Counterfactual Data Augmentation for Regression

by

Hossein Mohebbi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2025

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Data-driven modeling in real-world regression tasks often suffers from limited training samples, high collection costs, and noisy observations. While data augmentation has revolutionized fields such as computer vision and natural language processing by leveraging domain-specific symmetries, effective techniques for tabular regression remain elusive. Existing approaches, ranging from geometric interpolation to deep generative models, often fail to preserve the underlying noise structure of the data, leading to the generation of unrealistic samples that can degrade predictive performance.

This thesis proposes a novel framework called *Counterfactual Residual Data Augmentation (CRDA)*. Our method is founded on the theoretical principle of *Residual Invariance*, which posits that once a regressor has modeled the systematic component of the data, the remaining residual noise often remains stable under small perturbations of carefully selected features. We exploit this invariance to synthesize valid counterfactual samples, which are data points with perturbed features but preserved residual noise. We formalize this process through the lens of structural causal models, establishing conditions under which the residual is conditionally independent of specific feature subsets.

We provide a practical, model-agnostic algorithm that integrates feature selection heuristics and statistical safety checks to ensure augmentation is applied only when empirically beneficial. Through extensive evaluation across diverse benchmark datasets, we demonstrate that CRDA consistently reduces test error in data-scarce regimes. Specifically, our method reduces the Mean Squared Error (MSE) of Multi-Layer Perceptrons by an average of 22.9% and XGBoost regressors by 6.4%. Furthermore, comparisons against state-of-the-art baselines, including Mixup variants and diffusion-based generative models, reveal that CRDA offers a more robust and statistically grounded remedy for noise-prone, small-sample regression tasks. Finally, we provide a production-ready, open-source implementation of our framework to encourage applications in real-world tabular regression tasks.

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisor, Professor Pascal Poupart. His deep insight, patience, and continuous encouragement have been invaluable throughout my work. His guidance not only shaped this thesis but also fundamentally improved my approach to scientific research.

I would also like to extend my sincere thanks to my committee members, Yaoliang Yu and Hongyang Zhang, for their time, constructive feedback, and the insightful questions that helped refine this work.

I am deeply grateful to my colleagues and friends who made this journey memorable. A special thanks to Justin Xu, who has been a constant source of support and friendship. From working together on various projects at the Vector Institute to starting and finishing our Master's degrees side-by-side, his camaraderie has been essential. I also wish to thank Mohammed Abdulrahman for his collaboration and insights during our joint projects with Google.

Dedication

To my father,
who always believed in me and inspired me to pursue the path of academia and research.

To my mother,
for being my home, whose quiet sacrifices paved the way for this journey.

To my brother and my sister,
for their unconditional love and support throughout all my schooling.

And lastly to the beautiful people of Iran,
you are the roots that hold me upright, my heart beats with yours.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The rapid advancement of machine learning (ML) over the past decade has revolutionized fields ranging from computer vision to natural language processing. Central to this success is the availability of massive, high-quality datasets. However, in many high-stake domains, such as medicine, finance, and manufacturing, the data landscape is fundamentally different. These industries rely heavily on tabular data, structured data organized in rows and columns, where collecting large-scale labeled examples is often expensive, impractical, or ethically constrained.

While deep learning architectures have achieved superhuman performance in interpreting tasks, their application to tabular regression is frequently hindered by data scarcity and noisy observations. Unlike image or text data, where semantic-preserving augmentations are intuitive and well-established, tabular data lacks a universal set of invariant transformations. Consequently, supervised learning models in these domains often fail to fully capture the underlying behavior of real-world processes when training samples are limited.

1.1 The Problem

The central challenge described in this thesis is the "small data regime" of tabular regression. In such settings, the cost of acquiring new data points may be very difficult. For instance, in clinical trials, each data point represents a patient and a potentially costly or invasive procedure. In manufacturing, obtaining a labeled sample might require destroying a prototype during stress testing. When dataset sizes are too small, powerful regressors are

prone to overfitting and often learn noise rather than learning the underlying systematic components, leading to poor generalization on unseen data.

In fields like Computer Vision (CV) and Natural Language Processing (NLP), this data scarcity is often addressed through *data augmentation*. By applying transformations such as rotation, cropping, or back-translation, practitioners can artificially expand datasets because these domains possess known invariant symmetries (e.g., a rotated cat is still a cat). However, tabular regression lacks these obvious symmetries. Arbitrarily perturbing features in a table can destroy the semantic relationship between the inputs and the target, resulting in synthetic data that degrades, rather than improves, model performance. Existing solutions, such as deep generative models (GANs or VAEs), often struggle in this domain as well; they prioritize learning the joint probability distribution, but frequently fail to capture the precise conditional relationships required for accurate regression.

1.2 Proposed Approach

To address these limitations, we propose a novel technique: **Counterfactual Residual Data Augmentation (CRDA)**. This method is a model-agnostic framework designed to bolster regression performance under small data constraints. Unlike geometric augmentation methods that assume local linearity, or generative models that hallucinate entire samples, CRDA leverages the statistical structure of the prediction errors themselves.

The core intuition behind CRDA rests on the decomposition of an outcome into a *systematic component* (what the model can learn) and a *residual component* (the unexplained noise). To illustrate, consider the task of predicting house prices. A regression model might capture systematic drivers of value, such as location and square footage. The residual, or error, captures unobserved factors, such as a "bidding war" driven by a specific buyer's urgency. Our key insight is that this residual noise often exhibits *invariance* to perturbations in certain features. For instance, modifying a secondary feature, such as the finish of the garage floor, will change the systematic price of the house, but it is unlikely to alter the specific buyer's urgency (the residual). CRDA exploits this independence to synthesize valid *counterfactuals*, meaning we can generate a new training sample representing the same house with a different garage finish, updated with the new systematic price, while preserving the exact same "bidding war" residual.

By formalizing this intuition into a *Residual Invariance Principle*, CRDA allows us to systematically generate new, realistic training samples that respect the underlying data-generating process (DGP), effectively expanding the dataset without requiring additional real-world observations.

1.3 Thesis Contributions

The primary contributions of this thesis are as follows:

1. **A New Data Augmentation Framework:** We propose CRDA, a model-agnostic algorithm that synthesizes data by perturbing features while preserving instance-specific residuals. This method includes built-in safety mechanisms to ensure augmentation is only applied when beneficial.

2. **The Residual Invariance Principle:** We formalize how specific feature subsets can be perturbed without affecting the noise structure and under what conditions. We provide a causal interpretation of this principle and analyze how unobserved confounding can threaten its validity.

3. **Empirical Validation:** We conduct extensive experiments across nine benchmark datasets from standard repositories (UCI, PMLB, Kaggle). We demonstrate that CRDA consistently outperforms state-of-the-art competitors, including geometric augmentation methods and deep generative models. On average, our method yields a 22.9% reduction in Mean Squared Error (MSE) for MLP regressors and a 6.4% reduction for XGBoost models.

4. **Open Source Implementation:** To bridge the gap between theory and practice, we release `crda`, a production-ready Python package available on PyPI. This library provides a `scikit-learn` compatible interface for the proposed algorithm, enabling seamless integration into existing machine learning pipelines for applied research.

1.4 Thesis Outline

The remainder of this thesis is organized as follows:

- **Chapter 2** provides the necessary background on supervised regression, Structural Causal Models (SCMs), and the concept of counterfactuals. It establishes the mathematical notation and foundational concepts used throughout the work.

- **Chapter 3** surveys the landscape of data augmentation. We categorize existing approaches into heuristic methods, geometric interpolations, and deep generative models, discussing their limitations in the context of tabular regression.

- **Chapter 4** details the CRDA methodology. We present the formal problem definition, the proof of the Residual Invariance Principle, and a step-by-step breakdown of the algorithm, including the feature selection and safety-check mechanisms.
- **Chapter 5** outlines the experimental protocol. We describe the datasets, the baseline models (XGBoost and MLP), the competing augmentation methods, and the evaluation metrics used to validate our approach.
- **Chapter 6** presents a comprehensive analysis of the results. Beyond standard performance metrics, we include a deep-dive sensitivity analysis of the algorithm's hyperparameters, an ablation study of its components, and a validation of the independence assumptions.
- **Chapter 7** summarizes the findings, discusses the limitations regarding unobserved confounders, and suggests directions for future research.

Chapter 2

Background

This chapter establishes the fundamental theoretical frameworks upon which this thesis is built. We begin by defining the problem of supervised regression and the challenges inherent to learning from limited data, specifically the bias-variance trade-off. We then introduce Structural Causal Models (SCMs), providing the necessary formalism for understanding interventions and counterfactuals. Finally, we review the principles of data augmentation, contrasting its straightforward application in perceptual domains (like computer vision) with the unique difficulties presented by tabular data.

2.1 Supervised Regression

Supervised regression is the task of learning a mapping from input features to a continuous target variable. Formally, let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ be a dataset of n independent and identically distributed (i.i.d.) observations, where $\mathbf{x}_i \in \mathbb{R}^d$ represents the feature vector of dimension d and $y_i \in \mathbb{R}$ represents the continuous target label.

We assume these observations are generated by an underlying process:

$$Y = g(\mathbf{X}) + \epsilon \tag{2.1}$$

where $g : \mathbb{R}^d \to \mathbb{R}$ represents the systematic component of the relationship (the signal), and ϵ represents the stochastic noise or irreducible error, often assumed to have mean zero, $\mathbb{E}[\epsilon] = 0$.

The goal of a regression model $f_\theta(\mathbf{x})$, parameterized by θ, is to approximate the true function $g(\mathbf{x})$ by minimizing a specified loss function $\mathcal{L}$ over the training data:

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}\left(y_i, f_\theta(\mathbf{x}_i)\right) \tag{2.2}$$

In this work, we focus primarily on the **Mean Squared Error (MSE)**, defined as:

$$\mathcal{L}_{\text{MSE}}(y, \hat{y}) = (y - \hat{y})^2 \tag{2.3}$$

2.1.1 The Small Data Regime and Overfitting

The performance of data-driven models is heavily constrained by the sample size n. In the *small data regime*, where n is small relative to the dimensionality d or the complexity of the function class f_θ, models are prone to **overfitting**.

Overfitting occurs when the model learns to capture the stochastic noise ϵ specific to the training set $\mathcal{D}_{\text{train}}$, rather than the generalizable signal $g(\mathbf{x})$. This phenomenon can be analyzed through the lens of the bias-variance decomposition. For a test point $\mathbf{x}_0$, the expected squared prediction error is:

$$\mathbb{E}\left[(Y - \hat{f}(\mathbf{x}_0))^2\right] = \underbrace{\left(\mathbb{E}[\hat{f}(\mathbf{x}_0)] - g(\mathbf{x}_0)\right)^2}_{\text{Bias}^2} + \underbrace{\mathbb{E}\left[(\hat{f}(\mathbf{x}_0) - \mathbb{E}[\hat{f}(\mathbf{x}_0)])^2\right]}_{\text{Variance}} + \underbrace{\sigma_\epsilon^2}_{\text{Irreducible Error}} \tag{2.4}$$

In limited data settings, complex models (like neural networks or gradient-boosted trees) often exhibit low bias but high variance, meaning their predictions fluctuate wildly depending on the specific subset of data observed. Data augmentation is in essence a regularization technique designed to reduce this variance by artificially increasing the sample size n.

2.2 Structural Causal Models and Counterfactuals

Although standard regression captures the conditional expectation $\mathbb{E}[Y|\mathbf{X}]$, this is purely a correlation-based representation. We can turn to causality for another way to represent and interpret the underlying data-generating process.

2.2.1 Structural Causal Models (SCMs)

A Structural Causal Model (SCM) [21, 23] describes a system of variables and the causal mechanisms that determine their values. Formally, an SCM is a tuple $\mathcal{M} = (\mathcal{X}, \mathcal{Z}, F, P_{\mathcal{Z}})$, where:

- $\mathcal{X} = \{X_1, \ldots, X_m\}$ is the set of **endogenous** (observed) variables.
- $\mathcal{Z} = \{Z_1, \ldots, Z_m\}$ is the set of **exogenous** (noise) variables, which are determined by factors outside the model. $P_{\mathcal{Z}}$ defines a probability distribution over these background variables.
- $F = \{f_1, \ldots, f_m\}$ is a set of structural equations, where each f_i determines the value of X_i based on its causal parents $Pa_i \subseteq \mathcal{X} \setminus \{X_i\}$ and its specific noise term Z_i:

$$X_i \leftarrow f_i(Pa_i, Z_i) \tag{2.5}$$

The relationships defined by F induce a directed graph where edges represent direct causal influence. Unlike standard algebraic equations, the assignment operator $\leftarrow$ implies asymmetry: changing Pa_i changes X_i, but changing X_i does not affect Pa_i.

2.2.2 Interventions

An **intervention**, denoted by the $do(\cdot)$ operator, represents an external manipulation of the system that forces a variable to take a specific value, independent of its structural parents [21].

Applying the intervention $do(X_i = x')$ replaces the original structural equation $X_i \leftarrow f_i(Pa_i, Z_i)$ with the constant assignment $X_i \leftarrow x'$. In the causal graph, this operation removes all incoming edges to X_i, severing it from its natural causes while preserving its influence on its descendants. This allows us to predict the behavior of the system under hypothetical scenarios that may differ from the observed data distribution.

