Московский физико-технический институт
(национальный исследовательский университет)
Физтех-школа прикладной математики и информатики
Кафедра интеллектуальных систем

Набиев Мухаммадшариф Фуркатович

Выбор предсказательной модели в режиме многозадачного обучения с применением методов символьной регрессии

Выпускная квалификационная работа бакалавра

Научный руководитель:
к.ф.-м.н. О. Ю. Бахтеев

Москва — 2025

Abstract

This paper tackles the problem of constructing suboptimal models within multitask learning paradigm. Given a set of intrinsically related tasks, the goal is to uncover a shared structure—known as the inductive bias—across all tasks. By applying constrained evolutionary symbolic regression, we show that these models can be decomposed in a way that reveals the tasks' underlying structure. We conduct experiments on synthetic data and dataset of numbers MNIST, where the data structure, generation process, and optimal models are known. The results indicate that the models produced by the proposed method are indeed optimal for their respective tasks.

Contents

1 Introduction

The problem of establishing inductive bias [Mitchell, 2007] is a fundamental in machine learning. Inductive bias encapsulates the core assumptions that underpin the methodology adopted by a particular model in its predictive endeavors. For example, convolution captures locality of data and recurrent neural networks take into account sequential dependency [Cohen and Shashua, 2017, Dubinin and Effenberger, 2024]. It extends beyond the boundaries of explicitly observed data. Understanding and leveraging inductive bias is essential for enhancing model performance, especially in complex environments where data may exhibit diverse characteristics.

Inductive bias refers to a model's set of assumptions that guide it in making predictions on unseen data. These assumptions are essential for generalization: without them, a model would treat every possible hypothesis as equally likely, leading to overfitting or poor predictive performance [Mitchell, 2007]. Different datasets often possess unique distinguishing features that can be exploited for improved performance. For instance, models trained on image data may benefit from biases related to spatial hierarchies [Krizhevsky et al., 2017], while those handling sequential data [Hochreiter and Schmidhuber, 1997, Vaswani et al., 2017], such as time series or text, may rely on temporal dependencies. Recognizing and systematically integrating these biases into the model selection process can facilitate the identification of suboptimal yet effective architectures.

However, incorporating inductive biases into automated model selection remains a non-trivial challenge. In recent years, the field of machine learning has experienced rapid advancements driven by the development of sophisticated algorithms and architectures capable of tackling a wide range of tasks [Shehab et al., 2022, Ozbayoglu et al., 2020, Shekhar et al., 2020, Kumar et al., 2024], from natural language processing [Brown et al., 2020] to computer vision [Krizhevsky et al., 2012]. However, the design and optimization of these models often require significant expertise and

resources, leading to the rise of automated machine learning (AutoML) systems [He et al., 2021]. These systems are designed to simplify the process of choosing and tuning models, making machine learning tools more accessible and easier to use.

We use symbolic regression to infer inductive bias. Symbolic regression methods belong to a broader class of approaches that reduce the need for extensive manual tuning of model architectures. These methods aim to automatically discover symbolic representations using search strategies such as evolutionary algorithms [et al., 2020]. Few notable approaches are AutoML-Zero[Real et al., 2020] and PySR [Cranmer, 2023], which autonomously construct models using a genetic programming framework, assembling them from fundamental mathematical operations. These methods represent a significant departure from traditional model selection paradigms, which typically rely on predefined structures [Krizhevsky et al., 2017] or human intuition [Zoph and Le, 2016]. By allowing the model architecture to emerge organically from the data, AutoML-Zero and PySR minimizes biases introduced by prior knowledge, potentially leading to the discovery of novel architectures better suited to the task at hand [Dorrell et al., 2022, Liu et al., 2024].

By applying symbolic regression within a multitask learning framework, we aim to uncover shared inductive biases. Multitask learning (MTL) is closely linked to the notion of inductive bias. In fact, multitask learning can be viewed as a mechanism for introducing a meta-level inductive bias [Caruana, 1993, Baxter, 2000]—one that promotes the learning of shared representations across multiple related tasks. By jointly training on several tasks, the model is encouraged to capture common structures or features that are beneficial across them, effectively embedding a bias that reflects inter-task regularities. This shared inductive bias can improve generalization, especially when individual tasks have limited data or when the tasks exhibit complementary structure. Thus, multitask learning not only enhances performance through knowledge transfer but also serves as a principled way to shape the hypothesis space in alignment with the

underlying relationships among tasks [Zhang and Yang, 2017].

While MTL and symbolic regression can discover models with shared representations, these often risk encoding irrelevant features. To address this, the Information Bottleneck (IB) principle [Tishby et al., 2000] provides a framework for learning compressed, task-relevant representations by maximizing mutual information with outputs while minimizing it with inputs. In deep learning, is has applications such as explanation of how neural nets learn [Shwartz-Ziv and Tishby, 2017a], and methods such as Variational Information Bottleneck [Alemi et al., 2017]. There were also attempts to combine IB with MTL [Qian et al., 2020].

In this paper, we investigate how inductive biases inherent in the data can guide model selection within the multitask learning framework. We propose an automated methodology that combines evolutionary algorithms with symbolic regression to explore a diverse range of shared model architectures. A single model is trained with multiple task-specific heads corresponding to different datasets, enabling the discovery of shared representations that capture common structures across tasks. To ensure these representations remain compact and focused on relevant information, we incorporate the Information Bottleneck principle, which encourages learning compressed, task-relevant features, see fig. 1. This approach aims to improve generalizations and adaptability while reducing the need for extensive manual tuning by systematically searching for architectures that balance shared and task-specific components effectively.

Our contributions include:

1. We propose model selection criterion to identify inductive bias for a given set of tasks.

2. We provide theoretical justifications for the proposed method.

3. The experiments demonstrate that models achieving optimal trade-offs between accuracy, information, and complexity lie on the Pareto front.

