Evolving MultiFIX

Tackling Extreme Joint Modality Dependence in Deep Learning by Optimising Multimodal Features with GOMEA

Sander Britton

Delft University of Technology

CWI

TU Delft

Evolving MultiFIX

Tackling Extreme Joint Modality Dependence in Deep Learning by Optimising Multimodal Features with GOMEA

by

Sander Britton

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on
Thursday 13 November 2025 at 09:00 AM.

Student number:	5110947	
Project duration:	November 11, 2024 – November 13, 2025	
Thesis committee:	Prof. dr. Mathijs M. de Weerdt,	TU Delft, chair
	Dr. Bahareh Abdikivanani,	TU Delft, core member
	Prof. dr. Peter A.N. Bosman,	CWI, TU Delft, supervisor, core member
	Dr. Tanja Alderliesten,	LUMC, supervisor
	MSc. Mafalda Malafaia,	CWI, supervisor
	MSc. Johannes Koch,	CWI, supervisor

Cover:	Cover designed by Merel Smit
Style:	TU Delft Report Style, with modifications by Daan Zwaneveld

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

CWI
Centrum Wiskunde & Informatica

Preface

This thesis marks the conclusion of my academic journey at Delft University of Technology. What began with a Bachelor's in Electrical Engineering ended in a Master's in Computer and Embedded Systems Engineering. I can safely say that over the course of my studies, I have grown immensely, both academically and personally.

This thesis perfectly reflects that growth, representing the most challenging yet rewarding experience of my education. It taught me valuable lessons, many learned through extensive trial and error, that I will carry with me throughout my life. I would like to sincerely thank MSc Johannes Koch and MSc Mafalda Malafaia for their guidance, feedback, and support throughout this process, and Prof. dr. Peter Bosman and Dr. Tanja Alderliesten for their invaluable supervision.

I am also deeply grateful to my family and friends for their constant encouragement and belief in me. Special thanks go to my colleagues and fellow masters students during my time at CWI for their insightful feedback and making this journey both engaging and enjoyable. Finally, my heartfelt thanks go to my partner, whose kindness and unwavering support have kept me motivated throughout all these years.

Sander Britton
Amsterdam, November 2025

Summary

Multimodal machine learning models can exploit complementary information from multiple data modalities. MultiFIX (Multimodal Feature engIneering eXplainable artificial intelligence) is a framework designed to construct partially interpretable multimodal models, providing explanations for both modality-specific features and each modality its contribution to the final prediction. However, it was shown to not scale effectively for tasks with extreme joint-modality dependence.

This thesis proposes an alternative training strategy that integrates knowledge of the features to be engineered, expressed as feature targets that guide the learning process. The strategy improves upon baseline performance, even when the feature targets are non-ideal. Since ground-truth feature targets are typically unavailable in real-world settings, the feature targets are optimised using the Gene-pool Optimal Mixing Evolutionary Algorithm. The optimised feature targets, though only loosely aligned with the ground-truth features, enables the alternative training method to surpass baseline MultiFIX performance on a three-gated XOR task.

The same approach was evaluated on simpler tasks, such as the single XOR and AND problems, where it achieved slightly lower but still comparable performance to the already strong baselines. Results indicate that this computationally intensive approach is most beneficial for problems characterised by high joint-modality dependence and complex feature interactions. Interestingly, closer alignment between the optimised and ground-truth feature targets did not consistently lead to higher MultiFIX performance. Consequently, future improvements are likely to stem from refining how feature targets are integrated into the training process, rather than from further optimisation of the targets themselves.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
AE	Autoencoder
AI	Artificial Intelligence
AUROC	Area Under the Receiver Operating Characteristic
BAcc	Balanced Accuracy
BCE Loss	Binary Cross Entropy Loss
BST	Blockwise Supervised Training
CNN	Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Network
EA	Evolutionary Algorithm
E2E	End-to-End Training
FOS	Family Of Subsets
GOM	Gene-pool Optimal Mixing
GOMEA	Gene-pool Optimal Mixing Evolutionary Algorithm
GP	Genetic Programming
GP-GOMEA	Genetic Programming Gene-pool Optimal Mixing Evolutionary Algorithm
Grad-CAM	Gradient-weighted Class Activation Mapping
HPO	Hyperparameter Optimisation
IMG	Image
IMS	Interleaved Multi-start Scheme
IQR	Interquartile Range
LT	Linkage Tree
LR	Learning Rate
MBEA	Model-Based Evolutionary Algorithm
ML	Machine Learning
MLP	Multi-Layer Perceptron
MultiFIX	Multimodal Feature engIneering eXplainable artificial intelligence
NIS	No Improvement Stretch
NMI	Normalised Mutual Information
NN	Neural Network
SHAP	SHapley Additive exPlanations
TAB	Tabular
UPGMA	Unweighted Pair Grouping Method with Arithmetic-mean
WD	Weight Decay
XAI	Explainable Artificial Intelligence

Symbols

Symbol	Definition
α	The retention ratio parametrising the noise injection process
$\mathcal{A}$	Set of retention ratios
b	The block size used in the marginal blocks linkage model
$\mathcal{B}$	Set of 21 evenly spaced bins in the range $[0, 1]$
$circle$	The image feature denoting the presence of a circle in a given sample
$\mathfrak{F}$	Family of Subsets used in defining the linkage model
$\mathbf{F}^i$	Linkage set corresponding to index i
I_i	The true binary image feature target of the i^{th} data sample
$\mathbf{I}$	The set of true binary image feature targets of the i^{th} data samples
$\hat{\mathbf{I}}$	Approximation of the set of true binary image feature targets of the i^{th} data samples
l	The number of decision variables to be optimised (problem size)
$l_{\mathbf{F}^i}$	The number of decision variables in linkage set $\mathbf{F}^i$
L	The set of indices representing the decision variables as $L = \{0, 1, \ldots, l-1\}$
$n_{individuals}$	The number of individuals in the population of an EA
n_{BST}	The number of data samples used in BST
N	Denotes a count whose specific meaning depends on context: the number of negatives (for balanced accuracy), candidate solutions (in an EA), or samples (for loss computation). The relevant context indicates which definition applies.
P	The number of positives when determining balanced accuracy
p_m	The mutation probability in GOMEA
r_{BST}	The ratio of training samples that are used in BST
ρ_I	Absolute image-wise correlation between a set of feature targets and the true feature targets
ρ_T	Absolute tabular-wise correlation between a set of feature targets and the true feature targets
ρ^*	Average of the absolute modality-wise correlations between a set of feature targets and the true feature targets (feature correlation)
$\hat{\rho^*}$	Value in $\mathcal{B}$ that is closest to the feature correlation ρ^*
S	The number of distinct seeds used in creating noisy feature targets
$\mathbf{t}^{true}$	The true feature targets of all samples used in BST
$\hat{\mathbf{t}}^{noisy}$	A set of noisy feature targets of all samples used in BST
$\hat{\mathbf{t}}$	A set of feature targets of all samples used in BST, potentially equal to the true feature targets
T_i	The true binary tabular feature target of the i^{th} data sample
$\mathbf{T}$	The set of true binary tabular feature targets of the i^{th} data samples
$\hat{\mathbf{T}}$	Approximation of the set of true binary tabular feature targets of the i^{th} data samples
TN	The number of true negatives when determining balanced accuracy
TP	The number of true positives when determining balanced accuracy
$triangle$	The image feature denoting the presence of a circle in a given sample
x_i	The i^{th} tabular variable
$x_1 > x_2$	The tabular feature denoting whether tabular variable 1 is bigger than tabular variable 2 for a given sample
$x_3 > x_4$	The tabular feature denoting whether tabular variable 3 is bigger than tabular variable 4 for a given sample

1

Introduction

In the current era of big data, the rapid expansion of information from a wide range of sources, both in quantity and variety, has led to a heterogeneous data landscape [1, 2]. Domain experts, such as those in healthcare, routinely rely on heterogeneous data, like medical images, clinical notes, and lab results, often referred to as multiple data modalities, to make informed decisions. Additionally, single-modality models can be outperformed by multimodal modals, obtained through multimodal Machine Learning (ML) [3]. These multimodal models provide greater robustness and the ability to leverage complementary information [4].

Deep Neural Networks (DNN) are often used to achieve state-of-the-art performance [5, 6]. However, the black-box nature of DNNs, characterised by their lack of transparency, presents challenges in high-stakes domains. This is particularly evident in healthcare, where interpretability and trust are essential [7, 8]. Post-hoc explainability techniques, such as Gradient-weighted Class Activation Mapping (Grad-CAM) [9] and SHapley Additive exPlanations (SHAP) [10], can offer insights into the decision-making processes of complex models by providing activation heatmaps of input images that highlight prediction-relevant regions and by assigning importance values to individual features, respectively. However, these post-hoc explanations might be inaccurate and additionally do not explain how relevant information is being used. If an explanation model lacks accuracy, it undermines confidence in both the explanation itself and, by extension, the black box model it seeks to make interpretable [11]. Genetic Programming (GP) presents a promising alternative by evolving symbolic expressions that are inherently interpretable. These expressions can achieve performance comparable to black box methods and are particularly well-suited for tabular data [12, 13].

A key gap in current multimodal learning research is the lack of models that not only learn modality-specific features, but also perform feature fusion in an intrinsically explainable way. To address this, MultiFIX (Multimodal Feature engIneering eXplainable artificial intelligence) was proposed as a framework to build partially interpretable multimodal models [14]. For each data modality, a Deep Learning (DL) architectural block is trained to extract up to three potentially relevant features. The resulting features are concatenated and passed into the fusion DL block, which combines the features and produces the final prediction. Where possible, DL architectural blocks are replaced by intrinsically interpretable GP-generated symbolic expressions and otherwise explained using post-hoc explanations. This allows for both modality-specific and fusion explanations, offering a transparent view of how the model reaches its decisions and making it more suitable for critical domains.

MultiFIX has been shown to be performant on synthetic benchmark problems consisting of images and tabular data [14, 15, 16]. However, as the joint dependence between modalities to make highly accurate predictions increases, the proposed architecture can no longer be trained well with standard DNN training techniques [15]. While MultiFIX successfully solves a basic XOR problem involving one informative feature per modality, it fails to solve a more complex three-gated XOR task. This limitation illustrates that MultiFIX may not scale effectively to tasks with extreme joint-modality dependence.

To overcome this challenge, this thesis introduces an alternative training strategy for MultiFIX that incorporates knowledge about what features should be engineered, expressed as feature targets that guide the learning process. The hypothesis is that using these feature targets will allow MultiFIX to improve upon its baseline performance, even when those targets are imperfect. Since ground-truth features are only known for synthetic problems, feature targets are optimised using an Evolutionary Algorithm (EA). This creates an additional optimisation loop: the EA optimises feature targets, which then guides the alternative training strategy of the MultiFIX pipeline. Unlike gradient-based methods, EAs operate on populations of candidate solutions and are not reliant on local gradient information, making them well-suited for navigating complex, deceptive optimisation landscapes. This hybrid approach aims to bridge the gap between interpretability, multimodal fusion, and robust feature engineering.

1.1. Research Questions and Outline

The exploration of how modern EAs can be integrated into the MultiFIX framework to improve baseline performance on problems with extreme joint dependence between modalities is translated into the following main research question:

Main Research Question

To what extent can evolutionary algorithms optimise feature targets to improve the performance of MultiFIX on problems with extreme joint dependence between modalities?

First, Chapter 2 gives the necessary background to fully understand the contribution of this thesis. Then, Chapter 3 introduces the synthetic problems that will be used, together with their baseline results that this thesis aims to improve upon. The remainder of the thesis aims to answer the main research question and is structured by the following subquestions:

Research Questions

RQ1. How can feature knowledge, expressed as feature targets, be used in training MultiFIX and does it improve performance on tasks with extreme joint modality dependence?
RQ2. If the correlation to the ground-truth features is available, can GOMEA be used to approximate the ideal feature targets, and what parameter configuration yields the best performance?
RQ3. How can a proxy objective be designed to guide feature target optimisation in the absence of ground-truth features and does the proxy objective yield a fitness signal on tasks with extreme joint modality dependence?
RQ4. Does optimising the designed proxy objective with GOMEA yield feature targets that improve the performance of MultiFIX on tasks with extreme joint modality dependence, and why or why not?

Chapter 4 addresses RQ1 by describing how feature targets can be used in training MultiFIX. The method is evaluated first using ideal feature targets, which is available only in synthetic problems, and then with noisy variants to assess the robustness of the training strategy. Afterwards, Chapter 5, guided by RQ2, also assumes access to ground-truth features. It uses this to assess the efficiency of Gene-pool Optimal Mixing Evolutionary Algorithm (GOMEA), the chosen EA, on feature target optimisation in a simplified setting. Several parameter configurations are evaluated, with the results guiding the selection of GOMEA parameters. Chapter 6, guided by RQ3, follows with designing an optimisation objective that can be used as a proxy for when the ground-truth features are unknown, as is the case in real-world scenarios. Additionally, the fitness signal of this objective is estimated.

The following two chapters aim to answer RQ4. First, Chapter 7 shows the results of optimising said proxy objective through GOMEA for several synthetic problems. Chapter 8 then performs an ablation study to better understand the necessity and impact of specific components. Finally, Chapter 9 summarises the findings and aims to answer the main research question. Furthermore, suggestions for future work are given.

2

Background

This chapter provides the necessary background to understand the contributions of this thesis. Section 2.1 begins with an overview of multimodal learning, introducing the concept of multiple data modalities, the potential of their integration, and current state-of-the-art models. Afterwards, Section 2.2 introduces the MultiFIX framework as a recent approach designed to enhance interpretability in multimodal learning. A detailed description of MultiFIX is provided, explaining the pipeline, the training methodology, and its strategies for achieving explainability. Section 2.3 then introduces EAs. In particular, Section 2.4 highlights the GOMEA, a state-of-the-art method, aimed at handling linkage between problem variables. Finally, the test metrics used in this thesis are explained.

2.1. Multimodal Learning

In today's world, a continuous stream of data is produced by a wide array of sources and sensors, manifesting in diverse forms, referred to as modalities, such as images, textual content, audio signals, sensor readings, and numerical data [6, 17, 18]. The field of multimodal learning focuses on developing models that can jointly process and interpret these diverse information sources [18]. The goal is to achieve a deeper and more complete understanding of the data by leveraging the complementary and redundant information across modalities [17, 18, 19]. By combining different modalities, models can make more robust predictions and capture information that might be missing in individual modalities. Research confirms that multimodality often yields better results than using single modalities alone [6, 17].

A main challenge in multimodal classification is the need to capture inter-modality dependencies, i.e., the relationships and interactions between modalities that collectively determine the prediction. While modality-specific encoders are effective at extracting intra-modal patterns, many tasks require reasoning about how different modalities jointly influence the outcome [18, 20]. For example when using imaging and tabular modalities for a classification task, structured numerical features may provide context that alters the interpretation of visual cues, producing a richer and more accurate representation than either modality could in isolation [21]. Ignoring such cross-modal relationships risks overlooking complementary signals, which can reduce both predictive accuracy and interpretability [18, 21]. Recent work therefore emphasises that robust multimodal systems must explicitly model both intra- and inter-modality dependencies to achieve optimal performance [18, 20].

State-of-the-art multimodal models like TIP [22], STiL [23], and TIME [24] report strong performance improvements on joint image and tabular benchmarks. However, these works evaluate only aggregate performance and do not include targeted ablations of modality interactions. In other words, they improve overall accuracy but do not isolate whether the boost comes from better single-modality encoders or genuine cross-modal fusion. In fact, multimodal evaluation protocols typically lack any metric of inter-modality synergy. As one analysis notes, there is "no universal metric" for how much cross-modal information is captured, researchers usually just check if the multimodal model beats unimodal baseline performance [25]. In short, while state-of-the-art multimodal models like TIP [22], STiL [23], and TIME

[24] achieve state-of-the-art accuracy, current benchmarks do not explicitly test whether gains come from true inter-modality dependence or simply stronger unimodal features.

2.2. MultiFIX

MultiFIX (Multimodal Feature engIneering eXplainable artificial intelligence) was proposed as a framework to build partially interpretable multimodal models where the modality-specific features are engineered and explained, as well as the contribution of each modality to the final prediction [14]. It consists of one feature-engineering block per modality and one fusion block, as shown in Figure 2.1. This study focuses on two modalities: image and tabular data. Each feature-engineering block is tailored towards the properties of each data modality, using a Convolutional Neural Network (CNN) for the image modality and a Multilayer Perceptron (MLP) for the tabular modality. For the image modality, a pre-trained ResNet [26] is used to perform feature extraction, as computer vision tasks often benefit from rich, transferable representations learned on large-scale datasets [27]. The fusion block also uses an MLP to combine the engineered features. Detailed descriptions of the architectures are given in Appendix A.

Multiple training strategies can be employed for MultiFIX, with no single approach demonstrating a statistically significant advantage [14]. This thesis focuses on the end-to-end training strategy. In this strategy, the DL pipeline is trained in an end-to-end fashion using the traditional gradient-based Adam optimiser [28] and Binary Cross Entropy (BCE) Loss. End-to-end training is preferred due to it being an intuitive training strategy that performs well. Additionally, a pre-training step for image feature engineering is particularly relevant in this work. Specifically, an Autoencoder (AE) is trained in an unsupervised manner, using the ResNet component of the image feature engineering block as its encoder, to extract representative features of the image input within a linear latent space. The learned encoder weights are subsequently transferred to initialise the ResNet within the MultiFIX architecture. More details about the AE are given in Appendix B.

The inference stage follows the training stage. The MLPs are exchanged with symbolic expressions, which are intrinsically explainable. This is done through GP-GOMEA [29], which is a genetic programming algorithm in the GOMEA family of EAs, further explained in Section 2.4. The relevant DL architectural blocks are used to obtain the inputs and labels for GP-GOMEA. A similar exchange is not possible for the CNN, due to the dimension of the inputs and high complexity of the network. Instead, post-hoc explainability is performed in the form of Grad-CAM [9], which is a technique that highlights the regions of an input image that most strongly influence a model's output, in this case the engineered feature. It does this by using the gradients of the final convolutional layer with respect to the target to produce a heatmap. Overlaying this heatmap on the input image provides a visual understanding of the regions the model focuses on. Combined with the symbolic expressions, this helps explain how the model arrives at its decisions.

2.3. Evolutionary Algorithms

EAs are a class of optimisation techniques inspired by the principles of natural selection and genetics. Unlike traditional optimisation methods that rely on gradients or assumptions about the problem landscape, EAs are population-based and gradient-free, making them well-suited for exploring complex or deceptive search spaces. They work by evolving a population of N candidate solutions over multiple generations, guided by variation operators such as mutation and recombination, and selection mechanisms that favour better-performing individuals. The quality of each solution is evaluated using a fitness function, which maps the candidate's numerical encoding, as l decision variables, to a score reflecting its performance. The EA continues optimising this fitness value until the population converges or a predefined termination criterion is met, such as a maximum runtime, number of fitness evaluations, or a target fitness threshold. A diagram of the general working of an EA is given in Figure 2.2.

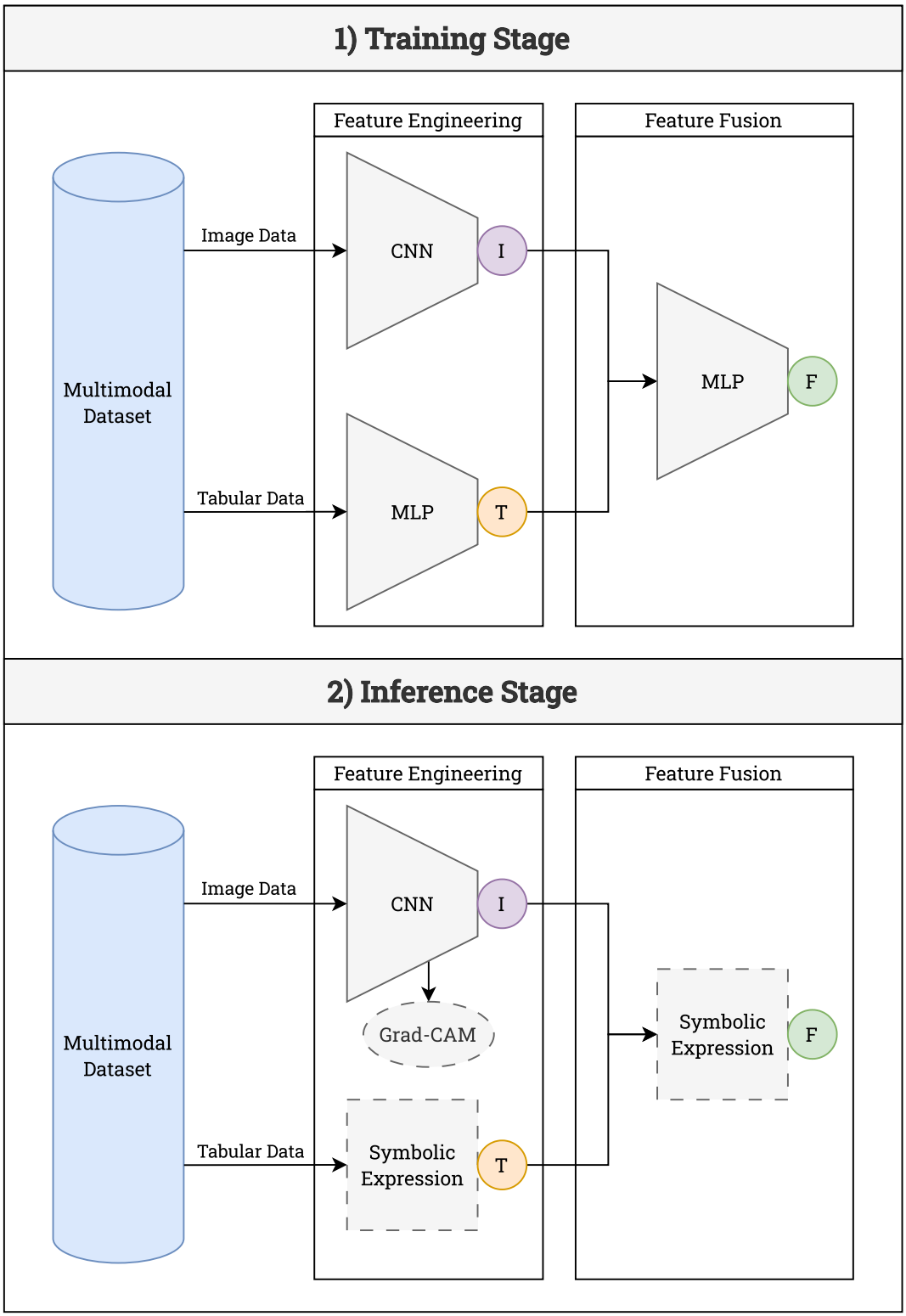


Figure 2.1: Overview of MultiFIX. Data from the multimodal dataset is passed into the feature engineering blocks. Feature vectors I and T are concatenated and passed to the fusion block to make the final prediction F in the Training Stage. In the Inference Stage, image features are explained through Grad-CAM, and both MLPs are replaced with symbolic expressions obtained with GP-GOMEA, using the DL blocks from the training stage to determine the input and labels on which the symbolic expressions are fitted.

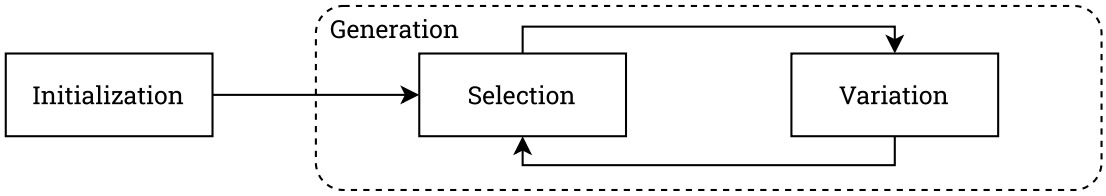


Figure 2.2: Diagram of general optimisation process of an EA.

2.4. Gene-pool Optimal Mixing Evolutionary Algorithm (GOMEA)

Selection guides the optimisation process towards better solutions, while variation introduces new individuals by modifying the existing population. Although recombination of "fit" parents often leads to improvements, inefficient mixing of building blocks can result in exponential increases in population size or the time required to solve a problem [30]. Model-Based Evolutionary Algorithms (MBEAs) aim to identify and effectively mix linked building blocks, which is crucial for achieving efficient and scalable EAs. One class of MBEAs is the family of Gene-pool Optimal Mixing Evolutionary Algorithms (GOMEAs) [31]. These algorithms have shown impressive performance on benchmarks and several practical use cases, such as optimising radiotherapy treatment plans [32], symbolic regression [29], and radiotherapy dose reconstruction [33].

This section starts by explaining the variation operator, Gene-pool Optimal Mixing (GOM), in Section 2.4.1. Secondly, Section 2.4.2 discusses several mutation operators. Section 2.4.3 follows up with an explanation of Forced Improvements (FI) and No Improvement Stretches (NIS). Next, Section 2.4.4 highlights the process of determining the population size. Section 2.4.5 subsequently explains the linkage model, which determines what variables to select for GOM. Finally, Section 2.4.6 explains how the population is initialised through probabilistically complete sampling.

2.4.1. Gene-pool Optimal Mixing (GOM)

GOMEA makes use of GOM as the variation operator. For each solution in the population, different sets of variables are iteratively considered, following a model called a linkage model. For each set, GOM selects a random parent from the previous generation as a donor. The offspring inherits the selected variables of the donor. GOM accepts donor variables only if they do not worsen the fitness of the solution. The linkage model can either be learned during optimisation or supplied by the user, potentially using knowledge about the problem structure, as elaborated on in Section 2.4.5.

2.4.2. Mutation Operators

Mutation is a variation operator that is often used in EAs. In GOMEA, mutation is an additional component that the user can easily switch on or off as desired, since it is not a necessity [34]. At every mixing step during the GOM procedure, mutation can be performed after copying values from a donor to the current solution and before fitness evaluation of the intermediate solution. Mutation is performed with some probability p_m only on the problem variables affected by the cross-over mask of the current mixing step. This probability can be set to a fixed value by the user, or set by one of the following methods: weak mutation or strong mutation [34]. Weak mutation uses a fixed mutation probability of $p_m = 1/l$, where l is the number of decision variables. Strong mutation uses an adaptive mutation probability $p_m = 1/l_{\mathbf{F}^i}$, where $l_{\mathbf{F}^i} = |\mathbf{F}^i|$ is the number of decision variables in the linkage set $\mathbf{F}^i$, i.e., the size of the cross-over mask. The expected number of mutations is much higher for strong mutation than for weak mutation.

2.4.3. Forced Improvement (FI) and No Improvement Stretches (NIS)

While GOM by itself does not ensure solution convergence [35], incorporating supplementary strategies such as FI can facilitate convergence [36]. FI is triggered when GOM is unable to alter the parent solution, which happens when all rounds of GOM worsened the offspring solution. In such cases, a second round of GOM is executed, this time using donor solutions selected either randomly from the elitist solution [31]. Moreover, FI is designed to enforce strict improvement and terminates as soon as a successful mixing step occurs. If FI also fails to improve the parent solution, it is replaced with the

elitist solution. A plateau in the search landscape can cause back and forth variable changes of parent solutions without improving their objective values, thus not triggering FI. This may be partly recognised when the elitist solution shows no improvement over many generations, leading to what is known as a NIS [34]. FI is also triggered when the NIS exceeds the threshold $1 + \lfloor \log_{10}(n_{individuals}) \rfloor$, which has been shown to give good results in the single-objective domain [35].

2.4.4. Population Size Tuning

Properly configuring the population size can significantly influence the performance of an EA [37]. Incorrectly setting this parameter, which is often a tedious and time-consuming task, can give a vastly wrong impression of the capabilities of an algorithm [38]. To eliminate the need for manual tuning, GOMEA can employ an Interleaved Multi-start Scheme (IMS), which runs multiple populations of different sizes independently and advances them in an interleaved fashion. However, this approach introduces computational overhead, which can become prohibitively costly when fitness evaluations are time-consuming. For this reason, an alternative population size tuning mechanism is opted for in this thesis. This process is elaborated on in Chapter 5.

2.4.5. Linkage Model

As mentioned before, the linkage model used in GOM can either be learned during optimisation or supplied by the user. When the linkage model is learned, it is based on the solutions in the population. The linkage tree (LT) structure is then used as the linkage model to capture the dependencies among problem variables. Let L denote the set of indices of all problem variables, i.e., $L = \{0, 1, \ldots, l-1\}$. An LT $\mathfrak{F}$ can be represented as a Family of Subsets (FOS) of L, i.e., $\mathfrak{F} = \{\mathbf{F}^0, \mathbf{F}^1, \ldots, \mathbf{F}^{|\mathfrak{F}-1|}\}$, where $\mathbf{F}^i \subseteq L$ [34]. Each subset defines a linkage group of variables that ideally share some form of dependency, but practically the common quality is that the group of variables is varied together. Respecting these dependencies is shown to improve performance [39].

Linkage Learning

The construction of an LT $\mathfrak{F}$ is achieved through a bottom-up hierarchical clustering algorithm known as the Unweighted Pair Grouping Method with Arithmetic-mean (UPGMA). Initially, each problem variable is treated as an individual linkage group, forming the leaf nodes of the LT. Specifically, for each variable index $i \in L$, a singleton linkage group is defined as $\mathbf{F}^i = \{i\}$. Subsequently, the algorithm iteratively merges the pair of linkage groups exhibiting the highest similarity, as quantified by a chosen distance metric. In this context, the Normalised Mutual Information (NMI) is employed as the similarity measure, where higher NMI values indicate stronger dependencies between variable subsets. At each iteration, the two linkage groups with the highest NMI are merged to form a new linkage group, which is then added to the LT. The original linkage groups involved in the merge are excluded from further consideration in subsequent iterations. This process continues until a single cluster encompassing all problem variables is formed, representing the root node of the LT.

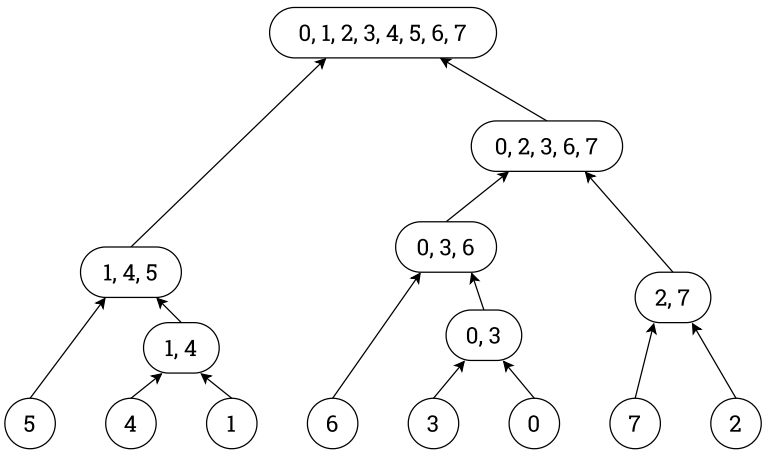


Figure 2.3: An example of a linkage tree with $l = 8$ problem variables [34]

Univariate and Marginal Blocks Linkage Models

In addition to the linkage tree, GOMEA also supports alternative linkage models that can be supplied by the user, allowing incorporation of problem-specific knowledge. The simplest such model is the univariate linkage model, in which each problem variable is considered independent. In this case, the FOS consists solely of singleton subsets, i.e., $\mathfrak{F}_{Univariate} = \{\{0\}, \{1\}, \ldots, \{l-1\}\}$. While this model disregards variable dependencies, it can be effective in problems where dependencies are weak or absent. Another option is the use of marginal blocks, where the FOS is manually defined as a set of variable groups believed to have interdependencies, often based on domain knowledge or problem structure. The FOS is defined as $\mathfrak{F}_{Marginal\ Blocks(b)} = \{\{0, \ldots, b-1\}, \ldots, \{l-b, \ldots, l-1\}\}$, where b is the block size. For example, in structured problems such as additively decomposable functions or models with modularity, grouping variables according to known subcomponents can significantly enhance performance.

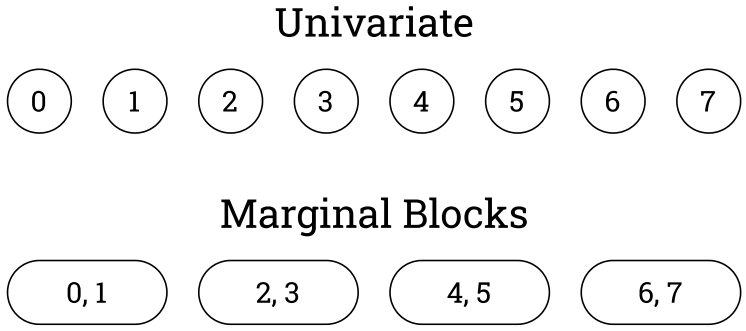


Figure 2.4: An example of a univariate and marginal blocks linkage models with $l = 8$ problem variables and marginal block size $b = 2$

2.4.6. Probabilistically Complete Sampling

Instead of initialising the population by drawing each binary gene of every individual from a Bernoulli distribution with $p = 0.5$, probabilistically complete sampling is employed [40]. Initialisation through probabilistically complete sampling aims to distribute values across the population such that each possible value of a variable appears as equally often as possible. For each variable, values are assigned to individuals in a way that greedily balances the frequency of each value in the population. Once all values are assigned, their positions are shuffled across individuals to remove any ordering bias. In the binary case, this results in each variable having an equal number of zeros and ones when the population size is odd. If the population size is odd, each variable will have one more zero or one.

2.5. Test Metrics

A given DL model can be evaluated for generalisation by testing it on a held-out test set. From this testing, several metrics can be obtained. This thesis focuses on the following test metrics: BCE Loss, Balanced Accuracy (BAcc), and Area Under the Receiver-Operating Characteristic curve (AUROC), all highlighted in the consequent subsections.

2.5.1. Binary Cross Entropy Loss (BCE Loss)

BCE Loss is defined in Equation (2.1), where y is the vector of ground-truth labels and x is the vector of predictions. In case of perfect classification, the loss will be 0 and in case of random guessing $\log(0.5) \approx 0.7$. It is possible for the BCE Loss to be higher than 0.7, because it penalises wrong predictions more heavily than it rewards good predictions.

$$l(x, y) = -\frac{1}{N} \sum_{n=1}^{N} y_n \cdot \log x_n + (1 - y_n) \cdot \log(1 - x_n) \tag{2.1}$$

2.5.2. Balanced Accuracy (BAcc)

BAcc is defined in Equation (2.2), where P and N are the number of positives and negatives respectively, after applying a threshold of 0.5 to the probabilities. TP and TN are the number of true positives and true negatives, respectively. It is a way of measuring classification accuracy that is resistant to imbalanced datasets. Ideal classification will give a balanced accuracy of 1.0, while random guessing in a binary classification problem will result in a balanced accuracy of 0.5.

$$BAcc = \frac{1}{2}(\frac{TP}{P} + \frac{TN}{N}) \tag{2.2}$$

2.5.3. Area Under the Receiver-Operating Characteristic curve (AUROC)

The receiver-operating characteristic curve is a visual representation of model performance across all thresholds. A curve can be drawn by calculating the True Positive Rate ($TPR = \frac{TP}{P}$) and the True Negative Rate ($TNR = \frac{TN}{N}$) for a set of possible thresholds. The area under the receiver-operating characteristic curve is taken to get a single scalar value that represents the probability that the model, if given a randomly chosen positive and negative example, will rank the positive higher than the negative. Ideal classification will, just like balanced accuracy, give an AUROC of 1.0 and random guessing will give 0.5. AUROC extends the leniency of the balanced accuracy metric by considering a range of thresholds instead of only a threshold of 0.5. Therefore, the AUROC is typically higher than the balanced accuracy.

3

Synthetic Problems and Baseline Results

This chapter introduces a set of synthetic multimodal problems designed to systematically evaluate model behaviour in a controlled setting. Synthetic datasets offer a flexible and interpretable environment to test hypotheses about feature extraction, modality interactions, and learning dynamics, allowing specific challenges such as feature redundancy and cross-modal interactions to be isolated and studied in detail.

The synthetic problems developed here combine image and tabular modalities with varying complexity. By constructing tasks based on logical operations of varying difficulty, such as simple conjunctions and XORs, it becomes possible to assess baseline model performance, diagnose failure cases, and motivate the development of more advanced methods presented in later chapters.

Section 3.1 first details the construction of the synthetic dataset, followed by a formal description of the problems under consideration in Section 3.2. Finally, baseline results obtained through hyperparameter optimisation are presented in Section 3.3, which highlights both the successes on simpler problems and challenges on more complex tasks.

