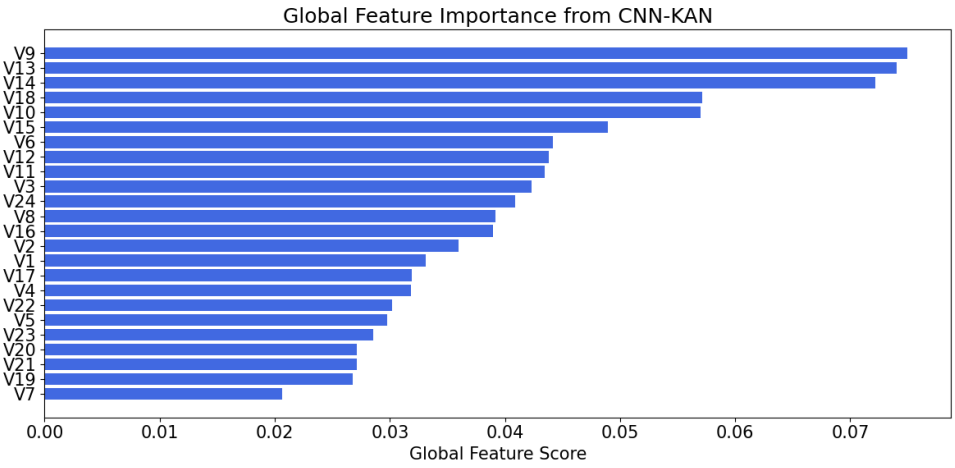
Finally, for the Wall Robot dataset, as shown in Table 4.11, the CNN branch overwhelmingly dominates in the best hybrid model with more than 85% relevance, resulting in the Grad-CAM feature relevance plot closely mirroring the final Global Feature Scores presented in Figure 4.8. Despite this strong spatial influence, a comparison with the correlation results in Table 4 reveals that while `V9` stands out as a consistently important feature across both the global scores and correlation rankings, the remaining features appear distributed without a clear pattern linking statistical correlation to feature relevance in the hybrid model. This further underscores the conclusion that classical correlation measures do not directly translate into importance within deep hybrid architectures, where feature interactions and model-specific learning dynamics dictate which variables ultimately drive predictions.



(a) Feature relevance from `kan_branch` (pyKAN scores).



(b) Feature relevance from `cnn_branch` (Grad-CAM scores).



(c) Global feature scores for the CNN-KAN hybrid model.

Figure 4.8: Visualization of feature relevance for the best CNN-KAN hybrid model on the Wall Robot dataset. The top charts show feature importance separately for the symbolic (`kan_branch`) and spatial (`cnn_branch`) pathways. The bottom chart presents the aggregated global feature scores, integrating both branches according to Global Feature Formula.

4.6 Summary of Insights and Discussion

Throughout this chapter, valuable insights have been gained into the behavior and explainability of the proposed CNN-KAN hybrid models. The graphical visualizations using Grad-CAM highlighted how, regardless of the synthetic image method, extra synthetic pixels often appear prominently in feature rankings. This effect arises from factors such as spatial proximity to real features and the limited capacity of relatively shallow CNN blocks to discriminate effectively within large images, particularly those produced by methods like TINTO. These observations relate to Hypotheses **H2** and **H3** (see Section 1.3), indicating that while CNNs can capture spatial patterns (**H2**), specific synthetic image designs (such as TINTO) can introduce artifacts that challenge explainability (**H3**).

In the broader analysis of feature relevance across IGTD, REFINED, and TINTO, it became evident that consistent patterns in top-ranked features are dataset- and branch-specific. For example, in the Puma dataset, certain features consistently surfaced as highly relevant in both branches, while in datasets like Wall Robot or

Treasury, identifiable patterns emerged only in a single branch, if at all. Moreover, the correlation study conducted earlier did not reveal a clear or consistent relationship between statistical correlations and the features ranked as important by the hybrid models, further emphasizing that feature importance in these architectures is strongly influenced by both data characteristics and architectural nuances. These findings support Hypothesis **H4** (see Section 1.3), demonstrating that interpretability tools and feature attribution can uncover nuanced behavior in hybrid architectures that traditional correlation analyses cannot fully explain.

The branch relevance analysis revealed that, in most cases, the CNN branch dominated the overall importance, prompting the investigation of concatenation strategies aimed at achieving a more balanced hybrid architecture while improving RMSE or accuracy metrics. Despite these efforts, the best-performing configurations often returned to an unbalanced state, with the CNN branch frequently retaining the majority of the relevance. Furthermore, more complex concatenation strategies, while sometimes beneficial for performance, required significantly longer training times to converge to optimal solutions. This partially validates Hypothesis **H5** (see Section 1.3), showing that while advanced concatenation mechanisms can enhance balance and performance, they also introduce practical challenges such as computational costs and convergence stability.