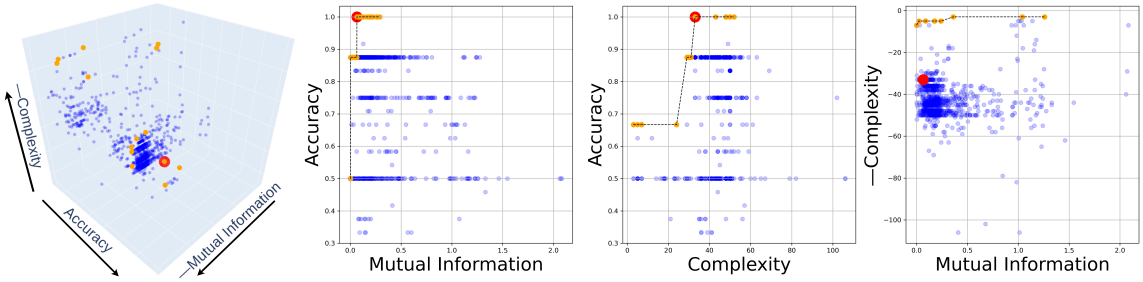

Figure 1: 3D and 2D scatter plots illustrate the performance of discovered models across three objectives: Mutual Information, Accuracy and Complexity. Each blue point corresponds to a candidate model, while the red point denotes the ground-truth model. The ground-truth model lies on Pareto-front for the Accuracy-Mutual Information and Accuracy-Complexity projections, indicating its optimality with respect to these trade-offs. In the Complexity-Mutual Information plane, the presence of models that are both low-complexity and low-information is expected, as such models represent simple expressions and ignore the input.

2 Problem statement

Multitask setting is based on an assumption that tasks can serve as mutual sources of inductive bias [Caruana, 1993]. Let $\mathfrak{T} = \{T_1, T_2, \ldots, T_m\}$ be a set of tasks. These tasks share a common internal structure but differ in their output heads, see fig. 2. Each task T_i has its corresponding dataset $\mathfrak{D}_i = (\mathbf{X}^i, \mathbf{y}^i)$ with an input objects $\mathbf{X}^i \in \mathbb{R}^{N_i \times n}$ and output targets $\mathbf{y}^i \in \mathcal{Y}$ where $\mathcal{Y} = \mathbb{R}^{N_i}$ for regression and $\mathcal{Y} = \{1, \ldots, K\}^{N_i}$ for classification.

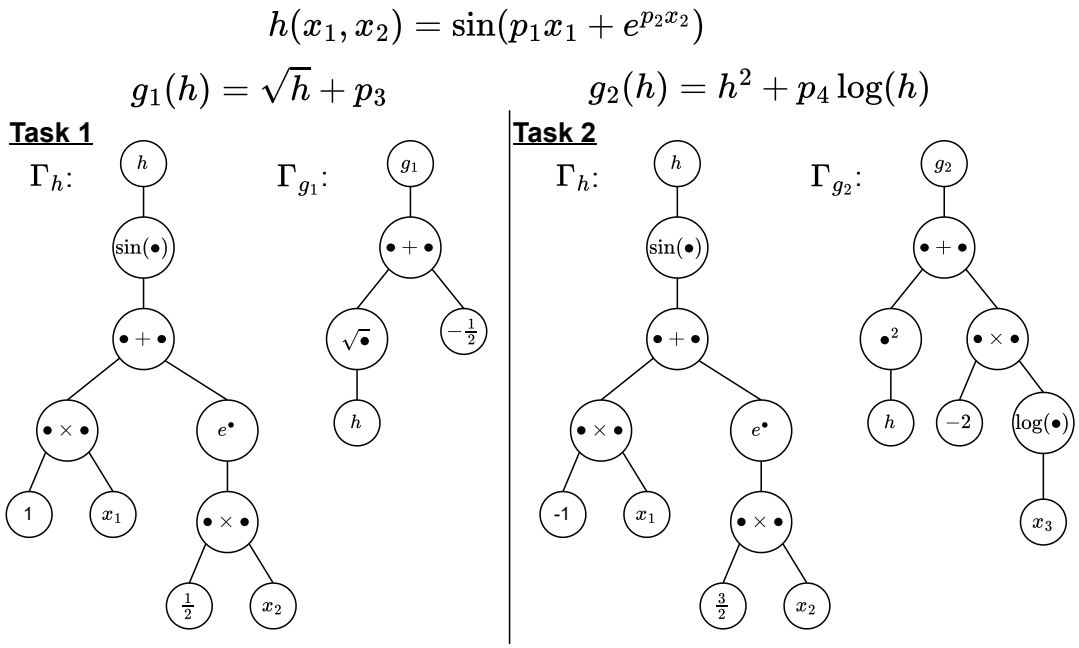

Figure 2: Multitask setting for $m = 2$ tasks. Dimension of hidden space $d = 1$. Parameters are task-specific and correspond to tasks.

A *model* $\mathbf{f}(\mathbf{x}, \mathbf{w})$ is defined as an expression with the following mapping $\mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}^m$, whose *structure* $\Gamma_{\mathbf{f}} \in \mathfrak{G}$ can be represented via expression tree [Rudoy and Strijov, 2013, Kulunchakov and Strijov, 2017] and $\mathfrak{G} := \{\Gamma_{\mathbf{f}} : \mathbf{f} \text{ is an expression}\}$. From this point onward, we omit $\mathbf{w}$, $\mathbb{R}^d$ and $\mathfrak{G}$ for clarity and simplicity.

Let $\mathfrak{F}$ be a parametric family of all models generated by symbolic regression algorithm with the constraint $\mathbf{f} = \mathbf{g} \circ \mathbf{h} := (g_1 \circ \mathbf{h}, \ldots, g_m \circ \mathbf{h})$, where $\mathbf{g}$ and $\mathbf{h}$ will be referred to as *decoder* and *encoder*.A prediction for i-th task will be located at the corresponding component $f_i = g_i \circ \mathbf{h}$.