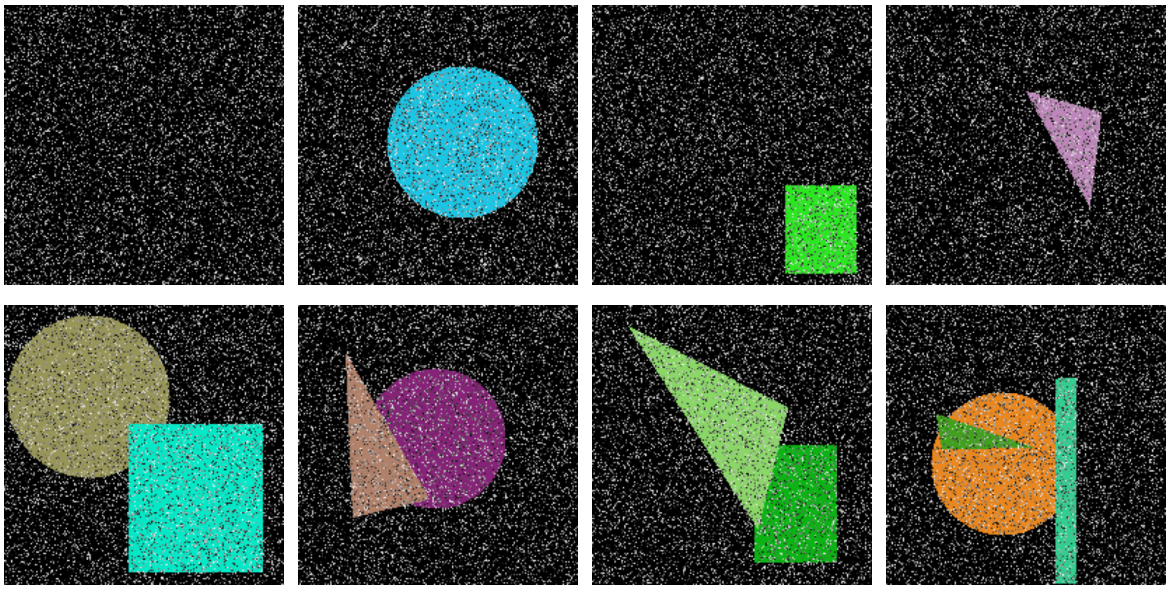


Figure 3.1: Examples of images for each combination of shape.

3.1. Synthetic Dataset

The dataset contains $1{,}000$ samples, each comprising aligned image and tabular data. The images are 200×200 RGB pixels and may include up to three types of shapes: rectangles, circles, and triangles. Each shape has a 50% probability of appearing, allowing for images with none, one, or multiple shapes. The shapes vary randomly in size, colour, and position. Noise was added by altering the colour of $10{,}000$ randomly selected pixels in each image. An example of the image data is shown in Figure 3.1. The tabular data consists of 10 continuous numerical features, each drawn independently from a uniform distribution in the range $[0, 1]$.

3.2. Synthetic Problems

This section introduces the problems focussed on in this thesis. The problems are: AND, single XOR, and three-gated XOR. Two variants of the three-gated XOR are given, where the first consists of two image features and one tabular feature, and the second consists of one image feature and two tabular features. The truth tables are given in Table 3.1. Ideal feature knowledge is available for the synthetic problems, since the underlying problem structure is known. This knowledge is used throughout this thesis to test the proposed solution. However, it should be noted that this knowledge is unknown in real-world scenarios.

The MultiFIX pipeline is designed to engineer a variable amount of features per modality. For simplicity, this thesis limits feature engineering to one feature per modality. For the AND and single XOR, the ideal image feature is the presence of a circle and the ideal tabular feature is whether tabular variable x_1 is bigger than x_2. However, for the three-gated XOR, the associative property is used to enforce one feature per modality: $XOR(circle, triangle, x_1 > x_2) = XOR(XOR(circle, triangle), x_1 > x_2)$. The same reasoning applies to the three-gated XOR with two tabular features.

Table 3.1: Truth tables for the four problems MultiFIX is subjected to. Image features $circle$ and $triangle$ indicate the presence of the respective shape in the image. Tabular features $x_1 > x_2$ and $x_3 > x_4$ represent whether the first tabular numerical feature is bigger than the second. The tabular numerical features are numbered from zero to nine.

(a) Truth table for AND problem

$circle$	$x_1 > x_2$	AND
0	0	0
0	1	0
1	0	0
1	1	1

(b) Truth table for XOR problem

$circle$	$x_1 > x_2$	XOR
0	0	0
0	1	1
1	0	1
1	1	0

(c) Truth table for three-gated XOR problem with two image features and one tabular feature

$circle$	$triangle$	$x_1 > x_2$	3-XOR
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

(d) Truth table for three-gated XOR problem with one image feature and two tabular features

$circle$	$x_1 > x_2$	$x_3 > x_4$	3-XOR
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

3.3. Baseline Results

Establishing a strong baseline is crucial for assessing model performance and guiding further experimentation. The baseline for this thesis will be the original MultiFIX pipeline after the training stage, i.e., the DL architecture and not the explainable alternative. For each synthetic multimodal tasks discussed in the previous section, Hyperparameter Optimisation (HPO) was conducted over a predefined grid search space as defined in Table 3.3. This included the amount of features each modality should engineer to let the model choose the number of relevant features. Data was split using a $80\%/20\%$ stratified train-test data split, with 5-fold stratified cross validation performed on the 80%. Models were trained in an end-to-end fashion using BCE loss for a maximum of 75 epochs with early stopping after 5 epochs with no improvement in the validation loss. Afterwards, evaluation was performed on the test set, obtaining the BCE loss, AUROC, and BAcc. Table 3.2 reports the results for the discussed setup.

The baseline results reveal clear performance patterns across the tasks. Simpler tasks, such as the AND and single XOR, achieve high AUROC and BAcc scores, while the more complex Three-gated

XOR tasks demonstrate substantially lower performance, highlighting the increased difficulty introduced by extreme dependency between features across modalities. Results also show that the complexity of the three-gated XOR increases when using two image features and one tabular feature instead of two tabular features and one image feature. Therefore, this thesis focuses on improving upon the baseline performance for the three-gated XOR task with two tabular features, as this problem represents the logical next step after the single XOR. In addition, the AND and single XOR tasks are also optimised using the proposed method, allowing comparison with problems for which the original MultiFIX setup already achieves strong performance.

Table 3.2: Baseline performance on synthetic multimodal tasks following HPO over a predefined grid, described in Table 3.3. Models were trained end-to-end with BCE loss using stratified 5-fold cross-validation on an $80\%/20\%$ stratified train-test data split. For each problem, **#IMG** and **#TAB** denote the number of image and tabular features, respectively, found through HPO. **LR** refers to the learning rate, **WD** to the weight decay. The following results are obtained through evaluation on the test set: **Loss** corresponds to the BCE loss, **AUROC** indicates the area under the receiver operating characteristic curve, and **BAcc** denotes the balanced accuracy.

Problem	**Hyperparameters**				**Results**		
	#IMG	*#TAB*	*LR*	*WD*	*Loss*	*AUROC*	*BAcc*
AND	3	1	1e-3	1e-4	0.139	0.987	0.930
XOR	2	2	1e-3	0.0	0.198	0.979	0.919
Three-gated XOR (two image features)	3	1	1e-3	1e-4	0.691	0.529	0.515
Three-gated XOR (two tabular features)	3	1	1e-3	1e-4	0.596	0.661	0.617

Table 3.3: HPO grid search space for obtaining the baseline results.

Hyperparameter	**Range**
#IMG	[0, 1, 2, 3]
#TAB	[0, 1, 2, 3]
LR	[1e-3, 1e-4, 1e-5]
WD	[0.0, 1e-3, 1e-4]

4

Using Feature Knowledge in Training MultiFIX

This chapter assumes that the ideal features are known and consequently investigates how this knowledge can enhance the training of MultiFIX. In the original setup, the fusion label provides only indirect supervision to the feature engineering DL blocks. The hypothesis is that incorporating knowledge of the ideal features to explicitly guide feature engineering will yield more effective features due to more direct supervision. The resulting features are, in turn, expected to improve overall MultiFIX performance beyond the baseline described in Chapter 3. It should be noted that ideal feature knowledge is only available for the previously defined synthetic problems and is typically unknown for real-world scenarios. The research question guiding this chapter is restated below.

Research Question

RQ1. How can feature knowledge be used in training MultiFIX and does it improve performance on tasks with extreme joint modality dependence?

The chapter proceeds as follows: first, Section 4.1 describes how feature knowledge is expressed in feature targets and consequently integrated into the training process of MultiFIX. Then, Section 4.2 outlines the evaluation methodology used to assess the performance of the resulting model. Finally, Section 4.3 presents an experiment analysing the relationship between feature target noise and MultiFIX performance.

4.1. Blockwise Supervised Training (BST)

Assuming perfect knowledge of the ideal features, each multimodal training sample can be annotated with binary feature targets, one per modality, indicating the presence of the corresponding ideal feature. In this thesis, each sample is annotated with exactly one feature label per modality, which is typically problem specific. However, this thesis focuses on one feature per modality to limit the scope of research. The set of true feature targets can be formalised for n_{BST} samples as a binary vector of targets $\mathbf{t}^{true}$ as illustrated in Equation (4.1), where I_i and T_i are the true binary feature targets of the i^{th} sample for the image and tabular modality, respectively.

$$\mathbf{t}^{true} = (I_0, T_0, \ldots, I_{n-1}, T_{n-1}) \in \{0,1\}^{2n_{BST}} \tag{4.1}$$

Consequently, these true feature targets can be used in Blockwise Supervised Training (BST). BST is a modular training strategy in which each block of MultiFIX is trained in isolation using intermediate supervision. An overview of the process is given in Figure 4.1. The feature targets are used to independently supervise the feature engineering blocks prior to training the fusion block, for which two

alternative strategies are considered in Section 4.1.1. BST can operate not only with true feature targets but also with noisy approximations, where certain samples include incorrectly annotated features. Therefore, the term *targets* is adopted instead of *labels* to reflect their function as learnable objectives that may not correspond to true feature annotations.

4.1.1. Idealised and Chained Fusion

The first option for training the fusion block in BST is idealised fusion. In this setting, the fusion block is trained using the feature targets as its direct input. This assumes perfect feature extraction, decoupling the fusion block from any errors made in the feature engineering blocks. As a result, the fusion block can focus solely on modelling inter-modality relationships in an idealised setting. The second option is chained fusion. Here, the trained feature engineering blocks are frozen, and their outputs are used as inputs to the fusion block, following the MultiFIX architecture. This approach is more representative of real-world inference, where the fusion block must operate on imperfect, noisy feature representations.

Each approach has its advantages and limitations. Idealised fusion provides a clean training signal for the fusion block, isolating it from feature extraction noise and enabling the modelling of optimal fusion behaviour. However, the idealised conditions under which the fusion block is trained are not replicated during inference, where the block must operate on the learned, and potentially imperfect, outputs of the feature engineering blocks. In contrast, chained fusion reflects the actual deployment scenario, promoting robustness to feature representation imperfections, but may suffer from error propagation if the feature engineering blocks are suboptimal. However, chained fusion penalises the feature engineering loss twice, by propagating the feature engineering errors to the fusion block. The aim of BST is to bias the DL blocks to the feature targets. This effect is less strong for chained fusion, because the fusion block is biased to the output of the feature engineering blocks, and not necessarily the feature targets. Therefore, the preference is given to idealised fusion.

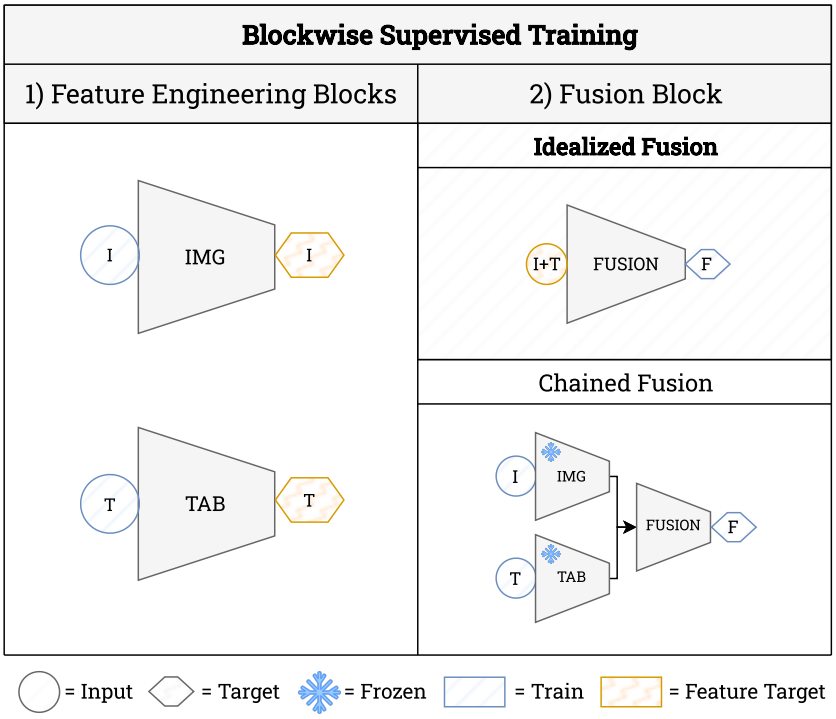


Figure 4.1: Overview of blockwise supervised training. First, the feature engineering blocks are trained in a supervised fashion using the feature targets. Secondly, the fusion block is trained. The first option to do this is idealised fusion: input the feature targets directly into the fusion block. The second option is chained fusion: freeze the weights of the feature engineering blocks and chain the output of these blocks to the input of the fusion block. The first option, idealised fusion, is preferred. Note the distinction between the shape for a target, which means what target is used during training, and the colour of the feature target, which is the set of feature targets, whether it is used as a target or a input.

4.1.2. Data Split and Blockwise Supervised Training Ratio

As stated in Section 3.1, the dataset contains $1{,}000$ samples, where each sample consists of aligned image and tabular data. Each sample also contains a ground-truth fusion label. A $80\%/20\%$ train-test split, stratified on the ground-truth fusion label, is performed. As mentioned before, when ideal feature knowledge is available, each training sample can be annotated with feature targets to allow for BST. However, ideal feature targets are unknown in real-world scenarios and should, as discussed later in this thesis, be optimised for. To limit the problem size of this optimisation, the BST ratio $r_{BST} \in (0, 1]$ is introduced. This ratio is used in obtaining the BST set by downsampling the training set to the BST ratio, also stratified to the ground-truth fusion label. An overview of this is given in Figure 4.2.

In obtaining the BST set, it is not possible to perform stratification on the true feature targets, since these are absent for real-world scenarios. Therefore, it is possible to have different feature distributions between sets. To better understand the potential variability, multiple data splits should be evaluated using different random seeds. Two different seeds should be defined in creating the data split, where the first seed is used for creating the $80\%/20\%$ train-test split, referred to as the *data split seed*, and the second seed is used for downsampling the train set, referred to as the *downsample seed*. The data split seed is set to 0 throughout this thesis, unless specified otherwise, to ensure all evaluations use the same test set and allow for direct comparison.

The selection of the BST ratio involves competing considerations. A higher ratio, and therefore bigger set, is generally associated with improved test performance under ideal labelling conditions, thereby reflecting the model's upper-bound capabilities. As mentioned before, the ultimate goal of this thesis is to optimise feature targets using an EA. Increasing the ratio also expands the number of feature targets that must be optimised for, thereby raising the complexity of the search space and the computational burden on the EA. These two factors are inherently in tension and thus the following range of BST ratios are investigated in this thesis: $r_{BST} \in \{0.25, 0.5, 0.75, 1.0\}$.

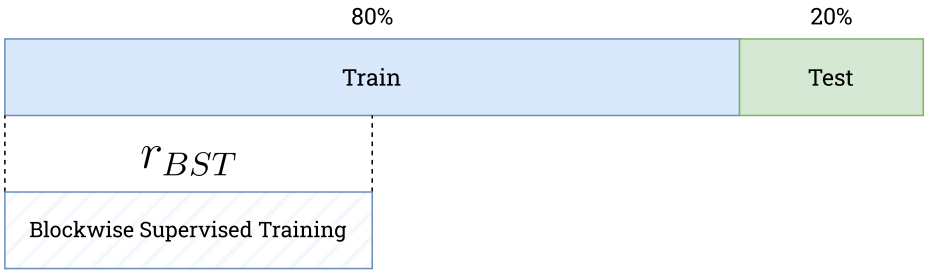


Figure 4.2: Overview of data split for Blockwise Supervised Training (BST). First, a $80\%/20\%$ stratified train-test split is performed. Then, the train set is downsampled to the BST ratio r_{BST} in a stratified fashion. This downsampled set is used in BST (see Figure 4.1).

4.1.3. Additional End-to-End Training

After BST, additional end-to-end training can be performed. The full DL architecture of MultiFIX, composed of the three DL blocks, uses the weights obtained from BST and is consequently trained end-to-end to predict the ground-truth fusion labels. It should be noted that parts that were frozen before BST are kept frozen. End-to-end training allows all blocks to be refined in a unified manner and potentially mitigating the effects of noisy or imperfect feature targets encountered during BST. Additional end-to-end training does not use feature targets, and is therefore not limited to training samples for which feature targets are available. Therefore, the entire train set is used for end-to-end training, instead of the downsampled BST set.

4.1.4. Optimisations

As discussed later in this thesis, the previously discussed training procedure will be used to perform fitness evaluation in an EA. During the optimisation process, fitness evaluation must be performed repeatedly, potentially thousands of times, making it the dominant contributor to overall runtime. Because fitness is evaluated so often, the computational cost of performing BST and potentially end-to-end training escalates rapidly, leading to significant runtime.

To mitigate this, several optimisations were introduced to reduce the evaluation time without sacrificing the quality of the resulting models. The primary computational bottleneck in the training pipeline is the ResNet used within the image feature-engineering block, due to its large number of parameters. Fortunately, this network is pre-trained and can be further fine-tuned in a task-specific manner before the EA is run. Following the approach described in [15], an AE is trained in an unsupervised fashion, where the ResNet forms the encoder. This pre-training step encourages the model to learn compact and informative latent representations of the input images. More details on the AE are given in Appendix B.

Once pre-training is completed, the ResNet component is frozen throughout the entire training process, and its outputs, i.e., the image embeddings, are cached. Consequently, for the image block only the lightweight output layer is trained. Therefore the computational load is significantly reduced by avoiding redundant forward passes through the full ResNet. This caching strategy preserves the quality of feature extraction while improving runtime efficiency.

In addition to architectural optimisations, software-level improvements were applied. Notably, PyTorch's automatic mixed precision was used to reduce the precision of floating-point operations from 32-bit to 16-bit wherever advantageous. This provides considerable speed-ups in both training and inference without compromising model accuracy, especially on modern GPUs optimised for mixed-precision computation.

4.1.5. Deep Learning Parameters

DL parameters can have a great impact on the performance of a model. Because of this, the parameters used in BST and additional end-to-end training were determined based on preliminary experimental tuning for the three-gated XOR task with two tabular features. This included performing grid search over a range of typical values for the learning rate and weight decay parameters, after which the results were manually analysed.

The results are obtained through testing with the true feature targets. It should once again be noted that this is not possible for real-world problems, where the true feature targets are not available. However, the ideal feature knowledge is used in this proof-of-concept to get a better idea of the upper limit of the proposed procedure.

First, the parameters of the feature engineering blocks were determined. These could then be used to determine the parameters for the fusion block. After determining all the parameters for BST, the parameters for additional end-to-end training were determined. The obtained DL settings are given in Table 4.1.

Table 4.1: Hyperparameter configurations used for Blockwise Supervised Training (BST) and additional end-to-end training. The Adam optimiser is employed across all training stages. Settings for the image, tabular, and fusion blocks pertain to the BST phase, while the end-to-end training settings apply to the additional end-to-end training of the integrated architecture. A consistent batch size is maintained across all procedures. All configurations were determined based on preliminary experimental tuning using the typically unavailable ground-truth feature targets.

DL Parameter	Image Block	Tabular Block	Fusion Block	End-to-End
Learning Rate	5e-3	5e-4	5e-3	5e-4
Weight Decay	1e-4	1e-4	1e-2	1e-2
Number of Epochs	60	40	25	25
Batch Size	200	200	200	200

4.1.6. Training Time

As discussed later in this thesis, the training procedure is used for fitness evaluation within the EA, which is the dominant contributor to the computational cost. It is therefore important to estimate training time both with and without additional end-to-end training. The training process was repeated $n = 100$ times using the DL parameters from Table 4.1 and the optimisations as described in the previous section on the hardware given in Table 4.3. The training time estimates are reported in Table 4.2. As expected, the training time increases as the BST ratio increases, because the BST set size increases, which should be considered when selecting the BST ratio.

Table 4.2: Training time estimates for Blockwise Supervised Training (BST) both with and without end-to-end training (E2E) for several BST ratios. The training process used the true feature targets and was repeated for $n = 100$ times.

BST Ratio	Time [s]	
	BST	*BST + E2E*
25%	0.822 ± 0.135	1.984 ± 0.269
50%	1.491 ± 0.195	2.700 ± 0.329
75%	2.200 ± 0.269	3.437 ± 0.355
100%	2.901 ± 0.351	4.027 ± 0.454

Table 4.3: Hardware and software specifications of the computational system used to conduct all experiments.

CPU	2 x Intel(R) Xeon(R) Silver 4110 CPU @ 2.10GHz (16 threads)
RAM	96 GB
GPU	4 x GeForce RTX 2080 Ti (11 GB GDDR6 Memory)
CUDA Version	11.7
PyTorch version	2.0.1

4.2. Testing True Feature Targets

After BST, all blocks can be assembled into the full MultiFIX DL architecture and be tested on the held-out test set. Every sample in the test set is inferred through the model and its output is evaluated against the ground-truth fusion labels. The following metrics, as detailed in Section 2.5, are used: BCE Loss, BAcc, AUROC. The true feature targets were tested on the three-gated XOR with two tabular features task for 8 data splitting seeds and 10 training seeds across several BST ratios, resulting in $n = 8 \cdot 10 = 80$ samples per BST ratio. Table 4.4 shows that the test results under ideal feature knowledge improve as the BST ratio increases, as hypothesised earlier in this thesis. In addition, performing end-to-end training after BST is shown to improve the test results over only applying BST.

Table 4.4: Test results after Blockwise Supervised Training (BST) with true feature targets on the three-gated XOR with two tabular features task, both with and without end-to-end training (E2E), for several BST ratios. The process was repeated for 8 data splitting seeds and 10 training seeds, resulting in $n = 8 \cdot 10 = 80$ samples per BST ratio. For each set of samples, the average and standard deviation of the Binary Cross Entropy Loss (BCE Loss), Area Under the Receiver-Operating Characteristic curve (AUROC), and Balanced Accuracy (BAcc) were obtained. The arrows indicate whether the metric should be minimised or maximised.

(a) Test results after BST using the true feature targets

BST Ratio	Test Results BST		
	BCE Loss ↓	*AUROC* ↑	*BAcc* ↑
25%	0.649 ± 0.042	0.717 ± 0.044	0.630 ± 0.044
50%	0.506 ± 0.026	0.848 ± 0.022	0.750 ± 0.033
75%	0.465 ± 0.055	0.876 ± 0.029	0.781 ± 0.038
100%	0.448 ± 0.071	0.900 ± 0.025	0.803 ± 0.034

(b) Test results after BST using the true feature targets and end-to-end training

BST Ratio	Test Results BST + E2E		
	BCE Loss ↓	*AUROC* ↑	*BAcc* ↑
25%	0.496 ± 0.053	0.842 ± 0.033	0.745 ± 0.032
50%	0.438 ± 0.050	0.881 ± 0.027	0.783 ± 0.037
75%	0.415 ± 0.051	0.896 ± 0.024	0.799 ± 0.034
100%	0.405 ± 0.051	0.904 ± 0.022	0.810 ± 0.031

4.3. Noisy Feature Targets

As shown in the previous section, BST is able to provide better-than-baseline MultiFIX performance when using the ground-truth feature targets. However, similar results can possibly be obtained when one or more feature targets are incorrectly assigned, i.e., for noisy feature targets. This section investigates how noise in the feature targets affects the performance of the resulting MultiFIX model after BST, both with and without end-to-end training.

First, the process of creating noisy feature targets is described in Section 4.3.1. Afterwards, Section 4.3.2 discusses how to determine the similarity between a set of noisy feature targets and the true feature targets by calculating the feature correlation. Finally, Section 4.3.3 tests the noisy feature targets in a similar fashion as for the true feature targets and analyses the effect of noise on the test results.

4.3.1. Creating Noisy Feature Targets

To generate noisy approximations of the true feature targets $\mathbf{t}^{true} \in \{0,1\}^{2n_{BST}}$ (see Equation (4.1)), a stochastic noise injection process parametrised by a retention ratio $\alpha \in [0,1)$ is defined. The retention ratio α determines how many true targets are retained and consequently how many are replaced with noise. Specifically, for a given α, a fraction α of the entries in $\mathbf{t}^{true}$ is selected uniformly at random and copied into a new vector $\hat{\mathbf{t}}^{noisy} = (\hat{I}_0, \hat{T}_0, \ldots, \hat{I}_{n_{BST}-1}, \hat{T}_{n_{BST}-1}) \in \{0,1\}^{2n_{BST}}$. The remaining $(1-\alpha)\cdot 2n_{BST}$ entries are replaced with samples from a Bernoulli distribution with $p = 0.5$, i.e., uniformly random binary values.

This process is repeated across $\alpha \in \mathcal{A} = \{0, \frac{1}{20}, \ldots, \frac{19}{20}\}$, a set of 20 evenly spaced retention ratios in $[0,1)$, and $S = 1{,}000$ distinct seeds for the three-gated XOR with two tabular features task, which results in $|\mathcal{A}| \cdot S = 20{,}000$ sets of noisy feature targets. All sets of noisy feature targets are created using a single data-split seed and downsample seed. The resulting $20{,}000$ feature targets, with differing levels of noise, can be used in investigating the effect of noise on MultiFIX performance.

4.3.2. Feature Correlation

To examine the difference between noise levels, it is important to quantify the amount of noise in a given set of feature targets. The retention ratio applied in noise injection serves as a measure of expected noise. However, because noise is added randomly, it is possible, though unlikely, that the injected values closely resemble the originals, resulting in less effective noise than expected from the retention ratio. Therefore, a more direct way of determining similarity to the ideal feature targets is necessary. Consequently, this section introduces the feature correlation.

When determining the correlation to the ground-truth feature targets, it is important to take into account that the complement of a feature is considered to be equally ideal. For example, if the image feature is whether or not a circle is present, it should be equally preferred to engineer the feature that is the absence of a circle. Therefore, the average absolute correlation $\rho^* \in [0,1]$ between a set of feature targets $\hat{\mathbf{t}}$ and the set of the true feature targets $\mathbf{t}^{true}$ is defined as follows.

Let $\mathbf{I}$ (see Equation (4.2)) and $\mathbf{T}$ (see Equation (4.3)) be the modality-specific subvectors of the true feature targets for the image and tabular modality, respectively. The same is done for the feature targets, resulting in $\hat{\mathbf{I}}$ (see Equation (4.4)) and $\hat{\mathbf{T}}$ (see Equation (4.5)). Then, the modality-wise absolute Pearson correlation is taken, resulting in ρ_I (see Equation (4.6)) and ρ_T (see Equation (4.7)). Taking the absolute ensures the property of complement invariance. These absolute correlations are averaged to obtain the final average absolute correlation ρ^* (see Equation (4.8)). In this thesis, this is simply referred to as the feature-target correlation. As utilised in a later chapter, it is also possible to use the continuously valued engineered features, as obtained from the feature engineering DL blocks, instead of the binary feature targets. This is referred to as the engineered-feature correlation.

$$\mathbf{I} = (I_0, \ldots, I_{n_{BST}-1}) \tag{4.2}$$

$$\mathbf{T} = (T_0, \ldots, T_{n_{BST}-1}) \tag{4.3}$$

$$\hat{\mathbf{I}} = (\hat{I}_0, \ldots, \hat{I}_{n_{BST}-1}) \tag{4.4}$$

$$\hat{\mathbf{T}} = (\hat{T}_0, \ldots, \hat{T}_{n_{BST}-1}) \tag{4.5}$$

$$\rho_I = |corr(\mathbf{I}, \hat{\mathbf{I}})| \tag{4.6}$$

$$\rho_T = |corr(\mathbf{T}, \hat{\mathbf{T}})| \tag{4.7}$$

$$\rho^* = \frac{1}{2}(\rho_I + \rho_T) \tag{4.8}$$

4.3.3. Testing Noisy Feature Targets

The aforementioned $20{,}000$ sets of noisy feature targets with differing levels of noise are tested on the three-gated XOR with two tabular features task to assess the effect of varying levels of noise on the learning ability of BST, both with and without end-to-end training. First, each set of noisy feature targets is used in BST and tested similar to the true feature targets in Section 4.2. This results in a BCE loss, AUROC, and BAcc for each entry. Afterwards, additional end-to-end training is performed and the resulting model is tested, resulting in a second set of test metrics. All sets of noisy feature targets are trained and tested on 5 train seeds.

The sets of noisy feature targets are grouped together by noise level to analyse the difference between test results. Grouping is performed assigning the correlation of the noisy feature targets $\rho^* \in [0, 1]$ to the closest point in a set of 21 evenly spaced bins $\mathcal{B} = \{0, \frac{1}{20}, \frac{2}{20}, \ldots, 1\}$ such that $\hat{\rho^*} = \arg\min_{b \in \mathcal{B}} |\rho^* - b|$.

In addition to the $20{,}000$ sets of noisy feature targets, the ground-truth feature targets are included as a reference. Results obtained from using the true feature targets are naturally placed in the bin corresponding to perfect correlation, i.e., a feature-target correlation of 1.

The true feature targets are tested for more than 5 train seeds to maintain roughly equal sample counts per bin. In the noise injection process, $1{,}000$ sets of noisy feature targets were generated per retention ratio, which loosely corresponds to the correlation bin. Each set of noisy feature targets is tested for 5 training seeds, resulting in roughly $5{,}000$ samples per bin. Therefore, the true feature targets are tested for $5{,}000$ training seeds to obtain a roughly similar sample count per bin.

For each bin, the mean and interquartile range (IQR) of all test metrics are computed and plotted, distinguishing between results with and without end-to-end training. This allows assessment of how noise in feature targets affects MultiFIX performance under both conditions. The procedure is repeated for each BST ratio, $r_{BST} \in 0.25, 0.5, 0.75, 1.0$, providing insight into the influence of the BST ratio on noise resilience. The results are given in Figure 4.3.

Discussion BST

When first looking at the plots without end-to-end training, it can be seen that a higher BST ratio is associated with better test results when only applying BST, which was also shown in Table 4.4a. Not only do the plots show that the test results are better for high feature-target correlations, but all levels of noise generally perform better with a higher BST ratio.

Next to that, the baseline is improved upon when using imperfect feature targets in only BST for ratios 100%, 75%, and 50%. A higher ratio also means that the baseline results are improved upon with noisier feature targets. Figure 4.3g shows that for a BST ratio of 25% the AUROC baseline is slightly improved upon when using feature targets that have a correlation of 0.8 or higher, while the BCE test loss and BAcc do not improve upon the baseline values.

Although Table 4.4a indicated that using true feature targets should, on average, improve both AUROC and BAcc for a BST ratio of 25%, the results in Figure 4.3g show the opposite, with lower performance across all metrics. In this case, the data points corresponding to the true feature targets (correlation 1.0) deviate from the overall trend, whereas the results in Table 4.4a align more closely. This deviation is not observed for a BST ratio of 50% (Figure 4.3e), but reappears in the BCE loss for BST ratios of 75% and 100%, as shown in Figure 4.3c and Figure 4.3a. For the latter, the effect is most pronounced: the BCE loss is, on average, lower when using feature targets with a correlation of 0.95 than when using the true feature targets, though this pattern does not extend to AUROC or BAcc.

The fact that this effect appears only in the BCE loss suggests that the model may be overly confident in its incorrect predictions, which are penalised more heavily by BCE than by accuracy-based metrics such as AUROC and BAcc. One reason for why this calibration issue occurs more strongly with the true feature targets than with feature targets of correlation 0.95 is that a small amount of noise acts as regularisation. The added noise introduces a slight bias but reduces overconfidence in wrong predictions, thereby improving the BCE loss while maintaining similar AUROC and BAcc values. As the dataset used for BST decreases in size with lower BST ratios, the model's variance increases while its bias decreases. Consequently, the beneficial regularisation effect of low-level noise diminishes, as observed for BST ratios of 75% and 50% in Figure 4.3c and Figure 4.3e, respectively. However, when the

BST dataset reduces further, variance dominates, and low-level noise once again exerts a beneficial regularising effect compared to using no noise, as observed for a BST ratio of 25% in Figure 4.3g.

Another reason for the discrepancies between the results for the true feature targets in Figure 4.3 and Table 4.4 is that the method of creating and testing true feature targets differs between experiments. Notably, the results in Table 4.4 are obtained over a range of data split seeds, while the experiment in this section used only one data split seed. The break of trend can also be partly explained by the different method of testing for the true feature targets in comparison to the method used for noisy feature targets. Analysing the predictions of the resulting models and comparing these for feature targets with correlation 0.95 and 1.0 could give more insight into this phenomenon.

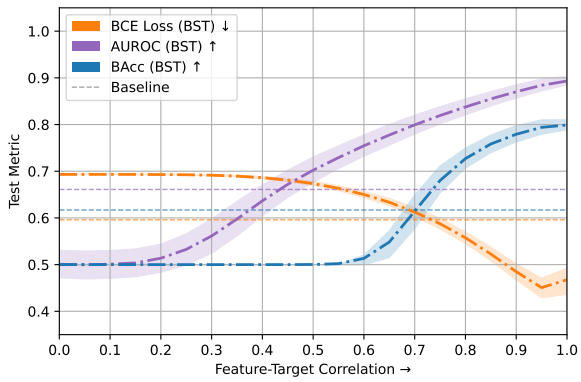
Discussion BST and End-to-End Training

When examining the plots with both BST and end-to-end training, it becomes clear that this approach generally outperforms only applying BST. In particular, AUROC and BAcc improve on average across all noise levels. However, at high noise levels, the test loss exceeds the value expected from random guessing (0.7). Despite this increase in BCE loss, models trained with high levels of noise still achieve better AUROC and BAcc values above random guessing performance (0.5 for both). This is due to the calibration issue as discussed before, where BCE loss penalises overconfident misclassifications more severely than accuracy-based metrics.

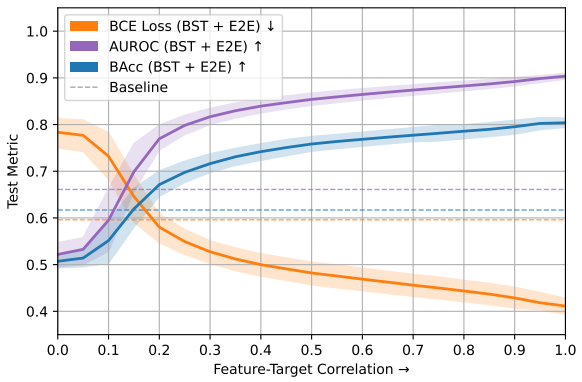
Furthermore, higher BST ratios require less correlated feature targets to achieve losses better than random guessing. Next to that, the discrepancies between the results of the true feature targets (correlation 1.0) between Table 4.4b and Figure 4.3 are minimal, with the results in this section generally being slightly better. Also the break of trend at perfect correlation is not as strong when compared to the plots without end-to-end training. However, all metrics for a BST ratio of 25% worsen for correlation of 1.0 in comparison to a correlation of 0.95, which is similar to what is shown in Figure 4.3g without end-to-end training. This effect weakens when the BST ratio increases.

Conclusion

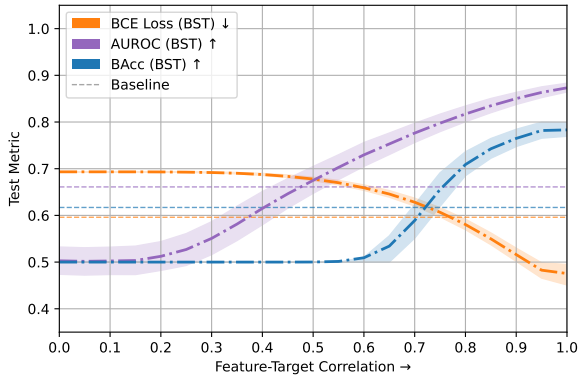
In conclusion, Figure 4.3 demonstrates that noisy feature targets are able to produce models through BST that outperform all baseline test metrics for the three-gated XOR with two tabular features problem, provided a BST ratio of at least 50%. Moreover, end-to-end training is shown to be generally beneficial, enabling feature targets with higher levels of noise to also exceed the baseline metrics, even at a BST ratio of 25%. In general, a higher BST ratio obtains better test results across all noise levels, both with and without end-to-end training. Although decent scores for AUROC and BAcc are obtained, the BCE test loss remains worse than 0.4, reflecting modest performance. This is likely due to the previously discussed calibration issue, which caused breaks of trend in BCE loss stemming from the regularising effect of low-level noise at certain BST ratios and led to BCE losses exceeding those expected from random guessing under high levels of feature target noise with end-to-end training.