2.2.3 Counterfactuals

Counterfactual reasoning allows us to answer retrospective "what-if" questions: *"Given that we observed outcome $Y = y$ for individual i, what would Y have been if X had been different?"*

Unlike simple interventions, which describe population-level averages, counterfactuals are specific to a particular instance (or unit) because they condition on the observed evidence. Calculating a counterfactual involves three steps [21]:

1. **Abduction:** Use the observed data $(\mathbf{x}, y)$ and the structural equations to infer the posterior distribution (or specific values) of the exogenous noise variables $\mathbf{z}$. This step "anchors" the noise to the specific reality of the observed instance.
2. **Action:** Modify the structural model $\mathcal{M}$ by applying the intervention $do(X = x')$, creating a modified model $\mathcal{M}_{x'}$.
3. **Prediction:** Compute the value of Y in the modified model $\mathcal{M}_{x'}$ using the exogenous noise $\mathbf{z}$ inferred during the abduction step.

2.3 Data Augmentation

Data augmentation is the process of creating new training samples from existing ones to improve model generalization. All augmentations leverage and inject some form of prior knowledge about the task's invariances to achieve this.

2.3.1 Invariance and Transformation

Ideally, an augmentation function $\mathcal{T}(\mathbf{x}, y) \to (\mathbf{x}', y')$ produces a new sample that remains valid under the true data distribution $P(X, Y)$.

- **Label-Preserving Augmentations:** In classification, we often seek transformations $\mathbf{x} \to \mathbf{x}'$ such that the label remains unchanged $(y' = y)$. For example, changing the colour of a car, does not change the fact that it is a car.
- **Label-Transforming Augmentations:** In regression, altering the input $\mathbf{x}$ typically requires a corresponding change in the target y. For instance, scaling the square footage of a house $(\mathbf{x})$ should fundamentally increase its price (y).

2.3.2 Challenges in Tabular Data

While augmentation is standard practice in unstructured data domains like Computer Vision (CV) and Natural Language Processing (NLP), it remains difficult for tabular regression due to the lack of known symmetries.

1. **Lack of Spatial/Temporal Structure:** Unlike pixels in an image or words in a sentence, tabular features (e.g., `Age`, `Income`, `BloodPressure`) do not have standard spatial relationships. Operations like "cropping" or "rotating" have no semantic meaning.
2. **Heterogeneous Features:** Tabular data often contains a mix of continuous, ordinal, and categorical variables with vastly different scales and distributions.
3. **Manifold Intrusion:** Simple interpolation techniques assume the data lies on a convex hull. In non-linear regression manifolds, interpolating between two valid points might land in a "low-density" region that represents an impossible state (e.g., a patient with the blood pressure of a child but the age of a senior).

This thesis addresses these limitations by taking inspiration from the counterfactual framework defined in Section 2.2.3 to perform augmentations that respect the underlying noise structure of the data. We interpret the regression residual as a proxy for the exogenous noise Z (Abduction), perturb the input features (Action), and recombine them to form a new valid sample (Prediction).

Chapter 3

Related Works

This chapter surveys the existing landscape of data augmentation strategies for tabular data. We categorize these approaches into heuristic interpolation methods, geometric manifold learning, deep generative modeling, and causal inference techniques. We conclude by situating our proposed method, Counterfactual Residual Data Augmentation (CRDA), within this landscape as a solution to the specific limitations of existing approaches in regression settings.

3.1 Heuristic and Interpolation-Based Augmentation

Early approaches to tabular augmentation primarily addressed the problem of class imbalance in classification tasks. The most prominent example is the Synthetic Minority Over-sampling Technique (SMOTE) [5]. SMOTE generates synthetic samples by interpolating linearly between a minority class sample and its k-nearest neighbors in feature space.

Extensions of this logic to regression tasks include SMOTE for Regression (SMOTER) [29] and SMOGN [4]. SMOGN combines over-sampling with the introduction of Gaussian noise, aiming to interpolate regions of the target variable Y that are under-represented (e.g., rare extreme values in housing prices).

While effective for balancing distributions, interpolation methods rest on the assumption of local linearity between samples. In complex regression manifolds where the decision boundary is highly non-linear or where the feature space is sparse, linear interpolation can generate synthetic points that fall off the data manifold, a phenomenon often referred to

as "manifold intrusion." Furthermore, these methods typically focus on the inputs X and interpolate Y deterministically, failing to account for the heteroskedastic noise inherent in real-world processes.

3.2 Geometric and Manifold-Based Approaches

A more recent wave of research attempts to formalize augmentation through geometric properties of the data manifold. A foundational concept here is *Mixup* [36], originally proposed for vision, which trains models on convex combinations of pairs of examples and their labels:

$$\tilde{x} = \lambda x_i + (1 - \lambda)x_j, \quad \tilde{y} = \lambda y_i + (1 - \lambda)y_j \tag{3.1}$$

While Mixup acts as an effective regularizer, naive application to regression can be detrimental if samples with vastly different target values are mixed, leading to unrealistic label assignments.

3.2.1 Manifold-Aware Adaptations

Several adaptations seek to constrain Mixup to valid regions of the tabular manifold:

- **RegMix** [11] optimizes the mixing policy to generate samples within estimated high-density regions of the data distribution, aiming to preserve the underlying structure.
- **C-Mixup** [34] addresses the risk of manifold intrusion by dynamically adjusting the sampling probability based on label similarity. It restricts mixing to pairs (x_i, x_j) that have similar y values, ensuring that the interpolated geometry remains locally consistent with the target function.

3.2.2 Anchor Regression

Moving beyond simple mixing, **Anchor Data Augmentation (ADA)** [27] extends the concept of Anchor Regression. ADA identifies "anchors" (by clustering) in the feature space and generates new samples using a first-order Taylor approximation around these anchors. This effectively assumes local linearity within clusters, however, in regimes where the underlying function is highly non-convex this assumption can fail and lead to poor augmentation.

3.3 Deep Generative Models

Deep generative models offer a different paradigm: rather than perturbing existing points, they attempt to learn the joint distribution $P(X, Y)$ and sample entirely new rows from it. This has been a vibrant area of research, particularly for privacy-preserving synthetic data generation.

3.3.1 Variational Autoencoders and GANs

Initial efforts adapted Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) to tabular data. **TVAE** and **CTGAN** [33] are state-of-the-art baselines in this category. CTGAN, for instance, uses mode-specific normalization to handle the multimodal distributions common in continuous tabular columns and employs a conditional generator to handle class imbalance.

3.3.2 Diffusion Models

More recently, diffusion models have shown superior performance in density estimation. **TabDDPM** [15] applies Denoising Diffusion Probabilistic Models to tabular data, demonstrating the ability to capture complex dependencies and higher-order feature interactions better than GANs.

While these models excel at generating realistic-looking data tables (i.e., $P(X)$ looks correct), they often struggle with the precise conditional relationship required for regression ($P(Y|X)$). By treating the target variable Y as "just another column" in the joint distribution, they fail to preserve the specific systematic-plus-noise decomposition of the data. Synthetic samples from these models often look plausible but contain "hallucinated" residuals—noise values that do not align with the features in a way consistent with the ground truth.

3.4 Residuals and Uncertainty Quantification

In the statistical literature, residuals have been leveraged extensively, though rarely for the explicit purpose of expanding training sets.

3.4.1 The Residual Bootstrap

The residual bootstrap [9] is a classical technique for estimating standard errors and confidence intervals. It involves fitting a model, calculating residuals, identifying the residual distribution, and resampling these residuals to create "bootstrap samples." While methodologically similar to our approach, the bootstrap is used almost exclusively for *inference* (uncertainty quantification) rather than *prediction* (improving model accuracy via data augmentation).

3.4.2 Conformal Prediction

Similarly, conformal prediction methods, such as Jackknife+ [3], utilize the distribution of held-out residuals to construct valid predictive confidence intervals with finite-sample guarantees. CRDA repurposes the fundamental statistical insight that residuals carry information about the noise distribution and repurposes it for augmentation.

3.5 Causal and Counterfactual Approaches

The most theoretically grounded approach to data augmentation stems from causal inference. Standard augmentation techniques often ignore the causal structure of the data generation process, which can lead to invalid samples.

3.5.1 Structural Invariance

Causal augmentation aims to preserve invariant relationships across environments [2]. For example, **CausalGAN** [14] allows sampling from specific interventional distributions given a known causal graph. In computer vision, this has been explored by intervening on object attributes (e.g., changing the background while keeping the object invariant).

3.5.2 Counterfactuals in Reinforcement Learning

Our work draws a direct parallel to theoretical results in Reinforcement Learning (RL) that investigated the identifiability of counterfactual next-state samples in dynamic systems [16]. Their work established that under mild assumptions (monotonicity and independence of

noise), the transition noise in a system is identifiable. Once the noise for a transition is fixed, it can be reused to predict the next state under a counterfactual action.

Connection to CRDA. CRDA translates this insight from RL dynamics to static tabular regression. We treat the regression residual as an exogenous noise variable in a Structural Causal Model (SCM). By verifying that this noise is independent of specific features, we can perform valid counterfactual interventions on those features while reusing the observed noise.

3.6 Summary

The literature indicates a gap in data augmentation for regression. Geometric methods rely on geometric assumptions that fail in complex manifolds, while deep generative models prioritize realistic marginal distributions over precise conditional logic. CRDA bridges this gap by applying a causal lens to the residual bootstrapping philosophy, allowing for the generation of samples that are both diverse (via feature perturbation) and statistically valid (via residual preservation).

Chapter 4

Method

In this chapter, we formally introduce **Counterfactual Residual Data Augmentation (CRDA)**, a novel framework designed to address the challenges of regression in data-scarce regimes. We begin by defining the problem formulation and the notation used throughout this work. We then establish the theoretical foundations of our approach, specifically the *Residual Invariance Principle*. Finally, we detail the algorithmic implementation of CRDA, including the feature selection heuristics and the statistical safety mechanisms employed to ensure robust augmentation.

4.1 Problem Formulation

We consider a standard supervised regression setting where we are given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ consisting of n independent and identically distributed (i.i.d.) samples. Each input vector $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$ consists of d features, and the target $y_i \in \mathcal{Y} \subseteq \mathbb{R}$ is a continuous variable.

We aim to learn a predictive function (regressor) $g : \mathcal{X} \to \mathcal{Y}$ that minimizes the expected loss, typically the Mean Squared Error (MSE), over the data distribution $P_{X,Y}$:

$$\mathcal{L}(g) = \mathbb{E}_{(\mathbf{x},y)\sim P_{X,Y}} \left[(y - g(\mathbf{x}))^2 \right]. \tag{4.1}$$

The ultimate goal is to approximate the true systematic function $g(x)$ effectively, even when the training set $\mathcal{D}$ is severely size-constrained. In this regime, minimizing the empirical risk alone often leads to overfitting, therefore, our objective is to leverage the residual

structure to generate valid auxiliary samples that improve generalization without requiring external data collection.

We propose to decompose the target variable y into a systematic component captured by the model and a stochastic residual component:

$$y_i = g(\mathbf{x}_i) + z_i, \tag{4.2}$$

where $g(\mathbf{x}_i)$ represents the conditional expectation $\mathbb{E}[Y|\mathbf{x}_i]$ (or an approximation thereof), and z_i represents the residual noise specific to instance i. This is effectively equivalent to modeling the data-generating process with an *additive-noise model (ANM)*. Our method exploits the statistical properties of z_i to generate valid counterfactual samples.

4.2 Theoretical Framework

The core theoretical contribution of CRDA is the formalization of *residual invariance*. This principle posits that in a well-specified regression model, the residual noise distribution remains stable under interventions on specific subsets of features. To formalize this, we rely on the language of Structural Causal Models (SCMs) [21].

4.2.1 The Residual Invariance Principle

Let the feature vector X be partitioned into two disjoint subsets, $X = (X_P, X_R)$, where:

- X_P: The **perturbable** features (those we intend to modify).
- X_R: The **remaining** features (those we hold fixed).

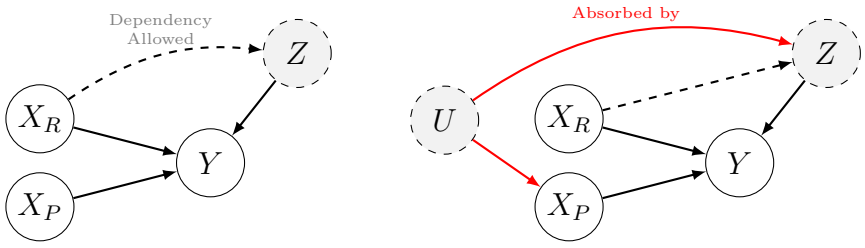
Let $g(X) = \mathbb{E}[Y|X]$ be the true conditional expectation function, and let $Z = Y - g(X)$ be the corresponding structural noise term.

Assumption 1 (Residual Invariance)**.** *We assume that the noise term Z is conditionally independent of the perturbable features X_P, given the fixed features X_R. Formally:* [1]

$$Z \perp\!\!\!\perp X_P \mid X_R \tag{4.3}$$

This assumption implies that while the noise Z may depend on X_R (e.g., heteroskedasticity where error variance scales with a fixed feature), it does not contain information unique to X_P once X_R is known. Under this assumption, we see that the noise distribution is invariant to specific perturbations.