For illustration, consider the case where $m = 2$ and $d = 1$. Suppose the two tasks are defined by the functions $g_1(h) = \sqrt{h} + p_3$ and $g_2(h) = h^2 + p_4 \log(h)$ both relying on a shared intermediate structure Γ_h, where $h(x_1, x_2) = \sin(p_1 x_1 + e^{p_2, x_2})$, see fig. 2. In this setup, the objective

becomes discovering such a shared encoder h.

Factoring a mapping $\mathbf{f}$ into $\mathbf{h}$ and $\mathbf{g}$ enables reuse of common intermediate features across tasks and shrinks the search space in symbolic regression: one first evolves a compact representation $\mathbf{h}$, then fits simpler heads $\mathbf{g}$. Theorem 2.1 ensures every $\mathbf{f}$ admits such factorization.

Theorem 2.1. *For any expression* $\mathbf{f} : \mathbb{R}^n \to \mathcal{Y}^m$ *there exists* $\mathbf{h} : \mathbb{R}^n \to \mathbb{R}^d$ *and* $\mathbf{g} : \mathbb{R}^d \to \mathcal{Y}^m$ *such that* $\mathbf{f} = \mathbf{g} \circ \mathbf{h}$.

Proof. There are two trivial cases when hidden space's dimension match dimension of the input space or the output space:

- $d = m$. We take $\mathbf{h} = \mathbf{f}$ and $\mathbf{g} = \mathrm{Id}$, hence $\mathbf{f}(\mathbf{x}) = \mathbf{g}(\mathbf{h}(\mathbf{x})) = \mathbf{g}(\mathbf{f}(\mathbf{x})) = \mathbf{f}(\mathbf{x})$.
- $d = n$. We take $\mathbf{h} = \mathrm{Id}$ and $\mathbf{g} = \mathbf{f}$, hence $\mathbf{f}(\mathbf{x}) = \mathbf{g}(\mathbf{h}(\mathbf{x})) = \mathbf{g}(\mathbf{x}) = \mathbf{f}(\mathbf{x})$

For non-trivial cases the decomposition can be obtained using Kolmogorov-Arnold theorem Kolmogorov [1961]. □

In search of such models, we employ an evolutionary search algorithm to discover a general model — up to constant factors — that can address the tasks $\mathfrak{T}$. The evolutionary process begins with a randomly initialized population of symbolic expressions. These expressions are composed of basic mathematical operations (e.g. addition, multiplication, exponential), and their structure is represented as expression trees. At each generation, individuals are evaluated across all tasks, and the best-performing models are retained. Variation is introduced through mutation:

1. Randomly select component from $[h_1, \ldots, h_d, g_1, \ldots, g_m]$.
2. Randomly select subtree `subtree` $\subset \Gamma$.
3. Replace it with newly generated of similar depth.

This promotes structural diversity while preserving overall model complexity.

Hypothesis: We refer to the structure of the optimal model's encoder as the inductive bias of the dataset. The encoder is assumed to capture fundamental properties of the dataset, thereby allowing us to interpret key characteristics of the data. Due to the nature of the given decomposition, it is necessary to impose certain constraints on the encoder. Without these restrictions, specific initial conditions may lead to degenerate cases in the decoder's structure.

Theorem 2.2. *Let* $\mathbf{f} : \mathbb{R}^n \to \mathcal{Y}^m$. *For a latent space* $\mathbb{R}^d$ *such that* $d \geq m$ *there exists a decomposition* $\mathbf{f} = \mathbf{g} \circ \mathbf{h}$ *such that* $\mathbf{g} = \mathrm{Id}_Z$ *and* $Z \cong \mathcal{Y}^m$.

Proof. Since $m \leq d$ we can redefine latent space as $Z \cong \mathcal{Y}^m$ and the given expression $\mathbf{f}$ by ignoring extra dimensions. If we take $\mathbf{h} = \mathbf{f} : \mathbb{R}^n \to Z$ and $\mathbf{g} = \mathrm{Id}_Z$ we get $\mathbf{f}(\mathbf{x}) = \mathbf{g}(\mathbf{h}(\mathbf{x})) = \mathrm{Id}_Z(\mathbf{f}(\mathbf{x})) = \mathbf{f}(\mathbf{x})$. □

By Theorem 2.2, we are required to constraint the dimensionality of a latent space. Among all encoders, our goal is to identify those that retain the most information about output while discarding as much irrelevant information from the input as possible. To this end, we incorporate the *Information Bottleneck*.

2.1 Information Bottleneck

Inspired by the role of intermediate representations in deep learning, we adopt the Information Bottleneck (IB) framework [Tishby and Zaslavsky, 2015] to analyze encoder structure $\Gamma_{\mathbf{h}}$. In this setting, the encoder $\mathbf{h}(\mathbf{X})$ is treated as a compressed representation of the input $\mathbf{X}$ that retains only the information necessary for predicting the output $\mathbf{y}$.

An encoder $\mathbf{h}$ is called sufficient with respect to $\mathbf{y}$ if:

$$\mathrm{I}(\mathbf{h}(\mathbf{X}), \mathbf{y}) = \mathrm{I}(\mathbf{X}, \mathbf{y}),$$

i.e., $\mathbf{h}(\mathbf{X})$ preserves all the information about $\mathbf{y}$ that is contained in $\mathbf{X}$. Minimal sufficient statistic, is a sufficient statistic that contains as little information about $\mathbf{X}$ as possible, formally defined as:

$$\mathbf{h}(\mathbf{X}) = \underset{\mathbf{h}'(\mathbf{X}):\, \mathrm{I}(\mathbf{h}'(\mathbf{X}),\mathbf{y})=\mathrm{I}(\mathbf{X},\mathbf{y})}{\arg\min} \mathrm{I}(\mathbf{h}'(\mathbf{X}), \mathbf{X}).$$

As exact solution exists only for special distributions, IB approach allows approximate solution via optimization problem [Shwartz-Ziv and Tishby, 2017b]:

$$\min_{p(\mathbf{h}(\mathbf{x})|\mathbf{x})} \mathrm{I}(\mathbf{X}, \mathbf{h}(\mathbf{X})) - \beta\, \mathrm{I}(\mathbf{h}(\mathbf{X}), \mathbf{y}).$$

The multiplier β controls how much relevant information should be captured.