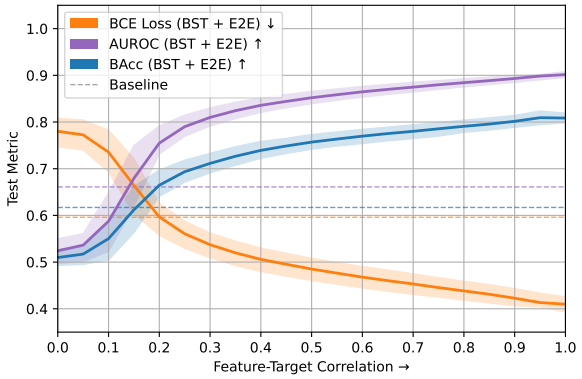
(a) BST ratio of 100% without end-to-end training



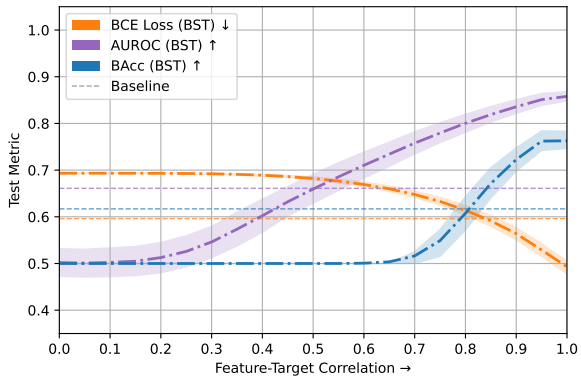
(b) BST ratio of 100% with end-to-end training



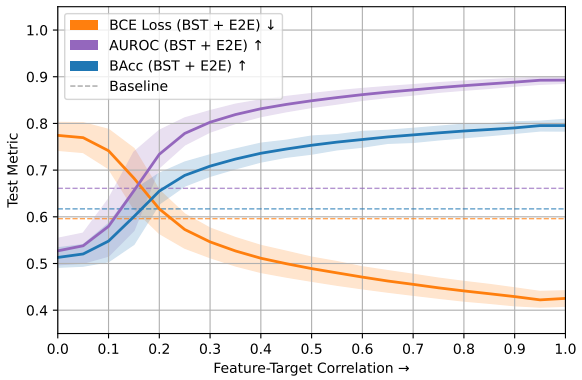
(c) BST ratio of 75% without end-to-end training



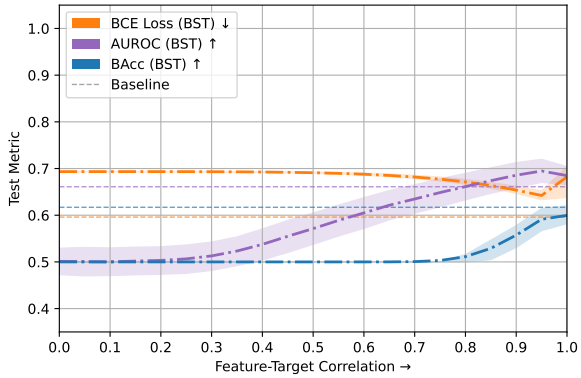
(d) BST ratio of 75% with end-to-end training



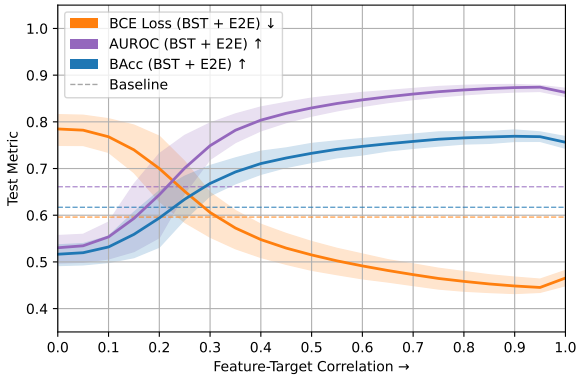
(e) BST ratio of 50% without end-to-end training



(f) BST ratio of 50% with end-to-end training



(g) BST ratio of 25% without end-to-end training



(h) BST ratio of 25% with end-to-end training

Figure 4.3: Correlation of feature targets used in Blockwise Supervised Training (BST) to the true feature targets versus the resulting test metric values, both before and after end-to-end training (in short referred to as E2E) for BST ratios 100%, 75%, 50%, and 25%. Test metrics include: Binary Cross Entropy Loss (BCE Loss), Area Under the Receiver-Operating Characteristic curve (AUROC), and Balanced Accuracy (BAcc). Noisy feature targets were created by injecting differing levels of noise to the true feature targets, which were binned according to their feature-target correlation. Each bin corresponds of roughly $5{,}000$ samples. The test results after BST and before end-to-end training are given in the left column of plots, where the means and interquartile ranges per bin are shown by the dash-dot lines and shaded areas, respectively. The test results after BST and end-to-end training are given in the right column of plots, which instead uses solid lines to show the means. Each plot also shows the baseline test metric values achieved from the baseline method as given in Table 3.2. The arrows in the labels denote whether a metric should be minimised or maximised.

5

Correlation-Based Optimisation of Feature Targets using GOMEA

This chapter builds upon the findings presented in the previous chapter. There, it was shown that using the true feature targets in BST enables MultiFIX to outperform its baseline on the three-gated XOR with two tabular features. However, the true feature targets are only known because the problems at hand are synthetic, and they are typically unknown for real-world scenarios. Therefore, this thesis aims to optimise approximations of the ground-truth feature targets. Such approximations have also been shown to improve MultiFIX performance over the baseline when used within BST, given a certain approximation quality. This chapter elaborates on the process of optimising such feature targets and is guided by the following research question.

Research Question

RQ2. If the correlation to the ground-truth features is available, can GOMEA be used to approximate the ideal feature targets, and what parameter configuration yields the best performance?

To address this question, Section 5.1 introduces the chosen EA, GOMEA, along with its parameters. An idealised version of the optimisation problem is used to evaluate GOMEA's effectiveness under various parameter configurations, with the goal of identifying the most suitable parameter settings. In this setup, the fitness of a candidate set of feature targets is obtained by calculating the correlation to the ground-truth feature targets of the three-gated XOR with two tabular features task.

The subsequent chapter will extend this work to a more realistic and computationally expensive setting, where fitness evaluation is performed without access to the true feature targets, closer to real-world conditions. However, since fitness evaluation based on feature-target correlation is relatively efficient, optimisation can be repeated across multiple configurations to better facilitate parameter selection.

Finally, Section 5.2 discusses the results of varying the population size and linkage model, whilst omitting mutation. The effect of introducing mutation on optimisation performance is then examined in Section 5.3. Together, these experiments provide a comprehensive understanding of how different GOMEA configurations influence its ability to approximate ideal feature targets from correlation-based fitness evaluations. This understanding forms the foundation for the more realistic optimisation scenario explored later this thesis, where the true feature targets are no longer accessible during optimisation.

5.1. GOMEA and Experimental Setup

The EA used in this work is GOMEA, which is a state-of-the-art EA with proven effectiveness in real-world problems [32]. A detailed explanation is provided in Section 2.4. GOMEA is tasked with finding one image and one tabular feature target for every sample in the BST set. Each candidate set of feature

targets is encoded as a binary vector $\hat{\mathbf{t}}$, as shown in Equation (5.1). Each vector has a problem size l, defined in Equation (5.2), which depends on the BST ratio and thus determines the dimensionality of the optimisation problem.

$$\hat{\mathbf{t}} = (\hat{I}_0, \hat{T}_0, \ldots, \hat{I}_{n-1}, \hat{T}_{n-1}) \in \{0,1\}^l \tag{5.1}$$

$$\begin{aligned} l &= n_{BST} \cdot n_{modalities} \\ &= n_{training} \cdot r_{BST} \cdot n_{modalities} \\ &= (1000 \cdot 80\%) \cdot r_{BST} \cdot 2 \\ &= 1600 \cdot r_{BST} \end{aligned} \tag{5.2}$$

First, a population of solutions is initialised through probabilistically complete sampling. For each solution in the population, GOM is performed on the sets of variables as defined in the linkage model. Donor variables are accepted only if they do not worsen the fitness of the solution. In this chapter, the fitness of the solution is determined by calculating the feature-target correlation to the true feature targets using the method as described in the previous chapter.

This process is repeated, resulting in multiple generations of the population. GOMEA terminates when a candidate solution in its population has reached the ideal feature-target correlation of 1.0 is achieved or when all solutions in the population have converged to be identical. The latter condition prevents the algorithm from running indefinitely after convergence.

The parameters that should be set for GOMEA are the following: population size, linkage model, and mutation operator. First, population size and linkage model are determined, since mutation is not essential to GOMEA and can therefore be disabled. The effect of mutation on optimisation effectiveness is investigated at the end of this chapter.

Typically, the linkage model is set to be a LT, which is learned during optimisation to capture dependencies between variables. However, the size of the LT grows considerably with the relatively large problem size, which can reduce the number of generations for a fixed evaluation or time budget. Therefore, two additional linkage models are considered: the univariate model and marginal blocks model. Both are smaller than the LT and do not require learning variable dependencies.

For the marginal blocks model, the block size is set equal to the number of features per data instance. This configuration is referred to as the instance blocks linkage model, since each linkage group contains all feature targets corresponding to a single data instance. As each instance has one feature per modality and two modalities in total, the block size b is set to two. Compared to the univariate model, instance blocks are expected to better capture dependencies between variables belonging to the same instance. However, the LT is hypothesised to provide the most accurate dependency modelling by learning relationships across different instances.

A grid search is conducted over combinations of linkage model and population size. For the linkage model, the following models are considered: univariate, instance blocks, and the LT. The population sizes considered are 20, 36, 72, and 144. Each combination of linkage model and population size is used for optimising the problem as described above, and each configuration is repeated for 30 different training seeds for every BST ratio $r_{BST} \in \{0.25, 0.5, 0.75, 1.0\}$. For each run, the number of evaluations is recorded, together with whether an ideal solution was found or if the population converged prematurely.

5.2. Results

The average number of evaluations required to find a perfectly correlated solution per parameter combination, together with what percentage of runs it was able to find such an ideal solution for, is given in Table 5.1. The univariate linkage model is shown to on average need the least number of evaluations to reach the optimal solution for the given problem across all BST ratios and population sizes. The results also show that both the univariate and the LT are able to solve the problem consistently.

The same can not be said for the instance blocks linkage model. Even though this linkage model was assumed to better model the dependencies between variables, the majority of runs with a population size of 20 lose all diversity before finding an ideal solution, therefore converging prematurely. Additionally, the runs that do find an ideal solution need considerably more evaluations to do so than the other linkage models.

Table 5.1: Average number of thousands of evaluations required to find ideal feature targets using correlation-based optimisation via GOMEA per combination of population size and linkage model. The percentage in brackets denotes the ratio of runs in which an ideal solution was found. Several parameter combinations are repeated for 30 training seeds and BST ratios 100%, 75%, 50%, and 25%. The objective function maximises the feature-target correlation to the true feature targets (see Equation (4.8)). Some runs converged before reaching the target fitness and were therefore left out the results. Problem size l was determined through Equation (5.2).

(a) BST ratio of 100% ($l = 1600$)

$r_{BST} = 100\%$		Linkage Model		
		Univariate	*Instance Blocks*	*Linkage Tree*
Population Size	*20*	142 (100%)	294 (7%)	152 (100%)
	36	239 (100%)	391 (80%)	252 (100%)
	72	464 (100%)	702 (97%)	532 (100%)
	144	896 (100%)	1,369 (100%)	1,137 (100%)

(b) BST ratio of 75% ($l = 1200$)

$r_{BST} = 75\%$		Linkage Model		
		Univariate	*Instance Blocks*	*Linkage Tree*
Population Size	*20*	100 (100%)	150 (13%)	109 (100%)
	36	168 (100%)	264 (87%)	191 (100%)
	72	313 (100%)	507 (100%)	405 (100%)
	144	623 (100%)	958 (100%)	850 (100%)

(c) BST ratio of 50% ($l = 800$)

$r_{BST} = 50\%$		Linkage Model		
		Univariate	*Instance Blocks*	*Linkage Tree*
Population Size	*20*	61 (100%)	119 (13%)	68 (100%)
	36	103 (100%)	170 (93%)	119 (100%)
	72	196 (100%)	302 (100%)	250 (100%)
	144	382 (100%)	582 (100%)	552 (100%)

(d) BST ratio of 25% ($l = 400$)

$r_{BST} = 25\%$		Linkage Model		
		Univariate	*Instance Blocks*	*Linkage Tree*
Population Size	*20*	27 (100%)	43 (27%)	31 (100%)
	36	45 (100%)	66 (93%)	53 (100%)
	72	87 (100%)	127 (100%)	109 (100%)
	144	165 (100%)	242 (100%)	226 (100%)

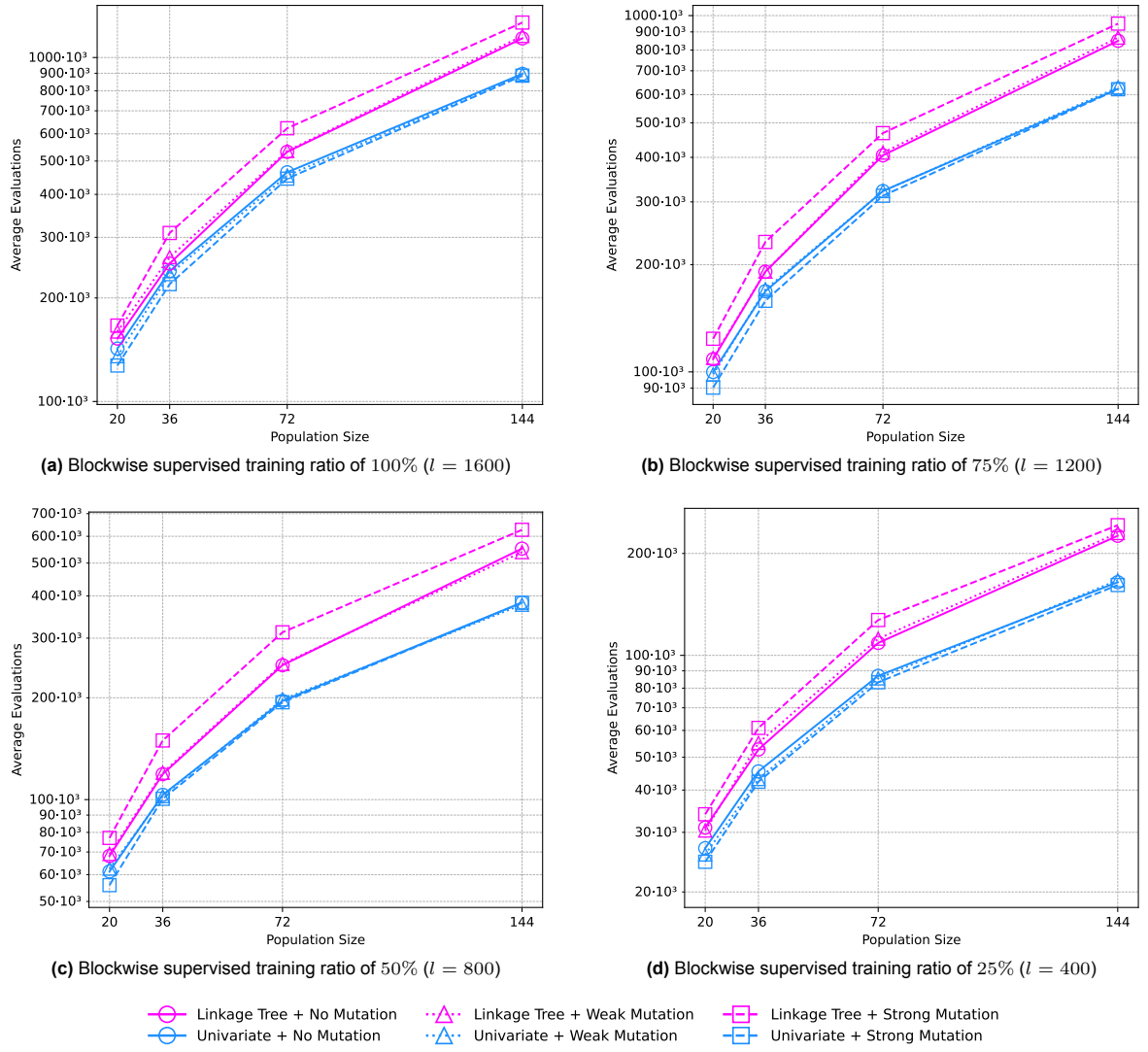
This may be explained by the existence of four distinct ideal solutions, which arises from the complement invariance property of the feature correlation measure, as it treats a feature and its complement as equally valid. Because the dataset consists of two modalities, each requiring an engineered feature, an ideal solution can correspond either to the true features or to any combination of their negated counterparts.

Each distinct ideal solution corresponds to an optimum in the search landscape. Consequently, when using the instance blocks linkage model, mixing between two solutions is likely to occur primarily when both are located near the same optimum. This also explains why a larger population size is required to reliably discover an optimal solution. In essence, the search process forms four subpopulations, each converging toward a different optimum. The univariate and LT models are less affected by this phenomenon because they allow more mixing between subpopulations.

In conclusion, the univariate linkage model combined with a population size of 20 is shown to be preferable when optimising the feature-target correlation to the ground-truth feature targets in the three-gated XOR task with two tabular features. This configuration consistently found an ideal solution with the least number of evaluations. However, this problem is idealised and the actual problem at hand is likely to be more challenging. Therefore, the aforementioned GOMEA parameters are seen as a lower bound. In practice, a larger population size and the more powerful LT might give better results.

5.3. Mutation Operators

As discussed in Section 2.4, mutation is an additional component in GOMEA. However, it can help GOMEA to optimise a given problem. To determine the effect of mutation on the correlation-based optimisation problem from this chapter, the previous experiment is repeated for both weak and strong mutation, as detailed in Section 2.4.2. Only the univariate and LT are considered, since the instance blocks model was shown to perform considerably worse. The average number of evaluations needed to reach an optimal solution per combination of linkage model, population size, and mutation operator are given in Figure 5.1, including the no mutation results from previous section. Unlike previous experiment, all runs were able to find an optimal solution. Therefore, this metric is omitted from illustration.



(a) Blockwise supervised training ratio of 100% ($l = 1600$)

(b) Blockwise supervised training ratio of 75% ($l = 1200$)

(c) Blockwise supervised training ratio of 50% ($l = 800$)

(d) Blockwise supervised training ratio of 25% ($l = 400$)

Figure 5.1: Average number of thousands of evaluations required to find ideal feature targets using correlation-based optimisation via GOMEA per combination of population size, linkage model, and mutation operator. The parameter combinations are repeated for 30 training seeds and BST ratios 100%, 75%, 50%, and 25%. The objective function maximises the feature-target correlation to the true feature targets (see Equation (4.8)). All runs found an optimal solution before prematurely converging to a non-ideal solution. Problem size l was determined through Equation (5.2).

The results indicate that the univariate linkage model combined with strong mutation performs best on average, although only marginally, across all tested populations sizes and BST ratios. It should be noted that the mutation probability for the combination of strong mutation and the univariate linkage model is 1, because the linkage sets are all of size 1. Therefore, for every round of GOM, it is checked whether the decision variable should be flipped or not, with the donor having no effect. This is a form of local search, where each solution disregards the rest of the population during GOM. Only when FI is triggered does mixing consider the current state of the population.

In contrast, the LT seems to not benefit from mutation, with strong mutation considerably increasing the amount of evaluations needed to reach an optimal solution. Weak mutation is shown to have little effect on the number of evaluations. Additionally, the LT on average needs more evaluations than the univariate linkage models across all tested population sizes and BST ratios. Similar results were obtained from the previous experiment.

6

Designing a Proxy Objective and Evaluating its Fitness Signal

The previous chapter showed that optimising for ideal feature targets through GOMEA is possible. It did this through an idealised version of the problem, where ideal feature knowledge was used to calculate the correlation of a candidate set of feature targets to the ground-truth feature targets. Consequently, this correlation was used as the optimisation objective. However, the true feature targets are typically unknown in real-world scenarios. Therefore, the research question guiding this chapter is stated as follows.

Research Question

RQ3. How can a proxy objective be designed to guide feature target optimisation in the absence of ground-truth features and does the proxy objective yield a fitness signal on tasks with extreme joint modality dependence?

Section 6.1 designs a proxy objective to assess feature target quality without using the ground-truth feature targets. Next chapter, this objective will be optimised through GOMEA. But first, Section 6.2 evaluates the fitness signal of the proxy objective to assess if optimising the proxy objective results in feature targets that correlate strongly with the true feature targets. Such feature targets should consequently improve MultiFIX performance when used in BST, as was discussed in Chapter 4. Finally, Section 6.3 makes a selection for what BST ratio should be used in the rest of the thesis.

6.1. Designing a Proxy Objective

The overarching goal is to design a proxy objective that, when optimised, yields a set of feature targets which result in better-than-baseline performance when used to train the MultiFIX model, as described in Chapter 4, in the absence of ground-truth feature targets. One intuitive strategy is to leverage the end-to-end training loss of the entire MultiFIX architecture after BST and end-to-end training as a measure of fitness. The idea is that the final training loss after training with a candidate set of feature targets gives an indirect measure of how well the feature targets align with the true feature targets. A relatively low loss would indicate that the feature targets used correlate relatively high with the true feature targets.

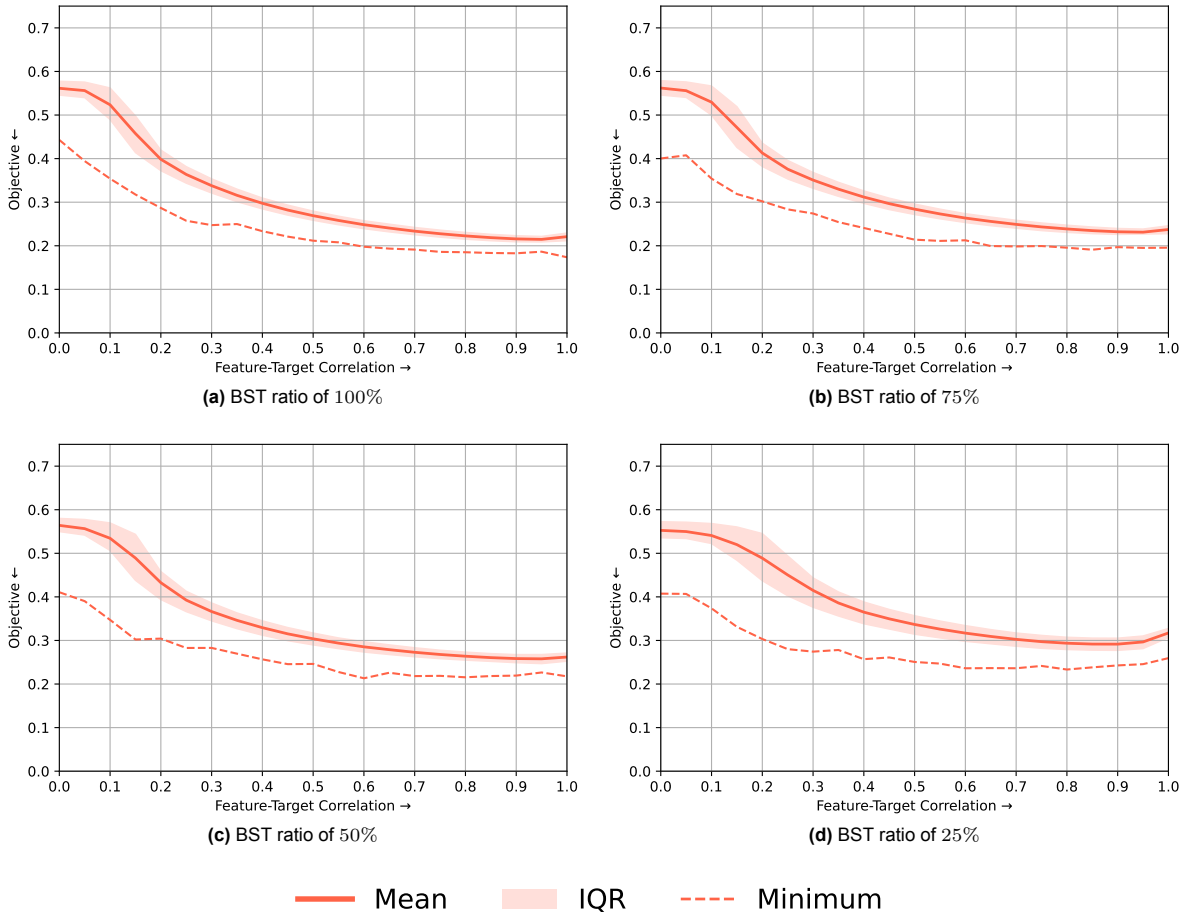
The train loss after end-to-end training is taken, since Chapter 4 showed that end-to-end training is generally beneficial when compared to only applying BST. This could be a direct effect of end-to-end training potentially correcting for noise encountered in feature targets. Additionally, the loss is determined for the entire train set, instead of just the BST set. This should give a better estimate of model performance and therefore give a better measure of feature target quality.

6.2. Evaluating the Fitness Signal

This section aims to evaluate the fitness signal of the proxy objective on the three-gated XOR task with two tabular features. Once again, the ground-truth feature label availability of the synthetic problem is used in assessing effectiveness. Specifically, the proxy objective is compared to the feature-target correlation to observe the relation between the two.

The same $20{,}000$ sets of noisy feature targets as in Section 4.3.1 are used to test for a range of noise levels. The correlation of each set of noisy feature targets is calculated by using the availability of ground-truth features. Consequently, each set of targets is binned according to its feature correlation, as described in more detail in Section 4.3.3.

The proxy objective for each set of noisy feature targets is obtained for 5 different random seeds. For each bin, the mean, IQR, and minimum objective value is calculated. The proxy objective of the ground-truth feature targets is also calculated, since the $20{,}000$ sets of noisy feature targets do not include the true feature targets. This process was repeated for $5{,}000$ different random seeds to balance the number of samples per bin, as was done in Section 4.3.3. The resulting objectives are binned and its mean, IQR, and minimum was obtained. This process was repeated for a BST ratio of 100%, 75%, 50%, and 25% to analyse the effect the BST ratio has on the fitness signal. The results are given in Figure 6.1.



(a) BST ratio of 100%

(b) BST ratio of 75%

(c) BST ratio of 50%

(d) BST ratio of 25%

Figure 6.1: Correlation of feature targets used in Blockwise Supervised Training (BST) to the true feature targets versus the objective for BST ratios 100%, 75%, 50%, and 25%. Noisy feature targets were created by injecting differing levels of noise to the true feature targets, which were binned according to their correlation to the true feature targets. Each bin consists of roughly $5{,}000$ samples. The means, interquartile ranges (IQRs), and minimum value per bin are shown by the solid lines, shaded areas, and dotted lines respectively. The arrows in the labels denote whether a metric should be minimised or maximised.

As hypothesised, the objective on average decreases when the feature-target correlation increases. This becomes more apparent as the BST ratio increases. Specifically, the average objective improves as the BST ratio increases for every level of correlation. However, the trend of improvement with rising feature-target correlation appears to break down at perfect correlation, that is, when using the true feature targets. This break of trend is most pronounced for a BST ratio 25%.

A similar phenomenon was observed in Section 4.3.3. Here, the BCE test loss was observed to be worse for ideal feature targets than for targets with a correlation of 0.95 for some configurations. In contrast, AUROC and BAcc were on average always the highest for perfect feature-target correlation, indicating a calibration issue. Since the designed proxy objective is also a form of BCE loss, the same explanation in Section 4.3.3 can be applied here.

The break in the trend occurs because the small amount of noise in the feature targets (with a correlation of 0.95) acts as a form of regularisation. This noise introduces a slight bias which, together with the high variance caused by the low BST ratio, leads to an improvement in the BCE loss. As the BST ratio increases and the BST set grows larger, the variance decreases, which in turn reduces the beneficial effect of the noise. Therefore, the effect is most apparent for a BST ratio of 25%.

Next to the the average trend depicted by the mean and IQR, the dashed lines represent the minimum values. These show a less smooth trend, as expected from the nature of the minimum operation, which does not average over a set of samples. Nevertheless, higher feature-target correlations still appear to correspond to lower minimum objective values. However, the differences in objective values between high correlation levels are relatively small, resulting in a plateau. Such plateaus may hinder optimisation beyond these regions.

Additionally, as discussed before, perfect correlation does not seem to consistently correspond with the global minimum. Because of this, it is not seen as the utopian point. Optimisation can get stuck in these local optima. For example, a BST ratio of 50% (Figure 6.1c) seems to ultimately prefer feature targets with correlation of 0.6 above targets with higher correlations. Nevertheless, it is possible for GOMEA to escape such optima, since it operates on a population of solutions.

To obtain a more stable fitness signal, it is possible to instead set the objective to be the average final end-to-end train loss over a range of random seeds. This should decrease the variance in objective values at a given feature-target correlation. However, this would multiply the fitness evaluation time with the size of the random seed range. Due to time constraints, this alternative fitness evaluation method was not used.

It should be noted that optimisation will likely explore different areas of the search space than those used in estimating the fitness signal. This is due to the algorithm exerting selective pressure toward feature targets with lower objective values, unlike the stochastic noise injection used to generate the $20{,}000$ noisy feature target sets for fitness signal estimation. Consequently, the optimisation is more likely to follow the trend indicated by the minimum objective values rather than the average trend.

In conclusion, the designed proxy objective, defined as the final training loss after end-to-end training, appears to provide a fitness signal suitable for optimising feature targets that correlate strongly to the true feature targets in the three-gated XOR task with two tabular features. This signal becomes more apparent as the BST ratio increases. However, at high feature-target correlations, optimisation may be hindered by objective plateaus and by the fact that the ideal solution does not necessarily correspond to the global minimum of the objective. A more stable fitness signal might potentially be obtained by averaging the final training loss over several random seeds, though this was not opted for due to time constraints. Finally, it should be noted that the fitness signal is merely an estimate, and optimisation will likely follow a different trend due to its selection pressure to lower objective values.

6.3. Blockwise Supervised Training Ratio Selection

All experiments until this point have tested the BST ratios of 100%, 75%, 50%, and 25% to evaluate their effect on various aspects of the three-gated XOR task with two tabular features. However, due to time constraints, it is not possible to optimise the proxy objective with GOMEA for all four BST ratios. Therefore, a single BST ratio is selected based on previous results.

As discussed in Section 4.1.2, the selection of the BST ratio involves competing considerations. In Section 4.3.3, higher ratios were shown to generally yield better-performing MultiFIX models across all levels of feature target noise, both with and without end-to-end training. Furthermore, the previous section indicated that higher ratios seem to provide a stronger fitness signal.

In contrast, lower BST ratios are computationally more efficient. Firstly, Section 4.1.6 showed that a lower ratio needs significantly less time to perform BST, regardless of whether end-to-end training is applied. Secondly, Chapter 5 demonstrated that GOMEA needs considerably less evaluations to find an ideal solution for idealised correlation-based optimisation, largely due to the bigger problem size that a larger BST ratio results in.

Taking all factors into account, a BST ratio of 50% was selected as the best trade-off for the three-gated XOR task with two tabular features. It allows for better-than-baseline MultiFIX models and a reliable fitness signal, while keeping computational costs lower than those of ratios of 75% and 100%. Its advantage over a BST ratio of 25% comes from the better performing model when trained with ground-truth feature targets. Additionally, the regularising effect of low levels of noise in feature targets during BST was shown to be minimal for a BST ratio of 50% compared to a ratio of 25%. This provides both a more reliable fitness signal and better performing model when using the true feature targets.

7

Optimising the Proxy Objective through GOMEA

To briefly recap, Chapter 4 demonstrated that incorporating feature knowledge in the form of feature targets can be used in BST and end-to-end training to obtain better-than-baseline MultiFIX performance on the three-gated XOR task with two tabular features. It also showed that imperfect feature targets can be sufficient, given a certain level of feature-target correlation to the typically unknown ground-truth features. Subsequently, Chapter 5 showed how to optimise for such feature targets through GOMEA by assuming the true feature targets were available. However, since this assumption does not hold in real-world scenarios, Section 6.1 introduced a proxy objective to enable optimisation in the absence of ideal feature knowledge. This chapter extends that work by testing whether optimising this objective through GOMEA yields useful feature targets, as formulated in the following research question.

Research Question

RQ4. Does optimising the designed proxy objective with GOMEA yield feature targets that improve the performance of MultiFIX on tasks with extreme joint modality dependence, and why or why not?

First, Section 7.1 elaborates on the experimental setup of this chapter. This setup is applied to the three-gated XOR task with two tabular features, in addition to the single XOR and AND tasks to further assess the effectiveness of the proposed method. The analysis methods are detailed in Section 7.2, while the discussion of the corresponding results is presented in Section 7.3. The findings and insights presented here provide the basis for the next chapter, which systematically evaluates the contribution of several individual components through an ablation study, directly addressing the "why or why not" aspect of the research question.

7.1. Experimental Setup

GOMEA is tasked with optimising feature targets $\hat{\mathbf{t}}$ that approximate the true feature targets $\mathbf{t}^{true}$, as discussed in more detail in Section 5.1. Since the BST ratio was determined to be 50% in Section 6.3, candidate solutions consist of $l = 1600 \cdot r_{BST} = 1600 \cdot 50\% = 800$ binary problem variables (see Equation (5.2)). The objective to be minimised is the final loss of end-to-end training after BST with the candidate feature targets, as detailed in Section 6.1. Optimisation is terminated once a total time budget of two weeks has passed.

The DL parameters for BST and end-to-end training used are those given in Table 4.1. Additionally, the GOMEA parameters that need to be set are the following: population size, linkage model, and mutation operator. Section 5.2 identified the univariate linkage model combined with a population size of 20 to need the least evaluations for finding an optimal solution in the idealised correlation-based optimisation problem. However, it also noted that these parameters should be seen as a lower bound, since the

real-world scenario is more challenging. Therefore, a population size of 36 is opted for in this chapter, together with the univariate linkage model. This should still allow for a decent number of generations in the given time budget. Finally, strong mutation is included, since Section 5.3 demonstrated that the combination of this mutation operator and a univariate linkage model benefitted optimisation of the idealised problem.

Moreover, three random seeds are specified: the data split seed, the downsample seed, and the train seed. The data split seed, used in splitting the data into a train and test set, is kept constant to enable consistent comparison of test results. The downsample seed is used in determining what samples from the train set are used in BST and is varied to evaluate the impact of different BST sets on performance. The train seed refers to the random seed used in all stochastic process in GOMEA, including the DL training in fitness evaluation. This seed is also varied to get a more accurate performance estimate. Specifically, four different combinations of downsample and train seeds are considered. This chapter focuses on the setting where both the downsample and train seed are set to 1, as this combination was found the best represent all runs. The results of all other seed combinations are given in Appendix C.

7.2. Analysis Methods

Each run is logged by recording the feature targets of all candidate solutions along with their corresponding objective values across generations. The population of initial solutions, referred to as generation 0, is also included. For every set of feature targets, BST is performed both with and without end-to-end training, after which the resulting models are evaluated on the held-out test set. The test results comprise the BCE loss, AUROC, and BAcc, both before and after additional end-to-end training. These results can, together with the objective values, be used to select a solution for the final model for real-world scenarios. However, in this chapter, the availability of ground-truth feature targets in the synthetic tasks is leveraged to further assess candidate solution quality. Importantly, these targets are not used during optimisation, but solely for evaluating the method's effectiveness.

First, the feature-target correlation of every candidate solution is computed and plotted against its corresponding objective value, resulting in a scatter plot where each marker represents a solution. This plot illustrates how the objective value relates to the feature-target correlation for the solutions obtained by GOMEA. The population's progression across generations is shown by colouring the solutions according to their generation, with the initial population (generation 0) highlighted using black outlines to distinguish it clearly.

Secondly, the engineered-feature correlation of each solution is determined. As discussed in Section 4.3.2, the method for computing feature correlation is not limited to binary targets. For each solution, the engineered-feature correlation is obtained by using its feature targets in both BST and end-to-end training, and then calculating the correlation between the resulting engineered features of the entire training set and the ground-truth feature targets. It is expected that these engineered features correlate more strongly with the true feature targets than the feature targets used during BST. To illustrate this, the objective value of each solution is again plotted, in a similar manner to the previous section, but now against the engineered-feature correlation instead of the feature-target correlation.

Thirdly, the test metrics are plotted against the engineered-feature correlation, using only the results obtained after BST without end-to-end training. As mentioned before, each solution has two sets of test metrics, one obtained after BST and one after additional end-to-end training. For this analysis, only the metrics from BST without end-to-end training are considered. The engineered features obtained after BST are used to compute the engineered-feature correlation. Each solution is represented by three markers, one for each test metric plotted against its corresponding engineered-feature correlation. Instead of colouring by generation, the markers are coloured according to metric type. Metrics corresponding to generation 0 once again uses black outlines, to highlight results obtained from essentially random search. Additionally, the baseline metrics from Section 3.3 are indicated by dashed horizontal lines to illustrate whether the obtained test metrics surpass the baseline performance.

Finally, the same approach is applied to the results obtained with both BST and end-to-end training. The engineered-feature correlation is now computed using the features resulting from end-to-end training. This plot illustrates how the solutions found by GOMEA correspond to the resulting MultiFIX performance on the held-out test set and how this performance compares with the baseline. Together with

the previous plots, it also shows whether selecting solutions based on their resulting test performances and objective values correspond with selecting solutions that accurately approximate ideal feature targets, albeit indirectly through its engineered features.

7.3. Results

The previously discussed experimental setup is first applied to the three-gated XOR task with two tabular features. Its results are discussed in Section 7.3.1. Next, Section 7.3.2 evaluates the proposed method on the single XOR problem to assess its effectiveness on simpler tasks that still exhibit extreme joint-modality dependence. Finally, Section 7.3.3 subjects the AND problem to the same setup to investigate its performance on tasks without extreme joint-modality dependence.

7.3.1. Three-Gated XOR With Two Tabular Features

The results obtained using the analysis methods described in Section 7.2 are summarised in Figure 7.1 for the three-gated XOR task with two tabular features. Each subfigure highlights a different aspect of the solutions obtained from optimisation and is examined in detail in the following subsections.