[1] $\perp\!\!\!\perp$ denotes statistical independence.



(a) Assumption Satisfied **(b) Assumption Violated**

Figure 4.1: Causal Visualization of Residual Invariance. (a) A valid structure where Z is independent of X_P given X_R. (b) An invalid structure where a latent confounder U creates a backdoor path between X_P and Z, violating Assumption 1. White nodes are observed variables, while shaded nodes are unobserved. Dashed lines imply an optional relationship.

4.2.2 Causal Interpretation and Confounding

To better understand the validity of Assumption 1, we can examine the underlying causal structure of the data. Figure 4.1 illustrates two scenarios.

In the valid scenario (Figure 4.1a), X_P influences Y, but the residual mechanism Z is structurally independent of X_P (conditioned on X_R). This allows us to intervene on X_P without breaking the relationship between the outcome and the noise.

However, a primary driver of distribution shift failures in real-world tabular data is *unobserved confounding*[24, 26]. When a latent confounder U exists, its influence on Y that is not explained by X is absorbed into the residual term Z. Consequently, Z acts as a noisy proxy for U. Figure 4.1b illustrates the case where U causes both X_P and Y, thereby creating a "backdoor path" $X_P \leftarrow U \rightarrow Y$. Here, a statistical dependence arises between X_P and Z, violating Assumption 1.

Therefore, the success of CRDA relies on correctly partitioning features into X_P and X_R such that no unobserved confounders link X_P to Z.

4.3 The CRDA Algorithm

Based on the theoretical framework established above, we propose the Counterfactual Residual Data Augmentation (CRDA) algorithm. The procedure consists of four main phases: Baseline Training, Feature Selection, Augmentation, and Validation. The complete procedure is detailed in Algorithm 1 and Figure 4.2 illustrates the end-to-end data flow,

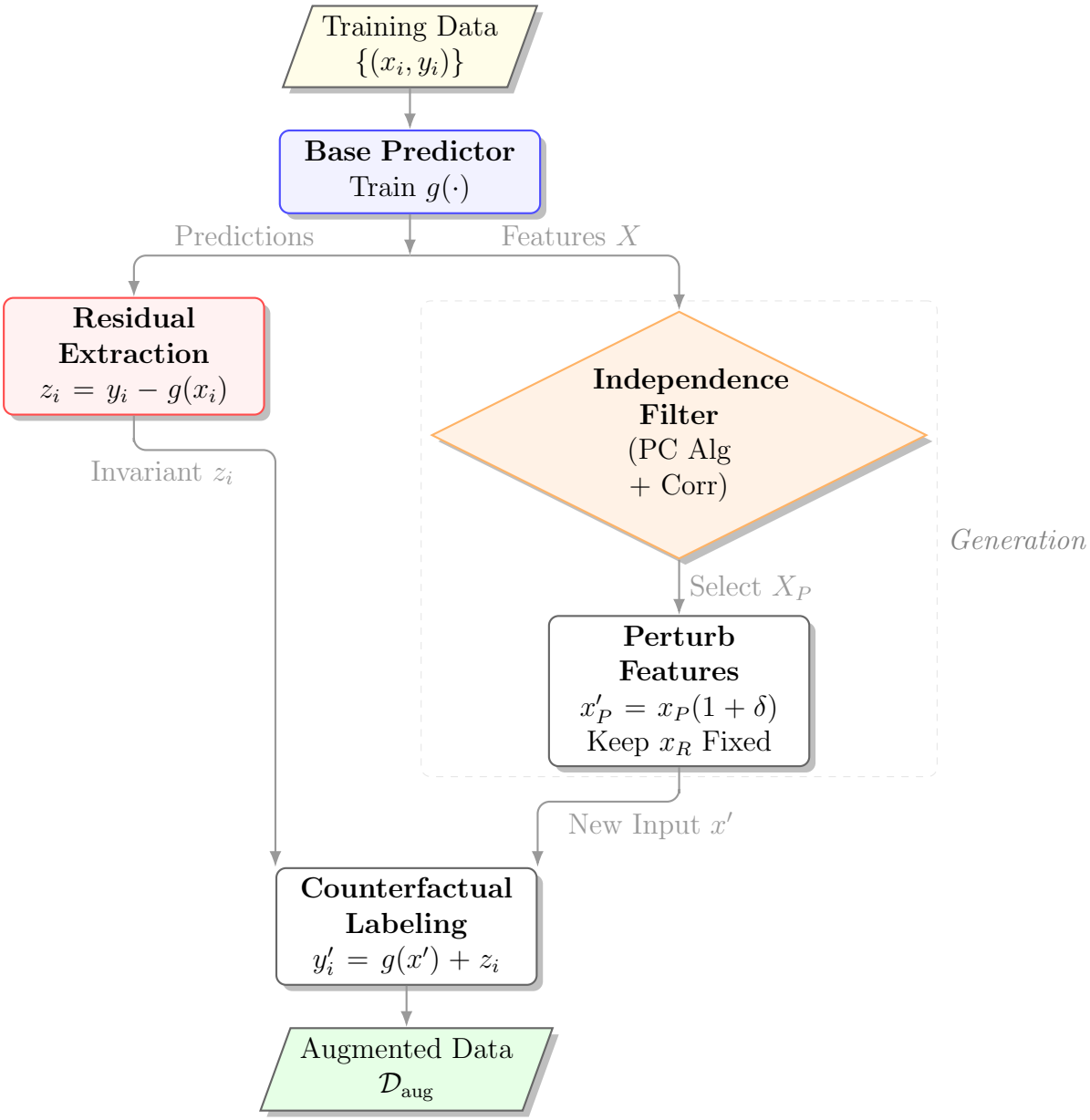


Figure 4.2: The Counterfactual Residual Data Augmentation (CRDA) Pipeline. The workflow proceeds top-to-bottom; the base model component $g(\cdot)$ first isolates the residual noise z_i. Simultaneously, the Independence Filter identifies safe features x_P. These are perturbed and recombined with the preserved residual to generate valid counterfactual samples.

showing how the systematic signal and stochastic residuals are decoupled, processed in parallel branches, and recombined to synthesize new data.

4.3.1 Phase 1: Baseline Training and Residual Extraction

We first split the available data $\mathcal{D}$ into training ($\mathcal{D}_{\text{train}}$) and testing ($\mathcal{D}_{\text{test}}$) sets. We train a baseline regressor $g(\cdot)$ on $\mathcal{D}_{\text{train}}$. In practice, this model can be chosen from a variety of families (e.g. MLP, XGBoost). Then, for every sample $(\mathbf{x}_i, y_i) \in \mathcal{D}_{\text{train}}$, we compute the residual:

$$z_i = y_i - g(\mathbf{x}_i). \tag{4.4}$$

This residual z_i serves as our empirical estimate of the noise term Z.

4.3.2 Phase 2: Feature Partitioning

To satisfy Assumption 1, we must identify which features belong to X_P (perturbable) and which must remain in X_R (fixed). Since the true causal graph is rarely known, we sequentially apply two empirical heuristics to screen for (approximate) conditional independence between features and the residual z:

1. **Causal Discovery (PC Algorithm):** We apply the Peter-Clark (PC) algorithm [28] to the joint set of variables $\{X, Y, Z\}$. Features that have a direct edge to Z in the discovered graph are flagged as dependent and assigned to X_R. The remaining features are still potential candidates for X_P.

2. **Pearson Correlation Test:** As a secondary filter, we calculate the Pearson correlation coefficient between each remainding candidate feature and z. Features with a statistically significant correlation (above a threshold) are assigned to X_R.

All features not rejected by these filters form the set X_P. If $X_P = \emptyset$, the algorithm terminates, returning the baseline model to avoid invalid augmentation.

4.3.3 Phase 3: Counterfactual Generation

For each training sample $(\mathbf{x}_i, y_i)$, we generate M synthetic samples (controlled by the hyperparameter AugDataSizeFactor). For each synthetic sample $m \in \{1, \ldots, M\}$:

1. We perturb the features in X_P using a uniform scaling factor $\delta \sim \text{Unif}[-p, p]$, where $p \in (0, 1)$ is the PERTURBATIONRANGE:

$$\mathbf{x}'_{i,P} = \mathbf{x}_{i,P} \cdot (1 + \delta). \tag{4.5}$$

2. We construct the new feature vector $\mathbf{x}'_i = (\mathbf{x}'_{i,P}, \mathbf{x}_{i,R})$, keeping X_R fixed.

3. We calculate the counterfactual label by passing the *new* input through the baseline model and adding the *old* residual:

$$y'_i = g(\mathbf{x}'_i) + z_i. \tag{4.6}$$

This process creates a new dataset $\mathcal{D}_{\text{aug}}$.

4.3.4 Phase 4: The Safety Mechanism

A critical component of CRDA is its "fail-safe" mechanism. Since the feature selection heuristics are imperfect, there is a risk that the augmentation could degrade performance (e.g., if a confounder was missed). To mitigate this, we perform a K-fold Cross-Validation (CV) on $\mathcal{D}_{\text{train}}$. In each fold, we train two models: one on the original fold data, and one on the fold data plus its CRDA-generated augmentations. We collect the validation errors for both models and perform a **Wilcoxon Signed-Rank Test** [32] on the paired errors.

If the test indicates that the augmented model's error is statistically significantly lower than the baseline ($p < \alpha$), we proceed to combine $\mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{aug}}$ and train a final model $g'(\cdot)$ on the expanded dataset to use on the test set. Otherwise, we discard the augmentations and return the baseline model. This ensures that CRDA is only applied when there is empirical evidence of its benefit.

Algorithm 1 Counterfactual Residual Data Augmentation (CRDA)

Require: Dataset $\mathcal{D}$, baseline regressor $g(\cdot)$.
Hyperparameters: PerturbationRange (p), AugDataSizeFactor (M).

Split $\mathcal{D}$ into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$.
Train baseline $g(\cdot)$ on $\mathcal{D}_{\text{train}}$.
Compute Residuals: $z_i \leftarrow y_i - g(\mathbf{x}_i)$ for all i.
Select partition (X_P, X_R):

- PC algorithm to remove features directly connected to Z.
- Correlation check to remove features strongly associated with Z.

if $X_P = \emptyset$ **then return** g ▷ No partition found
$\mathcal{D}_{\text{aug}} \leftarrow \emptyset$
for each $(\mathbf{x}_i, y_i) \in \mathcal{D}_{\text{train}}$ **do**
    for $m = 1$ to M **do**
        $\mathbf{x}'_{i,P} \leftarrow \mathbf{x}_{i,P} \cdot (1 + \text{Unif}[-p, p])$
        $\mathbf{x}'_i \leftarrow (\mathbf{x}'_{i,P}, \mathbf{x}_{i,R})$ ▷ Keep X_R fixed
        $y'_i \leftarrow g(\mathbf{x}'_i) + z_i$ ▷ Preserve residual
        $\mathcal{D}_{\text{aug}} \leftarrow \mathcal{D}_{\text{aug}} \cup \{(\mathbf{x}'_i, y'_i)\}$
Validation: Perform K-fold CV comparing *unaugmented* vs. *augmented* models.
Collect validation errors $\{e^{(k)}_{\text{unaug}}, e^{(k)}_{\text{aug}}\}_{k=1}^{K}$.
p-value $\leftarrow$ WilcoxonSignedRank$\big(\{e^{(k)}_{\text{unaug}}, e^{(k)}_{\text{aug}}\}\big)$
if p-value $< \alpha$ **then**
    Train new regressor g' on $\mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{aug}}$
    return g'
else
    return g ▷ Augmentation rejected

4.3.5 Iterative Nature & Ceiling

It is important to note that CRDA operates as an iterative refinement process rather than a standalone generator. It relies on an initial base regressor to extract the estimated residuals z that serve as proxies for the true noise. Consequently, the performance of the method is inherently coupled with the quality of this initial approximation. If the training data is so scarce that the base model fails to capture even the coarse systematic trends, the extracted residuals will contain signal leakage, rendering the augmentation ineffective. Thus, while CRDA can significantly improve sample efficiency, it has a performance ceiling dictated by the information content available in the original training set (a limitation we empirically analyze in Section 6.2).

4.4 Open Source Implementation

To encourage the adoption of our method in applied settings, we have released a production-ready implementation of Counterfactual Residual Data Augmentation as an open-source Python package. The library, named *crda*, is hosted on the Python Package Index (PyPI) and is designed to integrate seamlessly with the Scikit-Learn ecosystem, allowing the use of virtually any base regressor[2].

Our package provides a modular implementation of the CRDA pipeline described in Section 4.3, abstracting the complexity of the residual extraction, independence filtering, automated safety checks, and counterfactual generation steps.

The package can be installed directly via the package manager:

```
pip install crda
```

[2]Full documentation can be found at https://pypi.org/project/crda/

Chapter 5

Experimental Setup

To rigorously evaluate the efficacy of Counterfactual Residual Data Augmentation (CRDA), we conducted an extensive empirical study across a diverse suite of regression benchmarks. The primary objective was to assess whether CRDA could consistently reduce the Mean Squared Error (MSE) of standard regressors in data-scarce regimes without harm.