We aim to identify encoder structure $\Gamma_{\mathbf{h}}$that maximizes task-specific loss functions L_i across all given tasks. For the i-th task, mutual information between $\mathbf{h}(\mathbf{X}^i)$ and $\mathbf{y}^i$ is implicitly optimized by the loss: the more information we capture about $\mathbf{y}^i$, the better the predictions. Therefore, to enforce sufficiency, it is enough to minimize $\mathrm{I}(\mathbf{X}^i, \mathbf{h}(\mathbf{X}^i))$, encouraging the encoder to retain only the information relevant for the label,

$$\mathrm{I}(\mathbf{X}^i, \mathbf{h}(\mathbf{X}^i)) - \beta\, \mathrm{I}(\mathbf{h}(\mathbf{X}^i), \mathbf{y}^i) \approx \mathrm{I}(\mathbf{X}^i, \mathbf{h}(\mathbf{X}^i)) - \beta^* L(f_i(\mathbf{X}^i), \mathbf{y}^i).$$

2.1.1 Example: Convolution as a sufficient statistic

Let $(\mathbf{X}, \mathbf{y})$ be a dataset for image modality. Suppose there exists a model $f = g \circ \mathbf{h}$, where $\mathbf{h} := \mathbf{Conv}$ is a convolutional encoder, g is a task-specific decoder and $\mathbf{h}(\mathbf{X})$ is a sufficient statistic. Then, the composed model $f(\mathbf{X}) = g(\mathrm{Conv}(\mathbf{X}))$ approximates true function from inputs to outputs, using only the information retained in the convolutional representation.

Let $\mathfrak{H}$ be a set of all minimal sufficient statistics for the given task and $\mathrm{C} : \mathfrak{F} \to \mathbb{R}$ be a complexity function. The function C computes the

minimal number of bits needed—via Huffman coding on the frequencies of subexpressions—to encode all subexpressions within the encoder. Notably, variable and parameter indices are disregarded since the focus is on the structural complexity of the expression tree itself; only operations contribute to this complexity.

Hypothesis: Encoder structure $\Gamma_{\mathbf{h}^*}$ best aligned with the inductive bias of the dataset corresponds to the simplest minimial sufficient statistic:

$$\mathbf{h}^* = \arg\min_{\mathbf{h} \in \mathfrak{H}} \mathrm{C}(\mathbf{h}).$$

2.2 Optimization

Multi-criteria optimization is concerned with optimizing several objectives simultaneously. We have a following objectives $\mathbf{L} = (L_1, \ldots, \mathrm{I}_1, \ldots, \mathrm{C})$. The multi-criteria optimization problem is:

$$\min_{\mathbf{f}=(\mathbf{g}_1 \circ \mathbf{h}, \ldots, \mathbf{g}_m \circ \mathbf{h})} \left(L_1\left(f_1(\mathbf{X}^1; \mathbf{w}_1^*), \mathbf{y}^1\right), \ldots, \mathrm{I}\left(\mathbf{X}^1, \mathbf{h}(\mathbf{X}^1)\right), \ldots, \mathrm{C}(\mathbf{h})\right),$$
$$\text{s.t. } \mathbf{w}_i^* = \arg\min_{\mathbf{w}} L_i(f_i(\mathbf{X}^i; \mathbf{w}), \mathbf{y}^i).$$

Objective min is not well-defined because the feasible solution that minimizes all objectives might not exist. Therefore, the solution is usually interpreted in terms of the Pareto optimal points. Instead of computing the full Pareto front, it is common to solve a scalarized version using weighted sums of objectives, leading to a single objective formulation.

Hence, final optimization objective can be formulated as follows:

$$\Gamma_{\mathbf{h}} = \arg\min_{\mathbf{f}=(\mathbf{g}_1 \circ \mathbf{h}, \ldots, \mathbf{g}_m \circ \mathbf{h})} \frac{1}{m} \sum_{i=1}^{m} L_i(f_i(\mathbf{X}^i; \mathbf{w}_i^*), \mathbf{y}^i) + \lambda_1 \mathrm{I}(\mathbf{X}^i, \mathbf{h}(\mathbf{X}^i)) + \lambda_2 C(\mathbf{h}), \tag{1}$$

$$\text{s.t. } \mathbf{w}_i^* = \arg\min_{\mathbf{w}} L_i(f_i(\mathbf{X}^i; \mathbf{w}), \mathbf{y}^i). \tag{2}$$

Thus we define *inductive bias* as the structure $\Gamma_{\mathbf{h}}$ of $\mathbf{h}$. The goal is to construct models which general form can approximate the given tasks. In order to favor generalization an informational bottleneck approach was taken.

Having multi-task paradigm is crucial for our decomposition. Single-task setting will degenerate our solution to having $\mathbf{g} = \mathrm{Id}$.

Theorem 2.3. *For single-task setting there exists a solution* $\mathbf{f}$ *that has a decomposition* $f = g \circ h$*, where* $g = \mathrm{Id}$ *and* $h = f$*.*

Proof. In the single-task setting, any model f that maps inputs X to outputs Y can trivially be decomposed into an encoder-decoder form: $f = g \circ h$, where the encoder $h = f$ and $g = \mathrm{Id}$. This satisfies the required decomposition. □

Theorem 2.3 immediately gives us the following corollary.