Finally, when comparing the best architectures identified for each dataset, the CNN-KAN hybrids—with or without concatenation strategies—demonstrated strong performance, surpassing even the best author-provided baseline in three of the four cases. This confirms again Hypotheses **H1** and **H2**, indicating that replacing traditional MLPs with KAN modules improves interpretability (**H1**) and that integrating CNNs trained on synthetic images provides substantial performance gains (**H2**). The introduction of the Global Feature Score further enriched the analysis, providing an aggregated perspective on feature importance. As anticipated, these global scores closely mirrored the patterns observed in the CNN branch due to its dominant contribution of over 80% in most hybrid configurations but this also means that the ranking was not far from reality. This outcome supports Hypothesis **H6** (see Section 1.3), showing that a unified metric combining symbolic and spatial relevance can successfully bridge the gap between high performance and model transparency, thus enhancing interpretability and explainability in hybrid architectures.

Chapter 5

Conclusion and Future Work

This final chapter reflects on the key findings of the thesis and outlines directions for future work. It begins by summarizing the main conclusions drawn from the performance and explainability analysis of the proposed CNN-KAN hybrid models across diverse datasets. The insights highlight how the combination of symbolic and spatial reasoning components—enhanced with explainability tools—can lead to robust and interpretable predictions. The chapter then discusses limitations encountered throughout the study, followed by a structured set of future research opportunities that aim to extend the capabilities and generalizability of the hybrid framework.

5.1 Conclusion

This thesis set out to explore the development of hybrid neural network architectures combining CNN and KAN components for tabular data tasks, with a strong focus on balancing predictive performance and explainability. The experiments conducted across four diverse datasets yielded several important conclusions, both in terms of empirical results and methodological insights.

- The hybrid CNN-KAN architecture developed in this thesis has proven to be a highly effective modeling framework, consistently outperforming standalone CNN and KAN models in most datasets tested. Beyond its predictive performance, it offers strong interpretability and explainability, thanks to the integration of symbolic feature scores from the `pykan` components, spatial attribution via Grad-CAM in the CNN branch, and detailed branch relevance analysis. Together, these mechanisms provide valuable insights into how both symbolic and spatial pathways contribute to the hybrid model's predictions.
- The methodological framework adopted in this study provided a clear path to visualize not only the evolution of performance metrics during training but also how feature and branch relevance changed at each experimental stage. This transparency was essential for identifying strengths and weaknesses of each architectural configuration. Moreover, it was demonstrated that the proposed CNN–KAN hybrid models could achieve strong performance and reasonable balanced relevance with relatively few hyperparameters, particularly in the KAN components, underscoring the practical feasibility and efficiency of the approach.

- Despite the promising results achieved across all four datasets, it is still premature to generalize which synthetic image method—IGTD, REFINED, or TINTO—offers the best overall performance. Each dataset behaved independently, with no consistent pattern linking a particular image generator to superior outcomes. Similarly, the average correlation scores and individual feature correlations did not show clear relationships to the final model architecture, predictive performance, or the best-performing synthetic image method. These factors seem largely dataset-specific.
- An interesting and practical contribution of this work was the adaptation of the training strategy recommended by the authors of the `pykan` library. Using the LBFGS optimizer in full-batch mode with the MSE as the loss function proved highly effective not only for regression tasks but also for classification accuracy, despite common theoretical expectations suggesting other optimization strategies for classification problems.
- Regarding the concatenation strategies explored for hybrid architectures, the experiments confirmed their significant influence on balancing the contributions of the CNN and KAN branches. Strategies 1, 2, and 3 each provided performance benefits under different datasets and conditions. However, even in the best cases, the final branch relevance often remained strongly skewed toward the `cnn_branch`, rather than achieving the balanced integration initially envisioned. Strategy 4 consistently yielded the weakest results, likely due to the additional computational complexity it introduces, which, combined with the full-batch LBFGS training setup, may hinder convergence and stability.
- Another key outcome of this work was the validation of the Global Feature Score as a practical tool for ranking feature importance across hybrid models. Although the dominance of the `cnn_branch` led the global scores to resemble the Grad-CAM rankings in many cases, this similarity suggests that the Global Feature Score accurately reflects the models' learned patterns rather than diverging arbitrarily. This consistency underscores the Global Feature Score's potential as a reliable metric for interpretability and warrants further exploration on additional architectures and diverse datasets.