In this chapter, we detail the experimental protocol employed. We begin by describing the datasets and the preprocessing pipeline in Section 5.1. We then outline the baseline models and the state-of-the-art competitor methods used for comparison in Section 5.2. Finally, we provide a detailed account of the implementation, including the hyperparameter optimization strategies used, in Section 5.3.

5.1 Datasets

We selected nine real-world regression datasets from standard repositories including the University of California, Irvine (UCI) Machine Learning Repository [13], the Penn Machine Learning Benchmarks (PMLB) collection [20], and Kaggle [12]. These datasets were chosen to represent a variety of tabular domains (e.g., physics, biology, housing markets) and varying degrees of dimensionality (d) and sample size (n).

In addition to the real-world benchmarks, we created a synthetic dataset with a known DGP to analyze the method's behavior under controlled conditions.

5.1.1 Real-World Benchmarks

Table 5.1 summarizes the statistics of the nine real-world datasets used. The sample sizes range from as few as 768 samples (*Energy Efficiency*) to over 8,000 samples (*CPU Performance*), allowing us to test the augmentation method across different data regimes.

Table 5.1: Summary of evaluation datasets, including total sample count (n_{samples}), dimensionality (n_{features}), and source repository.

Dataset	$\mathbf{n}_{\text{samples}}$	$\mathbf{n}_{\text{features}}$	**Source**
CPU Performance	8,192	12	PMLB [17]
Satellite Image	6,435	36	PMLB [18]
Wind Power	6,574	14	UCI [10]
Synthetic Regression	1,000	10	PMLB [19]
Concrete Strength	1,005	8	UCI [35]
Energy Efficiency	768	9	UCI [31]
House Price	1,000	7	Kaggle [7]
Parkinson's Monitoring	5,875	20	UCI [30]
Wine Quality	5,318	11	UCI [8]

5.1.2 Synthetic Dataset

In order to ensure that we satisfy our residual invariance assumption, we generated data from a custom non-linear DGP defined as:

$$Y = X_1^2 + X_2 X_3 + Z, \qquad \text{where } Z \perp (X_1, X_2, X_3) \tag{5.1}$$

Here, the noise term Z is explicitly independent of the feature vector X, satisfying Assumption 1 by design. This controlled setting of known ground truth allows us to perform a sample-size scaling analysis.

5.1.3 Data Preprocessing and Splitting

To ensure consistent evaluation, we applied a standardized preprocessing pipeline to all datasets:

1. **Cleaning:** Duplicate rows and samples containing missing values (NaNs) were removed.
2. **Standardization:** All input features were standardized to have zero mean and unit variance.
3. **Data Scarcity Simulation:** To simulate varying levels of data availability, we generated five training subsets for each dataset, ranging in size from $n/5$ to n.
4. **Train-Test Split:** For every subset, data was split into 80% for training and 20% for testing.

5.2 Models and Baselines

We evaluated the performance of CRDA by applying it to standard base regressors and comparing the results against unaugmented baselines and alternative augmentation techniques.

5.2.1 Base Predictors

We utilized two distinct families of regressors as base models. These represent the most common modeling choices for tabular data:

- **Multi-Layer Perceptron (MLP):** A feed-forward neural network trained with the Adam optimizer and ReLU activations. Neural networks are often "data-hungry," making them ideal candidates for data augmentation.
- **XGBoost (XGB):** A gradient-boosted decision tree implementation [6]. Tree ensembles are generally robust on tabular data and serve as a strong baseline to improve upon.

Additionally, to test boundary cases and the robustness of the method, we conducted supplementary experiments using **Linear Regression** (a weak baseline) and **CatBoost** [25] (a strong baseline robust to overfitting).

5.2.2 Other Augmentation Methods

We benchmarked CRDA against a suite of state-of-the-art techniques for tabular data generation and augmentation:

1. **Geometric/Interpolation Methods:**
    - **C-Mixup [34]:** An adaptation of Mixup for regression that restricts interpolation to samples with similar labels to prevent manifold intrusion.
    - **Anchor Data Augmentation (ADA) [27]:** A method that clusters data points and generates samples using a first-order Taylor approximation, assuming local linearity.
2. **Deep Generative Models:**
    - **CTGAN [33]:** A Conditional Generative Adversarial Network designed for tabular data.
    - **TVAE [33]:** A Tabular Variational Autoencoder.
    - **TabDDPM [15]:** A denoising diffusion probabilistic model adapted for tabular data.

5.3 Implementation and Hyperparameters

All experiments were implemented in Python 3.11, utilizing `scikit-learn` [22] for MLP and Linear Regression, and the `xgboost` [6] library for gradient boosting. To ensure statistical reliability, all main experiments were repeated across *15 random seeds* and significance was assessed using 10-fold CV.

For hyperparameter optimization, we employed a nested tuning strategy. The base regressors were tuned first to ensure a strong baseline, followed by the tuning of CRDA-specific augmentation parameters.

Base Regressor Tuning

We tuned the MLP and XGBoost hyperparameters using `RandomizedSearchCV` [22] with 3-fold CV for 20 trials per dataset. We focused on the parameters most critical for the bias-variance trade-off while fixing standard architectural choices to textbook defaults.

MLP Regressor Configuration:

- **Fixed:** Solver (`adam`), Activation (`relu`), Batch size (32), Max iter (1,000), Early stopping (True).
- **Tuned:** Hidden layer sizes, L2 penalty (α), Initial learning rate, and Tolerance. The search spaces are detailed in Table 5.2.

Table 5.2: Search space for MLPRegressor hyperparameters.

Parameter	**Search Distribution / Grid**
`hidden_layer_sizes`	$\{(128, 64, 32), (128, 64), (64, 32), (64,)\}$
`alpha` (L2)	Log-Uniform $[10^{-5}, 10^{-3}]$
`learning_rate_init`	Log-Uniform $[10^{-3}, 10^{-2}]$
`tol`	Log-Uniform $[10^{-5}, 10^{-4}]$

XGBoost Regressor Configuration:

- **Fixed:** Objective (`reg:squarederror`), Tree method (`hist`), Estimators (1,000), Early stopping rounds (20).
- **Tuned:** Learning rate, Max depth, Min child weight, Subsample ratio, Colsample by tree, and Lambda (L2 reg). The search spaces are detailed in Table 5.3.

Table 5.3: Search space for XGBoostRegressor hyperparameters.

Parameter	**Search Distribution / Grid**
`learning_rate`	Log-Uniform $[10^{-3}, 10^{-1}]$
`max_depth`	$\{3, 4, 6\}$
`min_child_weight`	$\{1, 5\}$
`subsample`	$\{0.7, 1.0\}$
`colsample_bytree`	$\{0.7, 1.0\}$
`reg_lambda`	Log-Uniform $[10^{-3}, 10^{1}]$

CRDA Augmentation Tuning

CRDA introduces three specific "knobs" that control the augmentation process. These were tuned using the `Optuna` framework [1] with a Tree-structured Parzen Estimator (TPE)

sampling for 30 trials per dataset. The parameters and their ranges are described in Table 5.4.

Table 5.4: CRDA specific hyperparameters and their search ranges.

Parameter	**Description**	**Range**
`max_n_features`	The number of features to perturb simultaneously in a single counterfactual sample.	$\{1, 2, 3, 4, 5\}$
`aug_size_factor`	The ratio of synthetic samples generated relative to the original training set size.	$\{0.5, 0.75, 1.0, 1.25, 1.5\}$
`perturb_percent`	The maximum percentage magnitude (half-width) of the uniform perturbation applied to features.	$\{0.1, \ldots, 1.0\}$

5.4 Evaluation Metrics

The primary metric for performance is the Mean Squared Error (MSE) on the held-out test set. To quantify the improvement provided by CRDA over the baseline, we report the percentage change ($\Delta\%$):

$$\Delta\% = 100 \times \frac{\text{MSE}_{\text{CRDA}} - \text{MSE}_{\text{baseline}}}{\text{MSE}_{\text{baseline}}} \tag{5.2}$$

A negative $\Delta\%$ indicates a reduction in error (improvement). To ensure robustness, we used the Wilcoxon signed-rank test across the CV folds. The augmentation pipeline includes a significance gate (set at $\alpha = 0.05$) and if CRDA does not produce a statistically significant improvement on the validation folds, the method defaults to the baseline model to prevent performance degradation.

Chapter 6

Results and Analysis

In this chapter, we present a comprehensive evaluation of Counterfactual Residual Data Augmentation (CRDA). We begin by reporting the main predictive performance across the nine real-world benchmarks, analyzing the impact on both MLP and XGBoost regressors. We then compare our approach against state-of-the-art generative and geometric augmentation methods.

Beyond these metrics, we provide a deep dive into the mechanics of the algorithm. We analyze how performance scales with sample size, evaluate the sensitivity of the method to its hyperparameters, and empirically validate the core theoretical assumption of residual independence. Finally, we investigate boundary cases, specifically using weak linear models and stronger tree-based ensembles, to demonstrate the robustness of the CRDA framework.

6.1 Main Predictive Performance

Our primary evaluation metric is the Mean Squared Error (MSE) on held-out test sets. We compare the unaugmented baseline models against models retrained with the CRDA-augmented dataset.

6.1.1 Performance on MLP and XGBoost

The main results are summarized in Table 6.1 and Figure 6.1, demonstrating that CRDA yields consistent and often substantial reductions in test error.

Table 6.1: Augmentation results for XGB and MLP evaluated and averaged across 15 seeds with standard errors. Cells are green when data augmentation was more frequently selected to proceed according to the Wilcoxon signed rank test and red otherwise. $\Delta\%$ indicates the percentage change in MSE relative to the baseline. Lower is better ↓.