Relation Between Feature-Target Correlation and Objective Value

Figure 7.1a illustrates that the average feature-target correlation of the population tends to increase as the generations pass, with the final generation clearly improving upon the initial generation. Generation-to-generation improvement seems to decrease as the generations pass, with the first generation showing the biggest improvement.

GOMEA appears to explore different regions of the search space compared to those explored during the evaluation of the fitness signal in Section 6.2. This was already hypothesised due to the selection pressure the algorithm exerts on the objective value. Because of this, it finds feature targets which obtain a lower objective value than is expected from the estimated fitness signal in Figure 6.1c.

Moreover, GOMEA identifies non-ideal solutions whose objective values are lower than those expected from the ground-truth feature targets. Because of this, optimisation is not expected to find the ideal solution when given more budget. This can be combatted by averaging the objective value over several random seeds, as was proposed in the previous chapter, to limit overfitting to the train seed.

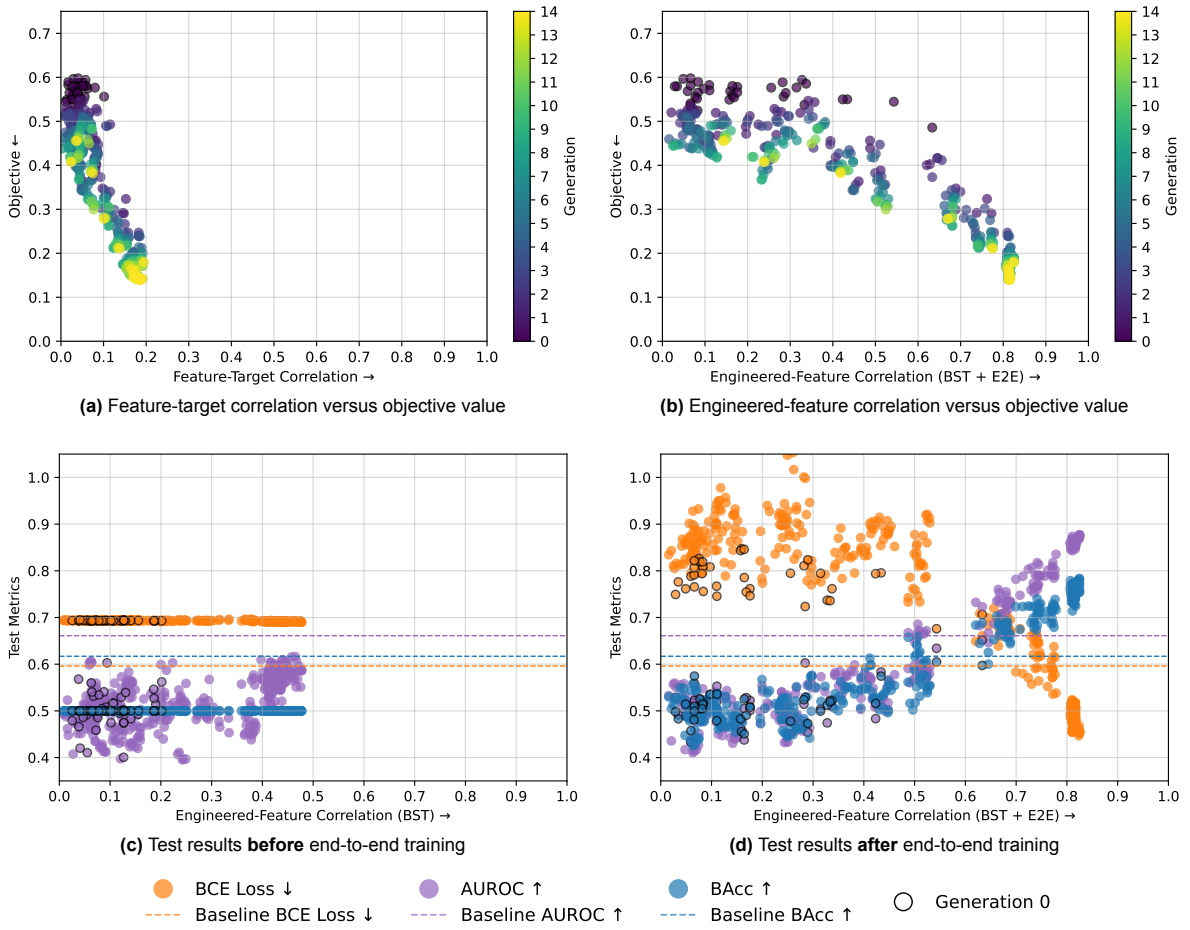
Relation Between Engineered-Feature Correlation and Objective Value

Together with the previous figure, Figure 7.1b illustrates how a solution its feature targets and its resulting engineered features relate in terms of correlation. These results indicate that the engineered features obtained after BST and end-to-end training correlate more strongly with the ground-truth feature targets than the feature targets used in BST. A solution's engineered-feature correlation therefore appears to be a better indicator of the resulting model's alignment with the underlying problem structure. Combining this correlation with the test performance of the MultiFIX models provides a more complete picture of the quality of the discovered solutions.

Before examining this relationship further, attention is first given to the initial population. Although the average solution in generation 0 shows relatively low feature-target and engineered-feature correlations, a few solutions achieve notably higher correlation scores. Optimisation clearly enhances these outliers across all metrics considered so far. Whether this trend extends to the remaining metrics is examined in the following sections.

Relation Between Engineered-Feature Correlation and Test Results Before End-to-End Training

The test results for each solution before end-to-end training are shown in Figure 7.1c. Each solution is represented by a marker for each test metric, plotted against the corresponding engineered-feature correlation without end-to-end training. This figure demonstrates that the baseline metrics are not improved upon and that the engineered-feature correlation is higher than the correlation of the feature targets used in BST, even without end-to-end training. However, the engineered-feature correlation is still lower than what is achieved after end-to-end training, indicating that end-to-end training provides additional benefits over using BST alone.



(a) Feature-target correlation versus objective value

(b) Engineered-feature correlation versus objective value

(c) Test results **before** end-to-end training

(d) Test results **after** end-to-end training

Figure 7.1: Subfigures resulting from optimising the proxy objective through GOMEA for the **three-gated XOR** task with two tabular features. GOMEA was run for a runtime of two weeks, with the downsample seed set to 1, and the train seed set to 1. Arrows denote whether the corresponding metric should be maximised or minimised. Markers corresponding the initial population of GOMEA are highlighted using black outlines. **(a)** Each dot represents a solution found by GOMEA, with the feature-target correlation plotted against the objective value. Dot colour indicates the generation of the solution. **(b)** Instead of feature-target correlation, the engineered-feature correlation after end-to-end training is plotted against the objective value. **(c)** Each dot represents a test metric plotted against the engineered-feature correlation of its corresponding solution, both *without* applying end-to-end training. The test metrics include: the Binary Cross Entropy (BCE) Loss, Area Under the Receiver-Operating Characteristic curve (AUROC), and Balanced Accuracy (BAcc). The dotted horizontal lines show the baseline value of the test metric in the same colour. **(d)** Instead of engineered-feature correlation and test metrics before end-to-end training, the results *after* end-to-end training are shown.

Analysis of individual metrics shows that both the BCE loss and BAcc remain at the values observed for random guessing. The AUROC, by contrast, shows considerable variability, with the average AUROC tending to increase with engineered-feature correlation. Nevertheless, higher correlations between feature targets and ground-truth features are required to surpass baseline performance using BST alone, as was also shown in Figure 4.3e.

Relation Between Engineered-Feature Correlation and Test Results After End-to-End Training

Finally, Figure 7.1d shows the engineered-feature correlation against the test results of each solution, both after end-to-end training. The figure clearly demonstrates that several solutions achieve better-than-baseline test metrics. Moreover, the top-performing solutions yield results that closely approximate the performance when using ground-truth feature targets, as reported in Table 4.4b.

Appendix C presents results for three additional seed combinations, which display similar characteristics. These results indicate that the outcomes in Figure 7.1 are unlikely to be a lucky run, as all runs produce solutions that improve upon baseline performance when end-to-end training is applied. The extent of improvement varies between runs, with the training seed appearing to have the most noticeable effect. The downsample seed seems to have little effect, likely because the feature distribution

is roughly consistent across different downsample seeds. It should be noted, however, that only four seed combinations were tested, so the results may be biased.

Examining the initial population reveals that a few solutions already approached baseline performance. As noted previously, generation-to-generation improvement decreases as evolution progresses. This trend is particularly evident among the top-performing solutions, which tend to form a cluster. This pattern likely arises from the population converging toward an elitist solution. Consequently, the population explores that region of the search space extensively, achieving only minor improvements and potentially overfitting to the training seed. Only a few generations of mixing appear sufficient to surpass baseline performance on the three-gated XOR task with tabular features.

Conclusion

In conclusion, GOMEA was able to identify sets of feature targets that, despite not closely approximating the ground-truth feature targets, improve upon baseline performance when used in BST and end-to-end training. Through BST, these feature targets appear to bias the model toward engineering features that correlate strongly with the true features, with end-to-end training further enhancing this effect. BST alone has proven insufficient, yielding test results only marginally better than random guessing. Moreover, optimisation was able to discover solutions with better-than-baseline performance after only a few generations, while subsequent generations produced diminishing improvements.

7.3.2. XOR

The results obtained using the analysis methods described in Section 7.2 are summarised in Figure 7.2 for the single XOR. Each subfigure highlights a different aspect of the solutions obtained from optimisation and is examined in detail in the following subsections. Additionally, results of the three-gated XOR task with two tabular features are used for comparisons.

Relation Between Feature-Target Correlation and Objective Value

Figure 7.2a shows that as the objective decreases the feature label correlation increases, which means GOMEA finds a similar fitness signal for the XOR problem as for the three-gated XOR problem. Despite the single XOR task being less complex, GOMEA elite solutions only display marginally better feature-target correlation. Moreover, two cluster seem to form on the objective value axis, where previously a more spread out distribution was displayed.

Relation Between Engineered-Feature Correlation and Objective Value

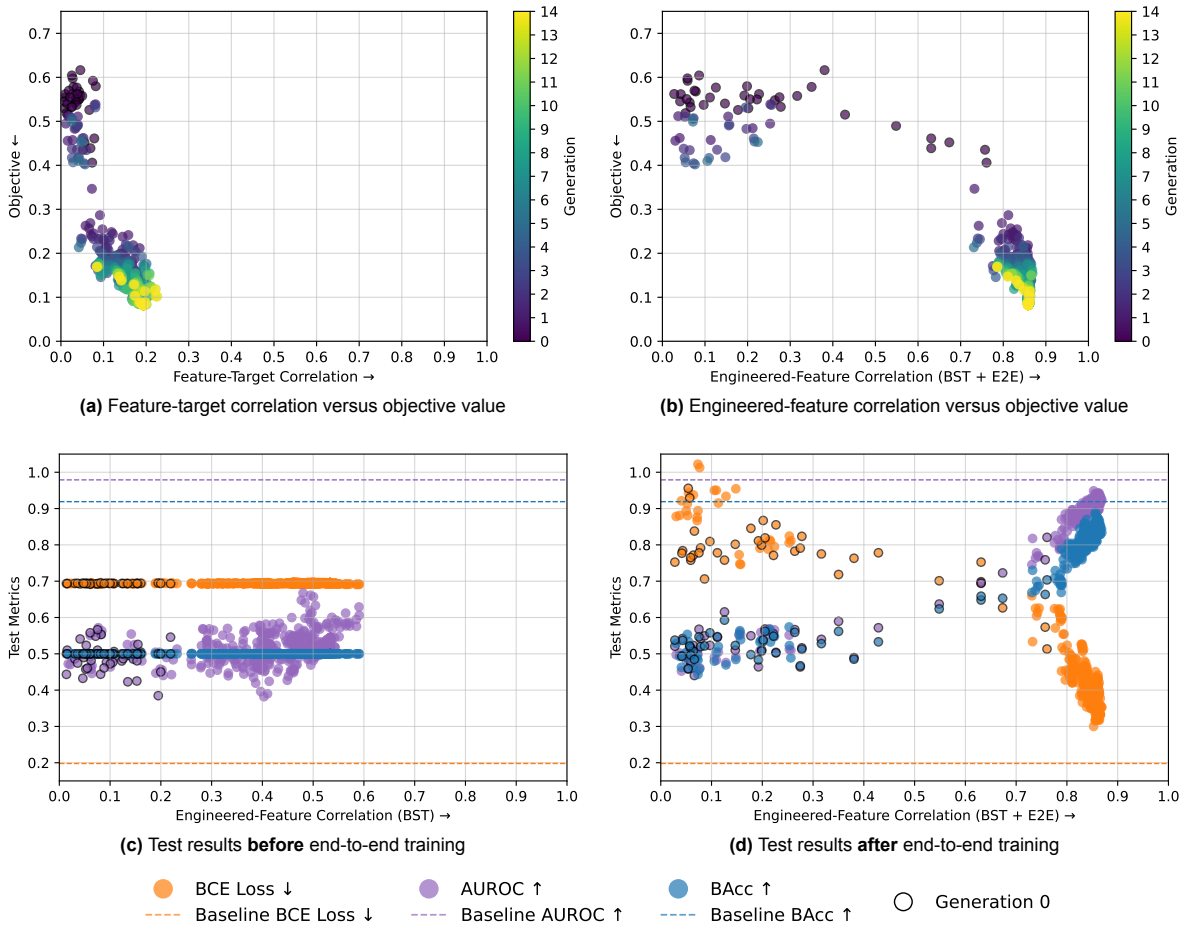
Figure 7.2b illustrates the distinction between clusters even more clearly, with the lower-objective cluster achieving much better engineered-feature correlations than the other cluster. This is likely due to the highly noisy feature-target sets destructively biasing the MultiFIX model through BST, consequently hindering the learning process in end-to-end training. Solutions with slightly better feature-target correlations or objective values do not exhibit this effect and therefore end up in the more performant cluster.

Several solutions of the initial population exhibit rather decent engineered-feature correlations. The first generation of mixing improves upon this the most, with consequent generations showcasing diminishing returns. This coincides with the observations for the previously discussed three-gated XOR task. Whether this extends to the test metrics is discussed in the following paragraphs.

Relation Between Engineered-Feature Correlation and Test Results Before End-to-End Training

The test results for each solution before end-to-end training are shown in Figure 7.2c. This figure shows a similar story as for the three-gated XOR, where only AUROC seems to improve with increasing engineered-feature correlation, while BCE loss and BAcc are only marginally better than the expected values from random guessing. Furthermore, none of the solutions approach the relatively strong baseline.

The distribution of solutions is also more evenly spread out, instead of the clear distinction of clusters seen in previous figures. This is due to the small variability on the test metrics axis and the engineered-feature correlation before end-to-end training also appearing to be relatively evenly distributed.



(a) Feature-target correlation versus objective value

(b) Engineered-feature correlation versus objective value

(c) Test results **before** end-to-end training

(d) Test results **after** end-to-end training

Figure 7.2: Subfigures resulting from optimising the proxy objective through GOMEA for the **single XOR** task. GOMEA was run for a runtime of two weeks, with the downsample seed set to 1, and the train seed set to 1. Arrows denote whether the corresponding metric should be maximised or minimised. Markers corresponding the initial population of GOMEA are highlighted using black outlines. **(a)** Each dot represents a solution found by GOMEA, with the feature-target correlation plotted against the objective value. Dot colour indicates the generation of the solution. **(b)** Instead of feature-target correlation, the engineered-feature correlation after end-to-end training is plotted against the objective value. **(c)** Each dot represents a test metric plotted against the engineered-feature correlation of its corresponding solution, both *without* applying end-to-end training. The test metrics include: the Binary Cross Entropy (BCE) Loss, Area Under the Receiver-Operating Characteristic curve (AUROC), and Balanced Accuracy (BAcc). The dotted horizontal lines show the baseline value of the test metric in the same colour. **(d)** Instead of engineered-feature correlation and test metrics before end-to-end training, the results *after* end-to-end training are shown.

Relation Between Engineered-Feature Correlation and Test Results After End-to-End Training

Finally, Figure 7.2d shows that GOMEA finds solutions that approach baseline performance without improving upon them. However, testing with the ground-truth features also obtained results that did not improve upon the baseline. Its performance was very similar to the results achieved by the top-performant solutions. Therefore, better-than-baseline performance was not expected.

Upper-limit performance showcased by the true feature targets can be improved by tailoring the DL parameters for the single XOR problem, since they are now fitted towards the three-gated XOR task with two tabular features. Likely, simply performing more epochs of end-to-end training will result in better MultiFIX performance. However, in real-world scenarios GOMEA will likely only be utilised when the baseline is found to be insufficient, which is unlikely to be the case for the single XOR task.

Conclusion

In conclusion, GOMEA was able to identify sets of feature targets that, despite not closely approximating the ground-truth feature targets, achieve similar performance as to using the ideal targets in BST and end-to-end training. Baseline performance was only approximated and not improved upon, which can

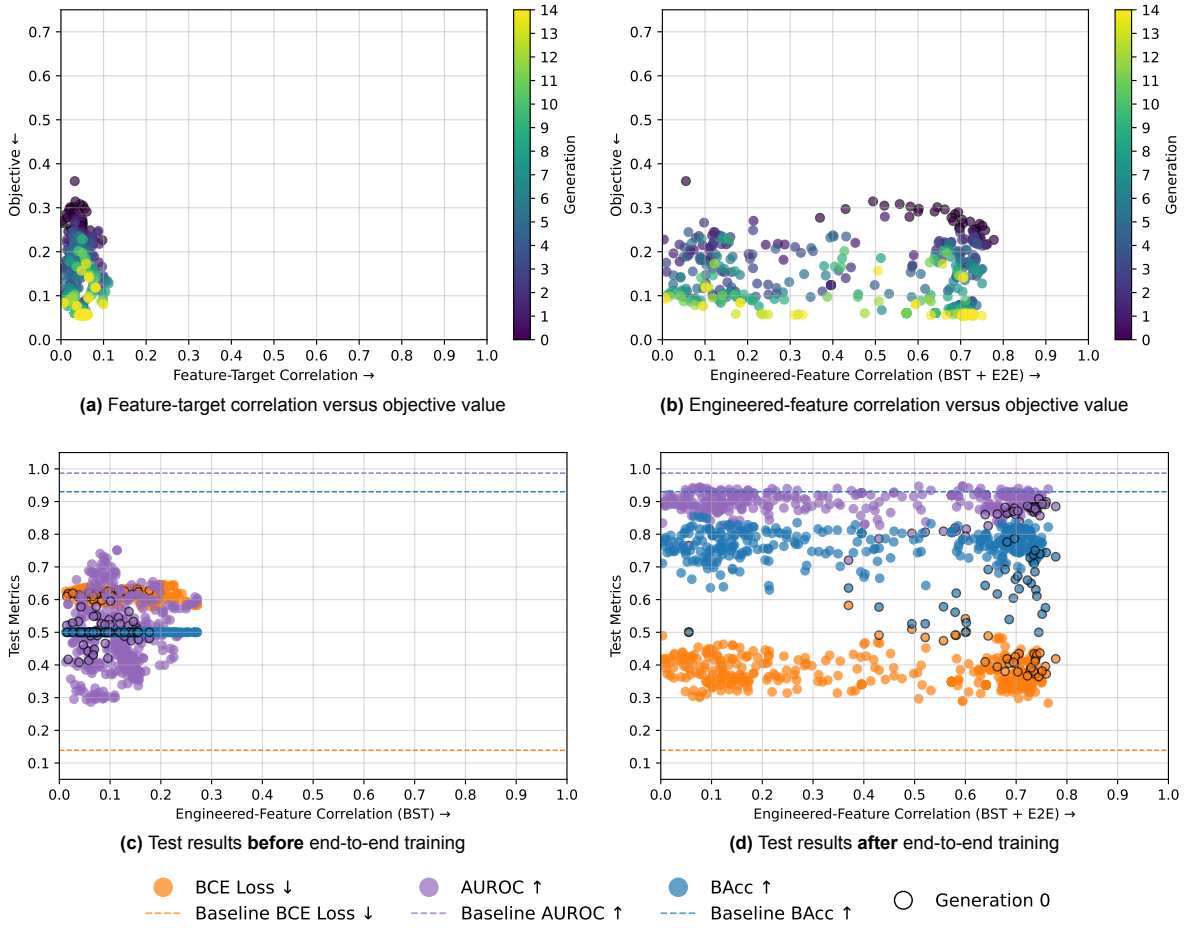
attributed to the DL parameters being tailored to the three-gated XOR task, instead of the single XOR. This can likely be tackled by simply performing more epochs of end-to-end training.

7.3.3. AND

The results obtained using the analysis methods described in Section 7.2 are summarised in Figure 7.3 for the AND problem, which does not exhibit extreme joint-modality dependence. Each subfigure highlights a different aspect of the solutions obtained from optimisation and is examined in detail in the following subsections. Additionally, results of the three-gated XOR task with two tabular features and single XOR are used for comparisons.

Relation Between Feature-Target Correlation and Objective Value

Figure 7.3a shows a different relation between the feature-target correlation and objective values of the observed solutions from the previously discussed problems. Specifically, it demonstrates that feature-target correlation remains relatively constant as the objective decreases. The feature-target correlation does not exceed the correlations observed in previous problems, despite the AND problem being of a lower complexity.



(a) Feature-target correlation versus objective value

(b) Engineered-feature correlation versus objective value

(c) Test results **before** end-to-end training

(d) Test results **after** end-to-end training

Figure 7.3: Subfigures resulting from optimising the proxy objective through GOMEA for the **AND** task. GOMEA was run for a runtime of two weeks, with the downsample seed set to 1, and the train seed set to 1. Arrows denote whether the corresponding metric should be maximised or minimised. Markers corresponding the initial population of GOMEA are highlighted using black outlines. **(a)** Each dot represents a solution found by GOMEA, with the feature-target correlation plotted against the objective value. Dot colour indicates the generation of the solution. **(b)** Instead of feature-target correlation, the engineered-feature correlation after end-to-end training is plotted against the objective value. **(c)** Each dot represents a test metric plotted against the engineered-feature correlation of its corresponding solution, both *without* applying end-to-end training. The test metrics include: the Binary Cross Entropy (BCE) Loss, Area Under the Receiver-Operating Characteristic curve (AUROC), and Balanced Accuracy (BAcc). The dotted horizontal lines show the baseline value of the test metric in the same colour. **(d)** Instead of engineered-feature correlation and test metrics before end-to-end training, the results *after* end-to-end training are shown.

Relation Between Engineered-Feature Correlation and Objective Value

Figure 7.3b shows also no relation between the objective values and the engineered-feature correlation after end-to-end training. The average objective value across a generation seems to improve, indicating that GOMEA is finding improvement. However, this does not translate in improved engineered-feature correlation, as shown by the best correlation being achieved by a solution from the initial population. On average, the engineered-feature correlation decreases over generations, suggesting that BST increasingly hinders feature learning as the objective value of the feature targets improves.

Relation Between Engineered-Feature Correlation and Test Results Before End-to-End Training

The test results for each solution before end-to-end training are shown in Figure 7.3c. The BAcc varies only marginally from the value expected from random guessing, similar to the previously discussed problems. However, the BCE test loss varies slightly more and averages to a value that is better than is expected from random guessing. Additionally, AUROC sees the most variance from all considered problems, with the average AUROC not increasing with the engineered-feature correlation. Finally, all metrics do not come close to the values obtained from the baseline.

In contrast to the previous figure, the engineered-feature correlation before end-to-end training seems to be clearly improved upon in Figure 7.3c. This indicates that an improved objective value more closely corresponds to the engineered-feature correlation when only BST applied. However, this correlation does not reach nearly the same level as observed for the more complex XOR problems.

Relation Between Engineered-Feature Correlation and Test Results After End-to-End Training

Finally, Figure 7.3d illustrates the test results after end-to-end training, compared against the correlations of the corresponding engineered features. It clearly shows the positive effect that end-to-end training has on the test metrics. The baseline values for the test metrics are not reached by the discovered solutions. However, the test values corresponding to using the true feature targets are the following: BCE loss of 0.257, AUROC of 0.957, and BAcc of 0.845. These values also do not surpass the baseline, and GOMEA is therefore not necessarily expected to produce solutions that achieve better MultiFIX performance. However, unlike in the XOR problems, the sets of feature targets discovered here achieve slightly lower performance than when using the ground-truth features.

Interestingly, test values seem to vary minimally across the observed range of engineered-feature correlations. This suggests that the quality of the feature targets has little to no influence on the resulting test performance. Even when the engineered features are uncorrelated with the true feature targets, the fusion network appears capable of combining them to make reasonably accurate predictions.

Conclusion

In conclusion, for the AND task, the objective value of a solution seems to not relate to its feature-target and engineered-feature correlation. Moreover, the quality of the feature targets and its resulting engineered features has little to no effect on the test results it obtains for the given task. Therefore, feature-target optimisation does not necessarily result in better-than-baseline performance.

Since the baseline approach already performs well on the AND problem, optimising feature targets through GOMEA provides limited additional benefit. Most of the performance improvement arises from end-to-end training itself. Consequently, comparable or even superior results may be achievable simply by extending the duration of end-to-end training, potentially making the proposed method a more general and effective strategy for tasks.

In general, optimising feature targets through GOMEA for use in BST and end-to-end training appears most beneficial when the dependency between modalities is strong, as demonstrated by the contrast between the AND and XOR problems. Furthermore, more complex feature interactions seem to gain more from this process, as shown by the single XOR and three-gated XOR tasks.

8

Ablation Study

The previous chapter demonstrated that optimising the proxy objective through GOMEA yields better-than-baseline performance on the three-gated XOR task with two tabular features. Building on these results, this chapter performs an ablation study to evaluate the impact of several individual components of the method. This analysis helps clarify which aspects contribute most to the observed improvements, thereby directly addressing the "why or why not" aspect of the following research question.

Research Question

RQ5. Does optimising the designed proxy objective with GOMEA yield feature targets that improves the performance of MultiFIX on tasks with extreme joint modality dependence, and **why or why not**?

First, the baseline and experimental setup for the ablation study are specified in Section 8.1. Secondly, Section 8.2 discusses the results after removing the additional end-to-end training in fitness evaluation. Thirdly, Section 8.3 replaces the tailored DL parameters with standard DL parameters, consequently considering its results. Finally, both components are removed in Section 8.4, including a discussion on how its performance compares to the baseline.

8.1. Baseline and Experimental Setup

First, it is important to state what is seen as the baseline of this ablation study. It should be mentioned that this baseline is the one obtained from optimising the proxy objective through GOMEA, as was done in Chapter 7, instead of the MultiFIX baseline stated in Chapter 3. Table 8.1 specifies what baseline values the no ablations variant obtained for easy comparison. The results as given in Figure 7.1 will be used for comparison in this chapter.

Each section specifies what it changes. After applying this change, the same optimisation process as in the previous chapter is started. The same seed combinations as for Chapter 7 are optimised for, with this chapter highlighting the same seed combinations, i.e., downsample and train seed both set to 1. The results of all seeds are given in Appendix C.

Table 8.1: Baseline results for the ablation study. As obtained from Figure 7.1c and Figure 7.1d, which illustrates the optimisation results of the three-gated XOR task with two tabular features and downsample and train seed set to 1. Values are estimates of the best value obtained per metric across all solutions found by GOMEA, both before and after end-to-end training (E2E). The arrows denote whether the metric should be minimised or maximised.

(a) Baseline test values

	BCE Loss ↓	AUROC ↑	BAcc ↑
BST	0.69	0.61	0.50
BST + E2E	0.45	0.87	0.78

(b) Baseline correlation values

Feature Targets ↑	Engineered Features ↑	
	BST	*BST + E2E*
0.19	0.47	0.82

8.2. Without End-to-End Training in Proxy Objective

Chapter 6 demonstrated that the final end-to-end training loss after BST and end-to-end training can serve as a proxy for determining how well the feature targets correlate with the ground-truth feature targets. Consequently, Chapter 7 showcased that optimising this proxy objective yields feature targets that result in better-than-baseline MultiFIX performance when used in BST and end-to-end training.

However, the step of end-to-end training may not be strictly necessary during the evaluation of the proxy objective. It may primarily serve to amplify the fitness signal already present when applying BST alone. If this is the case, omitting end-to-end training during fitness evaluation could save considerable computational time, as suggested by the results in Table 4.2.

To test this hypothesis, end-to-end training is omitted from fitness evaluation. The objective of a candidate set of feature targets is then defined as the BCE loss of the full DL architecture on the training set after applying the targets via BST, without end-to-end training. Evaluating the loss over the entire training set captures both how well the resulting model fits to the BST set and how well it generalises to data samples that were not used during BST. This should prevent overfitting to the BST set.

Apart from this modification, the same experimental setup described in Section 7.1 and the corresponding analysis methods detailed in Section 7.2 are used. Although end-to-end training is omitted during fitness evaluation, it is still applied in the analysis, as it has consistently proven beneficial, and final model selection would likely always involve end-to-end training. Baseline correlations are indicated with vertical lines, alongside the baseline test metrics, providing a clearer reference for comparing the solutions found in this section to those of the baseline. The results for the configuration with both the downsample and train seeds set to 1 are presented in Figure 8.1, while results for all four tested seed combinations are provided in Appendix C.2.1.

8.2.1. Discussion

Figure 8.1a illustrates that the obtained solutions substantially outperform the baseline in terms of feature–target correlation. The corresponding objective values are also considerably higher. This outcome is expected, as the objective function reflects the training loss for fewer epochs of training, by the omission of end-to-end training. Interestingly, this appears to enhance the quality of the discovered feature targets, potentially because the optimisation objective is more closely aligned with the underlying feature–target quality.

Furthermore, the optimisation process seems more capable of identifying higher-quality solutions when provided with a larger computational budget than the baseline approach. This can be attributed to the baseline achieving relatively low objective values for solutions exhibiting weak feature–target correlations. By comparison, Figure 8.1a suggests that there remains considerably more scope for improvement in terms of the objective, which may in turn lead to better overall solutions.

Figure 8.1b reveals that the previously observed improvement in feature–target quality does not extend to engineered-feature correlation. This is again likely due to the alternative objective not accounting for post–end-to-end training outcomes, which the baseline does. Nevertheless, a lower engineered-feature correlation does not necessarily imply reduced test performance.

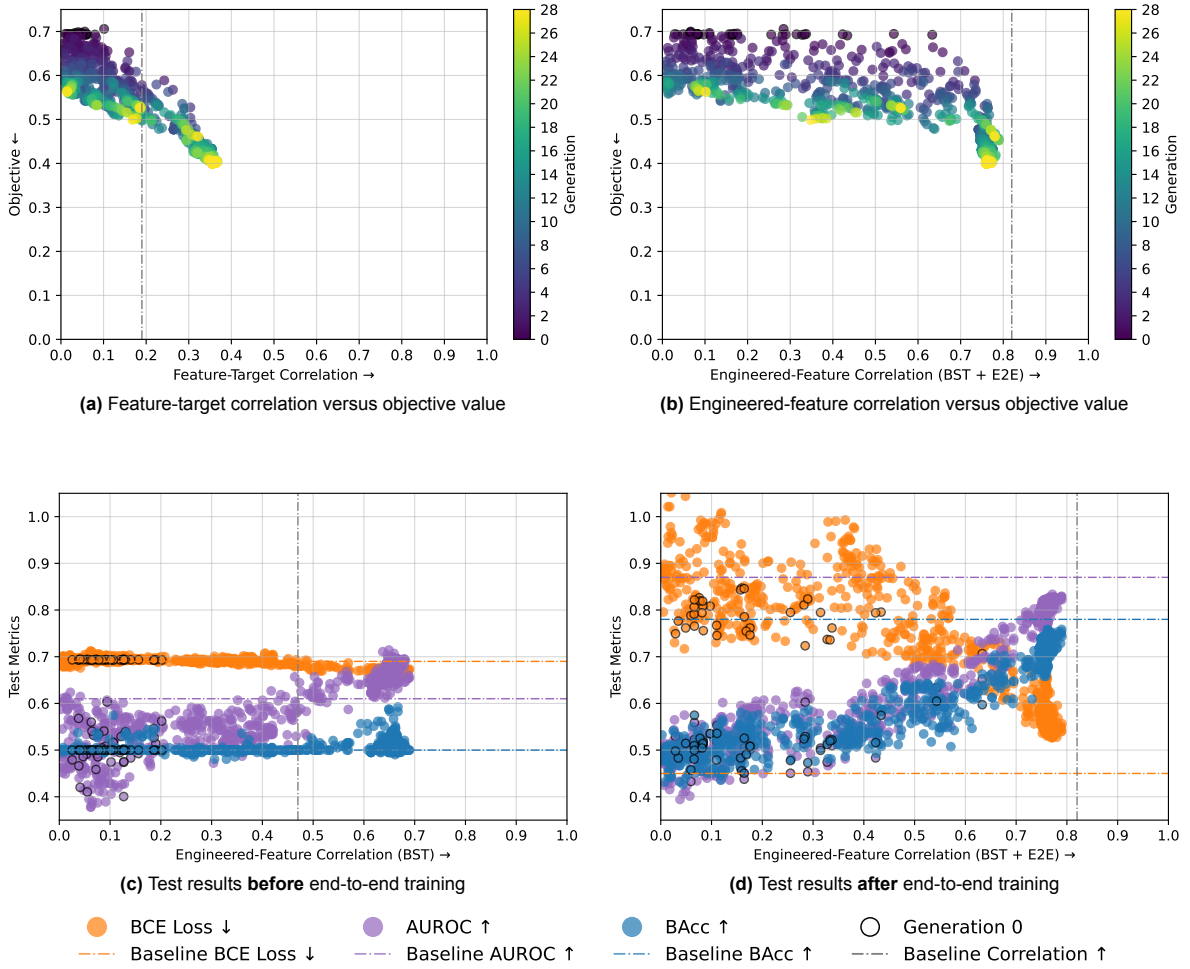
In contrast, Figure 8.1c shows that, prior to end-to-end training, the engineered-feature correlation is considerably higher than that of the baseline. This is consistent with the objective being better aligned with the performance of the BST component alone. Moreover, the test metrics are improved, with the AUROC even surpassing the baseline reported in Chapter 3, which is a result that was not previously observed without end-to-end training.

Finally, Figure 8.1d illustrates how the test metrics after end-to-end training compare to the baseline of the ablation study. The test results are slightly worse than the baseline values, despite several solutions having considerably better feature-target correlation. Again, this is likely due to the alternative objective being aligned less strongly with end-to-end performance. Nonetheless, the baseline performance from Chapter 3 is still exceeded.

8.2.2. Conclusion

In conclusion, using the alternative proxy objective without end-to-end training produces solutions that achieve considerably better-than-baseline feature-target correlations for the three-gated XOR task with two tabular features. However, this does not result in better test results after end-to-end training. This is likely due to the objective aligning less strongly with the results after end-to-end training, due to the omission of this training during fitness evaluation. Because of this, the baseline test results with BST alone were improved upon.

Using the alternative objective might improve upper-limit-performance by not resulting in close-to-zero values for solutions with low feature-target correlations. Nonetheless, this hypothesis requires further experimental validation before the alternative objective can be considered preferable to the baseline approach. Given a time budget of two weeks, the proxy objective incorporating end-to-end training remains the recommended strategy for optimising feature targets in the three-gated XOR task with two tabular features.



(a) Feature-target correlation versus objective value

(b) Engineered-feature correlation versus objective value

(c) Test results **before** end-to-end training

(d) Test results **after** end-to-end training

Figure 8.1: Subfigures resulting from optimising the alternative proxy objective, with **end-to-end training omitted** from fitness evaluation, through GOMEA for the three-gated XOR task with two tabular features. GOMEA was run for a runtime of two weeks, with the downsample seed set to 1, and the train seed set to 1. Arrows denote whether the corresponding metric should be maximised or minimised. Markers corresponding the initial population of GOMEA are highlighted using black outlines. The vertical dash-dotted lines denote the maximum corresponding correlation values obtained by the baseline approach. **(a)** Each dot represents a solution found by GOMEA, with the feature-target correlation plotted against the objective value. Dot colour indicates the generation of the solution. **(b)** Instead of feature-target correlation, the engineered-feature correlation after end-to-end training is plotted against the objective value. **(c)** Each dot represents a test metric plotted against the engineered-feature correlation of its corresponding solution, both *without* applying end-to-end training. The test metrics include: the Binary Cross Entropy (BCE) Loss, Area Under the Receiver-Operating Characteristic curve (AUROC), and Balanced Accuracy (BAcc). The dotted horizontal lines show the baseline value of the test metric in the same colour. **(d)** Instead of engineered-feature correlation and test metrics before end-to-end training, the results *after* end-to-end training are shown.

8.3. Standard DL Parameters

Chapter 7 used DL parameters specifically tuned for the three-gated XOR problem with two tabular features, in order to better approximate the upper-limit performance of the proposed method. However, these preliminary experiments relied on prior knowledge of the true feature targets, which is an assumption that is only valid for synthetic problems. In real-world scenarios, this knowledge is unavailable. Alternatively, HPO could be employed to identify the optimal DL parameters, but this would require repeatedly performing the computationally expensive GOMEA optimisation. Consequently, it is expected that standard DL parameters will be used in practice.