Availability of Code and Resources

All experiments, models, and analyses presented in this thesis have been implemented in Python notebooks, which are publicly available in the following GitHub repository:

`https://github.com/giomondragon/pykan/tree/master/gio_notebooks`

The repository includes:

- Data folders containing the datasets used in this thesis.
- Jupyter notebooks for generating synthetic images and performing grid searches across KAN, CNN, hybrid CNN–KAN architectures, and various concatenation strategies for each dataset.
- Notebooks that replicate the best experimental results and calculate the proposed Global Feature Score for each dataset.

- CSV files capturing detailed results from all grid search experiments.

This repository is intended to support reproducibility and to facilitate further research in hybrid architectures for tabular data analysis.

5.2 Limitations and Future Work

Despite the promising results of this study, several limitations remain that open up important directions for future work. These are summarized and addressed below:

- **Broader exploration of architectural parameters:** This study focused on a reduced grid of hyperparameters to demonstrate the feasibility of achieving strong performance with minimal tuning. However, several architectural elements were kept fixed—such as CNN-related aspects like depth, number of filters, and activation functions or KAN-specific parameters like `lamb_entropy` and add multiplication operations on nodes. Future work could expand this exploration to better understand how these parameters influence performance and interpretability.
- **Investigation of alternative optimization strategies:** Although the LBFGS optimizer used in conjunction with the PyKan library produced strong results, it is best suited to small datasets and comes with significant memory requirements. Its scalability to larger datasets remains uncertain. Exploring alternative optimizers such as AdamW or RMSprop—particularly in mini-batch training regimes—could provide insights into improving training efficiency and enabling the models to handle larger and more complex data.
- **Expansion of interpretability methods and adaptation of the Global Feature Score:** While Grad-CAM has been effective for visualizing feature importance in CNN branches, future work could incorporate additional interpretability techniques such as SHAP [10] or LIME [11], which provide feature attribution values suitable for ranking. Integrating these methods could not only enrich the explainability of hybrid architectures but also enable the adaptation of the Global Feature Score beyond CNN–KAN models, extending its applicability to other architectures and diverse tabular-to-image pipelines.
- **Development of advanced concatenation mechanisms:** Although several concatenation strategies were tested in this work, future research could explore more adaptive fusion techniques that apart from achieving better metric results it also truly balance the branches influence. For instance, designing a mechanism that dynamically adjusts the `alpha` scaling parameter based on the current relevance of each branch might improve balance and responsiveness during training.
- **Application of transfer learning in hybrid architectures:** Another promising direction involves leveraging transfer learning by pre-training the KAN and CNN branches as independent models on related tasks or subsets of the data. The learned parameters could then be loaded into the hybrid architecture for fine-tuning, potentially accelerating convergence, improving generalization, and reducing the computational burden of training the full hybrid pipeline from scratch.

- **Application to larger and more diverse tabular datasets:** The evaluation focused on small to medium-sized datasets with relatively low dimensionality. To test the robustness and scalability of the proposed hybrid models, future work should include larger, more complex datasets from real-world domains.
- **Exploration of alternative or enhanced image generation methods:** While this study focused on IGTD, REFINED, and TINTO for creating synthetic images, further research could investigate other tabular-to-image transformation methods or refine existing ones. For example, experimenting with different parameter settings—such as introducing blurring or varying the pixel intensity mapping in TINTO—might help capture more nuanced spatial relationships in the data.

References

[1] Y. Zhu, T. Brettin, F. Xia, *et al.*, "Converting tabular data into images for deep learning with convolutional neural networks," *Scientific Reports*, vol. 11, no. 1, p. 11 325, 2021. DOI: 10.1038/s41598-021-90923-y.

[2] O. Bazgir, R. Zhang, S. R. Dhruba, R. Rahman, S. Ghosh, and R. Pal, "Representation of features as images with neighborhood dependencies for compatibility with convolutional neural networks," *Nature Communications*, vol. 11, no. 1, p. 4391, 2020. DOI: 10.1038/s41467-020-18197-y.

[3] M. Castillo-Cara, R. Talla-Chumpitaz, R. García-Castro, and L. Orozco-Barbosa, "TINTO: Converting Tidy Data into Image for Classification with 2-Dimensional Convolutional Neural Networks," *SoftwareX*, vol. 22, p. 101 391, 2023. DOI: 10.1016/j.softx.2023.101391.

[4] J. Ferry, G. Laberge, and U. Aïvodji, "Learning Hybrid Interpretable Models: Theory, Taxonomy, and Methods," 2023. arXiv: 2303.04437 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2303.04437.