Dataset	**Sample Size**	**XGB ↓**			**MLP ↓**		
		$\text{MSE}_{\text{baseline}}$	MSE_{CRDA}	$\Delta\,\%$	$\text{MSE}_{\text{baseline}}$	MSE_{CRDA}	$\Delta\,\%$
CPU Performance	1638	0.00097 ± 0.00004	0.00089 ± 0.00004	-7.0 ± 2.7	0.00112 ± 0.00007	0.00087 ± 0.00003	-20.2 ± 3.2
	3276	0.00088 ± 0.00003	0.00079 ± 0.00002	-9.5 ± 2.2	0.00100 ± 0.00002	0.00085 ± 0.00002	-14.0 ± 1.5
	4914	0.00077 ± 0.00002	0.00072 ± 0.00001	-6.2 ± 1.2	0.00093 ± 0.00002	0.00082 ± 0.00001	-11.3 ± 1.5
	6552	0.00073 ± 0.00002	0.00069 ± 0.00001	-4.1 ± 1.6	0.00090 ± 0.00003	0.00079 ± 0.00001	-10.5 ± 2.2
	8190	0.00074 ± 0.00002	0.00070 ± 0.00001	-5.2 ± 1.9	0.00087 ± 0.00001	0.00078 ± 0.00001	-10.2 ± 0.8
Satellite Image	1287	0.01778 ± 0.00046	0.01697 ± 0.00051	-4.5 ± 1.4	0.02031 ± 0.00100	0.01629 ± 0.00057	-18.4 ± 3.1
	2574	0.01636 ± 0.00035	0.01576 ± 0.00040	-3.7 ± 0.8	0.01747 ± 0.00037	0.01455 ± 0.00039	-16.7 ± 1.5
	3861	0.01460 ± 0.00034	0.01390 ± 0.00034	-4.8 ± 0.8	0.01585 ± 0.00053	0.01211 ± 0.00035	-23.1 ± 1.7
	5148	0.01366 ± 0.00032	0.01300 ± 0.00030	-4.7 ± 1.1	0.01415 ± 0.00044	0.01076 ± 0.00029	-23.7 ± 1.2
	6435	0.01254 ± 0.00029	0.01186 ± 0.00026	-5.3 ± 0.8	0.01232 ± 0.00034	0.00989 ± 0.00028	-19.7 ± 1.0
Wind Power	1314	0.00742 ± 0.00028	0.00721 ± 0.00028	-2.8 ± 1.2	0.00752 ± 0.00024	0.00697 ± 0.00024	-7.2 ± 1.5
	2628	0.00602 ± 0.00012	0.00603 ± 0.00012	0.2 ± 0.6	0.00621 ± 0.00016	0.00562 ± 0.00011	-9.2 ± 1.4
	3942	0.00586 ± 0.00008	0.00578 ± 0.00008	-1.3 ± 0.4	0.00593 ± 0.00007	0.00539 ± 0.00008	-9.0 ± 0.7
	5256	0.00570 ± 0.00006	0.00562 ± 0.00007	-1.4 ± 0.4	0.00567 ± 0.00008	0.00533 ± 0.00008	-6.2 ± 0.4
	6570	0.00528 ± 0.00005	0.00522 ± 0.00005	-1.1 ± 0.3	0.00530 ± 0.00004	0.00500 ± 0.00004	-5.6 ± 0.5
Synthetic Regression	200	0.00652 ± 0.00043	0.00564 ± 0.00031	-12.0 ± 3.7	0.01993 ± 0.00172	0.01387 ± 0.00157	-28.8 ± 6.3
	400	0.00327 ± 0.00026	0.00312 ± 0.00022	-3.2 ± 2.5	0.00610 ± 0.00036	0.00384 ± 0.00031	-36.9 ± 3.0
	600	0.00264 ± 0.00008	0.00242 ± 0.00007	-7.9 ± 2.2	0.00321 ± 0.00026	0.00228 ± 0.00019	-27.9 ± 3.0
	800	0.00165 ± 0.00008	0.00161 ± 0.00009	-2.2 ± 2.0	0.00223 ± 0.00016	0.00140 ± 0.00008	-34.1 ± 4.0
	1000	0.00152 ± 0.00005	0.00145 ± 0.00006	-4.6 ± 2.8	0.00220 ± 0.00013	0.00123 ± 0.00007	-42.3 ± 3.1
Concrete Strength	201	0.00777 ± 0.00068	0.00701 ± 0.00063	-8.0 ± 3.6	0.01033 ± 0.00103	0.00793 ± 0.00050	-17.8 ± 5.7
	402	0.00493 ± 0.00035	0.00453 ± 0.00037	-8.4 ± 2.7	0.00635 ± 0.00053	0.00496 ± 0.00037	-19.8 ± 2.8
	603	0.00473 ± 0.00024	0.00427 ± 0.00024	-9.7 ± 2.2	0.00602 ± 0.00014	0.00494 ± 0.00014	-17.6 ± 2.5
	804	0.00365 ± 0.00017	0.00307 ± 0.00014	-15.7 ± 1.9	0.00497 ± 0.00026	0.00361 ± 0.00013	-24.8 ± 4.1
	1005	0.00290 ± 0.00010	0.00256 ± 0.00013	-12.2 ± 2.0	0.00422 ± 0.00024	0.00306 ± 0.00016	-26.9 ± 1.7
Energy Efficiency	153	0.00399 ± 0.00048	0.00344 ± 0.00050	-13.3 ± 7.7	0.00583 ± 0.00048	0.00426 ± 0.00046	-25.1 ± 6.8
	306	0.00233 ± 0.00014	0.00206 ± 0.00015	-12.2 ± 3.1	0.00321 ± 0.00014	0.00233 ± 0.00021	-28.1 ± 4.8
	459	0.00165 ± 0.00012	0.00143 ± 0.00011	-10.5 ± 5.9	0.00188 ± 0.00015	0.00106 ± 0.00013	-43.0 ± 4.6
	612	0.00128 ± 0.00007	0.00100 ± 0.00006	-19.3 ± 5.3	0.00091 ± 0.00008	0.00052 ± 0.00005	-40.7 ± 3.9
	765	0.00097 ± 0.00006	0.00076 ± 0.00007	-21.0 ± 4.4	0.00053 ± 0.00008	0.00035 ± 0.00003	-28.3 ± 4.3
House Price	200	0.00079 ± 0.00008	0.00064 ± 0.00005	-14.2 ± 4.8	0.00102 ± 0.00011	0.00057 ± 0.00007	-40.6 ± 4.8
	400	0.00033 ± 0.00002	0.00031 ± 0.00002	-5.4 ± 2.3	0.00041 ± 0.00003	0.00025 ± 0.00001	-37.0 ± 3.7
	600	0.00027 ± 0.00002	0.00026 ± 0.00002	-4.9 ± 2.7	0.00029 ± 0.00002	0.00020 ± 0.00002	-30.1 ± 3.8
	800	0.00024 ± 0.00001	0.00022 ± 0.00001	-9.9 ± 2.0	0.00023 ± 0.00001	0.00016 ± 0.00001	-30.3 ± 4.1
	1000	0.00020 ± 0.00001	0.00018 ± 0.00001	-6.5 ± 1.9	0.00019 ± 0.00001	0.00014 ± 0.00001	-27.0 ± 2.5
Parkinson's Monitoring	1175	0.00079 ± 0.00003	0.00072 ± 0.00003	-8.4 ± 2.4	0.00165 ± 0.00012	0.00101 ± 0.00006	-36.2 ± 3.9
	2350	0.00034 ± 0.00002	0.00032 ± 0.00001	-6.6 ± 2.8	0.00080 ± 0.00005	0.00054 ± 0.00003	-31.8 ± 2.5
	3525	0.00021 ± 0.00001	0.00020 ± 0.00001	-2.8 ± 3.4	0.00048 ± 0.00003	0.00030 ± 0.00002	-36.6 ± 4.0
	4700	0.00015 ± 0.00001	0.00014 ± 0.00001	-6.3 ± 2.4	0.00042 ± 0.00003	0.00021 ± 0.00001	-46.4 ± 4.1
	5875	0.00011 ± 0.00001	0.00011 ± 0.00001	1.7 ± 3.8	0.00026 ± 0.00002	0.00013 ± 0.00001	-47.2 ± 4.6
Wine Quality	1063	0.02057 ± 0.00056	0.02062 ± 0.00054	0.3 ± 0.8	0.02291 ± 0.00088	0.02284 ± 0.00129	-0.3 ± 3.3
	2126	0.01416 ± 0.00029	0.01429 ± 0.00029	1.0 ± 0.7	0.01539 ± 0.00026	0.01458 ± 0.00032	-5.2 ± 1.6
	3189	0.01391 ± 0.00019	0.01386 ± 0.00016	-0.3 ± 0.5	0.01478 ± 0.00023	0.01423 ± 0.00024	-3.6 ± 1.5
	4252	0.01332 ± 0.00024	0.01324 ± 0.00026	-0.6 ± 0.4	0.01386 ± 0.00027	0.01323 ± 0.00025	-4.4 ± 0.8
	5315	0.01332 ± 0.00012	0.01318 ± 0.00014	-1.1 ± 0.3	0.01397 ± 0.00016	0.01328 ± 0.00019	-5.0 ± 0.6

- **Multi-Layer Perceptron (MLP):** The neural network baseline benefits most significantly from our counterfactual augmentation. We observe an average MSE reduction of **22.9%** across all datasets. In specific high-dimensional or noisy domains, such as *Parkinson's Monitoring* and *House Price*, the error reduction exceeds 40%.

- **XGBoost (XGB):** The gradient boosting baseline, which is already highly robust in small-data regimes, shows a more modest but consistent improvement, with an average MSE reduction of **6.4%**.

Table 6.1 also highlights the efficacy of the Wilcoxon signed-rank significance gate (Line 15 of Algorithm 1). Cells colored in green indicate instances where the cross-validation gate correctly identified a statistically significant improvement ($p < 0.05$) and allowed augmentation to proceed. In rare cases (colored red), the method proceeded despite a lack of eventual test-set gain, though these instances are infrequent.

Figure 6.1 presents the average MSE change across all sample size partitions for each dataset. It further illustrates the extent of improvement that CRDA's new augmented data can bring to the base regressors test results.

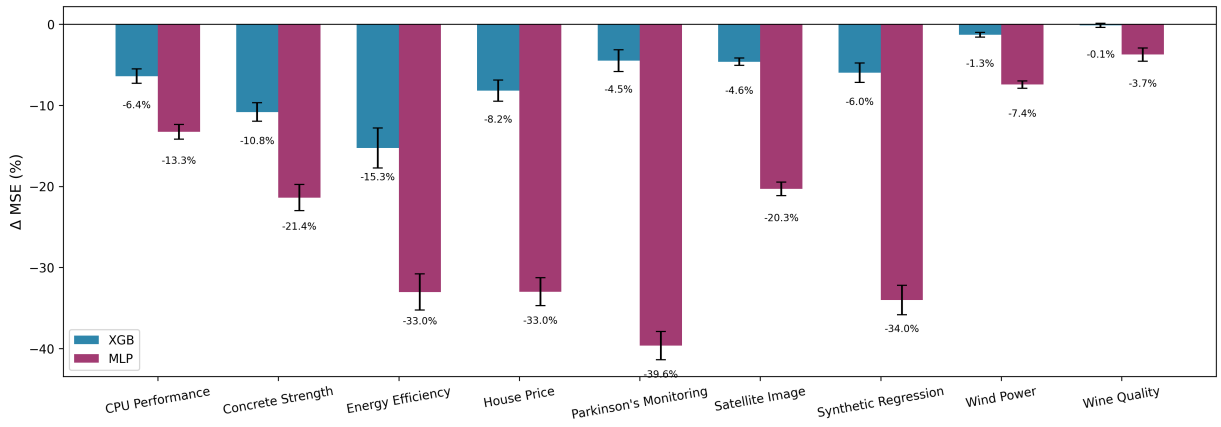
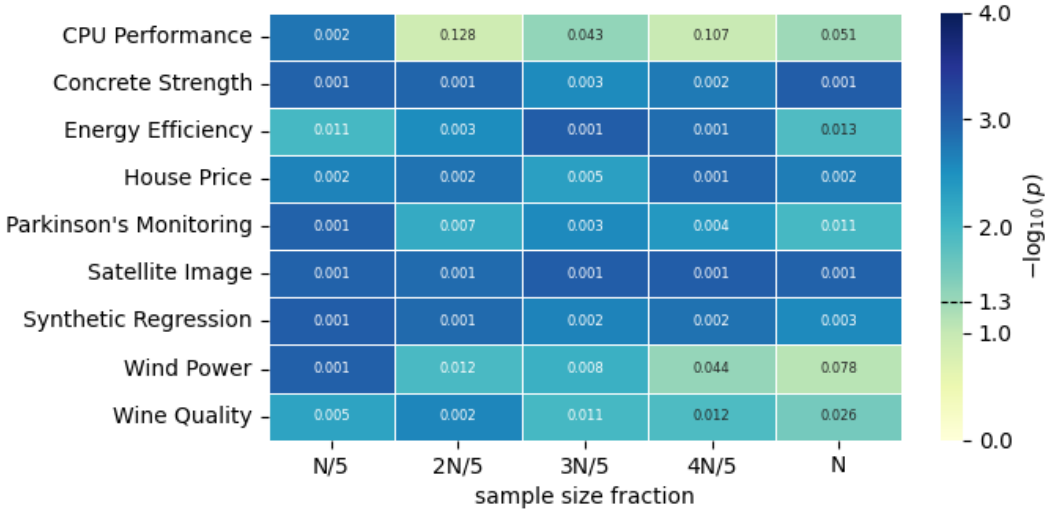
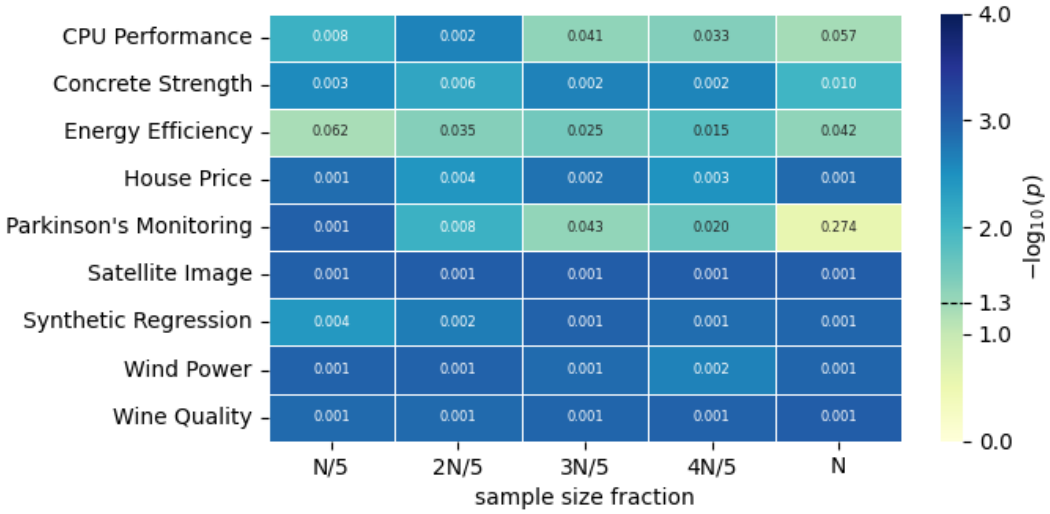


Figure 6.1: MSE percentage change for each dataset averaged over the five different training-subset sizes reported in Table 6.1 with error bars corresponding to standard error. Lower is better ↓.



(a) MLP Baseline



(b) XGB Baseline

Figure 6.2: Heatmaps of statistical significance ($-\log_{10}(p)$) across datasets and sample sizes. The dashed line on the colorbar indicates the $\alpha = 0.05$ threshold ($-\log_{10} p \approx 1.3$).

6.1.2 Statistical Significance

To assess the reliability of these improvements, we visualize the p-values from the Wilcoxon signed-rank tests across the experimental grid for every dataset $\times$ training–set–fraction. Figures 6.2a and 6.2b display heatmaps where color intensity represents $-\log_{10}(p)$ ($n_{\text{folds}} = 10$, $n_{\text{seeds}} = 15$ per cell). Darker regions indicate higher statistical significance.

The heatmaps reveal that across *both* baselines the majority of cells are darker than the $\alpha = 0.05$ cut-off, indicating that CRDA delivers a statistically significant reduction in test–MSE for most dataset/size combinations. Significance is strongest for smaller training sets and occasionally weakens as the full dataset is used (e.g. *CPU Performance* and *Wind Power* for MLP, *Parkinson's Monitoring* for XGB), but even at n the method remains significant in 7/9 datasets with at least one baseline. These results support the robustness of the performance gains reported, and validate the premise that augmentation is most critical when data is scarce.