This section investigates the impact of replacing the tailored DL parameters with standard ones on the optimisation of feature targets through GOMEA. Instead of the DL parameters specified in Table 4.1, all learning rates are set to 0.001, all weight decays to 0.0001, and each stage performs 30 training epochs. Apart from these adjustments, the experimental setup remains identical to that described in Section 7.1. As before, the vertical dash-dotted lines indicate the correlations achieved by the baseline. Results for the downsample and a training seed of 1 are shown in Figure 8.2, while results for all four tested seed combinations are provided in Appendix C.2.2.

8.3.1. Discussion

Figure 8.2a displays that the discovered solutions exhibit worse-than-baseline feature–target correlations. Additionally, visible markers are mostly limited to the first few generations. In this specific example, only one solution from the final generation is visible, as the population converged to a single solution by the tenth generation. Consequently, fewer distinct solutions are observed than would be expected after seventeen generations. Most solutions therefore overlap, resulting in the small number of visible markers. This premature convergences indicates that a larger population size may be needed to maintain diversity beyond the tenth generation.

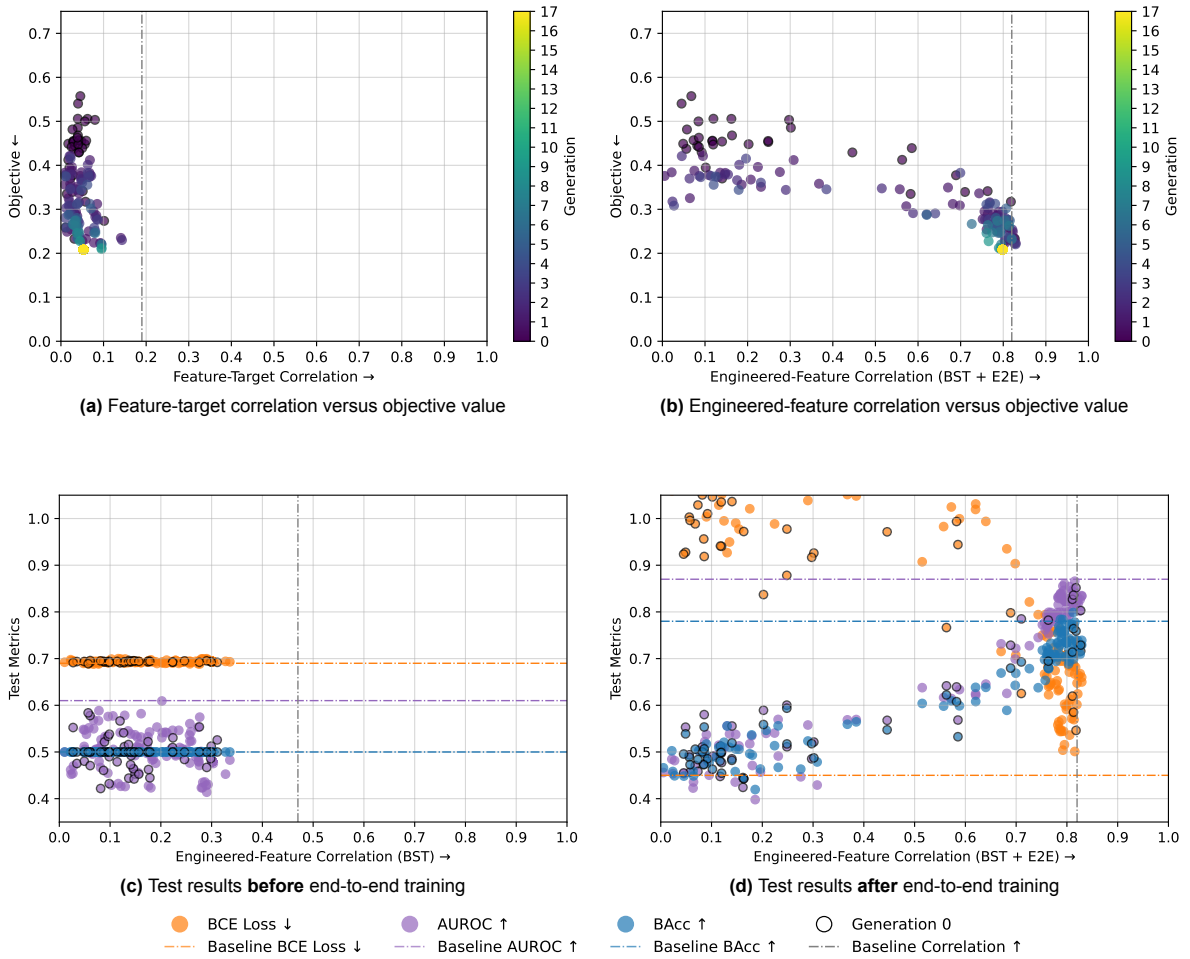
Figure 8.2b likewise shows relatively few visible markers, with the baseline engineered-feature correlation being slightly improved upon. This is likely due to the standard parameters enforcing more epochs of end-to-end training than was done in the baseline. However, the figure also indicates that the final point of convergence is suboptimal with respect to engineered-feature correlation. Interestingly, the initial population exhibits even stronger engineered-feature correlations than the baseline, with one solution achieving a correlation comparable to the best baseline result.

Figure 8.2c display that the test results before end-to-end training are very similar to the baseline, with only the engineered-feature correlation not approaching the baseline value. This is to be expected, since the standard DL parameters are not tailored to the specific task. Its main benefit seems to come from the additional epochs of end-to-end training, as illustrated by the previous figure.

Finally, Figure 8.2d illustrates the test results obtained after end-to-end training. The best observed AUROC and BAcc scores closely approximate the baseline values, with the latter even showing a slight improvement. However, a considerable difference in BCE loss is present. This may be attributed to the calibration issue discussed in Section 4.3.3, which appears to be more pronounced when using less suitable DL parameters compared with the tailored parameters applied in the baseline.

8.3.2. Conclusion

Using standard DL parameters to optimise feature targets for the three-gated XOR task with two tabular features results in considerably weaker feature–target correlations than those achieved with the baseline approach. Nevertheless, this reduction in correlation does not translate into poorer test performance, either before or after end-to-end training. Only the BCE test loss is notably worse when using standard parameters, possibly due to a stronger occurrence of the calibration issue. If this degradation is considered acceptable, then employing standard DL parameters for optimising feature targets in the three-gated XOR task yields results comparable to those obtained with tailored parameters.



(a) Feature-target correlation versus objective value

(b) Engineered-feature correlation versus objective value

(c) Test results **before** end-to-end training

(d) Test results **after** end-to-end training

Figure 8.2: Subfigures resulting from optimising the proxy objective through GOMEA with **standard DL parameters** for the three-gated XOR task with two tabular features. GOMEA was run for a runtime of two weeks, with the downsample seed set to 1, and the train seed set to 1. Arrows denote whether the corresponding metric should be maximised or minimised. Markers corresponding the initial population of GOMEA are highlighted using black outlines. The vertical dash-dotted lines denote the maximum corresponding correlation values obtained by the baseline approach. **(a)** Each dot represents a solution found by GOMEA, with the feature-target correlation plotted against the objective value. Dot colour indicates the generation of the solution. **(b)** Instead of feature-target correlation, the engineered-feature correlation after end-to-end training is plotted against the objective value. **(c)** Each dot represents a test metric plotted against the engineered-feature correlation of its corresponding solution, both *without* applying end-to-end training. The test metrics include: the Binary Cross Entropy (BCE) Loss, Area Under the Receiver-Operating Characteristic curve (AUROC), and Balanced Accuracy (BAcc). The dotted horizontal lines show the baseline value of the test metric in the same colour. **(d)** Instead of engineered-feature correlation and test metrics before end-to-end training, the results *after* end-to-end training are shown.

8.4. Standard DL Parameters and Without End-to-End Training

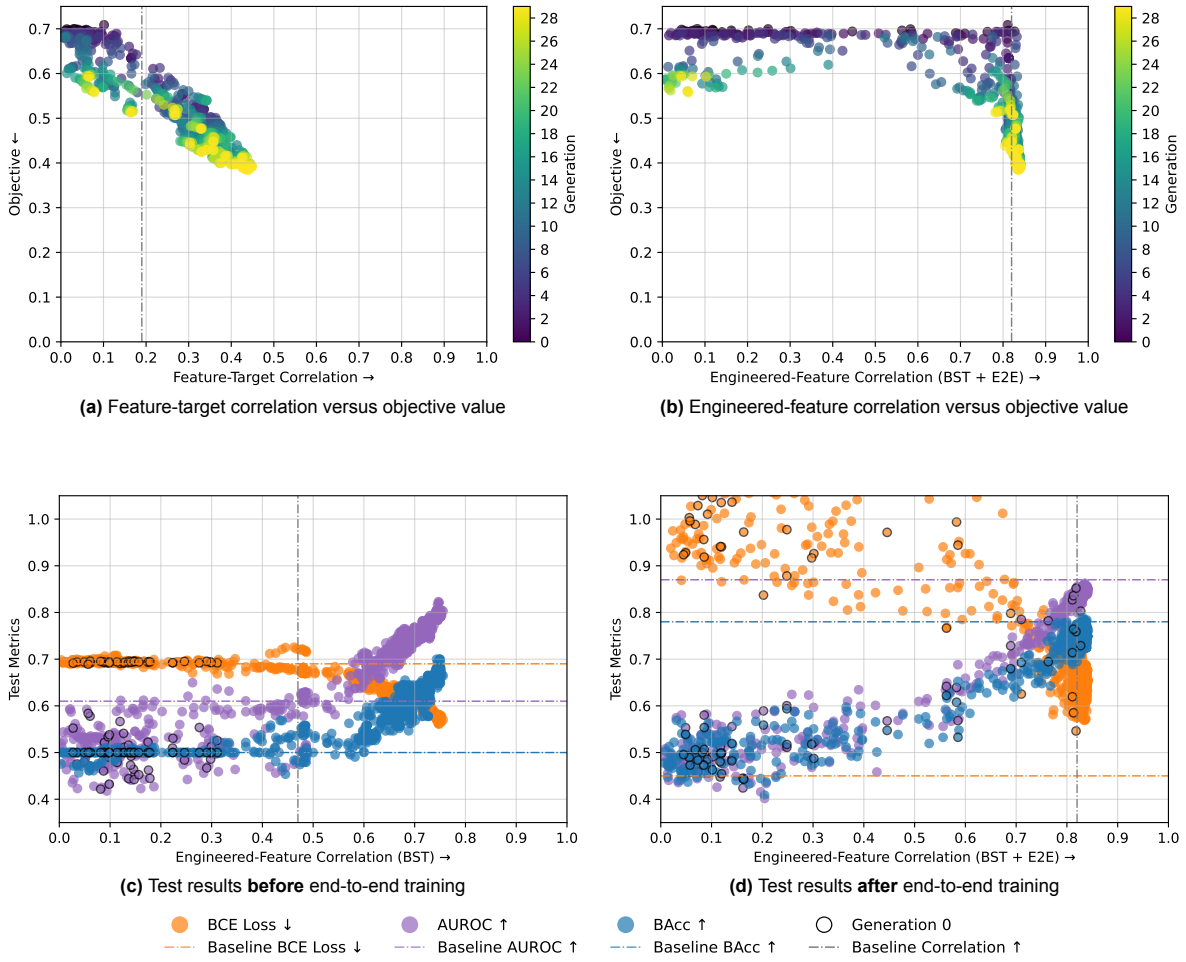
Finally, the prior two ablations are combined to analyse the effect it has on optimisation and its results. The previous section showed that standard DL parameters produce solutions that, despite obtaining worse feature correlations, obtain similar test results than when using tailored DL parameters. Removing end-to-end training during objective evaluation resulted in solutions that displayed considerably better-than-baseline feature-target correlations, whilst obtaining slightly worse test results after end-to-end training. This section analyses how combining these two changes compares to the baseline setting.

Other than the discussed changes, the same experimental setup is used as described in Section 7.1 is utilised. Once again, the vertical dash-dotted lines show the correlations obtained by the baseline. The results for the downsample and train seed of 1 are given in Figure 8.3. The results for all four tested seed combinations are given in Appendix C.2.3.

8.4.1. Discussion

Figure 8.3a shows that, as in Section 8.2, solutions are found that exhibit considerably stronger feature–target correlations than the baseline. In fact, these correlations appear to surpass those obtained when only omitting end-to-end training is from fitness evaluation. The use of standard DL parameters therefore seems to provide an additional advantage in optimising solutions that achieve high feature–target correlation, complementing the earlier observed benefit of shifting the optimisation objective.

Furthermore, Figure 8.3b reveals a slight improvement over the baseline in engineered-feature correlation after end-to-end training. As shown in Section 8.2, removing end-to-end training from objective evaluation negatively affects the engineered-feature correlation, due to a shift in optimisation pressure away from performance after end-to-end training. Hence, the observed improvement is likely a result of the standard DL parameters enforcing more epochs of end-to-end training, which, as demonstrated in Section 8.3, benefits the correlation between the engineered feature and the ground-truth features.



(a) Feature-target correlation versus objective value

(b) Engineered-feature correlation versus objective value

(c) Test results **before** end-to-end training

(d) Test results **after** end-to-end training

Figure 8.3: Subfigures resulting from optimising the proxy objective through GOMEA with **end-to-end training omitted** from fitness evaluation and using **standard DL parameters** for the three-gated XOR task with two tabular features. GOMEA was run for a runtime of two weeks, with the downsample seed set to 1, and the train seed set to 1. Arrows denote whether the corresponding metric should be maximised or minimised. Markers corresponding the initial population of GOMEA are highlighted using black outlines. The vertical dash-dotted lines denote the maximum corresponding correlation values obtained by the baseline approach. **(a)** Each dot represents a solution found by GOMEA, with the feature-target correlation plotted against the objective value. Dot colour indicates the generation of the solution. **(b)** Instead of feature-target correlation, the engineered-feature correlation after end-to-end training is plotted against the objective value. **(c)** Each dot represents a test metric plotted against the engineered-feature correlation of its corresponding solution, both *without* applying end-to-end training. The test metrics include: the Binary Cross Entropy (BCE) Loss, Area Under the Receiver-Operating Characteristic curve (AUROC), and Balanced Accuracy (BAcc). The dotted horizontal lines show the baseline value of the test metric in the same colour. **(d)** Instead of engineered-feature correlation and test metrics before end-to-end training, the results *after* end-to-end training are shown.

Figure 8.2c demonstrates a further advantage of the shift in optimisation objective by presenting the test results obtained before end-to-end training. As observed in Section 8.2, all baseline test values are improved upon, including the engineered-feature correlation. Moreover, Figure 8.3c indicates even greater improvements compared with the use of tailored DL parameters, surpassing the baseline values reported in Chapter 3. Notably, this represents the first instance of achieving better-than-baseline performance on the three-gated XOR task with two tabular features by using feature targets derived from GOMEA, without employing end-to-end training in the final model.

Finally, Figure 8.3d shows that, despite the improved engineered-feature correlation, overall performance does not surpass the baseline. Only the BAcc metric appears to improve marginally, consistent with the trend observed in Section 8.3. The AUROC scores remain comparable to the baseline, while the BCE loss is considerably worse. As noted before, this is likely due to the calibration issue being more pronounced when using standard DL parameters. Furthermore, as seen in the previous section and in Figure 8.3d, several solutions in the initial population achieve test results that closely approximate the best outcomes among all discovered solutions. Consequently, further optimisation yields only marginal improvements in the final model's performance.

8.4.2. Conclusion

In conclusion, combining the two ablations discussed in this chapter amplifies both the advantages and drawbacks observed when each is applied individually. Specifically, the feature–target correlation was further improved, leading to test results that surpassed the baseline performance described in Chapter 3, without the need for end-to-end training in the final model. However, the BCE loss after end-to-end training deteriorated further, indicating that the calibration issue is more pronounced for this configuration than in Section 8.3. Nevertheless, if the degraded BCE loss is acceptable, omitting end-to-end training during objective evaluation and using standard DL parameters in place of tailored parameters can be integrated into the baseline approach to achieve comparable performance on the three-gated XOR task with two tabular features.

9

Conclusion and Future Work

The previous chapter addressed the final research question, providing insights into this thesis' proposed method of optimising feature targets through GOMEA. Building on these results, this chapter seeks to answer the overarching research question by first summarising the key conclusions drawn in the preceding chapters. By revisiting these findings in a structured manner, the chapter highlights how each component of the research contributed to the main research question. After this summary, Section 9.2 looks at possible directions for future research and ways the methods and results from this thesis could be improved or expanded.

9.1. Conclusion

9.1.1. Research Question 1

Chapter 4 assumed knowledge of the ideal features for a given problem. While this was feasible for the synthetic problems described in Chapter 3, it does not hold for real-world scenarios. Consequently, the chapter explored how this feature knowledge could be incorporated into the training of MultiFIX, with the aim of improving its performance on tasks with extreme joint modality dependence. The specific research question addressed in this chapter was defined as follows.

Research Question

RQ1. How can feature knowledge be used in training MultiFIX and does it improve performance on tasks with extreme joint modality dependence?

The chapter demonstrated that knowledge of the ideal features can be used to generate ground-truth feature targets, which in turn guided the application of BST to bias the individual DL blocks of the MultiFIX architecture, optionally followed by end-to-end training. This approach was shown to improve MultiFIX performance on the three-gated XOR task with two tabular features, with end-to-end training yielding the highest performance.

Additionally, the chapter showed that using feature targets closely resembling the true feature targets leads to comparable performance on the task. This resemblance was quantified using feature correlation. Specifically, the *feature-target correlation* measured the similarity between the set of feature targets and the ground-truth feature targets, while the *engineered-feature correlation* measured how closely the engineered features produced by MultiFIX, after BST with the given feature targets, with or without end-to-end training, corresponded to the true feature targets.

In summary, the chapter demonstrated how feature knowledge can be made explicit through feature targets and incorporated into the training of MultiFIX through an alternative training strategy. It also evaluated the method's resilience to noise in feature targets, thereby providing an estimate of the minimum level of accuracy required to improve MultiFIX its baseline performance on the three-gated XOR task with two tabular features when the feature targets are optimised in later chapters.

9.1.2. Research Question 2

Chapter 5 also made use of the available ideal feature knowledge to construct an idealised version of the optimisation problem. This setup was employed to estimate the effectiveness of GOMEA in optimising feature targets. The chapter was guided by the following research question.

Research Question

RQ2. If the correlation to the ground-truth features is available, can GOMEA be used to approximate the ideal feature targets, and what parameter configuration yields the best performance?

The preceding chapter described how feature knowledge can be made explicit through the use of feature targets, which can then be used to train MultiFIX via BST. This chapter optimises these feature targets based on their correlation with the true feature targets, referred to as the feature-target correlation. As the calculation of this metric is relatively efficient, it enabled performance evaluation across a range of GOMEA configurations.

The chapter demonstrated that GOMEA successfully identified the ideal feature targets for this idealised problem. Furthermore, it identified several parameter configurations that were particularly efficient. These results serve both as an estimate of GOMEA its upper-limit performance and as a relatively inexpensive method for parameter selection.

9.1.3. Research Question 3

Ideal feature targets are typically unknown in real-world scenarios. Therefore, Chapter 6 demonstrated how to assess the quality of a set of feature targets, instead of using the ground-truth feature targets to determine their similarity through correlation. This quality assessment can then be used as an objective in optimising feature targets in a scenario that resembles the real-world scenario. The following research question formed the basis of this chapter.

Research Question

RQ3. How can a proxy objective be designed to guide feature target optimisation in the absence of ground-truth features and does the proxy objective yield a fitness signal on tasks with extreme joint modality dependence?

Chapter 6 demonstrated that the final training loss obtained after end-to-end training and BST for a given set of feature targets can serve as a proxy objective when ground-truth features are unavailable. Evaluation of the fitness signal showed that, for the three-gated XOR task with two tabular features, the objective values of different feature-target sets correlated with their respective feature-target correlations. This finding suggests that optimising the proxy objective could lead to feature targets that ultimately enhance MultiFIX its baseline performance.

9.1.4. Research Question 4

The previous chapter introduced a proxy objective designed to guide optimisation of feature targets that can improve the baseline performance of MultiFIX through BST and subsequent end-to-end training. Chapter 7 investigated the effectiveness of this optimisation using GOMEA across several synthetic problems, including the three-gated XOR task with two tabular features. The research question guiding this chapter is stated below, with the final part addressed in Chapter 8.

Research Question

RQ4. Does optimising the designed proxy objective with GOMEA yield feature targets that improves the performance of MultiFIX on tasks with extreme joint modality dependence, and why or why not?

Chapter 7 demonstrated that optimising the proxy objective successfully produced feature targets that enhanced baseline MultiFIX performance on the three-gated XOR task with two tabular features, even though the discovered solutions showed relatively low feature–target correlations. It was also shown that BST alone was insufficient for improving model performance, with end-to-end training appearing to yield the greatest model performance gains.

The optimisation process was further applied to the single XOR and AND tasks, where the baseline approach already performed relatively well. For these tasks, no solutions were found that achieved high feature–target correlations. Nonetheless, several sets of feature targets were discovered that approximated the baseline performance to some degree, with the single XOR task showing closer alignment than the AND task. Interestingly, in the AND task, the level of feature–target correlation and its corresponding engineered–feature correlation did not appear to relate to model performance, unlike the XOR tasks. Overall, optimising feature targets through GOMEA proved most beneficial in cases of strong modality dependence, where more complex feature interactions appeared to benefit most from this approach.

Chapter 8 then conducted an ablation study to assess the impact of key components within the feature-target optimisation method on the three-gated XOR task with two tabular features. First, end-to-end training was removed from the objective evaluation to substantially reduce computational cost during feature target optimisation. Although this led to solutions with higher feature–target correlations, their corresponding model after BST and end-to-end training obtains slightly worse test performance.

Next, the DL parameters, which were tailored to the specific task, were replaced with standard values. This simulates a real-world scenario in which HPO is deemed too expensive. Despite achieving weaker feature–target correlations, this configuration produced test results comparable to those from the tailored setup. Finally, combining both modifications amplified both the advantages and disadvantages observed when each was applied individually. The main disadvantage appears to be a calibration issue, where the model is overconfident in misclassifications.

9.1.5. Main Research Question

This section draws on the results and conclusions from the preceding chapters to answer the main research question of this thesis, stated below.

Main Research Question

To what extent can evolutionary algorithms optimise feature targets to improve the performance of MultiFIX on problems with extreme joint dependence between modalities?

In summary, the results demonstrated that GOMEA can optimise feature targets that, when used in BST and subsequent end-to-end training, outperform the baseline MultiFIX performance on the three-gated XOR task with two tabular features. This approach proved particularly effective for tasks characterised by strong joint-modality dependence, where complex feature interactions appeared to benefit most from the optimisation process. For simpler tasks such as the single XOR and AND problems, the improvements were smaller but still comparable to the baseline. Consequently, it is recommended to first apply the baseline MultiFIX approach to assess its performance before employing GOMEA-based feature target optimisation, as the latter is considerably more computationally expensive and does not guarantee improved performance.

These limitations could potentially be addressed by simplifying the optimisation process, for instance by omitting end-to-end training during objective evaluation to significantly reduce computational cost, and by using standard DL parameters alongside a larger number of end-to-end training epochs. This combination may yield performance more comparable to the baseline on tasks such as the AND and single XOR problems. Notably, higher feature-target correlations did not always translate to better overall performance, suggesting that future improvements are more likely to come from refining the training procedure rather than further tuning the evolutionary optimisation itself. Therefore, it is not recommended to focus on enhancing the performance of the evolutionary algorithm, but rather on improving how the resulting feature targets are integrated and trained within the MultiFIX framework.

9.2. Future Work

This section outlines several directions for future work building on the outcomes of this thesis. A key area for further investigation is the calibration issue, which may limit the practical applicability of the resulting models. The problem lies in the model its tendency to make highly confident but incorrect predictions, which is an undesirable property, particularly in safety-critical domains. A detailed analysis of the model its behaviour would be a good starting point for understanding the cause of this issue. This could involve examining the instances where the model fails and identifying potential patterns among the misclassified samples. Insights gained from such an analysis may reveal underlying shortcomings that, once addressed, could help reduce the relatively high BCE loss.

One potential solution is data augmentation, which can be used to increase the size of the dataset without increasing the problem complexity. For instance, when generating augmented samples, it can often be assumed that the features that should ideally be engineered in the original data remain the same in the augmented data. This means that no additional feature targets need to be optimised, keeping the effective problem size constant. Increasing the amount of data in this way could help mitigate the impact of the calibration issue. However, the validity of this assumption depends on both the specific problem and the chosen augmentation method. For example, when augmenting tabular data by adding small amounts of noise, a feature relation such as $x_1 > x_2$ in the original sample may no longer hold in the augmented version. The likelihood of such changes naturally depends on the magnitude of the noise applied. By contrast, it is generally less likely that an image feature is disrupted by simple transformations, such as rotating the image.

Furthermore, more complex problems can be explored. For instance, the three-gated XOR task with two image features was shown to yield worse baseline performance, indicating greater complexity than the variant with two tabular features. Optimising feature targets for this problem may reveal previously unseen shortcomings. In any case, it provides a valuable opportunity to better understand the limitations of the method proposed in this thesis. Naturally, a four-gated XOR task with two features per modality represents a logical next step after investigating the three-gated XOR with two image features.

Additionally, optimisation of multiple features per modality could be explored. This thesis limited each modality to a single feature, although it would intuitively make sense, for example, to optimise two tabular features and one image feature for the three-gated XOR task, as this more closely reflects the underlying problem structure and could potentially yield better results. The drawback is that increasing the number of features also increases the problem size, which may negatively affect performance. Nevertheless, this should be investigated to determine its actual impact. The baseline MultiFIX has demonstrated on several occasions that engineering more than one feature can be beneficial [15].

Finally, as noted earlier in this thesis, the engineered features often exhibit stronger correlations than the feature targets used in BST, particularly after end-to-end training. These engineered features could be leveraged to refine the feature targets during optimisation or even to replace the solution variables entirely. Such an approach might create a bootstrapping effect, where engineered features serve as feature targets to generate increasingly highly correlated features. Investigating this possibility represents a promising direction for improving the optimisation of feature targets.

References

[1] Nitin Arora et al. “Introduction to Big Data Analytics”. In: *Towards the Integration of IoT, Cloud and Big Data: Services, Applications and Standards*. Ed. by Vinay Rishiwal et al. Singapore: Springer Nature Singapore, 2023, pp. 1–18. ISBN: 978-981-99-6034-7. DOI: 10.1007/978-981-99-6034-7_1. URL: https://doi.org/10.1007/978-981-99-6034-7_1.

[2] I Made Putrama and Péter Martinek. “Heterogeneous data integration: Challenges and opportunities”. In: *Data in Brief* 56 (Oct. 2024), p. 110853. DOI: 10.1016/j.dib.2024.110853.

[3] Anil Rahate et al. “Multimodal Co-learning: Challenges, applications with datasets, recent advances and future directions”. In: *Information Fusion* 81 (2022), pp. 203–239. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2021.12.003. URL: https://www.sciencedirect.com/science/article/pii/S1566253521002530.

[4] Adrienne Kline et al. “Multimodal machine learning in precision health: A scoping review”. In: *npj Digital Medicine* 5.1 (2022), pp. 1–14. ISSN: 2398-6352. DOI: 10.1038/s41746-022-00712-8. URL: https://doi.org/10.1038/s41746-022-00712-8.

[5] Muhammad Adeel Azam et al. “A review on multimodal medical image fusion: Compendious analysis of medical modalities, multimodal databases, fusion techniques and quality metrics”. In: *Computers in Biology and Medicine* 144 (May 2022), p. 105253. DOI: 10.1016/j.compbiomed.2022.105253.

[6] Fei Zhao, Chengcui Zhang, and Baocheng Geng. “Deep Multimodal Data Fusion”. In: *ACM Comput. Surv.* 56.9 (Apr. 2024). ISSN: 0360-0300. DOI: 10.1145/3649447. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3649447.

[7] Andreas Holzinger et al. *What do we need to build explainable AI systems for the medical domain?* 2017. arXiv: 1712.09923 [cs.AI]. URL: https://arxiv.org/abs/1712.09923.

[8] Aurélie Pahud de Mortanges et al. “Orchestrating explainable artificial intelligence for multimodal and longitudinal data in medical imaging”. In: *NPJ digital medicine* 7.1 (2024). 195, p. 195. DOI: 10.1038/s41746-024-01190-w. URL: https://doi.org/10.1038/s41746-024-01190-w.

[9] Ramprasaath R. Selvaraju et al. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 618–626. DOI: 10.1109/ICCV.2017.74.

[10] Scott M. Lundberg and Su-In Lee. “A unified approach to interpreting model predictions”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS’17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 4768–4777. ISBN: 9781510860964.

[11] Cynthia Rudin. “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead”. In: *Nature Machine Intelligence* 1.5 (2019). 206, pp. 206–215. DOI: 10.1038/s42256-019-0048-x. URL: https://doi.org/10.1038/s42256-019-0048-x.

[12] Binh Tran, Bing Xue, and Mengjie Zhang. “Using Feature Clustering for GP-Based Feature Construction on High-Dimensional Data”. In: *Genetic Programming*. Ed. by James McDermott et al. Cham: Springer International Publishing, 2017, pp. 210–226. ISBN: 978-3-319-55696-3.

[13] Marco Virgolin, Tanja Alderliesten, and Peter A.N. Bosman. “On explaining machine learning models by evolving crucial and compact features”. In: *Swarm and Evolutionary Computation* 53 (2020), p. 100640. ISSN: 2210-6502. DOI: https://doi.org/10.1016/j.swevo.2019.100640. URL: https://www.sciencedirect.com/science/article/pii/S2210650219305036.

[14] Mafalda Malafaia et al. “A Step towards Interpretable Multimodal AI Models with MultiFIX”. In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. GECCO ’25 Companion. NH Malaga Hotel, Malaga, Spain: Association for Computing Machinery, 2025, pp. 2001–2009. ISBN: 9798400714641. DOI: 10.1145/3712255.3734292. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3712255.3734292.

[15] Mafalda Malafaia et al. *MultiFIX: An XAI-friendly feature inducing approach to building models from multimodal data*. 2024. arXiv: 2402.12183 [cs.AI]. URL: https://arxiv.org/abs/2402.12183.

[16] Mafalda Malafaia et al. “Learning multimodal explainable AI models from medical images and tabular data: proof of concept”. In: *Medical Imaging 2025: Image Processing*. Ed. by Olivier Colliot and Jhimli Mitra. Vol. 13406. International Society for Optics and Photonics. SPIE, 2025, p. 1340612. DOI: 10.1117/12.3040402. URL: https://doi.org/10.1117/12.3040402.

[17] Maciej Pawłowski, Anna Wróblewska, and Sylwia Sysko-Romańczuk. “Effective Techniques for Multimodal Data Fusion: A Comparative Analysis”. In: *Sensors* 23.5 (2023). ISSN: 1424-8220. DOI: 10.3390/s23052381. URL: https://www.mdpi.com/1424-8220/23/5/2381.

[18] Tadas Baltrusaitis, Chaitanya Ahuja, and Louis-Philippe Morency. “Multimodal Machine Learning: A Survey and Taxonomy”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 41.2 (Feb. 2019), pp. 423–443. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2018.2798607. URL: https://doi-org.tudelft.idm.oclc.org/10.1109/TPAMI.2018.2798607.

[19] Songtao Li and Hao Tang. *Multimodal Alignment and Fusion: A Survey*. 2024. arXiv: 2411.17040 [cs.CV]. URL: https://arxiv.org/abs/2411.17040.

[20] Divyam Madaan et al. “Jointly modeling inter- & intra-modality dependencies for multi-modal learning”. In: *Proceedings of the 38th International Conference on Neural Information Processing Systems*. NIPS ’24. Vancouver, BC, Canada: Curran Associates Inc., 2025. ISBN: 9798331314385.

[21] Jing Gao et al. “A Survey on Deep Learning for Multimodal Data Fusion”. In: *Neural Computation* 32.5 (May 2020), pp. 829–864. ISSN: 0899-7667. DOI: 10.1162/neco_a_01273. eprint: https://direct.mit.edu/neco/article-pdf/32/5/829/1865303/neco_a_01273.pdf. URL: https://doi.org/10.1162/neco%5C_a%5C_01273.

[22] Siyi Du et al. *TIP: Tabular-Image Pre-training for Multimodal Classification with Incomplete Data*. 2024. arXiv: 2407.07582 [cs.CV]. URL: https://arxiv.org/abs/2407.07582.

[23] Siyi Du et al. *STiL: Semi-supervised Tabular-Image Learning for Comprehensive Task-Relevant Information Exploration in Multimodal Classification*. 2025. arXiv: 2503.06277 [cs.CV]. URL: https://arxiv.org/abs/2503.06277.

[24] Jiaqi Luo, Yuan Yuan, and Shixin Xu. *TIME: TabPFN-Integrated Multimodal Engine for Robust Tabular-Image Learning*. 2025. arXiv: 2506.00813 [cs.CV]. URL: https://arxiv.org/abs/2506.00813.

[25] Maciej Pawłowski, Anna Wróblewska, and Sylwia Sysko-Romańczuk. “Effective Techniques for Multimodal Data Fusion: A Comparative Analysis”. In: *Sensors* 23.5 (Feb. 2023), p. 2381. ISSN: 1424-8220. DOI: 10.3390/s23052381. URL: http://dx.doi.org/10.3390/s23052381.

[26] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV]. URL: https://arxiv.org/abs/1512.03385.

[27] Alexander Kolesnikov et al. *Big Transfer (BiT): General Visual Representation Learning*. 2020. arXiv: 1912.11370 [cs.CV]. URL: https://arxiv.org/abs/1912.11370.

[28] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG]. URL: https://arxiv.org/abs/1412.6980.

[29] Thalea Schlender et al. *Improving the efficiency of GP-GOMEA for higher-arity operators*. 2024. arXiv: 2402.09854 [cs.NE]. URL: https://arxiv.org/abs/2402.09854.

[30] Dirk Thierens. “Scalability Problems of Simple Genetic Algorithms”. In: *Evolutionary Computation* 7.4 (Dec. 1999), pp. 331–352. ISSN: 1063-6560. DOI: 10.1162/evco.1999.7.4.331. eprint: https://direct.mit.edu/evco/article-pdf/7/4/331/1493126/evco.1999.7.4.331.pdf. URL: https://doi.org/10.1162/evco.1999.7.4.331.

[31] Arkadiy Dushatskiy et al. *Parameterless Gene-pool Optimal Mixing Evolutionary Algorithms*. 2021. arXiv: 2109.05259 [cs.NE]. URL: https://arxiv.org/abs/2109.05259.

[32] Anton Bouter. "Optimal Mixing Evolutionary Algorithms for Large-Scale Real-Valued Optimization". PhD thesis. Delft University of Technology, 2023. URL: https://resolver.tudelft.nl/uuid:0e03913c-898e-4392-8de5-072a7ead7fd6.

[33] Marco Virgolin et al. "Symbolic regression and feature construction with GP-GOMEA applied to radiotherapy dose reconstruction of childhood cancer survivors". In: *Proceedings of the Genetic and Evolutionary Computation Conference*. GECCO '18. Kyoto, Japan: Association for Computing Machinery, 2018, pp. 1395–1402. ISBN: 9781450356183. DOI: 10.1145/3205455.3205604. URL: https://doi.org/10.1145/3205455.3205604.

[34] Ngoc Hoang Luong, Han La Poutré, and Peter A.N. Bosman. "Multi-objective Gene-pool Optimal Mixing Evolutionary Algorithm with the Interleaved Multi-start Scheme". In: *Swarm and Evolutionary Computation* 40 (2018), pp. 238–254. ISSN: 2210-6502. DOI: https://doi.org/10.1016/j.swevo.2018.02.005. URL: https://www.sciencedirect.com/science/article/pii/S2210650217304765.

[35] Peter A.N. Bosman and Dirk Thierens. "More concise and robust linkage learning by filtering and combining linkage hierarchies". In: *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*. GECCO '13. Amsterdam, The Netherlands: Association for Computing Machinery, 2013, pp. 359–366. ISBN: 9781450319638. DOI: 10.1145/2463372.2463420. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/2463372.2463420.

[36] Peter A.N. Bosman and Dirk Thierens. "Linkage neighbors, optimal mixing and forced improvements in genetic algorithms". In: *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*. GECCO '12. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 2012, pp. 585–592. ISBN: 9781450311779. DOI: 10.1145/2330163.2330247. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/2330163.2330247.

[37] Anton Bouter and Peter A. N. Bosman. "A Joint Python/C++ Library for Efficient yet Accessible Black-Box and Gray-Box Optimization with GOMEA". In: *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*. GECCO '23 Companion. ACM, July 2023, pp. 1864–1872. DOI: 10.1145/3583133.3596361. URL: http://dx.doi.org/10.1145/3583133.3596361.

[38] Marco Virgolin et al. "Scalable genetic programming by gene-pool optimal mixing and input-space entropy-based building-block learning". In: *Proceedings of the Genetic and Evolutionary Computation Conference*. GECCO '17. Berlin, Germany: Association for Computing Machinery, 2017, pp. 1041–1048. ISBN: 9781450349208. DOI: 10.1145/3071178.3071287. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3071178.3071287.

[39] Ngoc Hoang Luong, Han La Poutré, and Peter A.N. Bosman. "Exploiting Linkage Information and Problem-Specific Knowledge in Evolutionary Distribution Network Expansion Planning". In: *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*. GECCO '15. Madrid, Spain: Association for Computing Machinery, 2015, pp. 1231–1238. ISBN: 9781450334723. DOI: 10.1145/2739480.2754682. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/2739480.2754682.