Corollary 2.3.1. *Let there be* $m > 1$ *tasks, and suppose each task* T_i *is modeled by* $f_i(\mathbf{x}) = [\mathbf{f}(\mathbf{x}, \mathbf{w}_i)]_i$ *where all task-specific heads share the same expression-tree structure:* $\Gamma_{g_i} = \Gamma_{g_j}$ *for every* $i, j \in \{1, \ldots, m\}$*. Then one can choose* $\mathbf{h}(\mathbf{x}, \mathbf{w}) = \mathbf{f}(\mathbf{x}; \mathbf{w})$, $\mathbf{g}(\mathbf{x}) = \mathbf{x}$*, so that for each task* $f_i(\mathbf{x}) = (\mathbf{g} \circ \mathbf{h})(\mathbf{x}; \mathbf{w}_i) = \mathbf{h}(\mathbf{x}; \mathbf{w}_i)$*.*

3 Computational experiment

We conducted experiments across three fundamentally distinct types of tasks. The first type was based on the circles dataset [Pedregosa et al., 2011] and included one binary classification task and two regression tasks, each associated with a different decoder. The second type involved synthetic autoregressive (AR) data, while the third included both classification and regression tasks on 4-pixel images and MNIST [Deng, 2012].

For circles and 4-pixel datasets we analytically computed the mutual information between the input and the ground truth representations. This allowed us to test the hypothesis that our analytically derived ground truth representations lie on the Pareto front.

The primary goal of these experiments was to assess whether our framework could identify representations that act as minimal sufficient statistics for multiple tasks under mutual information constraints. By imposing analytically defined mutual information thresholds and penalizing excess complexity, we evaluated whether effective multitask encoders compress inputs just enough to retain only task-relevant content. This experimental design enabled us to observe how the model negotiates the trade-off between informativeness and complexity, and to investigate whether shared inductive biases naturally emerge across diverse tasks.

3.1 Experiments

3.1.1 Circles

Motivated by the single-task setup described in Theorem. 2.3 using synthetic circle datasets, we extend the setting to three distinct tasks, each defined by a different decoder: $[r > 0]$, $r + 0.5$, and $\sqrt{r} - 0.5$ where r is the radius of the circle. The underlying hypothesis is that, despite the varying decoders, the encoder $\mathbf{h}$ will consistently learn a second-order

representation that captures the circular structure of the data. The decision boundary for datasets was chosen to be a circle for its simplicity and practicality.

First we discovered models without using any kind of regularizers. Then we utilized mutual information and complexity (see Table. 1). The experiments, indeed, show that we can infer second-order equation. Models from 1-5 were discovered using only task-specific losses. And as we can see it has very high mutual information and complexity. They also have complex expressions for encoder. When using mutual information and complexity (models 6-10) we were able to discover structure of the hidden space.

First, we discovered models without applying any form of regularization. Subsequently, we incorporated mutual information and complexity as additional objectives (see Table 1). The experiments demonstrate that this approach allows us to infer a second-order equation. Models 1-5 were trained using only task-specific losses, which resulted in high mutual information and complexity values, as well as complex encoder expressions. In contrast, models 6-10, trained with mutual information and complexity regularization, revealed a more structured and interpretable models.

Models 6, 8, and 10 demonstrate high accuracy with minimal complexity, aligning well with the structure of the task. These regularized models achieve compact representations while preserving predictive performance.

3.1.2 Sequential data

We constructed simple synthetic datasets following the autoregressive dependency $y_t = \alpha y_{t-1} + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. By varying the parameter α, we generated datasets with different temporal dynamics. The models discovered on these datasets, along with their associated mutual information and complexity, are summarized in Table 3.1.2.

	#	Accuracy	MSE_1	MSE_2	$\text{I}(\mathbf{h}(\mathbf{X}), \mathbf{X})$	C
No reg.	1	1.00	$4.63 \cdot 10^{-2}$	$2.22 \cdot 10^{-2}$	2.98	22
	2	1.00	$6.52 \cdot 10^{-2}$	$3.60 \cdot 10^{-3}$	4.20	34
	3	0.82	$3.56 \cdot 10^{-2}$	$4.68 \cdot 10^{-3}$	4.82	18
	4	1.00	$2.10 \cdot 10^{-2}$	$6.24 \cdot 10^{-3}$	2.98	43
	5	0.57	$2.62 \cdot 10^{-2}$	$3.90 \cdot 10^{-2}$	2.13	38
Reg.	6	0.83	$4.08 \cdot 10^{-14}$	$2.26 \cdot 10^{-15}$	1.85	12
	7	0.80	$6.36 \cdot 10^{-2}$	$2.11 \cdot 10^{-2}$	1.56	16
	8	0.77	$7.14 \cdot 10^{-16}$	$1.14 \cdot 10^{-4}$	1.86	12
	9	0.78	$6.33 \cdot 10^{-2}$	$2.12 \cdot 10^{-2}$	1.56	16
	10	0.80	$6.57 \cdot 10^{-2}$	$2.80 \cdot 10^{-2}$	1.86	10

Table 1: Found models and metrics on circle dataset. The explicit forms of the models can be found in Appendix 5.

#Datasets	Models	$\text{I}(\mathbf{h}(\mathbf{X}), \mathbf{X})$	C
1, 2, 5	$p_0 x_t$	5.83	5
10	$p_0 sin^2(x_t) x_t$	3.14	18

Table 2: Found models for synthetic autoregression dataset.

3.1.3 Image data

We also conducted experiments on image data, focusing on both the MNIST dataset and a synthetic 4-pixel dataset.

Synthetic 4-pixel dataset. We constructed a benchmark consisting of eight tasks based on 4-pixel binary inputs: six binary classification tasks and two regression tasks. In the classification tasks, each label indicates whether a specific pattern appears within the 4 input pixels. For example, the input [0, 1, 0, 1] containes the pattern [0, 1, 0], resulting in a classification label of 1 and a regression output of 1. Similarly, the input [1, 1, 1, 1] contains the pattern [1, 1, 1], yielding a classification label of 1 and a regression output of 2. The regression tasks are defined

such that each output is the sum of two binary indicators—one over the first three pixels and another over the last three pixels—reflecting the presence of predefined patterns. These tasks aim to evaluate whether the learned encoder representations capture locality in the input. The results presented in Table 3 confirm that the discovered encoders do indeed exhibit locality.