6.1.3 Comparison with State-of-the-Art Methods

We benchmarked CRDA against a suite of specialized regression augmentation techniques (C-Mixup, Anchor Data Augmentation) and deep generative models (TabDDPM, TVAE, CTGAN). The results, presented in Table 6.2, demonstrate the pitfalls of existing approaches and CRDA's superior stability in performance.

Geometric methods like ADA and C-Mixup assume local linearity or manifold continuity, and while they provide gains on certain tasks, they exhibit catastrophic failure modes in others (e.g., increasing MSE by over 100% on *Synthetic Regression* and *Parkinson's*). Similarly, deep generative models significantly degrade performance more often, struggling to capture the precise conditional distribution $P(Y|X)$ required for regression, leading to "hallucinated" noise. In contrast, CRDA's residual-preserving mechanism ensures that synthetic samples remain faithful to the underlying noise structure. Across all datasets, CRDA is the only method that reliably improves performance for both XGB and MLP models without the risk of significant degradation.

Table 6.2: The percent MSE change (Δ%) for XGB and MLP base regressors. We compare CRDA against specialized regression augmentations (C-Mixup, ADA) and generative models (TabDDPM, TVAE, CTGAN). Results averaged across 10 seeds, reporting standard error. Lower is better ↓.

Dataset	**Model**	**% MSE Change ↓**					
		$\Delta_{\text{C-Mixup}}$	Δ_{ADA}	Δ_{TabDDPM}	Δ_{TVAE}	Δ_{CTGAN}	Δ_{CRDA}
CPU Performance	XGB	1.7 ± 1.5	1.9 ± 0.6	36.5 ± 4.0	23.6 ± 3.2	47.5 ± 3.5	**-1.4 ± 0.7**
	MLP	-0.9 ± 1.0	-0.6 ± 1.0	27.3 ± 4.8	30.6 ± 4.8	141.4 ± 18.6	**-12.0 ± 1.0**
Satellite Image	XGB	6.4 ± 1.7	1.4 ± 1.1	10.7 ± 1.4	13.1 ± 2.3	8.6 ± 1.3	**-0.7 ± 0.7**
	MLP	-1.0 ± 2.1	3.3 ± 2.4	9.9 ± 3.0	21.8 ± 4.3	50.5 ± 5.6	**-23.3 ± 1.5**
Wind Power	XGB	-1.9 ± 0.6	-0.2 ± 0.3	-0.4 ± 0.5	4.9 ± 0.8	8.7 ± 1.3	**-2.6 ± 0.3**
	MLP	4.9 ± 3.4	14.7 ± 2.0	-7.1 ± 1.4	4.9 ± 1.4	18.7 ± 2.4	**-8.0 ± 1.1**
Synthetic Regression	XGB	141.5 ± 31.8	18.3 ± 3.6	25.0 ± 8.2	117.0 ± 20.0	158.4 ± 23.5	**2.3 ± 1.8**
	MLP	78.1 ± 19.2	16.7 ± 6.4	-19.2 ± 2.7	68.0 ± 11.3	191.5 ± 29.8	**-33.3 ± 3.8**
Concrete Strength	XGB	**-2.8 ± 1.7**	-0.1 ± 1.4	-1.1 ± 3.0	8.1 ± 2.7	26.1 ± 4.4	-1.7 ± 1.9
	MLP	-4.8 ± 2.9	-3.8 ± 1.5	-5.9 ± 2.9	34.8 ± 6.3	135.1 ± 17.7	**-15.4 ± 2.3**
Energy Efficiency	XGB	-18.0 ± 9.1	-20.7 ± 3.5	3.4 ± 7.7	-18.3 ± 8.0	**-25.0 ± 6.5**	-10.7 ± 3.8
	MLP	11.9 ± 14.2	-22.9 ± 4.1	11.1 ± 11.1	131.8 ± 32.0	353.7 ± 52.2	**-32.5 ± 5.4**
House Price	XGB	-12.8 ± 12.6	**-42.9 ± 7.4**	-13.4 ± 12.9	297.5 ± 93.6	817.7 ± 142.1	-12.8 ± 3.6
	MLP	-51.0 ± 5.1	**-52.6 ± 5.8**	-4.3 ± 16.1	996.5 ± 286.4	3036.1 ± 373.9	-42.3 ± 3.4
Parkinson's Monitoring	XGB	105.1 ± 13.2	89.5 ± 11.1	286.7 ± 25.9	434.7 ± 56.5	596.9 ± 55.2	**-0.3 ± 1.9**
	MLP	102.6 ± 34.9	19.5 ± 10.1	164.6 ± 27.1	660.6 ± 128.0	1280.3 ± 139.3	**-51.1 ± 5.1**
Wine Quality	XGB	-2.0 ± 0.5	-0.0 ± 0.6	-2.6 ± 0.7	-0.5 ± 0.6	0.6 ± 0.9	**-2.8 ± 0.4**
	MLP	13.1 ± 5.2	23.6 ± 2.5	**-5.5 ± 0.6**	-1.2 ± 0.9	2.2 ± 0.5	-2.9 ± 0.8

6.2 Sample Size Scaling Analysis

To understand how CRDA's effectiveness varies with data availability, we conducted a scaling experiment using a synthetic dataset with a known Data Generating Process (DGP): $Y = X_1^2 + X_2X_3 + Z$, where $Z \perp X$.

Figure 6.3 illustrates the relationship between sample size and MSE reduction. We identify three distinct regimes:

1. **Low Data Regime (< 2.5k):** The baseline model is too weak to estimate the systematic function $g(X)$ accurately. Consequently, the calculated residuals $z = y - g(x)$ contain significant signal leakage, violating the residual independence assumption. CRDA offers minimal benefit here.

2. **The "Sweet Spot" (2.5k – 20k):** The baseline model captures the main trend, and the residuals are good approximations of the true noise Z. Here, CRDA provides the maximum benefit by densifying the feature space around limited training points.

3. **Saturation Regime (> 30k):** The dataset is sufficiently large that the baseline model has essentially converged. Augmentation provides diminishing returns.

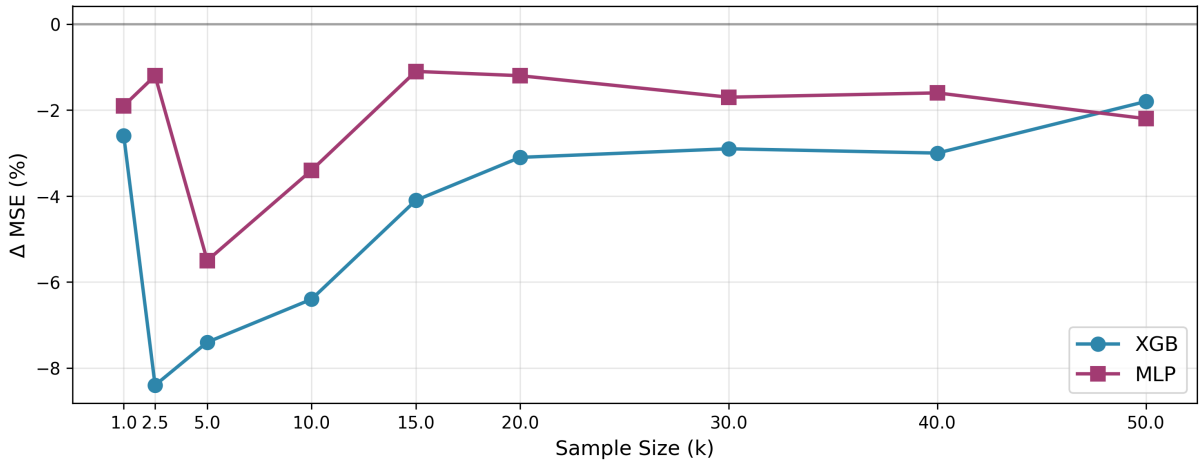


Figure 6.3: Sample-size scaling on synthetic data with a known DGP. A "sweet spot" is observed where the base learner is competent enough to isolate residuals, but the data is sparse enough to require augmentation.

6.3 Sensitivity Analysis

The CRDA framework introduces three distinct hyperparameters (or "knobs") that govern the augmentation process. Understanding how these parameters influence model performance is crucial for practical application. The three parameters are:

1. `aug_data_size_factor`: The ratio of synthetic counterfactual samples generated relative to the original training set size. A factor < 1 suggests undersampling to correct class imbalances (though less relevant for regression), while > 1 focuses on variance reduction.

2. `max_n_features_to_perturb`: The maximum number of features X_P perturbed simultaneously for a single sample. This controls the complexity of the counterfactual intervention.

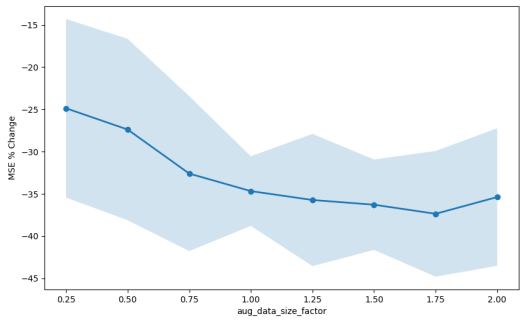
3. `max_perturb_percent`: The magnitude of the uniform noise δ applied to the features, defined as the half-width of the interval $[-p, +p]$.

To isolate the marginal effect of each parameter, we conducted a sensitivity sweep on the *House Price* dataset. We fixed two parameters at their modal best values (determined via Optuna) and varied the third. The effects ont the MSE percentage change for MLP and XGB baselines are presented in Figures 6.4 and 6.5, respectively. The sensitivity profiles reveal distinct inductive biases between neural and tree-based architectures.

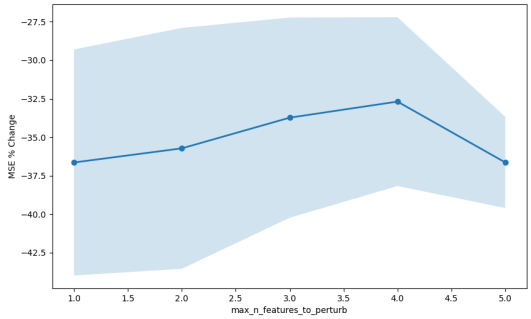
Augmentation Size (`aug_data_size_factor`): As seen in Figure 6.4a, the MLP regressor benefits monotonically from larger augmentation factors. Neural networks are high-variance estimators in small-data regimes; providing a critical mass of synthetic samples helps smooth the decision boundary and prevent overfitting. In contrast, XGB (Figure 6.5a) is less sensitive to the sheer volume of data, showing diminishing returns beyond a factor of $1.0\times$.

Feature Perturbation Count (`max_n_features_to_perturb`): Figure 6.4b shows that MLPs degrade in performance if too many features are perturbed simultaneously (e.g., > 2). This likely stems from the manifold hypothesis: perturbing too many dimensions pushes the synthetic sample off the data manifold, creating realistic but unlikely feature combinations that confuse the neural network. However, XGB benefits from higher-dimensional perturbations (Figure 6.5b), likely because decision trees operate on axis-aligned splits and benefit from exploring feature interactions more broadly.

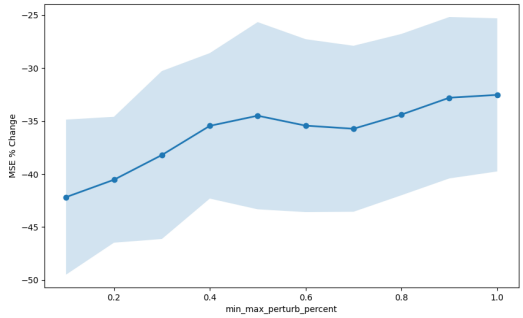
Perturbation Magnitude (`max_perturb_percent`): The magnitude of the perturbation also diverges between baselines. Larger scales help XGB discover more diverse synthetic points (Figure 6.5c), but MLP tends to prefer smaller shifts in order to maintain stable gradients in training (Figure 6.4c).



(a) Size Factor

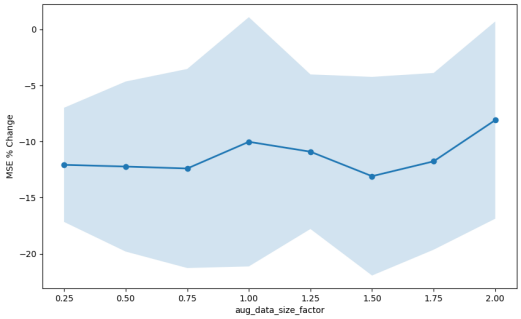


(b) Features Perturbed

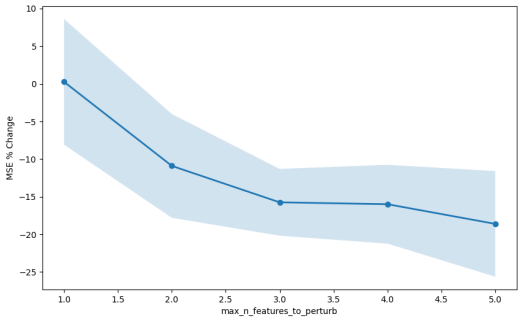


(c) Perturbation %

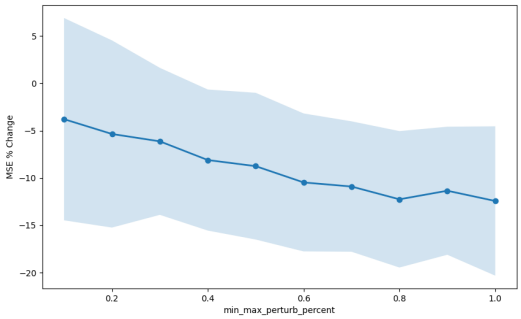
Figure 6.4: CRDA parameter sensitivity for the **MLP** baseline on House Price. MLPs favor larger dataset sizes but fewer simultaneous feature perturbations at lower magnitudes.