[40] Anton Bouter, Dirk Thierens, and Peter A. N. Bosman. "The Pitfalls and Potentials of Adding Gene-invariance to Optimal Mixing". In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. GECCO '25 Companion. NH Malaga Hotel, Malaga, Spain: Association for Computing Machinery, 2025, pp. 1918–1926. ISBN: 9798400714641. DOI: 10.1145/3712255.3734293. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3712255.3734293.

A

MultiFIX Architecture Description

The following figures give a description on how the architecture of MultiFIX. Figure A.3 gives a general overview of how the pipeline is set up. The architecture of the CNN is explained in more detail in Figure A.1. The MLP in the feature induction is described in Figure A.2. Finally, the feature fusion MLP is illustrated in Figure A.4.

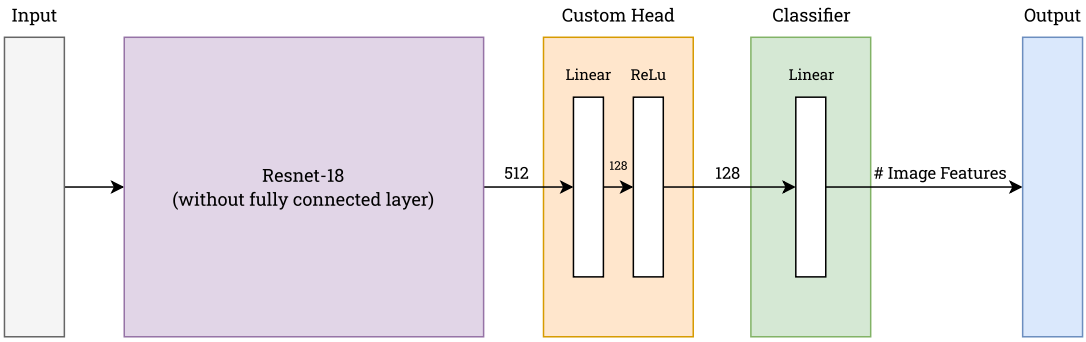


Figure A.1: Detailed description of the image block architecture, denoted as 'CNN' in Figure A.3. The weights are pre-loaded from Torchvision.

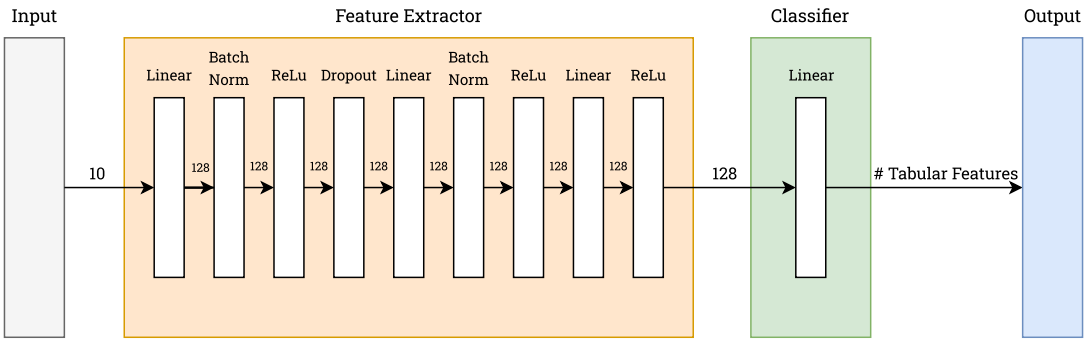


Figure A.2: Detailed description of the tabular block architecture, denoted as 'MLP' in the 'Feature Induction' block in Figure A.3. Dropout rate used is 0.125.

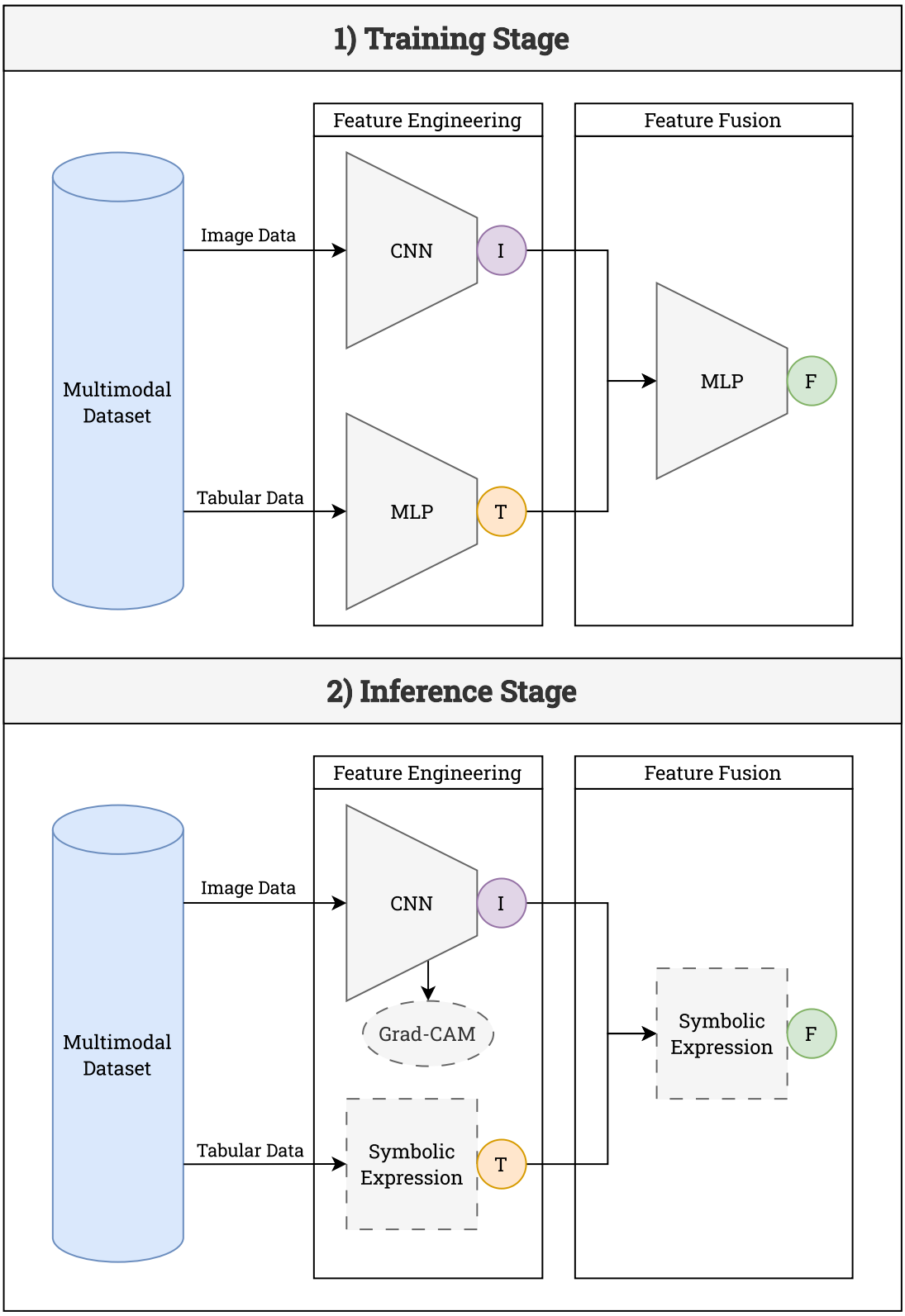


Figure A.3: Overview of MultiFIX. Data from the multimodal dataset is passed into the feature engineering blocks. Feature vectors I and T are concatenated and passed to the fusion block to make the final prediction F in the Training Stage. In the Inference Stage, image features are explained through Grad-CAM, and both MLPs are replaced with symbolic expressions obtained with GP-GOMEA, using the DL blocks from the training stage to determine the input and labels on which the symbolic expressions are fitted.

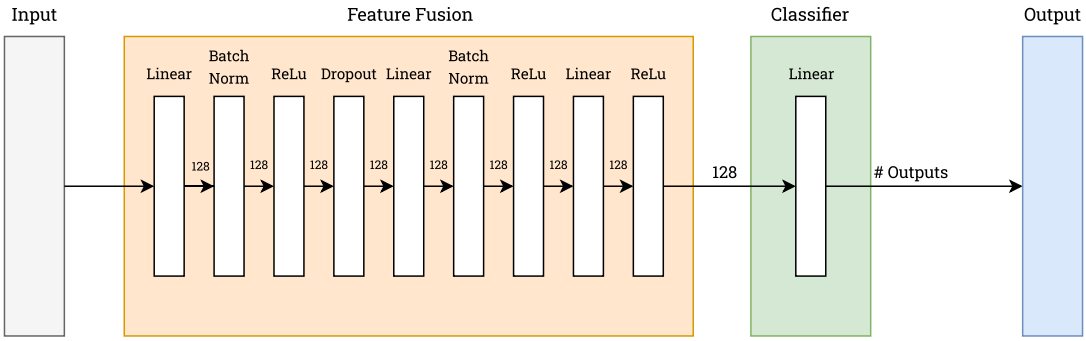


Figure A.4: Detailed description of the fusion block architecture, denoted as 'MLP' in the 'Feature Fusion' block in Figure A.3. Dropout rate used is 0.125. Input size depends on the number of image and tabular features. The features are concatenated before fusion, so the input size will simply be the sum of the number of image and tabular features. The output size depends on the problem. It can be adapted to work with multi-label classification, multi-class classification, or regression. However, the loss function should be adapted accordingly.

B

Autoencoder Pre-Training

The previous chapter showed that the image DL block mainly consists of the ResNet component, which forms the primary computational bottleneck during training. Therefore, an AE is trained and its encoder weights are transferred to the ResNet. Afterwards, the ResNet component is frozen and its outputs, i.e., the image embeddings, are cached.

This supplementary chapter highlights the AE pre-training step in the MultiFIX pipeline. First, Appendix B.1 details the architecture of the AE. Next, Appendix B.2 discusses the experimental setup, followed by the results in Appendix B.3, which highlights the loss progression and displays some image reconstructions.

B.1. Autoencoder Architecture

The AE consists of an encoder and a decoder. The encoder consists of the ResNet block, as illustrated in Figure A.1. The decoder reconstructs the input image from the latent representation, which has a dimensionality of 128. It first maps this latent vector through fully connected layers into a tensor of size $512 \times 7 \times 7$ after which a sequence of four residual decoding blocks progressively increases the spatial resolution while reducing the number of channels. Finally, a transposed convolutional layer produces a three-channel image, and a sigmoid activation ensures that the output pixel values lie within the range $[0, 1]$.

B.2. Experimental Setup

In training the AE, the following experimental setup is utilised. First, the Adam optimiser is used in training, with a learning rate of 0.0001 and no weight decay, i.e., set to zero. Then, the Mean Squared Error (MSE) loss is used to determine the loss. This is done over the entire training set. Therefore, only the data split seed needs to be determined. Finally, training is performed for 100 epochs, after which the final encoder and decoder weights are saved to reuse in MultiFIX.

B.3. Results

The MSE loss per epoch of training is illustrated in Figure B.1. This figure shows that the loss quickly decreases in the first few epochs, after which it steadily improves. Final epochs seem to provide marginal improvements. Figure B.2 showcases some random samples and their reconstructed images to further assess the quality of the AE. This shows that the AE is able to reconstruct the images rather well, with the prominent features, i.e., the presence of a circle, rectangle, and triangle, being clearly visible for this small set of samples.

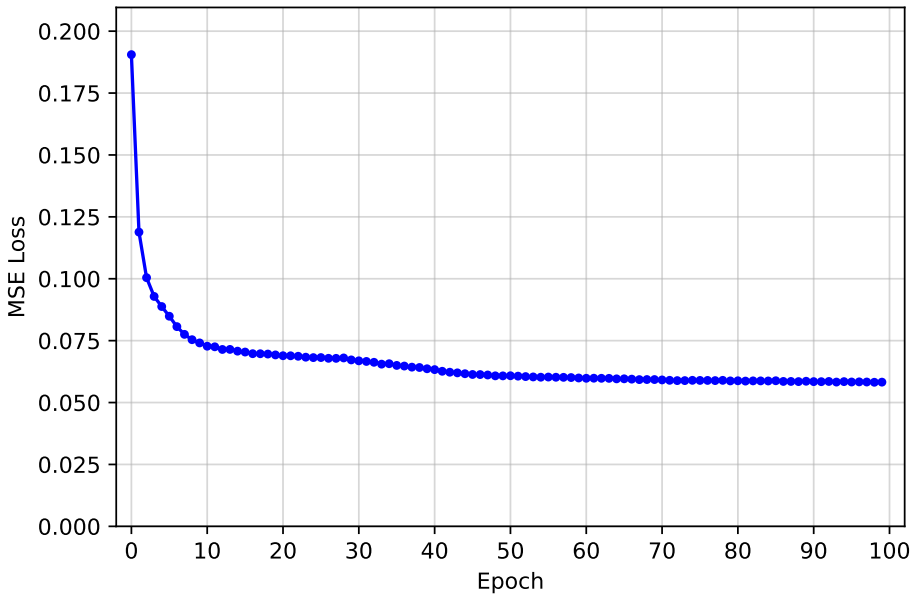


Figure B.1: Mean Squared Error (MSE) Loss progression during training over the epochs

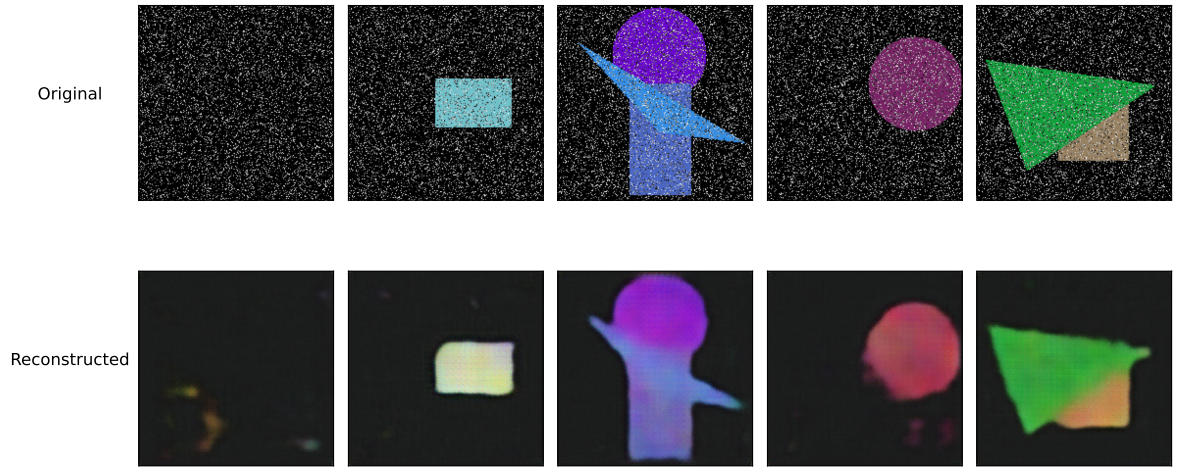


Figure B.2: Five random samples with their original image and the reconstructed image after passed through the trained autoencoder

C

Results for Additional Seeds

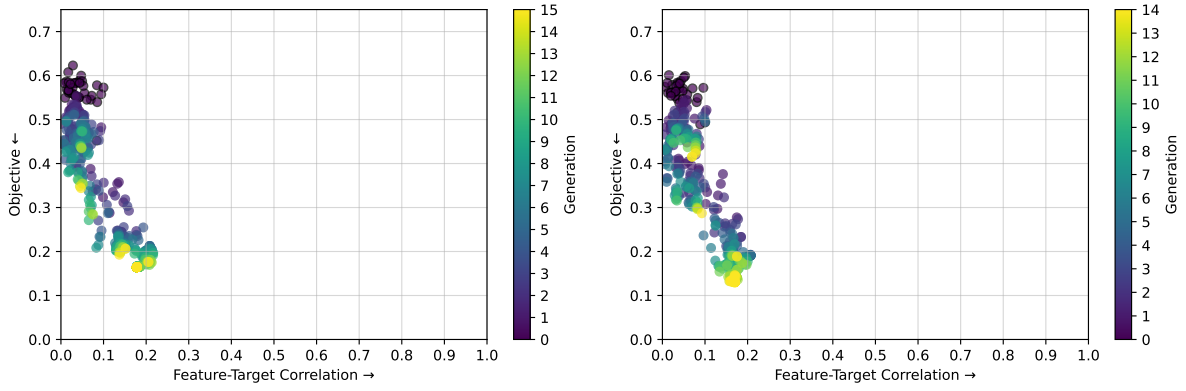
This appendix chapter gives the results of all seeds for which results were obtained for optimising the proxy objective through GOMEA. This was done in Chapter 7 and in Chapter 8. In Chapter 7 the three-Gated XOR with two tabular features, single XOR, and AND problems were optimised. In Chapter 8, three ablations were tested for, where first, the end-to-end training in the fitness evaluation is omitted, secondly standard DL parameters were used, and finally both ablations were applied.

For optimising the proxy objective through GOMEA, three different seeds need to be specified. First, the data split seed determines how train and test data is split, which is set to 0 throughout the entire thesis. Next, the downsample seed determines what training samples are used in BST. Finally, the train seed is used for training the DL blocks and for random number generation in GOMEA, which is kept the same for both applications. Four combinations of downsample and train seeds were used in obtaining results. This was done to get an idea of the distribution of results and not focusing on one lucky or unlucky set of seeds. Specifically, the following four downsample–train seed pairs were used: $(0, 0), (0, 1), (1, 0),$ and $(1, 1)$. First, the additional results for Chapter 7 are given and finally the additional results of Chapter 8 is given.

C.1. Optimising the Proxy Objective through GOMEA

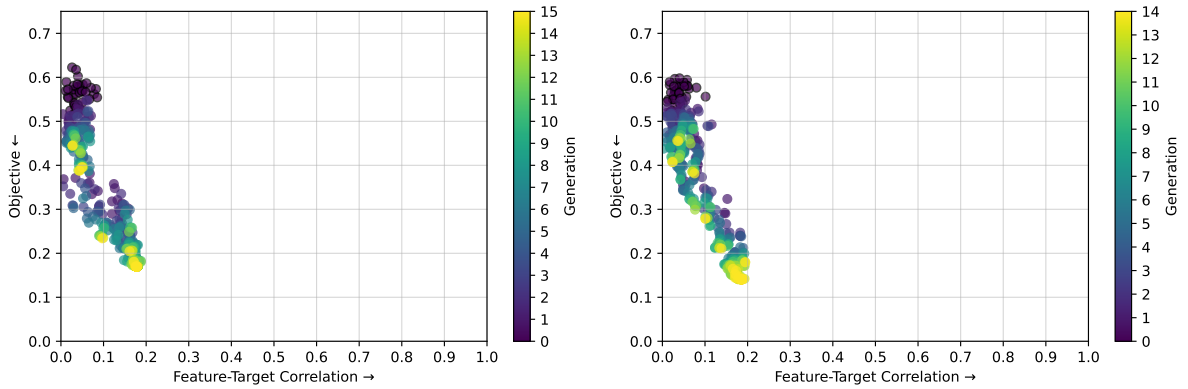
C.1.1. Three-Gated XOR with Two Tabular Features

Figure C.1, Figure C.2, Figure C.3, and Figure C.4 are the results of optimising the proxy objective for the three-gated XOR task with two tabular features, as discussed in Section 7.3.1 for all tested seed combinations.



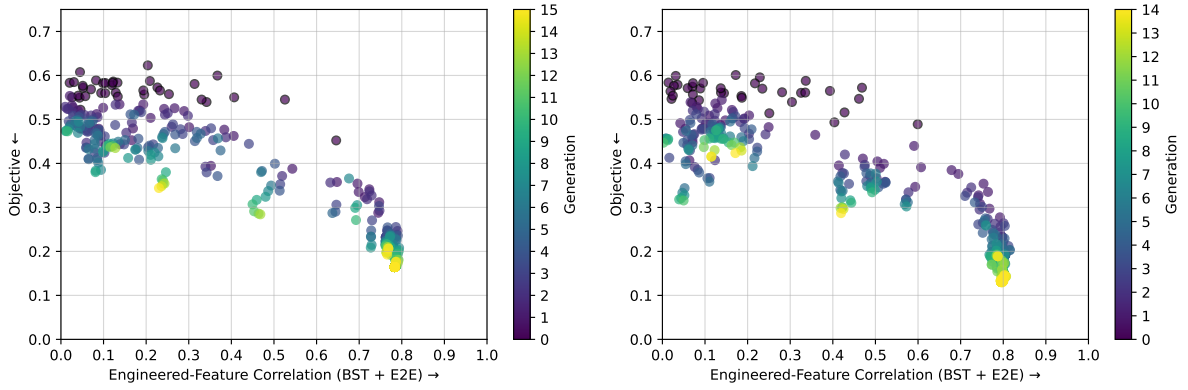
(a) Downsample seed: 0; Train seed: 0

(b) Downsample seed: 0; Train seed: 1

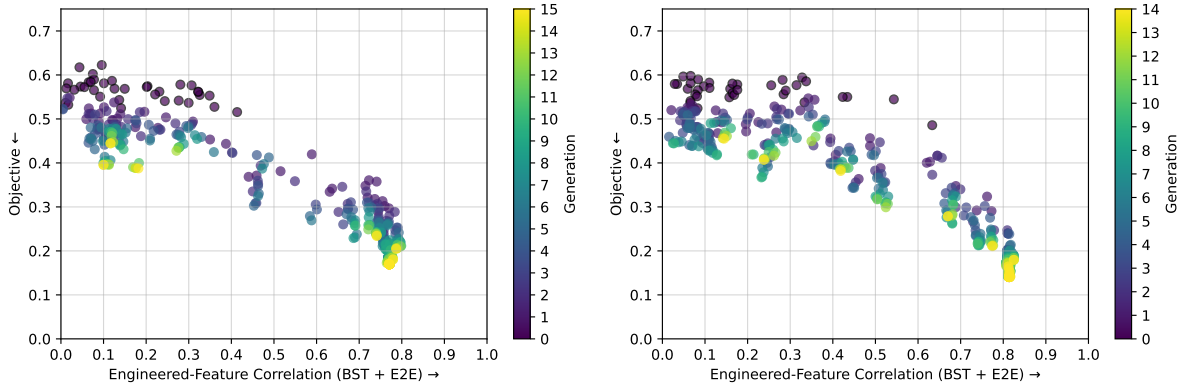


(c) Downsample seed: 1; Train seed: 0 **(d)** Downsample seed: 1; Train seed: 1

Figure C.1: Each dot represents a solution found by GOMEA for the three-gated XOR task with two tabular features, with the objective value plotted against the feature–target correlation. Each subplot corresponds to one run with a specific downsample seed and training seed, with a maximum runtime of two weeks per run. The dot colour indicates the generation of the solution, with generation 0 denoting the initial population, highlighted by a black outline. Arrows denote whether the corresponding metric should be maximised or minimised.

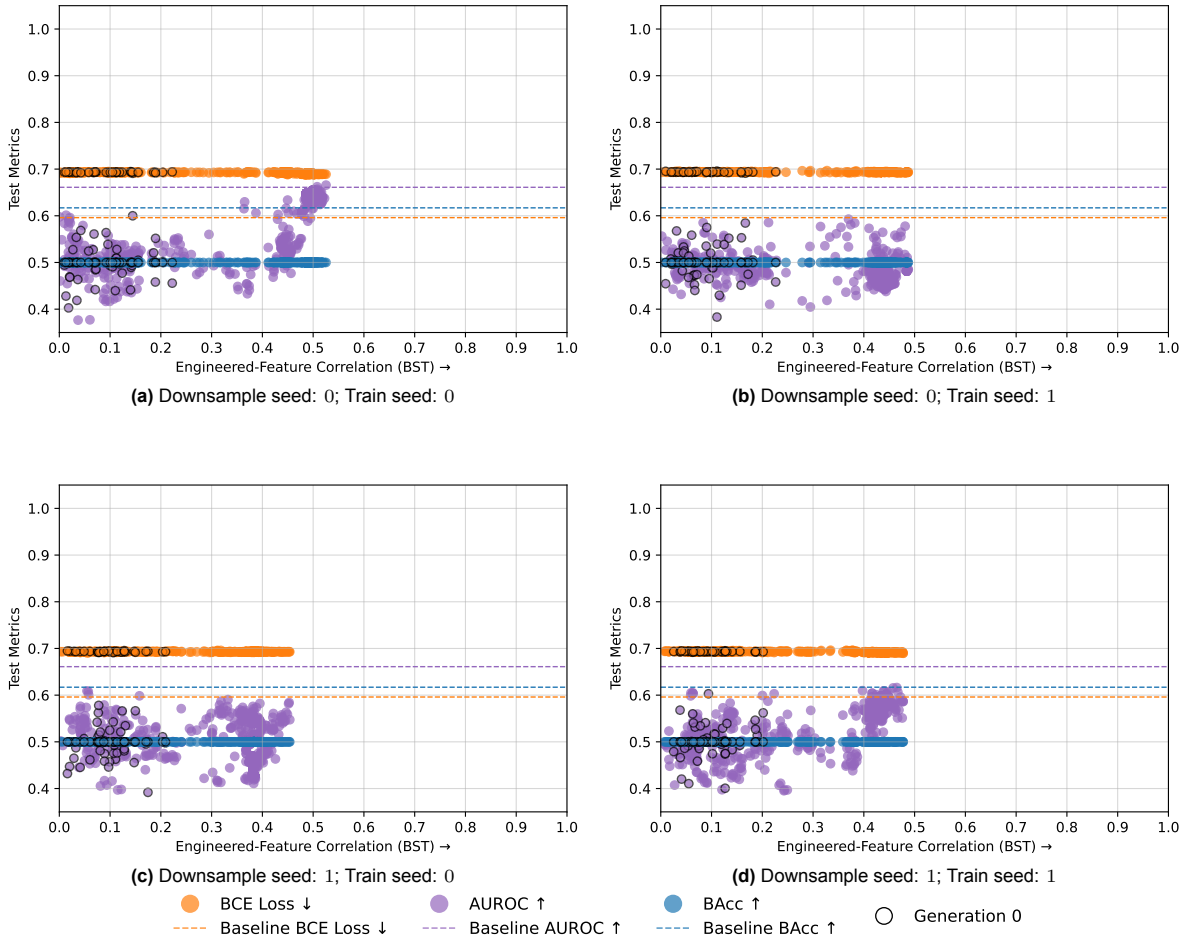


(a) Downsample seed: 0; Train seed: 0 **(b)** Downsample seed: 0; Train seed: 1



(c) Downsample seed: 1; Train seed: 0 **(d)** Downsample seed: 1; Train seed: 1

Figure C.2: Each dot represents a solution found by GOMEA for the three-gated XOR task with two tabular features, with the objective value plotted against the engineered-feature correlation after Blockwise Supervised Training (BST) and End-to-End Training (E2E). Each subplot corresponds to one run with a specific downsample seed and training seed, with a maximum runtime of two weeks per run. The dot colour indicates the generation of the solution, with generation 0 denoting the initial population, highlighted by a black outline. Arrows denote whether the corresponding metric should be maximised or minimised.



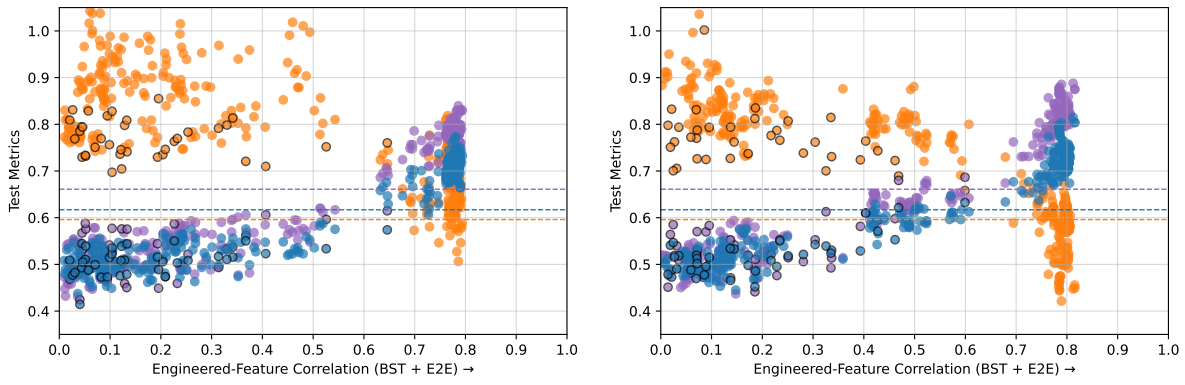
(a) Downsample seed: 0; Train seed: 0

(b) Downsample seed: 0; Train seed: 1

(c) Downsample seed: 1; Train seed: 0

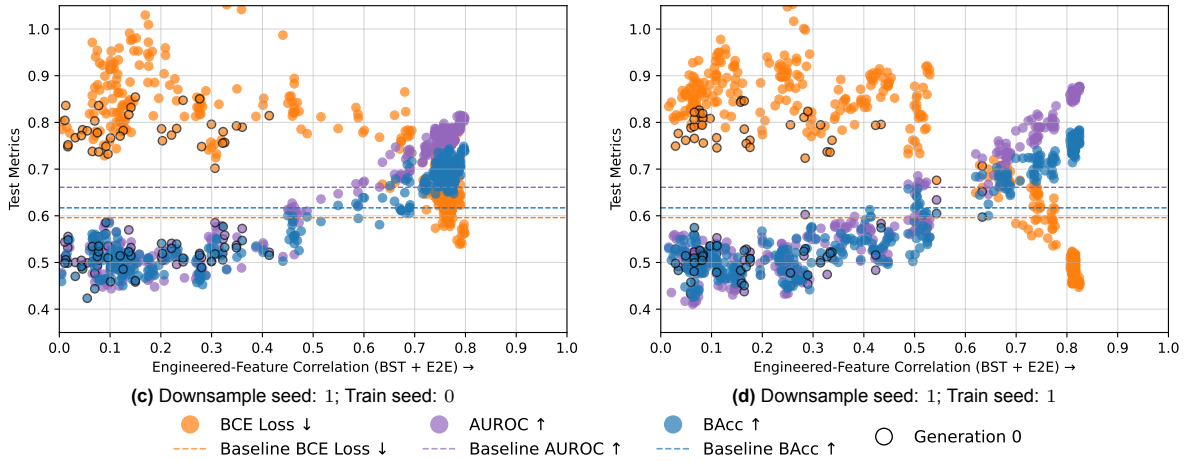
(d) Downsample seed: 1; Train seed: 1

Figure C.3: Each dot represents a test metric value for a solution found by GOMEA for the three-gated XOR task with two tabular features plotted against its engineered-feature correlation **before** end-to-end training. The test metrics include: Binary Cross Entropy Loss (BCE), Area Under the Receiver-Operating Characteristic curve (AUROC), and Balanced Accuracy (BAcc). The dotted horizontal lines show the baseline value of the test metric in the same colour. Each subplot corresponds to one run with a specific downsample seed and training seed, with a maximum runtime of two weeks per run. The dot colour indicates the generation of the solution, with generation 0 denoting the initial population, highlighted by a black outline. Arrows denote whether the corresponding metric should be maximised or minimised.



(a) Downsample seed: 0; Train seed: 0

(b) Downsample seed: 0; Train seed: 1



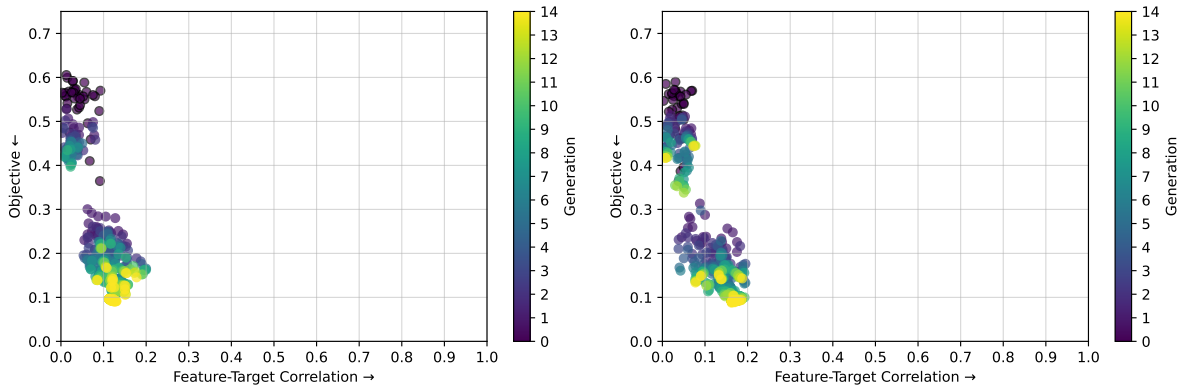
(c) Downsample seed: 1; Train seed: 0

(d) Downsample seed: 1; Train seed: 1

Figure C.4: Each dot represents a test metric value for a solution found by GOMEA for the three-gated XOR task with two tabular features plotted against its engineered-feature correlation **after** end-to-end training. The test metrics include: Binary Cross Entropy Loss (BCE), Area Under the Receiver-Operating Characteristic curve (AUROC), and Balanced Accuracy (BAcc). The dotted horizontal lines show the baseline value of the test metric in the same colour. Each subplot corresponds to one run with a specific downsample seed and training seed, with a maximum runtime of two weeks per run. The dot colour indicates the generation of the solution, with generation 0 denoting the initial population, highlighted by a black outline. Arrows denote whether the corresponding metric should be maximised or minimised.

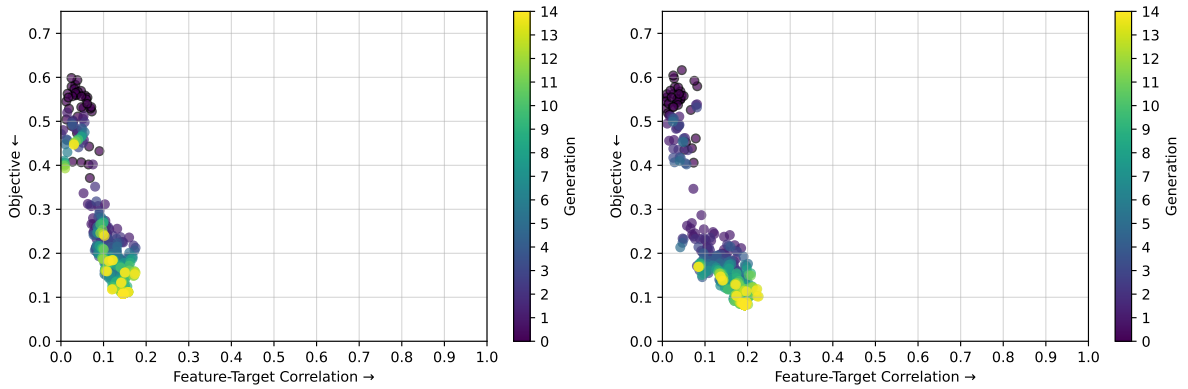
C.1.2. XOR

Figure C.5, Figure C.6, Figure C.7, and Figure C.8 are the results of optimising the proxy objective for the single XOR task, as discussed in Section 7.3.2 for all tested seed combinations.



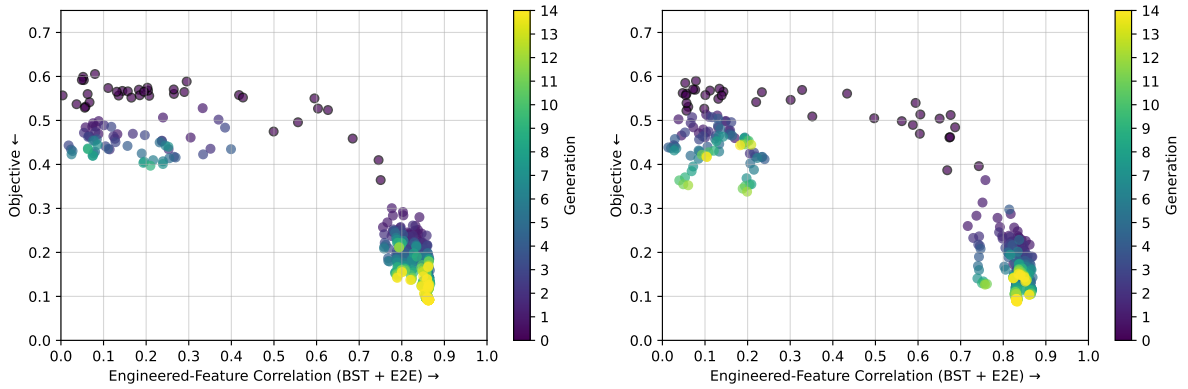
(a) Downsample seed: 0; Train seed: 0

(b) Downsample seed: 0; Train seed: 1

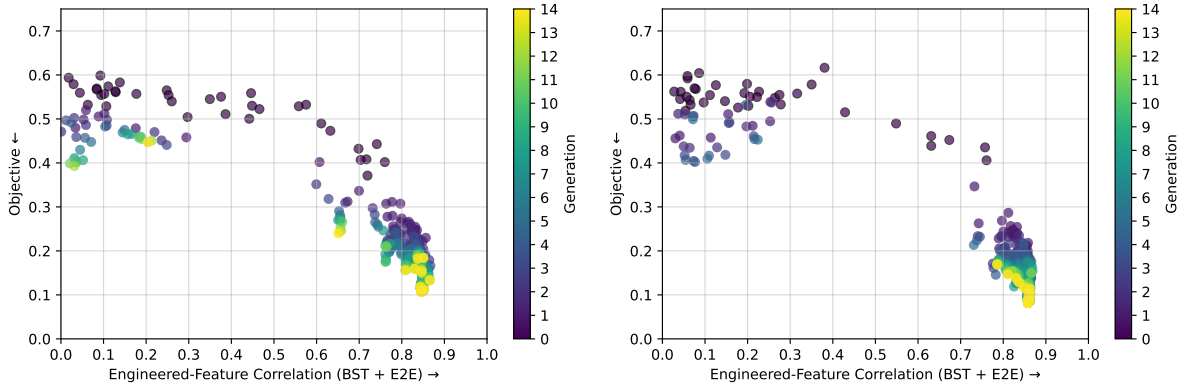


(c) Downsample seed: 1; Train seed: 0 **(d)** Downsample seed: 1; Train seed: 1

Figure C.5: Each dot represents a solution found by GOMEA for the single XOR task, with the objective value plotted against the feature–target correlation. Each subplot corresponds to one run with a specific downsample seed and training seed, with a maximum runtime of two weeks per run. The dot colour indicates the generation of the solution, with generation 0 denoting the initial population, highlighted by a black outline. Arrows denote whether the corresponding metric should be maximised or minimised.