[5] F. J. Lara-Abelenda, D. Chushig-Muzo, A. M. Wägner, M. Tayefi, and C. Soguero-Ruiz, "Interpretable and multimodal fusion methodology to predict severe hypoglycemia in adults with type 1 diabetes," *Engineering Applications of Artificial Intelligence*, vol. 144, p. 110 142, 2025, ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2025.110142.

[6] M. Castillo-Cara, J. Martinez-Gomez, J. Ballesteros-Jerez, I. García-Varea, R. García-Castro, and L. Orozco-Barbosa, "MIMO-Based Indoor Localisation With Hybrid Neural Networks: Leveraging Synthetic Images From Tidy Data for Enhanced Deep Learning," *IEEE Journal of Selected Topics in Signal Processing*, vol. 19, no. 3, pp. 559–571, 2025. DOI: 10.1109/JSTSP.2025.3555067.

[7] Z. Liu, Y. Wang, S. Vaidya, *et al.*, "KAN: Kolmogorov-Arnold Networks," 2025. arXiv: 2404.19756 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2404.19756.

[8] D. Basina, J. R. Vishal, A. Choudhary, and B. Chakravarthi, "KAT to KANs: A Review of Kolmogorov-Arnold Networks and the Neural Leap Forward," 2024. arXiv: 2411.10622 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2411.10622.

[9] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626. DOI: 10.1109/ICCV.2017.74.

[10] S. Lundberg and S.-I. Lee, *A unified approach to interpreting model predictions*, 2017. arXiv: 1705.07874 [cs.AI]. [Online]. Available: https://arxiv.org/abs/1705.07874.

[11] M. T. Ribeiro, S. Singh, and C. Guestrin, *"why should i trust you?": Explaining the predictions of any classifier*, 2016. arXiv: 1602.04938 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1602.04938.

[12] F. J. Lara-Abelenda, D. Chushig-Muzo, P. Peiro-Corbacho, *et al.*, "Transfer learning for a tabular-to-image approach: A case study for cardiovascular disease prediction," *Journal of Biomedical Informatics*, vol. 165, p. 104 821, 2025, ISSN: 1532-0464. DOI: 10.1016/j.jbi.2025.104821.

[13] S. Lee, I.-Y. Kwak, K. Lee, *et al.*, "Table2Image: Interpretable Tabular Data Classification with Realistic Image Transformations," 2025. arXiv: 2412.06265 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2412.06265.

[14] H. A. Alenizy and J. Berri, "Transforming tabular data into images via enhanced spatial relationships for CNN processing," *Scientific Reports*, vol. 15, no. 1, p. 17 004, 2025. DOI: 10.1038/s41598-025-01568-0. [Online]. Available: https://doi.org/10.1038/s41598-025-01568-0.

[15] A. B. Malayeri and M. B. Khodabakhshi, "Concatenated convolutional neural network model for cuffless blood pressure estimation using fuzzy recurrence properties of photoplethysmogram signals," *Scientific Reports*, vol. 12, no. 1, p. 6633, 2022. DOI: 10.1038/s41598-022-10244-6. [Online]. Available: https://doi.org/10.1038/s41598-022-10244-6.

[16] O. N. Oyelade, E. A. Irunokhai, and H. Wang, "A twin convolutional neural network with hybrid binary optimizer for multimodal breast cancer digital image classification," *Scientific Reports*, vol. 14, no. 1, p. 692, 2024. DOI: 10.1038/s41598-024-51329-8. [Online]. Available: https://doi.org/10.1038/s41598-024-51329-8.

[17] X. Wang, L. Sun, C. Lu, and B. Li, "A Novel Transformer Network with a CNN-Enhanced Cross-Attention Mechanism for Hyperspectral Image Classification," *Remote Sensing*, vol. 16, no. 7, p. 1180, 2024. DOI: 10.3390/rs16071180. [Online]. Available: https://www.mdpi.com/2072-4292/16/7/1180.

[18] R. Talla-Chumpitaz, M. Castillo-Cara, L. Orozco-Barbosa, and R. García-Castro, "A novel deep learning approach using blurring image techniques for Bluetooth-based indoor localisation," *Information Fusion*, vol. 91, pp. 173–186, 2023, ISSN: 1566-2535. DOI: 10.1016/j.inffus.2022.10.011.

[19] V. Gómez-Martínez-Martínez, F. J. Lara-Abelenda, P. Peiro-Corbacho, D. Chushig-Muzo, C. Granja, and C. Soguero-Ruiz, "LM-IGTD: a 2D image generator for low-dimensional and mixed-type tabular data to leverage the potential of convolutional neural networks," 2024. arXiv: 2406.14566 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2406.14566.