(a) Size Factor



(b) Features Perturbed



(c) Perturbation %

Figure 6.5: CRDA parameter sensitivity for the **XGB** baseline on House Price. XGBoost shows diminishing returns when it comes to more data, but benefits from more feature perturbations and stronger magnitudes.

6.4 Validation of Residual Independence Assumption

The theoretical validity of CRDA hinges on Assumption 1: that the residual noise Z is conditionally independent of the perturbable features X_P, given the fixed features X_R. If this assumption is violated (e.g., due to unobserved confounding), perturbing X_P while holding Z constant would yield invalid counterfactuals.

To empirically validate that our feature selection filter (the PC Algorithm combined with Pearson correlation checks) successfully identifies features satisfying this condition, we utilized Mutual Information (MI).

We calculated the MI between the model residuals Z and the input features X, grouping them into two categories:

1. **Selected (X_P):** Features approved by CRDA for perturbation.
2. **Rejected (X_R):** Features flagged by CRDA as being dependent on the residual.

MI serves as a non-parametric estimator of the KL-Divergence $D_{KL}(P(Z, X)||P(Z)P(X))$; values closer to zero indicate independence.

The results, averaged over 15 seeds using an XGBoost regressor, are presented in Table 6.3. In datasets where strong feature-residual dependencies exist (e.g., *House Price*, *Energy Efficiency*), the filter is highly effective. The rejected features display MI scores nearly 3× higher than the selected features. This confirms that CRDA correctly partitions the feature space, isolating the "safe" columns for variation from those entangled with the noise. In datasets like *Wind Power* or *Wine Quality*, the MI scores are uniformly low, suggesting that the residuals are naturally independent of the features; here, CRDA correctly permits a wider range of perturbations.

Table 6.3: Validation of Feature-Residual Independence via Mutual Information (MI). We report the MI (in nats) between residuals Z and features. The *Divergence Ratio* (MI_{Rej}/MI_{Sel}) indicates how much stronger the dependence is for rejected features compared to selected ones. Higher ratios indicate effective filtering.

Dataset	**Selected Features (X_P)** (Lower is better)	**Rejected Features (X_R)** (Higher implies dependence)	**Divergence Ratio** (MI_{Rej}/MI_{Sel})
House Price	$\mathbf{0.0056 \pm 0.0020}$	0.0155 ± 0.0023	$\mathbf{2.75}\times$
Energy Efficiency	$\mathbf{0.0054 \pm 0.0012}$	0.0160 ± 0.0023	$\mathbf{2.94}\times$
Parkinson's Monitoring	$\mathbf{0.0054 \pm 0.0006}$	0.0103 ± 0.0007	$\mathbf{1.92}\times$
Synthetic Regression	$\mathbf{0.0073 \pm 0.0013}$	0.0136 ± 0.0025	$\mathbf{1.85}\times$
Concrete Strength	0.0203 ± 0.0024	0.0320 ± 0.0035	$1.58\times$
CPU Performance	0.0136 ± 0.0010	0.0144 ± 0.0014	$1.06\times$
Wine Quality	0.0110 ± 0.0009	0.0136 ± 0.0011	$1.23\times$
Wind Power	0.0065 ± 0.0007	0.0084 ± 0.0007	$1.29\times$
Satellite Image	0.1031 ± 0.0013	0.1149 ± 0.0009	$1.11\times$

6.5 Ablation Studies

To demonstrate that the performance gains are driven by our specific counterfactual mechanism and not merely by injecting noise, we compared CRDA against two simplified baselines:

1. **Global Perturbation:** This baseline perturbs *all* features randomly ($X_P = X$), effectively bypassing the PC-algorithm and correlation checks. This tests the importance of the feature selection step.
2. **Label Invariance:** This baseline perturbs features but keeps the target label fixed ($y' = y$), rather than updating it via $y' = g(x') + z$. This tests the importance of the residual preservation mechanism.

Table 6.4 presents the results on three representative datasets. Global Perturbation often leads to performance degradation (positive $\Delta\%$), particularly for XGBoost. By perturbing features that are correlated with the residual, this naive approach breaks the noise structure of the data. Label Invariance also performs poorly; by changing inputs without adjusting outputs, it creates physically impossible samples that confuse the regressor. CRDA consistently outperforms both, confirming that both intelligent feature selection and residual-based label updates are necessary components.

Table 6.4: Ablation results on Synthetic Regression, Energy Efficiency, and Parkinson's Monitoring datasets. Values represent the percentage change in MSE ($\Delta\%$) relative to the unaugmented baseline (lower is better). Results are averaged over 5 seeds.

Dataset	**Model**	**MSE $\Delta\%$ Change ($\downarrow$)**		
		Global Perturbation	Label Invariance	**CRDA**
Synthetic Regression	MLP	-16.12 ± 4.30	-12.44 ± 32.4	**-38.94 ± 4.02**
	XGB	+1.21 ± 2.10	-1.02 ± 1.33	**-3.62 ± 1.93**
Energy Efficiency	MLP	-14.50 ± 3.86	-2.65 ± 2.47	**-38.84 ± 5.99**
	XGB	-7.15 ± 9.75	-5.28 ± 8.78	**-17.45 ± 5.16**
Parkinson's Monitoring	MLP	-13.50 ± 8.90	+0.55 ± 19.0	**-58.40 ± 5.16**
	XGB	+0.36 ± 1.57	-3.09 ± 4.35	**-7.82 ± 2.47**

6.6 Boundary Cases

A robust data augmentation method must satisfy two boundary conditions; it should not degrade performance when applied to weak models where residuals are poorly defined, and it should still provide value when applied to state-of-the-art models that are difficult to improve.

6.6.1 Safety Check: Linear Regression

We applied CRDA to a simple Linear Regression model. Linear models often fail to capture the systematic signal $g(X)$ fully, causing significant signal leakage into the residuals Z. Consequently, Z remains highly dependent on X, violating our core assumption. In such cases, the CRDA safety mechanisms (independence filters and the Wilcoxon gate) should trigger and prevent augmentation.

As seen in Table 6.5, across all datasets and sample sizes, the p-values for the Wilcoxon signed-rank test were consistently greater than our 0.05 threshold. CRDA correctly identified that augmentation would likely harm performance and abstained (rejected augmentation) in all of the cases. We still report the $\Delta\,\%$ if we had ignored the filter and observe how the MSE generally increased, proving the necessity of the safety mechanism. CRDA therefore protects against weaker baselines, illustrating how model-agnostic does not imply *always helpful*.

Table 6.5: Augmentation results for Linear Regression on all datasets and sample sizes over 15 seeds ± standard errors. Cells are green when data augmentation was selected to proceed according to the Wilcoxon signed rank test and red otherwise. Lower is better for the Δ MSE % change ↓.

Dataset	**Size**	**Linear Regression**			
		$\text{MSE}_{\text{baseline}}$	MSE_{CRDA}	Δ % ↓	p-value
CPU Performance	1638	0.011094 ± 0.000870	0.011066 ± 0.000842	0.04 ± 0.58	0.461 ± 0.035
	3276	0.010935 ± 0.000576	0.011012 ± 0.000604	0.58 ± 0.41	0.506 ± 0.038
	4914	0.009789 ± 0.000295	0.009784 ± 0.000296	-0.05 ± 0.23	0.452 ± 0.038
	6552	0.009834 ± 0.000360	0.009887 ± 0.000361	0.55 ± 0.18	0.515 ± 0.029
	8190	0.009705 ± 0.000347	0.009709 ± 0.000346	0.05 ± 0.10	0.456 ± 0.040
Satellite Image	1287	0.042119 ± 0.000653	0.042185 ± 0.000658	0.16 ± 0.14	0.240 ± 0.037
	2574	0.041148 ± 0.000505	0.041131 ± 0.000504	-0.04 ± 0.07	0.264 ± 0.032
	3861	0.040646 ± 0.000388	0.040666 ± 0.000393	0.05 ± 0.05	0.275 ± 0.025
	5148	0.040154 ± 0.000304	0.040183 ± 0.000296	0.08 ± 0.05	0.393 ± 0.037
	6435	0.040492 ± 0.000291	0.040509 ± 0.000288	0.04 ± 0.05	0.347 ± 0.031
Wind Power	1314	0.007335 ± 0.000262	0.007339 ± 0.000261	0.06 ± 0.10	0.501 ± 0.042
	2628	0.006363 ± 0.000130	0.006368 ± 0.000130	0.08 ± 0.05	0.468 ± 0.028
	3942	0.006580 ± 0.000098	0.006583 ± 0.000098	0.04 ± 0.04	0.493 ± 0.029
	5256	0.006583 ± 0.000084	0.006584 ± 0.000084	0.00 ± 0.03	0.498 ± 0.025
	6570	0.006175 ± 0.000050	0.006175 ± 0.000049	-0.00 ± 0.02	0.528 ± 0.033
Synthetic Regression	200	0.023265 ± 0.000989	0.023317 ± 0.000979	0.29 ± 0.60	0.306 ± 0.029
	400	0.022073 ± 0.000552	0.022101 ± 0.000552	0.13 ± 0.26	0.365 ± 0.032
	600	0.021332 ± 0.000483	0.021358 ± 0.000490	0.12 ± 0.16	0.385 ± 0.039
	800	0.015924 ± 0.000382	0.015945 ± 0.000386	0.12 ± 0.09	0.381 ± 0.038
	1000	0.015908 ± 0.000365	0.015911 ± 0.000361	0.03 ± 0.12	0.419 ± 0.031
Concrete Strength	201	0.016621 ± 0.001047	0.016592 ± 0.001043	-0.15 ± 0.28	0.362 ± 0.035
	402	0.017469 ± 0.000866	0.017431 ± 0.000886	-0.33 ± 0.31	0.352 ± 0.031
	603	0.017323 ± 0.000472	0.017336 ± 0.000490	0.03 ± 0.25	0.406 ± 0.028
	804	0.016711 ± 0.000551	0.016726 ± 0.000563	0.06 ± 0.16	0.452 ± 0.028
	1005	0.015719 ± 0.000346	0.015722 ± 0.000348	0.01 ± 0.08	0.477 ± 0.024
Energy Efficiency	153	0.003620 ± 0.000316	0.003650 ± 0.000315	1.02 ± 0.92	0.413 ± 0.043
	306	0.002928 ± 0.000118	0.002926 ± 0.000118	-0.03 ± 0.35	0.398 ± 0.038
	459	0.002872 ± 0.000120	0.002873 ± 0.000119	0.04 ± 0.22	0.453 ± 0.040
	612	0.002615 ± 0.000083	0.002621 ± 0.000081	0.29 ± 0.30	0.452 ± 0.035
	765	0.002667 ± 0.000054	0.002673 ± 0.000057	0.21 ± 0.19	0.436 ± 0.025
House Price	200	0.000103 ± 0.000006	0.000103 ± 0.000006	0.77 ± 0.52	0.344 ± 0.029
	400	0.000100 ± 0.000004	0.000101 ± 0.000004	0.10 ± 0.18	0.463 ± 0.037
	600	0.000100 ± 0.000004	0.000100 ± 0.000004	0.07 ± 0.12	0.515 ± 0.023
	800	0.000104 ± 0.000003	0.000104 ± 0.000003	0.03 ± 0.09	0.476 ± 0.046
	1000	0.000103 ± 0.000003	0.000104 ± 0.000003	0.22 ± 0.16	0.445 ± 0.040
Parkinson's Monitoring	1175	0.004750 ± 0.000108	0.004754 ± 0.000108	0.08 ± 0.10	0.425 ± 0.041
	2350	0.004672 ± 0.000093	0.004681 ± 0.000096	0.18 ± 0.14	0.333 ± 0.025
	3525	0.004651 ± 0.000093	0.004650 ± 0.000092	-0.02 ± 0.06	0.355 ± 0.027
	4700	0.004618 ± 0.000074	0.004620 ± 0.000074	0.05 ± 0.05	0.449 ± 0.029
	5875	0.004655 ± 0.000053	0.004655 ± 0.000052	0.01 ± 0.03	0.429 ± 0.023
Wine Quality	1063	0.021526 ± 0.000636	0.021544 ± 0.000647	0.07 ± 0.23	0.393 ± 0.025
	2126	0.015126 ± 0.000310	0.015131 ± 0.000307	0.04 ± 0.05	0.452 ± 0.032
	3189	0.015448 ± 0.000199	0.015461 ± 0.000197	0.09 ± 0.05	0.443 ± 0.026
	4252	0.014830 ± 0.000261	0.014839 ± 0.000260	0.06 ± 0.04	0.487 ± 0.034
	5315	0.014894 ± 0.000162	0.014896 ± 0.000164	0.01 ± 0.03	0.505 ± 0.024

6.6.2 Robustness Check: CatBoost

Finally, we evaluated CRDA against **CatBoost** [25], a state-of-the-art gradient boosting method designed specifically for tabular data. CatBoost uses oblivious trees and ordered boosting to prevent overfitting, making it an exceptionally strong baseline that is difficult to improve upon via simple augmentation.