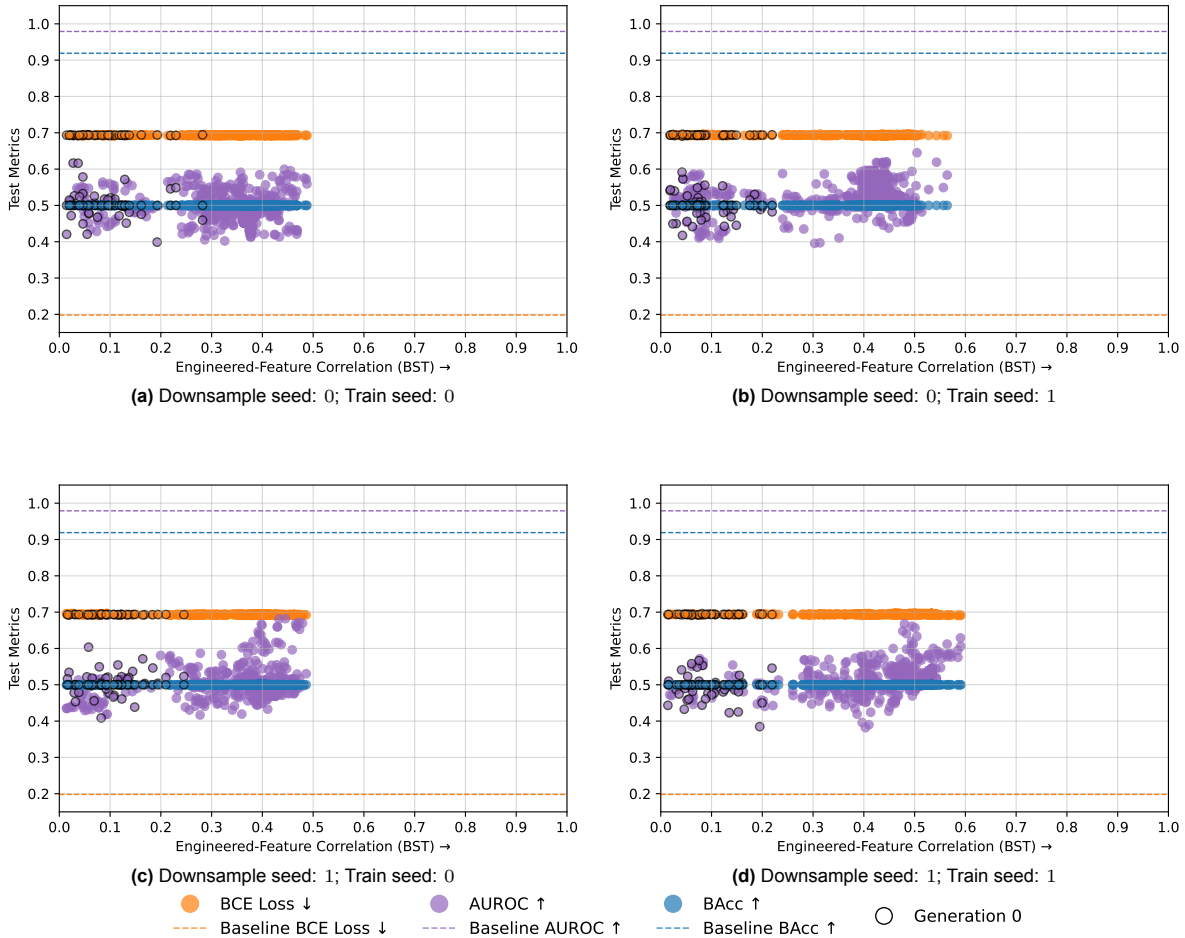


(a) Downsample seed: 0; Train seed: 0 **(b)** Downsample seed: 0; Train seed: 1



(c) Downsample seed: 1; Train seed: 0 **(d)** Downsample seed: 1; Train seed: 1

Figure C.6: Each dot represents a solution found by GOMEA for the single XOR task, with the objective value plotted against the engineered-feature correlation after Blockwise Supervised Training (BST) and End-to-End Training (E2E). Each subplot corresponds to one run with a specific downsample seed and training seed, with a maximum runtime of two weeks per run. The dot colour indicates the generation of the solution, with generation 0 denoting the initial population, highlighted by a black outline. Arrows denote whether the corresponding metric should be maximised or minimised.



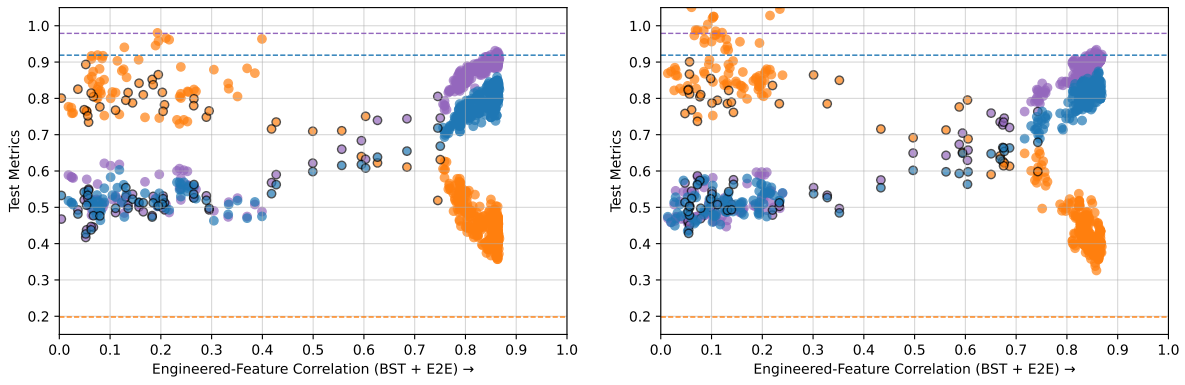
(a) Downsample seed: 0; Train seed: 0

(b) Downsample seed: 0; Train seed: 1

(c) Downsample seed: 1; Train seed: 0

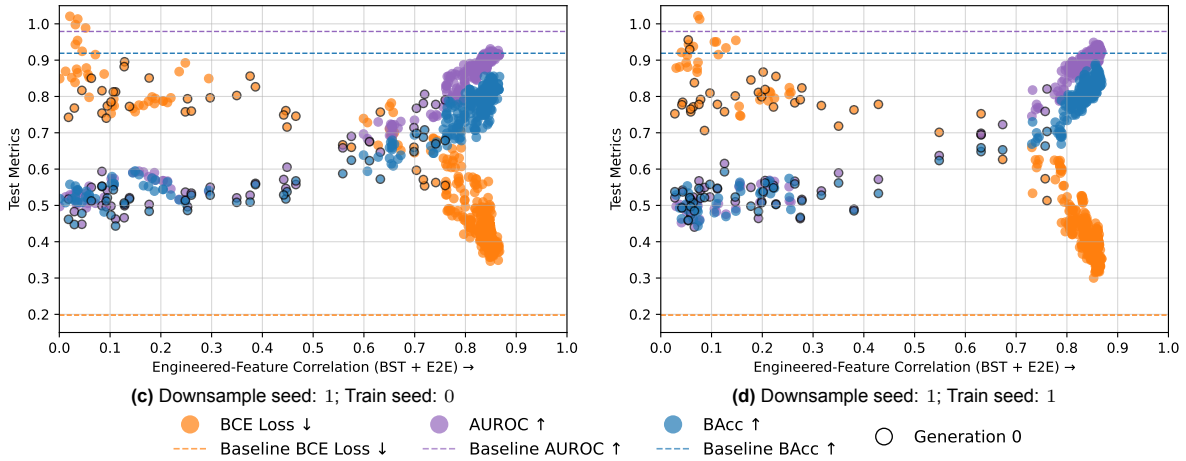
(d) Downsample seed: 1; Train seed: 1

Figure C.7: Each dot represents a test metric value for a solution found by GOMEA for the single XOR task plotted against its engineered-feature correlation **before** end-to-end training. The test metrics include: Binary Cross Entropy Loss (BCE), Area Under the Receiver-Operating Characteristic curve (AUROC), and Balanced Accuracy (BAcc). The dotted horizontal lines show the baseline value of the test metric in the same colour. Each subplot corresponds to one run with a specific downsample seed and training seed, with a maximum runtime of two weeks per run. The dot colour indicates the generation of the solution, with generation 0 denoting the initial population, highlighted by a black outline. Arrows denote whether the corresponding metric should be maximised or minimised.



(a) Downsample seed: 0; Train seed: 0

(b) Downsample seed: 0; Train seed: 1



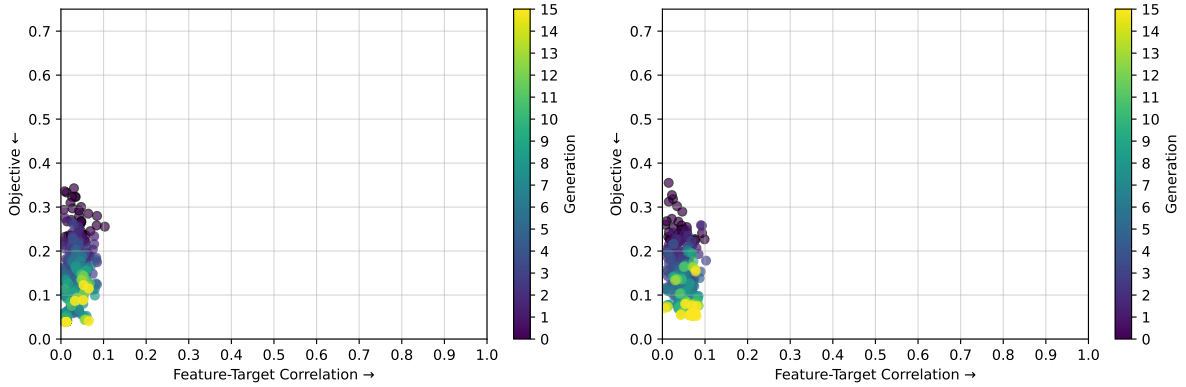
(c) Downsample seed: 1; Train seed: 0

(d) Downsample seed: 1; Train seed: 1

Figure C.8: Each dot represents a test metric value for a solution found by GOMEA for the single XOR task plotted against its engineered-feature correlation **after** end-to-end training. The test metrics include: Binary Cross Entropy Loss (BCE), Area Under the Receiver-Operating Characteristic curve (AUROC), and Balanced Accuracy (BAcc). The dotted horizontal lines show the baseline value of the test metric in the same colour. Each subplot corresponds to one run with a specific downsample seed and training seed, with a maximum runtime of two weeks per run. The dot colour indicates the generation of the solution, with generation 0 denoting the initial population, highlighted by a black outline. Arrows denote whether the corresponding metric should be maximised or minimised.

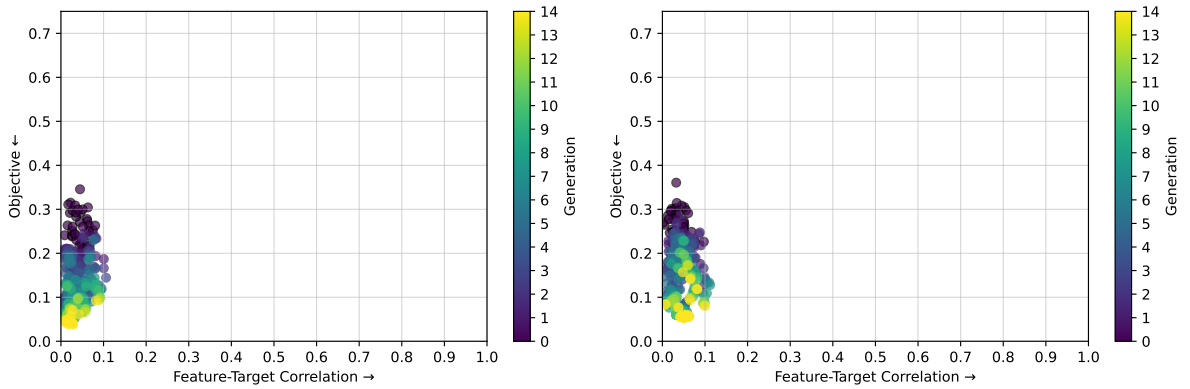
C.1.3. AND

Figure C.9, Figure C.10, Figure C.11, and Figure C.12 are the results of optimising the proxy objective for the AND problem, as discussed in Section 7.3.3 for all tested seed combinations.



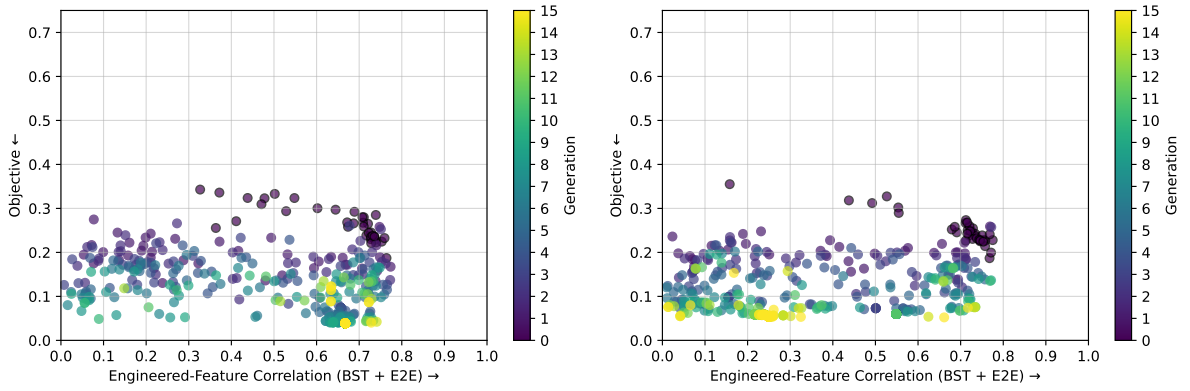
(a) Downsample seed: 0; Train seed: 0

(b) Downsample seed: 0; Train seed: 1

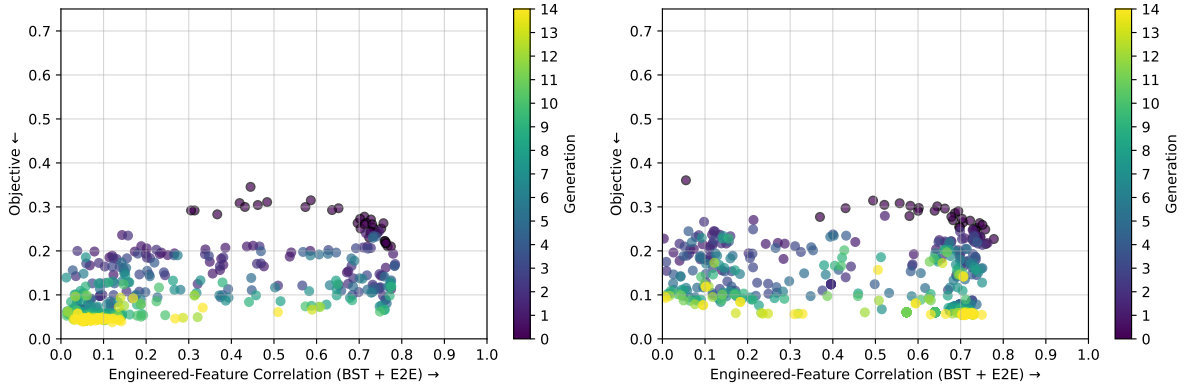


(c) Downsample seed: 1; Train seed: 0 **(d)** Downsample seed: 1; Train seed: 1

Figure C.9: Each dot represents a solution found by GOMEA for the AND task, with the objective value plotted against the feature–target correlation. Each subplot corresponds to one run with a specific downsample seed and training seed, with a maximum runtime of two weeks per run. The dot colour indicates the generation of the solution, with generation 0 denoting the initial population, highlighted by a black outline. Arrows denote whether the corresponding metric should be maximised or minimised.

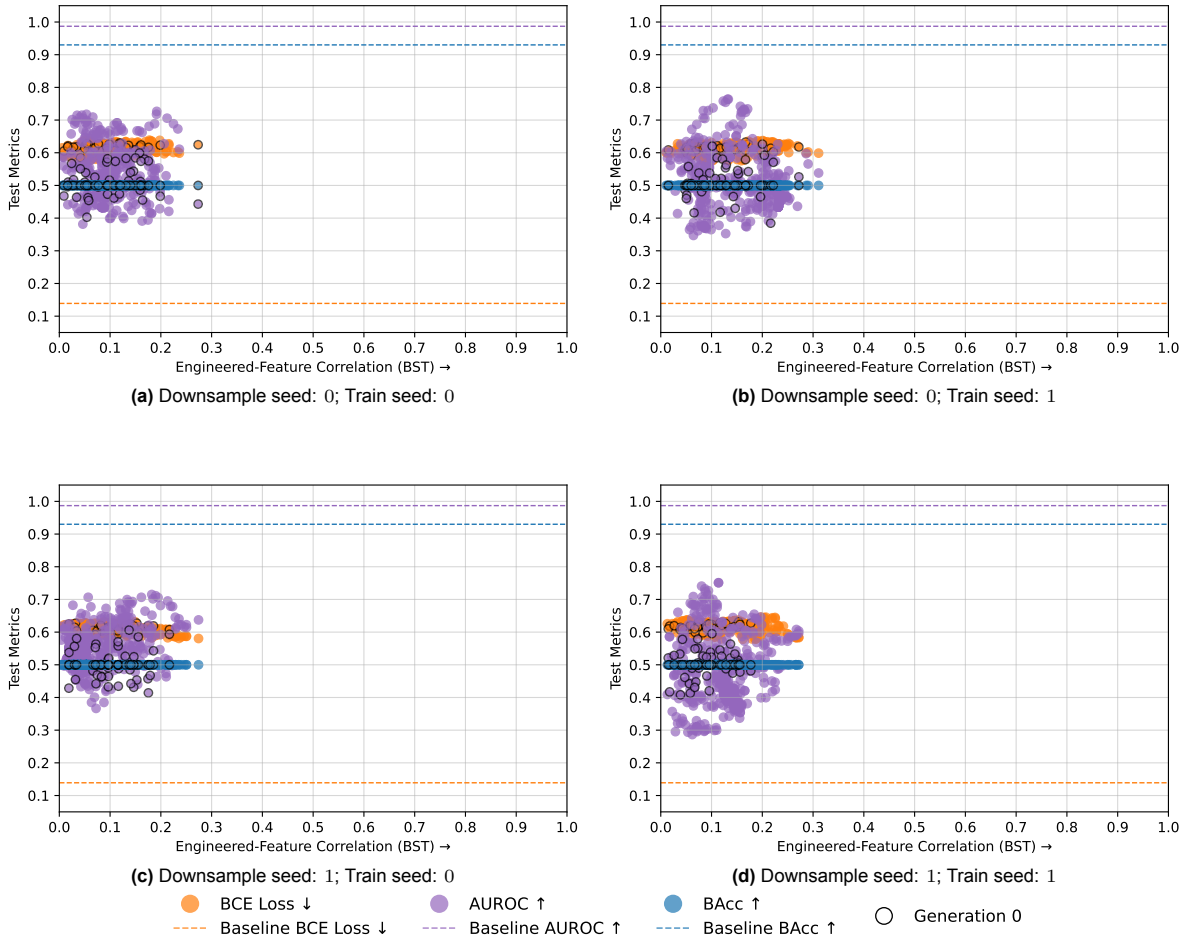


(a) Downsample seed: 0; Train seed: 0 **(b)** Downsample seed: 0; Train seed: 1



(c) Downsample seed: 1; Train seed: 0 **(d)** Downsample seed: 1; Train seed: 1

Figure C.10: Each dot represents a solution found by GOMEA for the AND task, with the objective value plotted against the engineered-feature correlation after Blockwise Supervised Training (BST) and End-to-End Training (E2E). Each subplot corresponds to one run with a specific downsample seed and training seed, with a maximum runtime of two weeks per run. The dot colour indicates the generation of the solution, with generation 0 denoting the initial population, highlighted by a black outline. Arrows denote whether the corresponding metric should be maximised or minimised.



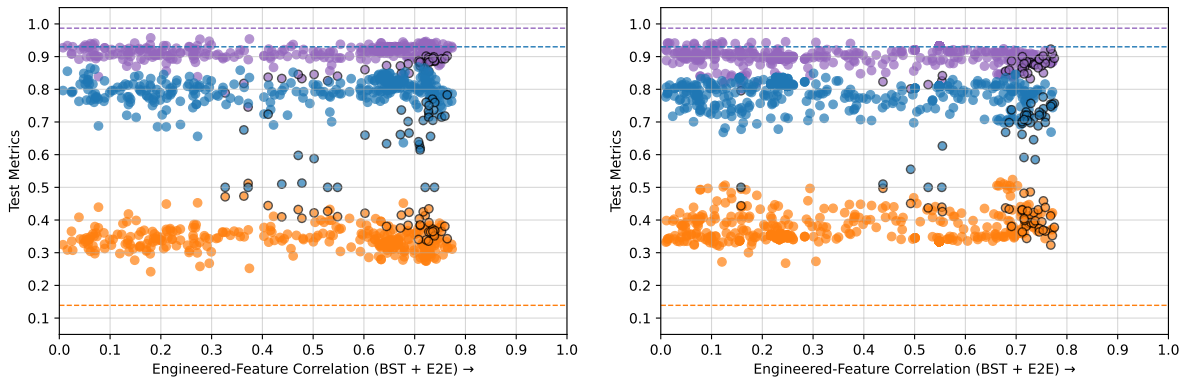
(a) Downsample seed: 0; Train seed: 0

(b) Downsample seed: 0; Train seed: 1

(c) Downsample seed: 1; Train seed: 0

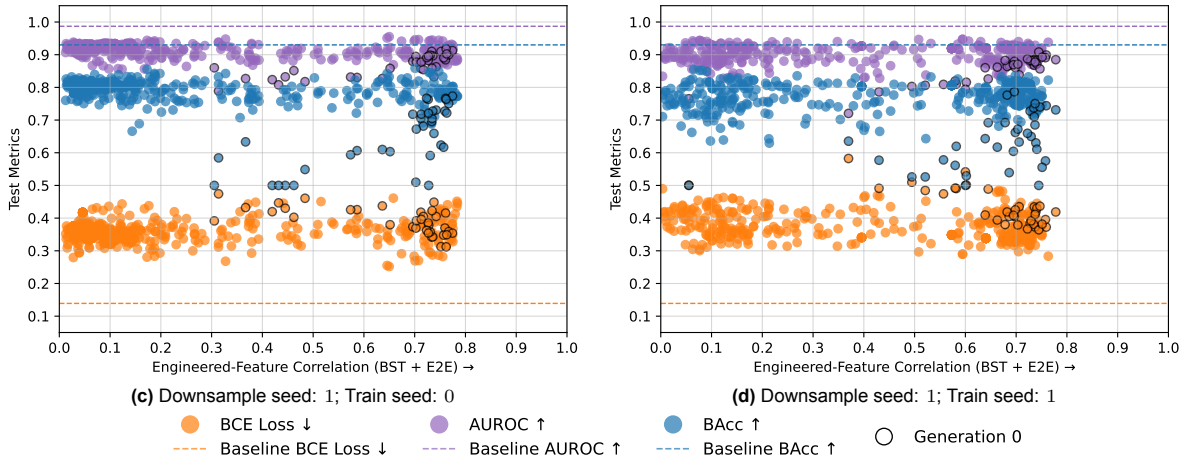
(d) Downsample seed: 1; Train seed: 1

Figure C.11: Each dot represents a test metric value for a solution found by GOMEA for the AND task plotted against its engineered-feature correlation **before** end-to-end training. The test metrics include: Binary Cross Entropy Loss (BCE), Area Under the Receiver-Operating Characteristic curve (AUROC), and Balanced Accuracy (BAcc). The dotted horizontal lines show the baseline value of the test metric in the same colour. Each subplot corresponds to one run with a specific downsample seed and training seed, with a maximum runtime of two weeks per run. The dot colour indicates the generation of the solution, with generation 0 denoting the initial population, highlighted by a black outline. Arrows denote whether the corresponding metric should be maximised or minimised.



(a) Downsample seed: 0; Train seed: 0

(b) Downsample seed: 0; Train seed: 1



(c) Downsample seed: 1; Train seed: 0

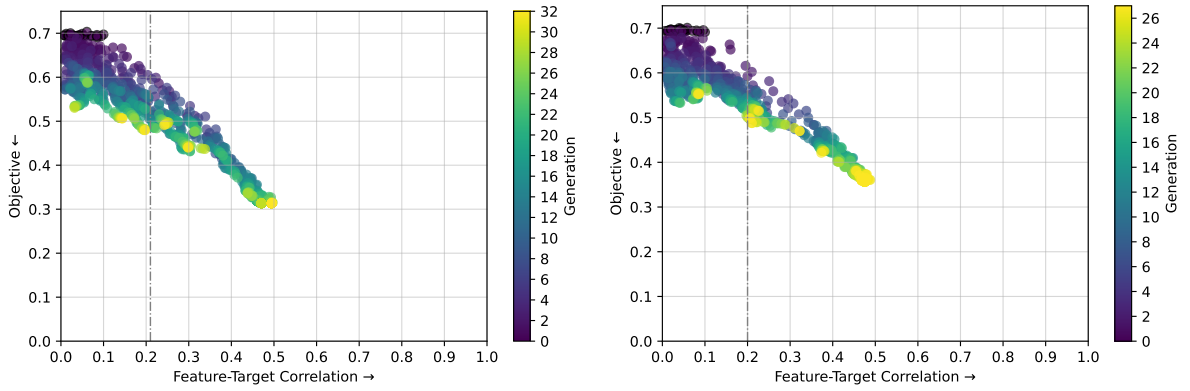
(d) Downsample seed: 1; Train seed: 1

Figure C.12: Each dot represents a test metric value for a solution found by GOMEA for the AND task plotted against its engineered-feature correlation **after** end-to-end training. The test metrics include: Binary Cross Entropy Loss (BCE), Area Under the Receiver-Operating Characteristic curve (AUROC), and Balanced Accuracy (BAcc). The dotted horizontal lines show the baseline value of the test metric in the same colour. Each subplot corresponds to one run with a specific downsample seed and training seed, with a maximum runtime of two weeks per run. The dot colour indicates the generation of the solution, with generation 0 denoting the initial population, highlighted by a black outline. Arrows denote whether the corresponding metric should be maximised or minimised.

C.2. Ablation Study

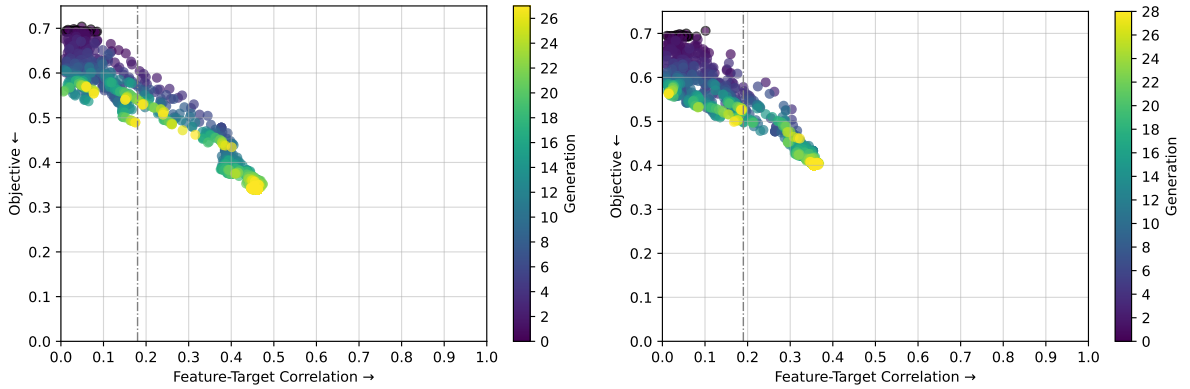
C.2.1. Without End-to-End Training in Proxy Objective

Figure C.13, Figure C.14, Figure C.15, and Figure C.16 are the results of optimising the proxy objective for the three-gated XOR task with two tabular features with no end-to-end training in the fitness evaluation, as discussed in Section 8.2, for all tested seed combinations.



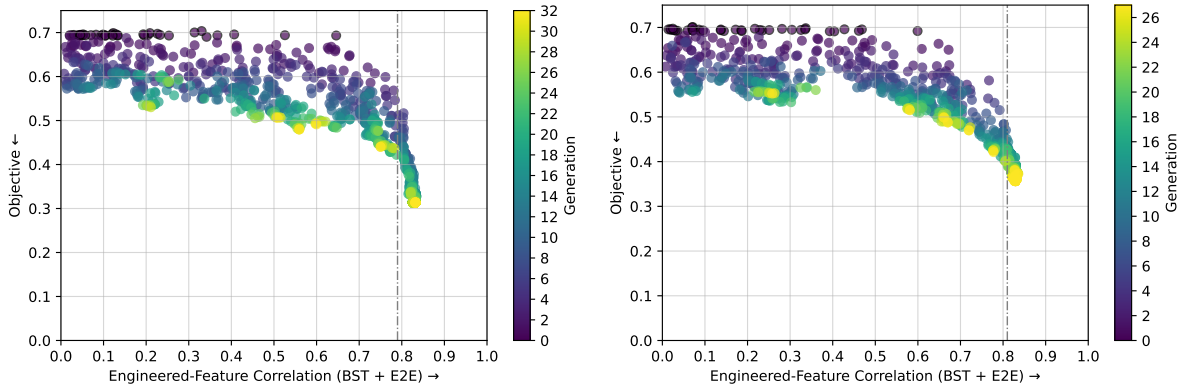
(a) Downsample seed: 0; Train seed: 0

(b) Downsample seed: 0; Train seed: 1

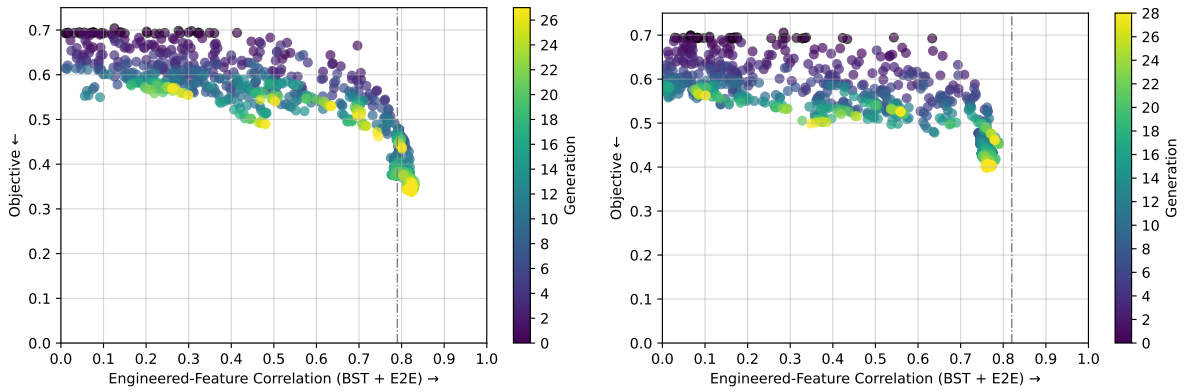


(c) Downsample seed: 1; Train seed: 0 **(d)** Downsample seed: 1; Train seed: 1

Figure C.13: Each dot represents a solution found by GOMEA for the three-gated XOR task with two tabular features, without end-to-end training in the fitness evaluation, where the objective value is plotted against the feature–target correlation. Each subplot corresponds to one run with a specific downsample seed and training seed, with a maximum runtime of two weeks per run. The dot colour indicates the generation of the solution, with generation 0 denoting the initial population, highlighted by a black outline. Arrows denote whether the corresponding metric should be maximised or minimised.

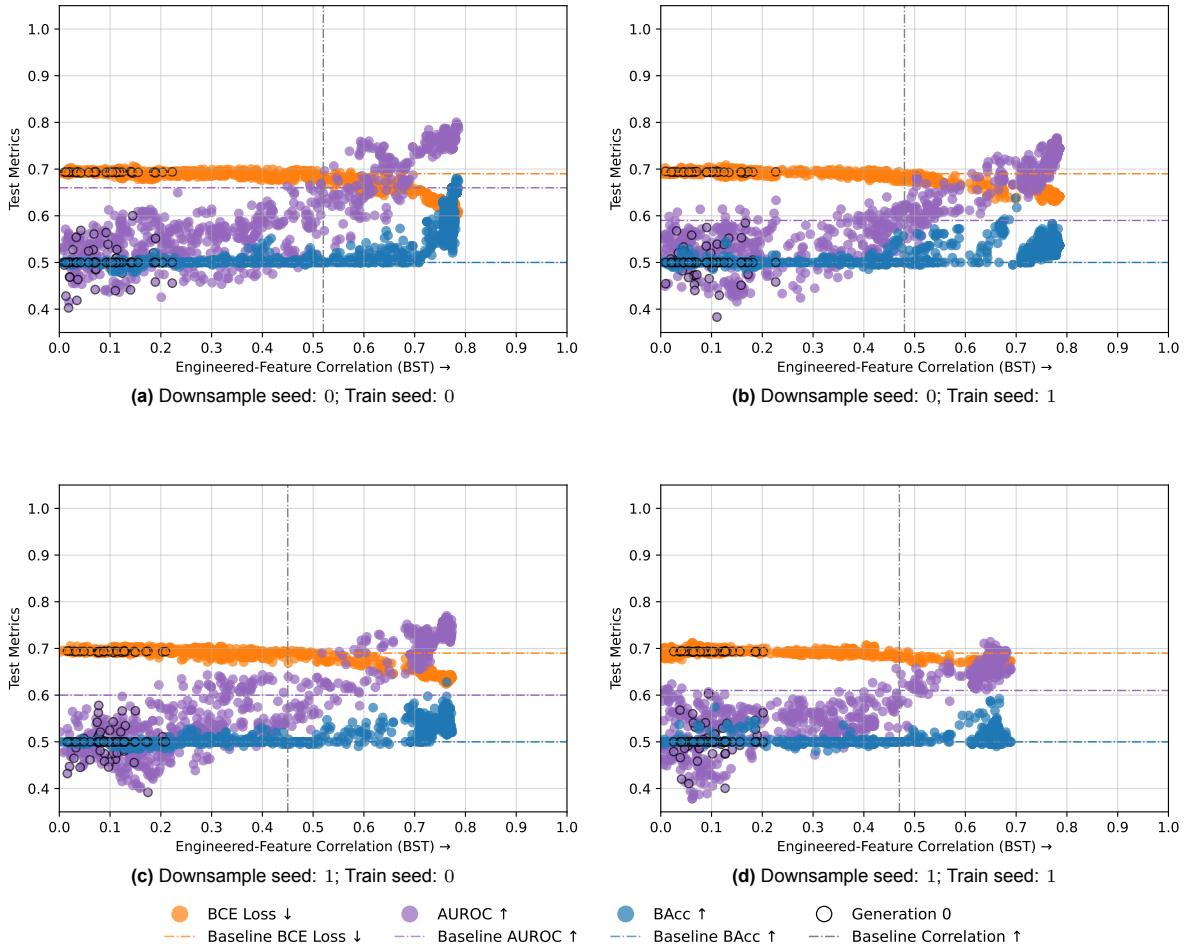


(a) Downsample seed: 0; Train seed: 0 **(b)** Downsample seed: 0; Train seed: 1



(c) Downsample seed: 1; Train seed: 0 **(d)** Downsample seed: 1; Train seed: 1

Figure C.14: Each dot represents a solution found by GOMEA for the three-gated XOR task with two tabular features, without end-to-end training in the fitness evaluation, where the objective value is plotted against the engineered-feature correlation after Blockwise Supervised Training (BST) and End-to-End Training (E2E). Each subplot corresponds to one run with a specific downsample seed and training seed, with a maximum runtime of two weeks per run. The dot colour indicates the generation of the solution, with generation 0 denoting the initial population, highlighted by a black outline. Arrows denote whether the corresponding metric should be maximised or minimised.