All discovered models demonstrate locality, meaning they rely on a pixel

#	Average accuracy	Average MSE	$\mathrm{I}(\mathbf{h}(\mathbf{X}), \mathbf{X})$	C
1	0.98	$3.5 \cdot 10^{-3}$	9.57	36
2	0.92	$8.19 \cdot 10^{-4}$	9.14	30
3	0.97	$5.96 \cdot 10^{-3}$	8.58	43
4	0.94	$1.67 \cdot 10^{-13}$	9.96	36
5	0.94	$3.93 \cdot 10^{-2}$	8.30	45

Table 3: Discovered models and their performence on the synthetic 4-pixel dataset. The explicit forms of the models can be found in Appendix 5.

and its nearby neighbors (no farther than 2 units away). Each model achieves a favorable trade-off between accuracy, complexity and mutual information.

MNIST dataset. We further evaluated our approach on MNIST using a simple neural network with a single hidden layer, see fig. 3. Each hidden neuron was assigned a mask that restricts its receptive field to a local neighborhood of 8 pixels around a given pixel, mimicking the behavior of convolution. The model was trained with a mutual information regularizer, implemented via the entropy of a Gauusian distribution. For comparison, we repeated the experiment using randomly assigned masks.

We observed that models with deterministic masks consistently outperformed those with random masks along the Pareto front, demonstrating better trade-offs between accuracy and complexity.

The results demonstrate that our method discovers simpler models without compromising predictive accuracy. By incorporating mutual

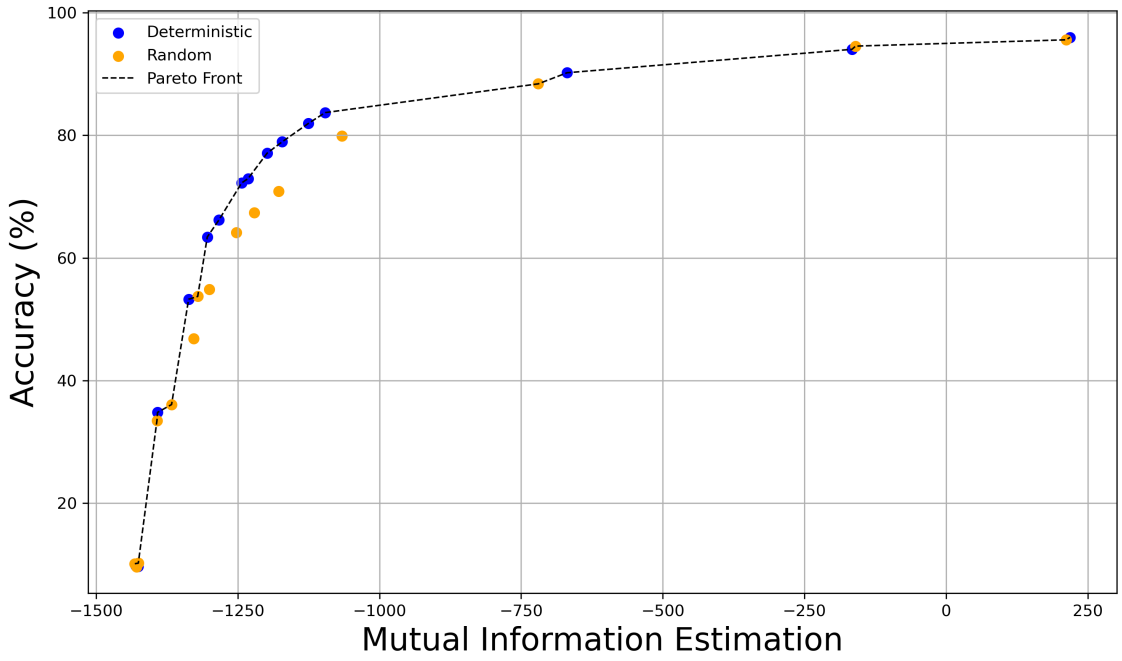

Figure 3: Models constructed for MNIST.

information and complexity regularization, we guide the learning process toward representations that capture relevant patterns in the data. The structure of the learned models reveals the inductive biases inherent in the dataset.

4 Conclusion

In this work, we addressed the problem of constructing suboptimal models within the multitask learning paradigm. Given a set of intrinsically related tasks, our objective was to uncover a shared structure—referred to as the inductive bias—common across all tasks. Using constrained evolutionary symbolic regression, we demonstrated that these models could be decomposed in a manner that revealed the underlying structure of the data. We validated our approach through experiments on synthetic data and the MNIST dataset, where the data generation process, structure, and optimal models were known. The results showed that the models produced by the proposed method captured the shared inductive bias, confirming the method's ability to recover interpretable and optimal structures in a multitask setting.

A promising direction is to integrate the proposed method in AutoML pipelines, where the system automatically infers the inductive biases inherent in a given dataset and selects or configures models accordingly. By analyzing symbolic encoders optimized for compression and predictive performance, one could extract structural properties of the data and use these insights to guide model architecture choices.

References

Tom Michael Mitchell. The need for biases in learning generalizations. 2007. URL https://api.semanticscholar.org/CorpusID:3237155.

Nadav Cohen and Amnon Shashua. Inductive bias of deep convolutional networks through pooling geometry, 2017. URL https://arxiv.org/abs/1605.06743.

Igor Dubinin and Felix Effenberger. Fading memory as inductive bias in residual recurrent networks. *Neural Networks*, 173:106179, 2024. ISSN 0893-6080. doi:https://doi.org/10.1016/j.neunet.2024.106179. URL https://www.sciencedirect.com/science/article/pii/S0893608024001035.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, May 2017. ISSN 0001-0782. doi:10.1145/3065386. URL https://doi.org/10.1145/3065386.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997. ISSN 0899-7667. doi:10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems*, 2017. URL https://api.semanticscholar.org/CorpusID:13756489.