We evaluated performance at three fixed sample sizes ($N = \{300, 500, 700\}$) to observe behavior across different data availabilities.

Table 6.6 presents the percentage change in MSE ($\Delta\%$). We observe three distinct behaviors:

- **Consistent Gains:** On *House Price* and *Wind Power*, CRDA significantly reduces MSE across all sample sizes (peaking at -22.8% for *House Price*), demonstrating that CRDA behaves robustly for these tasks regardless of sample size.
- **Late-Stage Gains:** *CPU Performance* requires a sufficient number of samples to model the residual. It shows no benefit at $N = 300$ but improves substantially as data increases, reaching -13.0% at $N = 700$.
- **Sweet-Spot Behavior:** Datasets such as *Parkinson's Monitoring*, *Energy Efficiency*, and *Synthetic Regression* exhibit a "sweet spot" around $N = 500$, where the augmentation provides the most benefit ($\approx$ 4-5% reduction) before CatBoost potentially saturates the signal at larger sample sizes.

Table 6.6: Percentage change in MSE ($\Delta\%$) for *CatBoost* at fixed sample sizes. Values represent the mean $\Delta\%$ across 15 seeds $\pm$ standard error.

Dataset	$N = 300$	$N = 500$	$N = 700$
House Price	-22.80 $\pm$ 0.60	-18.87 $\pm$ 0.52	-14.11 $\pm$ 0.50
CPU Performance	1.92 $\pm$ 0.91	-7.34 $\pm$ 0.58	-13.04 $\pm$ 0.90
Parkinson's Monitoring	-1.44 $\pm$ 0.61	-5.13 $\pm$ 0.60	-1.33 $\pm$ 0.46
Energy Efficiency	-1.65 $\pm$ 0.86	-4.95 $\pm$ 1.01	-1.89 $\pm$ 0.90
Synthetic Regression	-2.31 $\pm$ 0.57	-4.01 $\pm$ 0.60	-1.15 $\pm$ 0.66
Wind Power	-3.78 $\pm$ 0.38	-2.54 $\pm$ 0.24	-2.47 $\pm$ 0.29
Satellite Image	-0.95 $\pm$ 0.53	-2.08 $\pm$ 0.41	-1.95 $\pm$ 0.30
Wine Quality	-0.09 $\pm$ 0.39	-1.62 $\pm$ 0.29	-1.94 $\pm$ 0.21
Concrete Strength	-0.95 $\pm$ 0.56	-0.69 $\pm$ 0.39	0.46 $\pm$ 0.39

Chapter 7

Conclusion and Future Work

In this thesis, we addressed the challenge of training regression models in data-scarce regimes. Although modern machine learning has achieved remarkable success in computer vision and natural language processing, largely driven by the availability of massive datasets and domain-specific data augmentation techniques, tabular regression has arguably lagged behind. While, real-world tabular data is often expensive to collect, heterogeneous, and noisy, it also lacks the invariant symmetries (such as rotation or translation) that make augmentation straightforward in other domains.

To bridge this gap, we introduced **Counterfactual Residual Data Augmentation (CRDA)**, a model-agnostic framework that synthesizes new training samples by leveraging the statistical invariance of residual noise. By decomposing the target variable into a systematic component (learned by a base regressor) and a stochastic residual, CRDA allows us to generate valid counterfactuals; data points with perturbed features but preserved noise structures.

7.1 Summary of Contributions

The primary contributions of this research can be summarized as follows:

1. **The Residual Invariance Principle:** We formalized the theoretical basis for augmentation in regression through Assumption 1. We posited that for a subset of features X_P, the residual noise Z remains conditionally independent given the remaining features X_R. This insight shifts the perspective of augmentation from ge-

ometric interpolation (as seen in Mixup) to statistical sampling from an invariant noise distribution.

2. **The CRDA Algorithm:** We developed a practical, end-to-end pipeline that implements this principle. Crucially, the algorithm also includes automated safety mechanisms, specifically the PC algorithm for feature selection and the Wilcoxon signed-rank test for validation, that allow the method to identify when augmentation is beneficial and, more importantly, when it should abstain to prevent performance degradation.

3. **Empirical Validation across Architectures:** We demonstrated that CRDA is effective for both neural and tree-based models.

    - For **Multi-Layer Perceptrons (MLPs)**, which are notoriously sensitive to data scarcity, CRDA reduced the Mean Squared Error (MSE) by an average of **22.9%** across nine benchmarks.
    - For **XGBoost**, a strong tree-based model, CRDA achieved a consistent average reduction of **6.4%**, proving that even robust ensembles can benefit from the densification of the feature space.

4. **Robustness against State-of-the-Art Augmentation:** When compared against deep generative models (CTGAN, TVAE, TabDDPM) and geometric augmentation methods (ADA, C-Mixup), CRDA exhibited superior stability and improvement. While the most competitive methods occasionally suffered catastrophic failure modes (increasing error by over 100% in sparse regimes), CRDA consistently improved or maintained baseline performance.

5. **Open Source Implementation:** We released `crda`, a pip-installable Python package that implements the proposed framework. This tool provides a `scikit-learn` compatible interface, making residual-based augmentation accessible to practitioners and facilitating future research.

7.2 Limitations

Despite the strong empirical results, the proposed framework relies on assumptions that may not hold in every setting. A critical analysis of these limitations is essential for the responsible application of the method.

7.2.1 Dependence on the Base Predictor

The validity of the generated counterfactuals hinges on the quality of the estimated residuals, which are defined as $\hat{z} = y - \hat{g}(x)$. If the base predictor $\hat{g}$ is high-bias (underfitted), the "residual" will contain significant systematic signal rather than pure noise. As demonstrated in our sample-size scaling analysis (Section 6.2), in very low-data regimes ($n < 2,500$ for the synthetic task), the base model is too weak to disentangle signal from noise, rendering CRDA ineffective. This creates a "cold start" problem where one needs enough data to train a decent base model before one can augment that data to train a better model.

7.2.2 Risk of Unobserved Confounding

The core assumption $P(Z|X_P, X_R) = P(Z|X_R)$ implies that the features chosen for perturbation (X_P) are not confounded with the residual by some unobserved latent variable U. While we employ the PC algorithm and correlation tests as *risk-control heuristics* to filter out dependent features, these tests are not infallible, particularly in non-linear settings or with limited samples. If a feature in X_P is causally linked to Z via a backdoor path, perturbing it while holding Z constant breaks the data generating process, potentially leading to out-of-distribution samples that harm the model.

7.2.3 Scope of Application

Currently, CRDA is strictly defined for regression tasks. The concept of an additive residual $y = g(x) + z$ does not map straightforwardly to classification tasks, where the target is a discrete class label and the "noise" is often conceptualized as aleatoric uncertainty rather than a scalar value.

7.3 Future Work

The limitations outlined above suggest several promising avenues for future research.

7.3.1 Extension to Classification

Extending the principle of residual invariance to classification is the most immediate next step. While discrete labels do not have additive residuals, one could operate in the *latent space* of the model. For example, in a neural network classifier, the pre-softmax logits could be decomposed into a systematic component and a residual vector. Alternatively, one could leverage *probability residuals* (the difference between one-hot labels and predicted probabilities) to generate counterfactual probability vectors, which could then be used in a knowledge distillation framework (e.g., training a student model on soft targets).

7.3.2 Proximal Causal Inference

To better address the risk of unobserved confounding, future iterations of CRDA could integrate techniques from Proximal Causal Inference. Instead of simply discarding features that appear correlated with the residual (as our current filter does), one could use proxy variables to explicitly model the confounding influence. This would allow the method to "adjust" the residual during perturbation rather than requiring strict invariance, potentially unlocking a larger set of features for augmentation.

7.3.3 Privacy-Preserving Augmentation

Deep generative models like TabDDPM are often cited for their utility in generating synthetic data for privacy preservation. Since CRDA generates new samples by changing existing features and existing residuals, it arguably poses a privacy risk (i.e. it does not generate "new" people, but rather "counterfactual versions" of existing people). Investigating whether CRDA can be adapted to provide differential privacy guarantees, perhaps by adding noise to the residuals themselves rather than reusing them exactly, would make the method applicable to highly sensitive domains like healthcare and finance.

7.4 Closing Remarks

In an era where models are increasingly large and data-hungry, the ability to squeeze more information out of existing datasets is invaluable. This thesis has presented CRDA not just as an engineering trick to lower MSE, but as a principled approach to data augmentation rooted in the statistical properties of noise. By respecting the residual structure of the

data, we move closer to training models that are robust not just to the data they have seen, but to the data they *could* have seen.

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

[2] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[3] Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. Predictive inference with the jackknife+. *The Annals of Statistics*, 49(1):486–507, 2021.

[4] Paula Branco, Luís Torgo, and Rita P Ribeiro. Smogn: a pre-processing approach for imbalanced regression. In *First international workshop on learning with imbalanced domains: Theory and applications*, pages 36–50. PMLR, 2017.

[5] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[6] Tianqi Chen. Xgboost: A scalable tree boosting system. *Cornell University*, 2016.

[7] Kaggle Community. House Prices: Advanced Regression Techniques (Dataset), 2024.

[8] Paulo Cortez, Antonio Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Wine Quality, 2009. UCI Machine Learning Repository.

[9] Bradley Efron. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics: Methodology and distribution*, pages 569–593. Springer, 1992.

[10] John Haslett and Adrian E. Raftery. Irish Wind Speed (Malin Head, 1961–1978), 1989. Daily average wind speeds at 12 Irish stations.

[11] Seong-Hyeon Hwang and Steven Euijong Whang. Regmix: Data mixing augmentation for regression. *arXiv preprint arXiv:2106.03374*, 2021.

[12] Kaggle. Kaggle: Home of Data Science and Machine Learning. urlhttps://www.kaggle.com.

[13] Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. The UCI Machine Learning Repository, 2023.

[14] Murat Kocaoglu, Christopher Snyder, Alexandros G Dimakis, and Sriram Vishwanath. Causalgan: Learning causal implicit generative models with adversarial training. *arXiv preprint arXiv:1709.02023*, 2017.

[15] Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models. In *International conference on machine learning*, pages 17564–17579. PMLR, 2023.

[16] Chaochao Lu, Biwei Huang, Ke Wang, José Miguel Hernández-Lobato, Kun Zhang, and Bernhard Schölkopf. Sample-efficient reinforcement learning via counterfactual-based data augmentation. *CoRR*, abs/2012.09092, 2020.

[17] Randal S. Olson, William La Cava, Patryk Orzechowski, Ryan J. Urbanowicz, and Jason H. Moore. PMLB Dataset 227_cpu_small, 2017. Penn Machine Learning Benchmarks, version 2025-05-16.

[18] Randal S. Olson, William La Cava, Patryk Orzechowski, Ryan J. Urbanowicz, and Jason H. Moore. PMLB Dataset 294_satellite_image, 2017. Penn Machine Learning Benchmarks, version 2025-05-16.

[19] Randal S. Olson, William La Cava, Patryk Orzechowski, Ryan J. Urbanowicz, and Jason H. Moore. PMLB Dataset 623_fri_c4_1000_10, 2017. Synthetic Friedman #4 variant; Penn Machine Learning Benchmarks.

[20] Randal S. Olson, William La Cava, Patryk Orzechowski, Ryan J. Urbanowicz, and Jason H. Moore. Pmlb: a large benchmark suite for machine learning evaluation and comparison. *BioData Mining*, 10(36):1–13, Dec 2017.

[21] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2 edition, 2009.

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

[23] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. The MIT Press, 2017.

[24] Parjanya Prashant, Seyedeh Baharan Khatami, Bruno Ribeiro, and Babak Salimi. Scalable out-of-distribution robustness in the presence of unobserved confounders. *arXiv preprint arXiv:2411.19923*, 2024.

[25] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

[26] Abbavaram Gowtham Reddy, Celia Rubio-Madrigal, Rebekka Burkholz, and Krikamol Muandet. When shift happens–confounding is to blame. *arXiv preprint arXiv:2505.21422*, 2025.

[27] Nora Schneider, Shirin Goshtasbpour, and Fernando Perez-Cruz. Anchor data augmentation. *Advances in Neural Information Processing Systems*, 36:74890–74902, 2023.

[28] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2nd edition, 2000.

[29] Luís Torgo, Rita P Ribeiro, Bernhard Pfahringer, and Paula Branco. Smote for regression. In *Portuguese conference on artificial intelligence*, pages 378–389. Springer, 2013.

[30] Athanasios Tsanas and Max A. Little. Parkinsons Telemonitoring, 2009. UCI Machine Learning Repository.

[31] Athanasios Tsanas and Angeliki Xifara. Energy Efficiency, 2012. UCI Machine Learning Repository.

[32] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83, 1945.

[33] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. *Advances in neural information processing systems*, 32, 2019.

[34] Huaxiu Yao, Yiping Wang, Linjun Zhang, James Y Zou, and Chelsea Finn. C-mixup: Improving generalization in regression. *Advances in neural information processing systems*, 35:3361–3376, 2022.

[35] I-Cheng Yeh. Concrete Compressive Strength, 1998. UCI Machine Learning Repository.

[36] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.