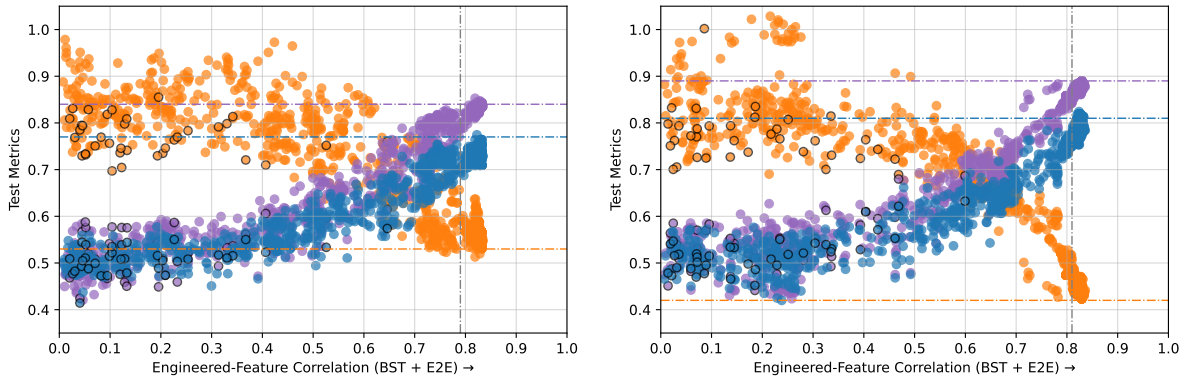
(a) Downsample seed: 0; Train seed: 0

(b) Downsample seed: 0; Train seed: 1

(c) Downsample seed: 1; Train seed: 0

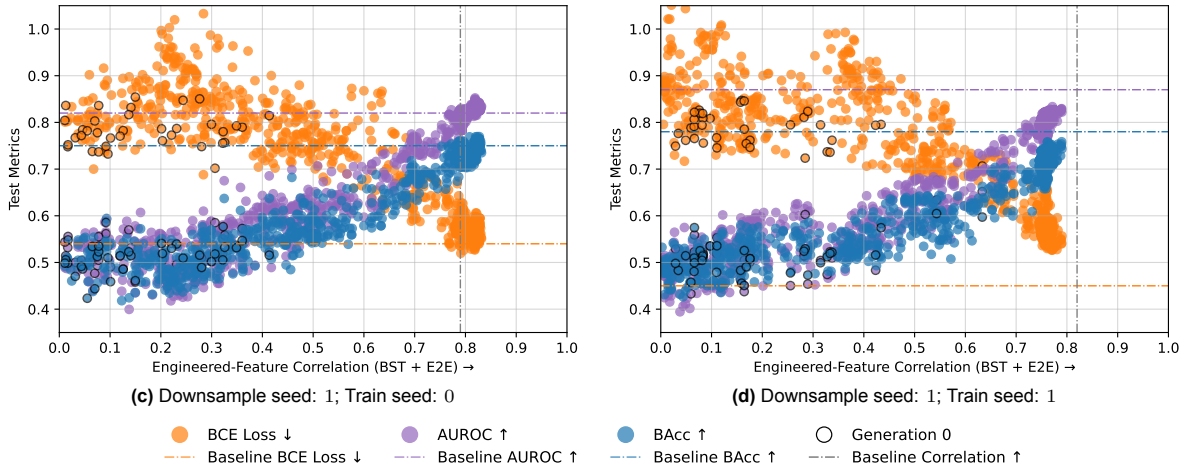
(d) Downsample seed: 1; Train seed: 1

Figure C.15: Each dot represents a test metric value for a solution found by GOMEA for the three-gated XOR task with two tabular features, without end-to-end training in the fitness evaluation, plotted against its engineered-feature correlation **before** end-to-end training. The test metrics include: Binary Cross Entropy Loss (BCE), Area Under the Receiver-Operating Characteristic curve (AUROC), and Balanced Accuracy (BAcc). The dash-dotted horizontal lines show the baseline values of the values obtained with no ablations, as shown in Figure C.4. Each subplot corresponds to one run with a specific downsample seed and training seed, with a maximum runtime of two weeks per run. The dot colour indicates the generation of the solution, with generation 0 denoting the initial population, highlighted by a black outline. Arrows denote whether the corresponding metric should be maximised or minimised.



(a) Downsample seed: 0; Train seed: 0

(b) Downsample seed: 0; Train seed: 1



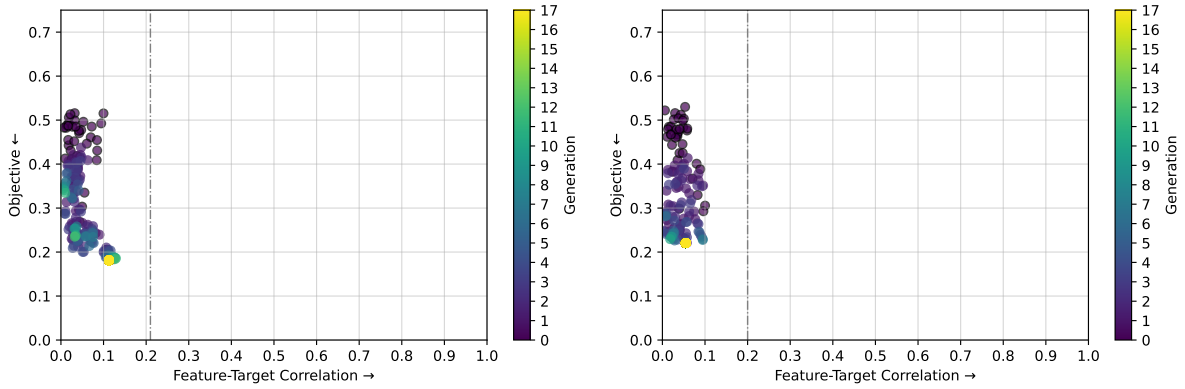
(c) Downsample seed: 1; Train seed: 0

(d) Downsample seed: 1; Train seed: 1

Figure C.16: Each dot represents a test metric value for a solution found by GOMEA for the three-gated XOR task with two tabular features, without end-to-end training in the fitness evaluation, plotted against its engineered-feature correlation **after** end-to-end training. The test metrics include: Binary Cross Entropy Loss (BCE), Area Under the Receiver-Operating Characteristic curve (AUROC), and Balanced Accuracy (BAcc). The dash-dotted horizontal lines show the baseline values of the values obtained with no ablations, as shown in Figure C.4. Each subplot corresponds to one run with a specific downsample seed and training seed, with a maximum runtime of two weeks per run. The dot colour indicates the generation of the solution, with generation 0 denoting the initial population, highlighted by a black outline. Arrows denote whether the corresponding metric should be maximised or minimised.

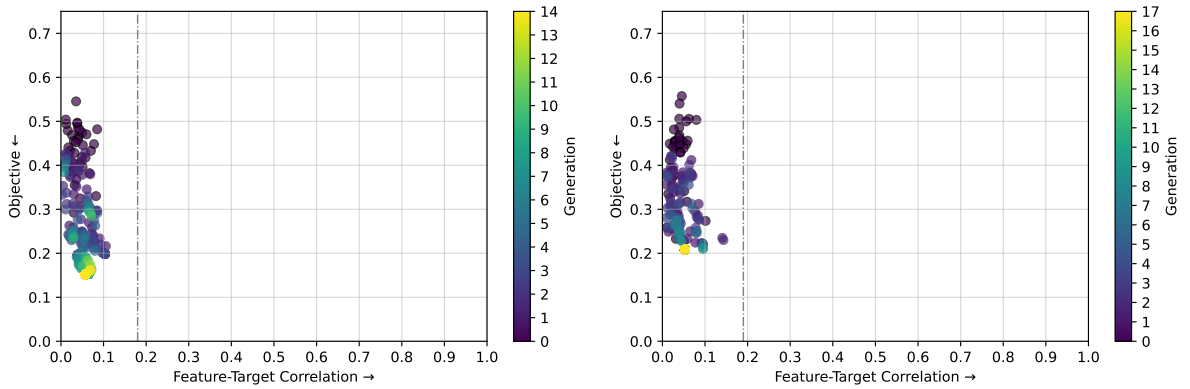
C.2.2. Standard DL Parameters

Figure C.17, Figure C.18, Figure C.19, and Figure C.20 are the results of optimising the proxy objective for the three-gated XOR task with two tabular features, with standard DL parameters, as discussed in Section 8.3, for all tested seed combinations.



(a) Downsample seed: 0; Train seed: 0

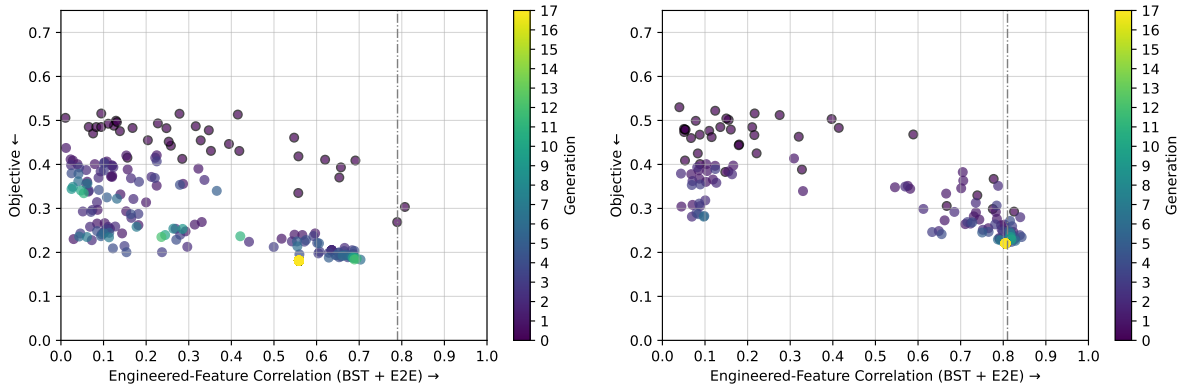
(b) Downsample seed: 0; Train seed: 1



(c) Downsample seed: 1; Train seed: 0

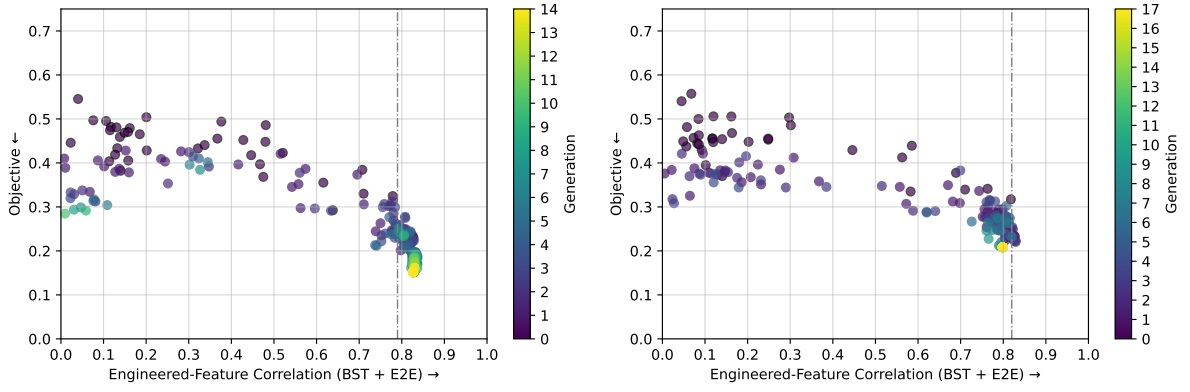
(d) Downsample seed: 1; Train seed: 1

Figure C.17: Each dot represents a solution found by GOMEA for the three-gated XOR task with two tabular features, with standard DL parameters, where the objective value is plotted against the feature–target correlation. Each subplot corresponds to one run with a specific downsample seed and training seed, with a maximum runtime of two weeks per run. The dot colour indicates the generation of the solution, with generation 0 denoting the initial population, highlighted by a black outline. Arrows denote whether the corresponding metric should be maximised or minimised.



(a) Downsample seed: 0; Train seed: 0

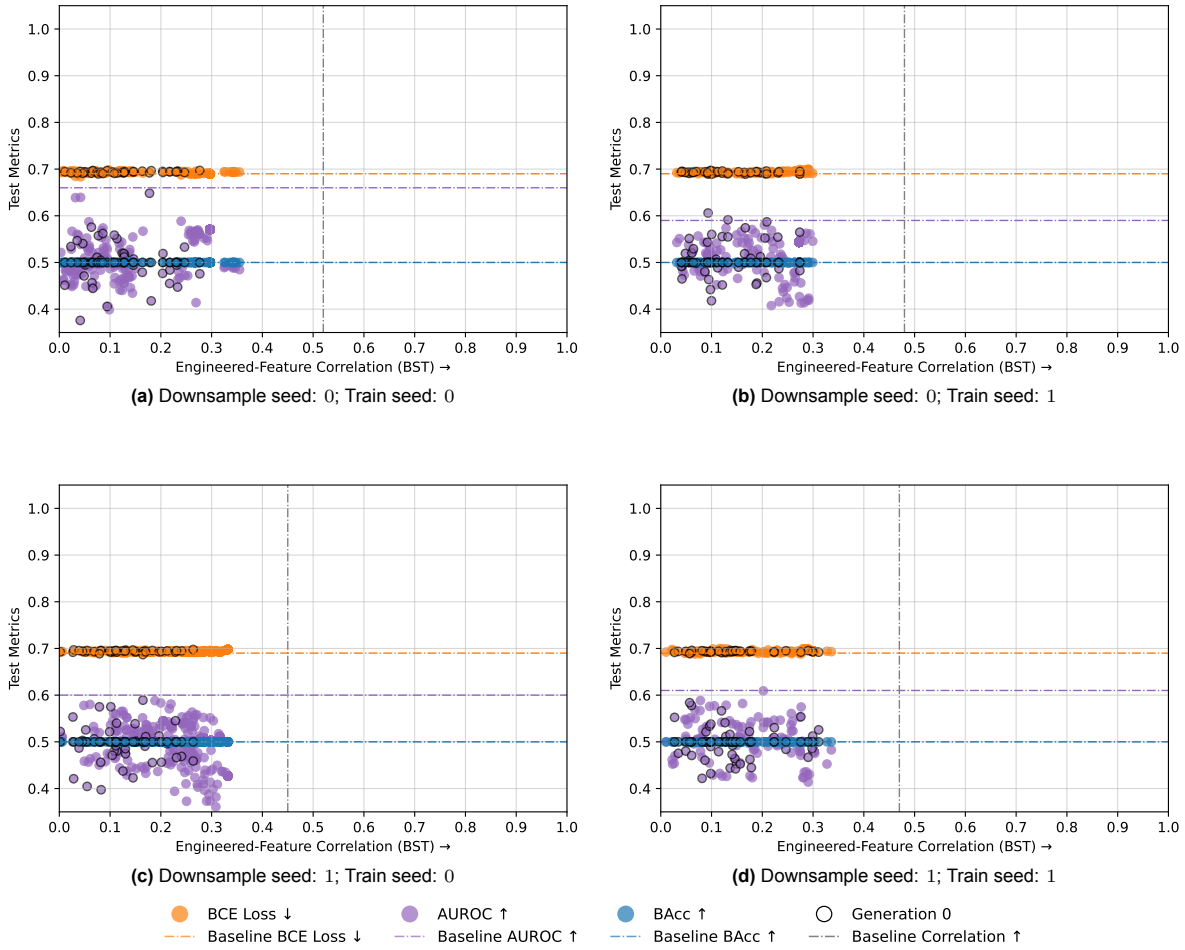
(b) Downsample seed: 0; Train seed: 1



(c) Downsample seed: 1; Train seed: 0

(d) Downsample seed: 1; Train seed: 1

Figure C.18: Each dot represents a solution found by GOMEA for the three-gated XOR task with two tabular features, with standard DL parameters, where the objective value is plotted against the engineered-feature correlation after Blockwise Supervised Training (BST) and End-to-End Training (E2E). Each subplot corresponds to one run with a specific downsample seed and training seed, with a maximum runtime of two weeks per run. The dot colour indicates the generation of the solution, with generation 0 denoting the initial population, highlighted by a black outline. Arrows denote whether the corresponding metric should be maximised or minimised.



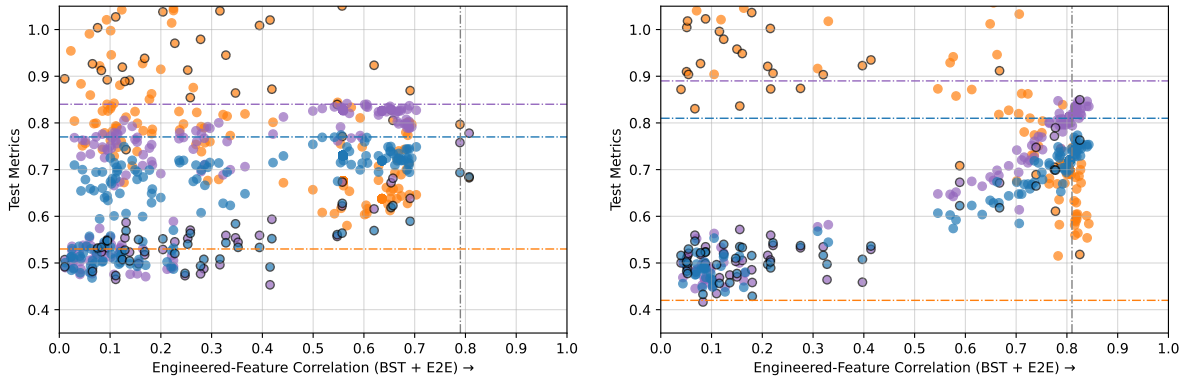
(a) Downsample seed: 0; Train seed: 0

(b) Downsample seed: 0; Train seed: 1

(c) Downsample seed: 1; Train seed: 0

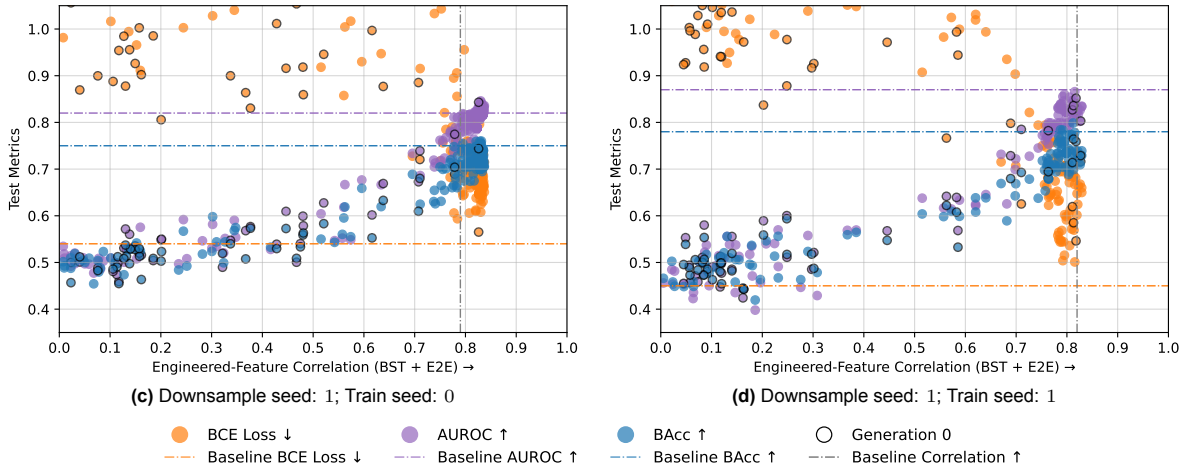
(d) Downsample seed: 1; Train seed: 1

Figure C.19: Each dot represents a test metric value for a solution found by GOMEA for the three-gated XOR task with two tabular features, with standard DL parameters, plotted against its engineered-feature correlation **before** end-to-end training. The test metrics include: Binary Cross Entropy Loss (BCE), Area Under the Receiver-Operating Characteristic curve (AUROC), and Balanced Accuracy (BAcc). The dash-dotted horizontal lines show the baseline values of the values obtained with no ablations, as shown in Figure C.4. Each subplot corresponds to one run with a specific downsample seed and training seed, with a maximum runtime of two weeks per run. The dot colour indicates the generation of the solution, with generation 0 denoting the initial population, highlighted by a black outline. Arrows denote whether the corresponding metric should be maximised or minimised.



(a) Downsample seed: 0; Train seed: 0

(b) Downsample seed: 0; Train seed: 1



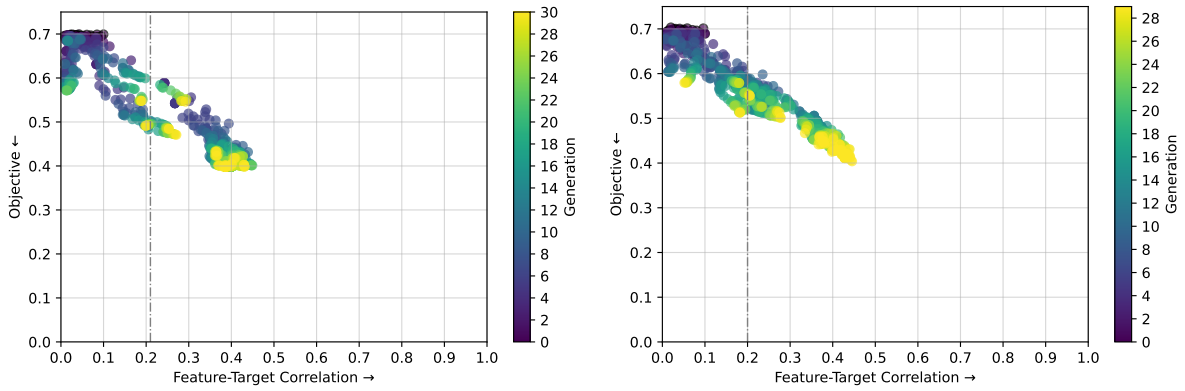
(c) Downsample seed: 1; Train seed: 0

(d) Downsample seed: 1; Train seed: 1

Figure C.20: Each dot represents a test metric value for a solution found by GOMEA for the three-gated XOR task with two tabular features, with standard DL parameters, plotted against its engineered-feature correlation **after** end-to-end training. The test metrics include: Binary Cross Entropy Loss (BCE), Area Under the Receiver-Operating Characteristic curve (AUROC), and Balanced Accuracy (BAcc). The dash-dotted horizontal lines show the baseline values of the values obtained with no ablations, as shown in Figure C.4. Each subplot corresponds to one run with a specific downsample seed and training seed, with a maximum runtime of two weeks per run. The dot colour indicates the generation of the solution, with generation 0 denoting the initial population, highlighted by a black outline. Arrows denote whether the corresponding metric should be maximised or minimised.

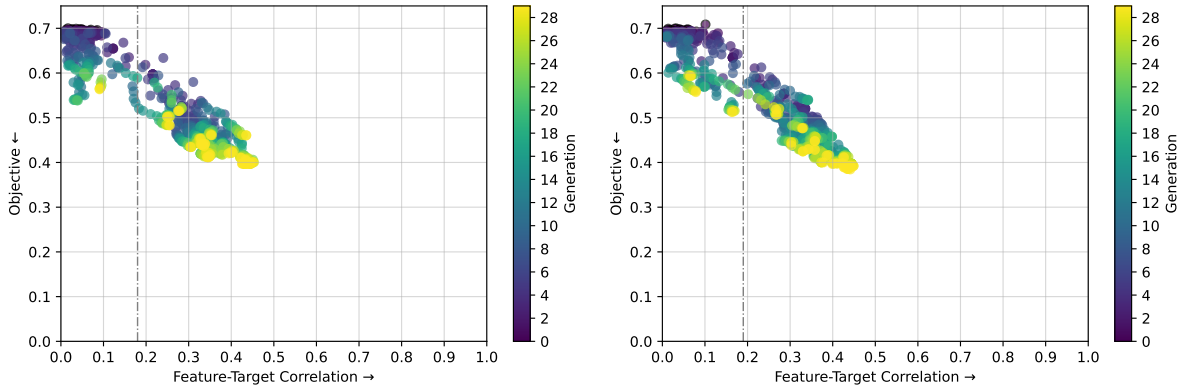
C.2.3. Standard DL Parameters and Without End-to-End Training

Figure C.21, Figure C.22, Figure C.23, and Figure C.24 are the results of optimising the proxy objective for the three-gated XOR task with two tabular features, with no end-to-end training in the fitness evaluation and using standard DL parameters, as discussed in Section 8.4, for all tested seed combinations.



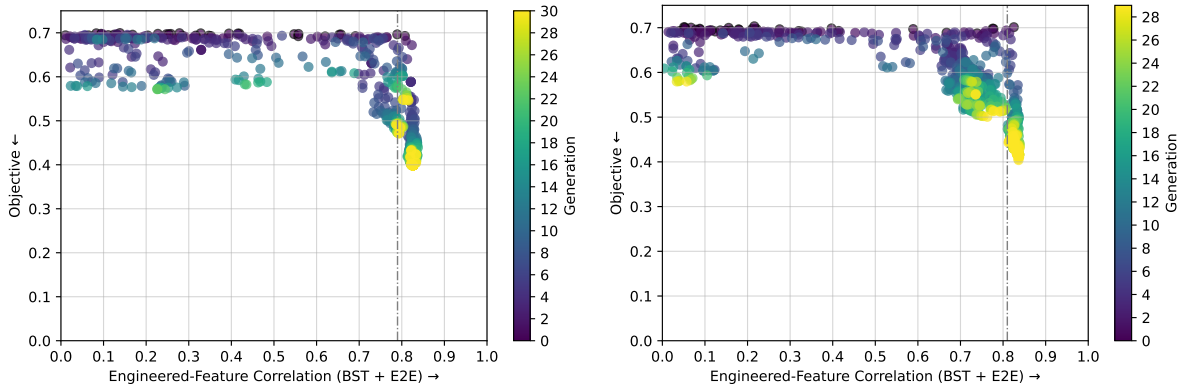
(a) Downsample seed: 0; Train seed: 0

(b) Downsample seed: 0; Train seed: 1

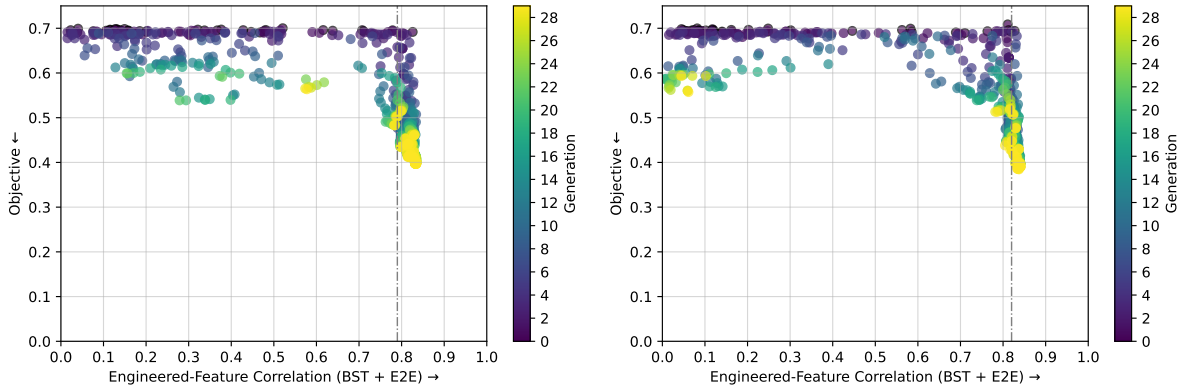


(c) Downsample seed: 1; Train seed: 0 **(d)** Downsample seed: 1; Train seed: 1

Figure C.21: Each dot represents a solution found by GOMEA for the three-gated XOR task with two tabular features, with standard DL parameters and without end-to-end training in the fitness evaluation, where the objective value is plotted against the feature–target correlation. Each subplot corresponds to one run with a specific downsample seed and training seed, with a maximum runtime of two weeks per run. The dot colour indicates the generation of the solution, with generation 0 denoting the initial population, highlighted by a black outline. Arrows denote whether the corresponding metric should be maximised or minimised.

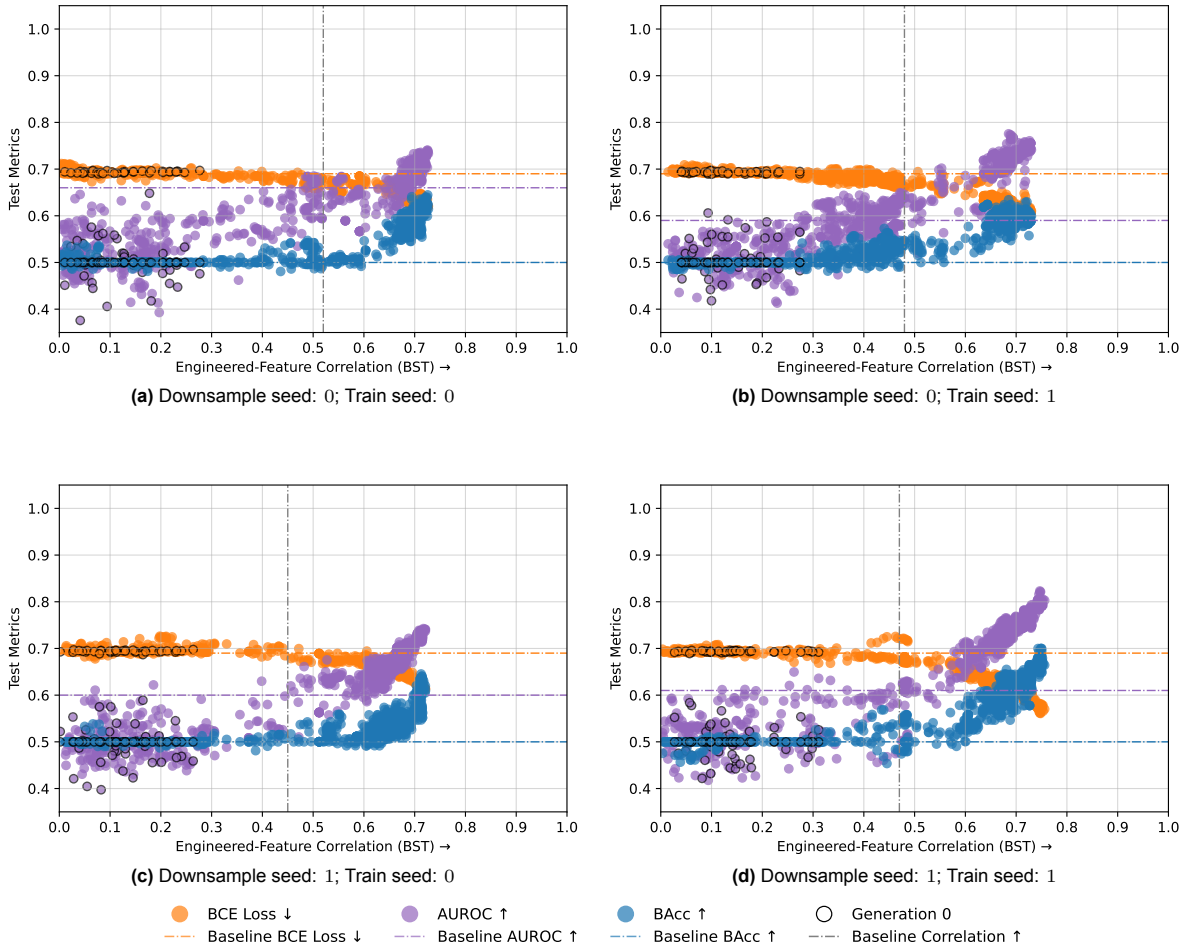


(a) Downsample seed: 0; Train seed: 0 **(b)** Downsample seed: 0; Train seed: 1



(c) Downsample seed: 1; Train seed: 0 **(d)** Downsample seed: 1; Train seed: 1

Figure C.22: Each dot represents a solution found by GOMEA for the three-gated XOR task with two tabular features, with standard DL parameters and without end-to-end training in the fitness evaluation, where the objective value is plotted against the engineered-feature correlation after Blockwise Supervised Training (BST) and End-to-End Training (E2E). Each subplot corresponds to one run with a specific downsample seed and training seed, with a maximum runtime of two weeks per run. The dot colour indicates the generation of the solution, with generation 0 denoting the initial population, highlighted by a black outline. Arrows denote whether the corresponding metric should be maximised or minimised.



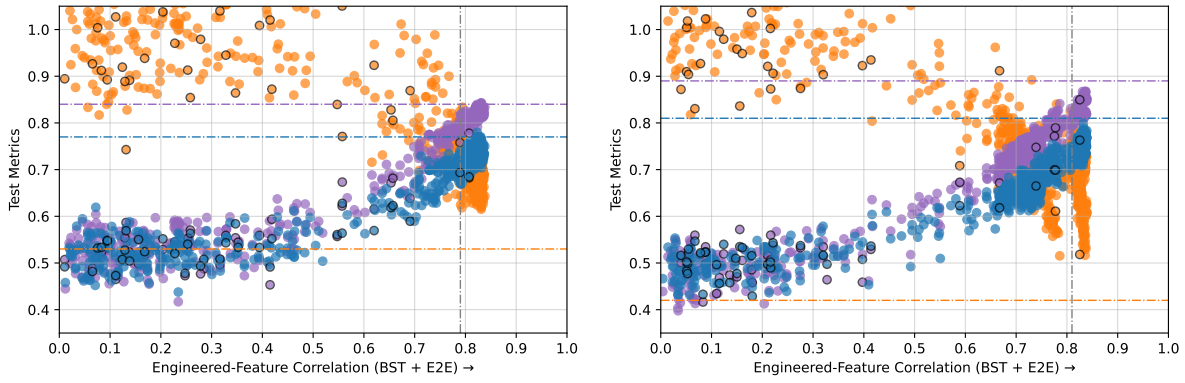
(a) Downsample seed: 0; Train seed: 0

(b) Downsample seed: 0; Train seed: 1

(c) Downsample seed: 1; Train seed: 0

(d) Downsample seed: 1; Train seed: 1

Figure C.23: Each dot represents a test metric value for a solution found by GOMEA for the three-gated XOR task with two tabular features, with standard DL parameters and without end-to-end training in the fitness evaluation, plotted against its engineered-feature correlation **before** end-to-end training. The test metrics include: Binary Cross Entropy Loss (BCE), Area Under the Receiver-Operating Characteristic curve (AUROC), and Balanced Accuracy (BAcc). The dash-dotted horizontal lines show the baseline values of the values obtained with no ablations, as shown in Figure C.4. Each subplot corresponds to one run with a specific downsample seed and training seed, with a maximum runtime of two weeks per run. The dot colour indicates the generation of the solution, with generation 0 denoting the initial population, highlighted by a black outline. Arrows denote whether the corresponding metric should be maximised or minimised.



(a) Downsample seed: 0; Train seed: 0

(b) Downsample seed: 0; Train seed: 1

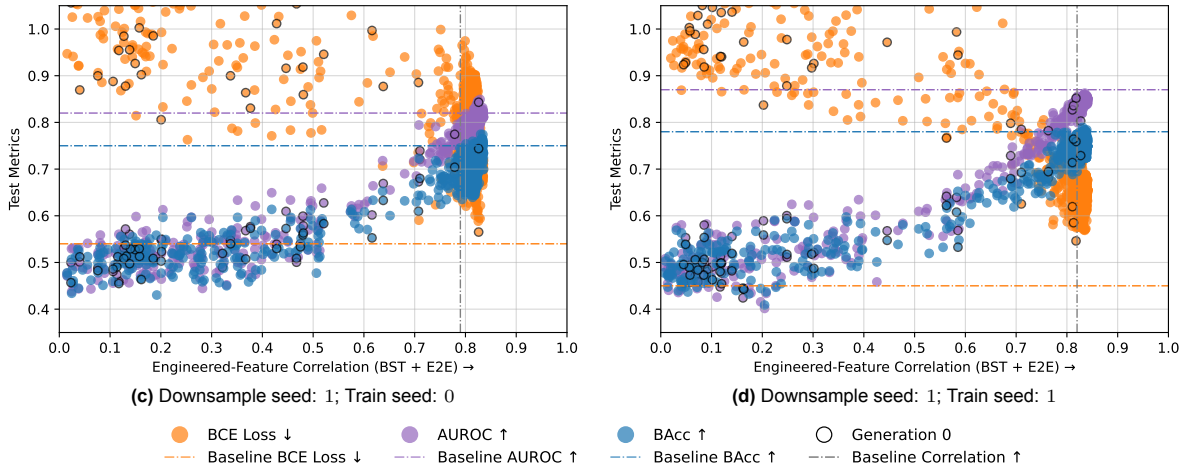


Figure C.24: Each dot represents a test metric value for a solution found by GOMEA for the three-gated XOR task with two tabular features, with standard DL parameters and without end-to-end training in the fitness evaluation, plotted against its engineered-feature correlation **after** end-to-end training. The test metrics include: Binary Cross Entropy Loss (BCE), Area Under the Receiver-Operating Characteristic curve (AUROC), and Balanced Accuracy (BAcc). The dash-dotted horizontal lines show the baseline values of the values obtained with no ablations, as shown in Figure C.4. Each subplot corresponds to one run with a specific downsample seed and training seed, with a maximum runtime of two weeks per run. The dot colour indicates the generation of the solution, with generation 0 denoting the initial population, highlighted by a black outline. Arrows denote whether the corresponding metric should be maximised or minimised.