Mohammad Shehab, Laith Abualigah, Qusai Shambour, Muhannad A. Abu-Hashem, Mohd Khaled Yousef Shambour, Ahmed Izzat Alsalibi, and Amir H. Gandomi. Machine learning in medical applications: A review of state-of-the-art methods. *Comput-*

ers in Biology and Medicine, 145:105458, 2022. ISSN 0010-4825. doi:https://doi.org/10.1016/j.compbiomed.2022.105458. URL https://www.sciencedirect.com/science/article/pii/S0010482522002505.

Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer. Deep learning for financial applications : A survey. *Applied Soft Computing*, 93:106384, 2020. ISSN 1568-4946. doi:https://doi.org/10.1016/j.asoc.2020.106384. URL https://www.sciencedirect.com/science/article/pii/S1568494620303240.

Himanshu Shekhar, Sujoy Seal, Saket Kedia, and Amartya Guha. Survey on applications of machine learning in the field of computer vision. In Jyotsna Kumar Mandal and Debika Bhattacharya, editors, *Emerging Technology in Modelling and Graphics*, pages 667–678, Singapore, 2020. Springer Singapore. ISBN 978-981-13-7403-6.

S. Pradeep Kumar, Meenakshi Diwakar, Jaishika S, P. Arthi, V. Samuthira Pandi, and Shobana D. The application of machine learning to natural language processing: Modern advances in the study of human language. In *2024 International Conference on Cybernation and Computation (CYBERCOM)*, pages 561–566, 2024. doi:10.1109/CYBERCOM63683.2024.10803211.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Xin He, Kaiyong Zhao, and Xiaowen Chu. Automl: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212:106622, January 2021. ISSN 0950-7051. doi:10.1016/j.knosys.2020.106622. URL http://dx.doi.org/10.1016/j.knosys.2020.106622.

Cranmer et al. Discovering symbolic models from deep learning with inductive biases. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Esteban Real, Chen Liang, David So, and Quoc Le. Automl-zero: Evolving machine learning algorithms from scratch. In *International conference on machine learning*, pages 8007–8019. PMLR, 2020.

M. Cranmer. Interpretable machine learning for science with pysr and symbolicregression.jl. *ArXiv*, abs/2305.01582, 2023. URL https://api.semanticscholar.org/CorpusID:258436785.

Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

William Dorrell, Maria Yuffa, and Peter E. Latham. Meta-learning the inductive bias of simple neural circuits. In *International Conference on Machine Learning*, 2022. URL https://api.semanticscholar.org/CorpusID:254018268.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljacic, Thomas Y. Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks. *ArXiv*, abs/2404.19756, 2024. URL https://api.semanticscholar.org/CorpusID:269457619.

Rich Caruana. Multitask learning: a knowledge-based source of inductive bias. In *Proceedings of the Tenth International Conference on International Conference on Machine Learning*, ICML'93, page 41–48, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc. ISBN 1558603077.

Jonathan Baxter. A model of inductive bias learning. *J. Artif. Int. Res.*, 12(1):149–198, March 2000. ISSN 1076-9757.

Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34:5586–5609, 2017. URL https://api.semanticscholar.org/CorpusID:11311635.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *ArXiv*, physics/0004057, 2000. URL https://api.semanticscholar.org/CorpusID:8936496.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *ArXiv*, abs/1703.00810, 2017a. URL https://api.semanticscholar.org/CorpusID:6788781.

Alexander A. Alemi, Ian Fischer, and Joshua V. Dillon. Deep variational information bottleneck. *ArXiv*, abs/1612.00410, 2017. URL https://api.semanticscholar.org/CorpusID:204922497.

Weizhu Qian, Bowei Chen, and Franck Gechter. Multi-task variational information bottleneck. *ArXiv*, abs/2007.00339, 2020. URL https://api.semanticscholar.org/CorpusID:220280268.

G. I. Rudoy and V. V. Strijov. Algorithms for inductive generation of superpositions for approximation of experimental data. *Informatics and Applications*, 7(1):44–53, 2013.

A.S. Kulunchakov and V.V. Strijov. Generation of simple structured information retrieval functions by genetic algorithm without stagnation. *Expert Systems with Applications*, 85:221–230, 2017. ISSN 0957-4174. doi:https://doi.org/10.1016/j.eswa.2017.05.019. URL https://www.sciencedirect.com/science/article/pii/S0957417417303354.

Andrey Nikolaevich Kolmogorov. *On the representation of continuous functions of several variables by superpositions of continuous functions of a smaller number of variables.* American Mathematical Society, 1961.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5, 2015. URL https://api.semanticscholar.org/CorpusID:5541663.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information, 2017b. URL https://arxiv.org/abs/1703.00810.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. doi:10.1109/MSP.2012.2211477.

5 Appendix

Here we list found models.

5.1 Circles

Model 1:

$$
\begin{aligned}
h_0 &= x_0(x_1 + x_1),\\
h_1 &= (x_0 + p_0) + x_1,\\
g_0 &= \big(p_1 + h_1^2\, p_0^2\big) + p_0(p_1 + p_1)(h_0 + h_0),\\
g_1 &= p_1\, h_0 + p_1 + h_1 p_2,\\
g_2 &= p_2 + (p_1 + h_0)(h_0\, p_0).
\end{aligned}
$$

Model 2:

$$
\begin{aligned}
h_0 &= (x_0 + p_1)p_0,\\
h_1 &= \sqrt{x_1^2} + p_2 p_0,\\
g_0 &= \big(p_0 + h_0^2\big) + \Big(p_1^4 + (h_1 + p_0)h_1\Big),\\
g_1 &= h_0^2 + h_1,\\
g_2 &= p_2 h_1 + \sqrt{(h_0 + h_0)^2}.
\end{aligned}
$$

Model 3:

$$
\begin{aligned}
h_0 &= p_1\big(p_2 + p_0 + x_0\big),\\
h_1 &= x_1,\\
g_0 &= h_1 p_0 + \big(h_1 h_0 + p_2^2\big),\\
g_1 &= h_0^2 + p_0^2 + \big(p_1 p_2 + p_1\big) + p_2 h_1^2,\\
g_2 &= \big(p_2 + h_0^2 + \sqrt{h_1^2}\big)p_2.
\end{aligned}
$$