[20] M. M. Ferdaus, M. Abdelguerfi, E. Ioup, *et al.*, "KANICE: Kolmogorov-Arnold Networks with Interactive Convolutional Elements," 2024. arXiv: 2410.17172 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2410.17172.

[21] F. Lv, Q. Wei, Y. Huang, T. Tuncer, S. Dogan, and F. Özyurt, "PoolKANNeXt: A new pooling-based Kolmogorov Arnold convolutional neural network," *Alexandria Engineering Journal*, vol. 128, pp. 144–152, 2025. DOI: 10.1016/j.aej.2025.05.073. [Online]. Available: https://doi.org/10.1016/j.aej.2025.05.073.

[22] T. Wang and Q. Lin, *Hybrid predictive model: When an interpretable model collaborates with a black-box model*, 2019. arXiv: 1905.04241 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1905.04241.

[23] G. Ranjbaran, D. R. Recupero, C. K. Roy, and K. A. Schneider, "C-shap: A hybrid method for fast and efficient interpretability," *Applied Sciences*, vol. 15, no. 2, 2025, ISSN: 2076-3417. DOI: 10.3390/app15020672. [Online]. Available: https://www.mdpi.com/2076-3417/15/2/672.

[24] J. Liu, D. Gonzalez Fernandez, M. Castillo-Cara, and R. García Castro, *TINTOlib: A Python Library for Transforming Tabular Data into Synthetic Images for Deep Neural Networks*, version v1.0.6, Jun. 2025. DOI: 10.5281/zenodo.15753688.

[25] Manuel Castillo-Cara and Raúl García-Castro and Jiayun Liu, *TINTOlib official documentation*, Link: https://tintolib.readthedocs.io/en/latest/, [Online: accessed 1-jul-2025], 2025.

[26] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, pp. 503–528, 1989. DOI: 10.1007/BF01589116. [Online]. Available: https://doi.org/10.1007/BF01589116.

Annex

Average Absolute Correlation Scores

This annex provides the detailed results of the average absolute correlation analysis performed for all four datasets examined in this thesis. As discussed in Section 3.2, these tables show the average absolute correlation value of each feature with respect to all other features in the same dataset. Although correlation did not directly determine feature importance in the hybrid models, these results offer valuable background for interpreting the observed feature relevance patterns, especially when comparing symbolic and spatial contributions across architectures.

The following tables summarize the average absolute correlation scores for:

- The Treasury dataset (Table 1)
- The Puma dataset (Table 2)
- The Forex dataset (Table 3)
- The Wall Robot dataset (Table 4)

Feature	Avg Abs Correlation
3M-Rate-AuctionAverage	0.881946
checkableDeposits	0.872843
bankCredit	0.871836
5Y-CMaturityRate	0.866276
30Y-CMortgageRate	0.844544
tradeCurrencies	0.838322
loansLeases	0.834138
3M-Rate-SecondaryMarket	0.826873
3Y-CMaturityRate	0.825911
currency	0.825009
moneyStock	0.816039
savingsDeposits	0.815735
demandDeposits	0.801799
federalFunds	0.801674
1Y-CMaturityRate	0.562426

Table 1: Average absolute correlation score per feature in the Treasury dataset.

Feature	Avg Abs Correlation
tau2	0.011663
theta2	0.010963
theta1	0.009622
theta3	0.007331
thetad2	0.007282
tau1	0.006976
thetad3	0.006474
thetad1	0.005836

Table 2: Average absolute correlation score per feature in the Puma dataset.

Feature	Avg Abs Correlation
Ask_Low	0.845501
Bid_Low	0.845448
Bid_Close	0.843024
Ask_Close	0.843024
Ask_Open	0.841695
Bid_Open	0.841624
Ask_High	0.840112
Bid_High	0.840089
Bid_Volume	0.378499
Ask_Volume	0.368663

Table 3: Average absolute correlation score per feature in the FOREX dataset.

Feature	Avg Absolute Correlation
V9	0.216733
V10	0.201590
V22	0.201490
V1	0.200496
V7	0.200118
V8	0.194734
V21	0.193048
V23	0.189889
V24	0.184586
V4	0.178936
V6	0.178921
V5	0.177320
V18	0.175391
V11	0.173374
V14	0.164253
V19	0.161838
V12	0.159364
V17	0.155663
V3	0.144565
V20	0.142113
V13	0.140473
V15	0.134190
V2	0.105729
V16	0.098139

Table 4: Average absolute correlation score per feature in the Wall Robot Navigation dataset.