Model 4:

$$
\begin{aligned}
h_0 &= p_2 x_0 x_1,\\
h_1 &= x_1 + \big(2p_2 + p_0 + x_0 + p_1\big),\\
g_0 &= \big(h_0 + p_0 + h_0\big)p_0 + h_1^2,\\
g_1 &= (p_1 + h_0)^4 + p_0 p_1 h_1,\\
g_2 &= 2p_1 h_1^2 + h_0 + p_2^2.
\end{aligned}
$$

Model 5:

$$
\begin{aligned}
h_0 &= x_0 + x_1 x_0 p_1 + x_1 + p_0,\\
h_1 &= p_2^2,\\
g_0 &= h_0\big(h_0 + p_0\big) + \big(h_0 + p_0\big)^2 + 2h_1,\\
g_1 &= h_1 + 4p_0^2 h_0^2\, p_1^2,\\
g_2 &= p_1.
\end{aligned}
$$

Model 6:

$$
\begin{aligned}
h_0 &= p_1,\\
h_1 &= x_0^2 + x_1^2,\\
g_0 &= \big(h_0 h_1 + 2p_0\big)h_1^2 + h_0,\\
g_1 &= p_0 + \sqrt{h_1} + p_1 p_2 p_0 + p_0^2 p_1 + p_1,\\
g_2 &= p_0 + \sqrt{h_0^2 \sqrt{h_1}}.
\end{aligned}
$$

Model 7:

$$
\begin{aligned}
h_0 &= \big(p_2 x_0 + x_1\big)^2,\\
h_1 &= p_1,\\
g_0 &= h_0 + h_1 + 4h_0^2 (h_0 + p_0)^2,\\
g_1 &= p_2 p_1 + \sqrt{h_0} p_1 + p_0,\\
g_2 &= p_0 + h_0(h_1 + p_2 h_0).
\end{aligned}
$$

Model 8:

$$
\begin{aligned}
h_0 &= x_1^2 + x_0^2,\\
h_1 &= p_0,\\
g_0 &= h_1 h_0 + p_1\big(h_0 + p_1\big) + h_0^2 p_0^2,\\
g_1 &= \sqrt{h_0} + h_1,\\
g_2 &= p_2\big(h_0 p_0 + p_1 + \sqrt{h_0}\big).
\end{aligned}
$$

Model 9:

$$
\begin{aligned}
h_0 &= p_2,\\
h_1 &= \big(x_1 + x_0 p_0\big)^2,\\
g_0 &= \big(p_2^2 + p_0\big) h_1^2 p_2^2 + 2h_1 + p_2,\\
g_1 &= p_2 + \sqrt{h_1}\big(p_2^2 + p_1 h_1\big),\\
g_2 &= h_0\sqrt{p_1 + h_1 + h_0^2}.
\end{aligned}
$$

Model 10:

$$
\begin{aligned}
h_0 &= \big(x_1 + x_0\big)^2,\\
h_1 &= p_2,\\
g_0 &= \big(p_1 + h_0\big)\big(h_1 + p_0 + h_1 h_0\big) + 2p_1,\\
g_1 &= \sqrt{h_0} h_1^2 + h_1 h_0 p_0 + p_1,\\
g_2 &= h_0 p_0 + p_1\big(h_1 + p_2\big)\sqrt{h_0}.
\end{aligned}
$$

5.2 4-pixels

Model 1:

$$
\begin{aligned}
h_0 &= x_1p_3 - x_2 - p_5 - x_0,\\
h_1 &= x_2p_2 + x_3,\\
g_0 &= \big(h_0 + p_1h_0 - h_1h_0 - \max(h_1, p_0)\big)\big(\max(h_1h_0, p_4 + h_1) + h_0 + p_4p_1\big),\\
g_1 &= h_0\big(p_1(p_4 - h_1) - \max(p_3, h_0)\big).
\end{aligned}
$$

Model 2:

$$
\begin{aligned}
h_0 &= x_1,\\
h_1 &= \max\big(p_4, x_2 - x_0\big) + x_2p_5 - x_3,\\
g_0 &= \max\big(h_1, \max(p_5, h_1)\big) - \max\big(p_3, h_1p_0\big) - h_0p_2(h_1 - h_0),\\
g_1 &= \max\big(h_1 - p_5 - p_0, p_1\big) + p_1 + 2p_3h_0(p_3 + h_1)(p_0 + p_5).
\end{aligned}
$$

Model 3:

$$
\begin{aligned}
h_0 &= \big(x_1 + p_2\big)\big(p_3x_3 + p_1x_2\big),\\
h_1 &= x_1 + p_1x_0,\\
g_0 &= \max\big(h_1 + p_0 - p_5h_0, p_5 - p_4\big),\\
g_1 &= -p_4 + p_5 - \max(h_1, h_1) - (h_0 + p_2) + p_0.
\end{aligned}
$$

Model 4:

$$
\begin{aligned}
h_0 &= x_1p_3 - x_3\big(p_0 + x_2\big),\\
h_1 &= x_0 + x_2p_2,\\
g_0 &= h_0(h_0 + h_1) + (h_0 + p_4) - p_1p_5\max(h_1, h_1),\\
g_1 &= (p_1p_2 + p_4 + h_0 + h_1)\max\big(p_1p_4, \max(h_1h_0, p_4h_0)\big).
\end{aligned}
$$

Model 5:

$$
\begin{aligned}
h_0 &= p_3\big(x_2 - p_4\big)\big(p_2 + x_0 + x_1\big),\\
h_1 &= x_1\big(p_1 - x_3\big),\\
g_0 &= (p_5 - h_0)(p_0 + h_1) - p_4,\\
g_1 &= \max\big(h_0 + h_1, h_0\big) + \max\big(h_1 p_5, p_4\big) - \big(h_1^2 + p_3\big).
\end{aligned}
$$