CoFrNets: Interpretable Neural Architecture Inspired by Continued Fractions

Anonymous Author(s)
Affiliation
Address
email

Abstract

In recent years there has been a considerable amount of research on local post hoc explanations for neural networks. However, work on building interpretable neural architectures has been relatively sparse. In this paper, we present a novel neural architecture, CoFrNet, inspired by the form of continued fractions which are known to have many attractive properties in number theory, such as fast convergence of approximations to real numbers. We show that CoFrNets can be efficiently trained as well as interpreted leveraging their particular functional form. Moreover, we prove that such architectures are universal approximators based on a proof strategy that is different than the typical strategy used to prove universal approximation results for neural networks based on infinite width (or depth), which is likely to be of independent interest. We experiment on nonlinear synthetic functions and are able to accurately model as well as estimate feature attributions and even higher order terms in some cases, which is a testament to the representational power as well as interpretability of such architectures. To further showcase the power of CoFrNets, we experiment on 6 real datasets spanning tabular, text and image modalities, and show that they are either comparable or significantly better than other interpretable models and multilayer perceptrons, sometimes approaching the accuracies of state-of-the-art models.

1 Introduction

> "It is simple. The minute I heard the problem, I knew that the answer was a continued fraction. Which continued fraction, I asked myself. Then the answer came to my mind."

This was the response of the mathematics genius Ramanujan to Mahalanobis, who was astounded how he was able to solve the difficult Strand puzzle [29] almost instantaneously. Besides showcasing the genius of Ramanujan, the puzzle also showcases the power of continued fractions. Continued fractions (CFs), typically represented as a sequence that looks like a ladder: $a_0 + \frac{b_1}{a_1 + \frac{b_2}{a_2 + \cdots}}$, can represent any real number and any analytic function, including trignometric functions, polynomials, the exponential function, power functions, and special functions like the gamma, hypergeometric, and Bessel functions [8]. To represent arbitrary real numbers, it is sufficient for the a_ks to be non-negative integers and for the $b_k = 1$. Analytic functions, which can be represented as a power series, can also be represented in this form. Moreover, CFs are the best rational approximations to a number/function in a certain sense [30]. Rational approximations are obtained by curtailing the fraction just before the "+" sign. For instance, in the example CF above a_0, $\frac{a_0 a_1 + b_1}{a_1}$, and $\frac{a_0 a_1 a_2 + a_0 b_2 + a_2 b_1}{a_1 a_2 + b_2}$ are three different rational approximations. Additional properties of CFs are discussed in Section 3.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Given the desirable properties of CFs and noticing their ladder-like structure, we propose a neural architecture inspired by CFs illustrated in Figure 1. In place of the a_ks, linear functions of the input $x \in \mathbb{R}^p$ are computed by taking the inner product of x with weight vector $w_k \in \mathbb{R}^p$ in each layer k (or step of the ladder).[1] The reciprocal of the function thus far is applied as a nonlinearity in each layer. We refer to this **Co**ntinued **Fr**action-inspired neural network as CoFrNet (pronounced 'coffer net'). (Like coffers in building architecture [49], the proposed neural architecture is made up of repeating structures.) Although more complicated functions than linear could be used in each layer, we show in Section 5 that linear functions are sufficient for universal approximation with a finite number of such ladders. The proof follows a different strategy than typical results on universal approximation of neural networks that rely on the results of Cybenko [9], Hornik [20], and Zhou [26], and may be of independent interest.

The proposed architecture is "simple": there are only p weights to be learned in each layer as opposed to a quadratic number for standard architectures such as multilayer perceptron (MLP) or those with densely connected layers.[2] A key differentiation from other neural architectures is that the input is passed into every layer [14, 18, 15, 47].[3] Moreover, the nonlinearity $\frac{1}{z}$ is different from more commonly used nonlinearities such as sigmoid, ReLU, and polynomials. The CF representation is much more compact than directly representing a polynomial of the same degree, which requires exponentially many coefficients. We later show how the CF representation for analytic functions permits the architecture to be made human-interpretable.

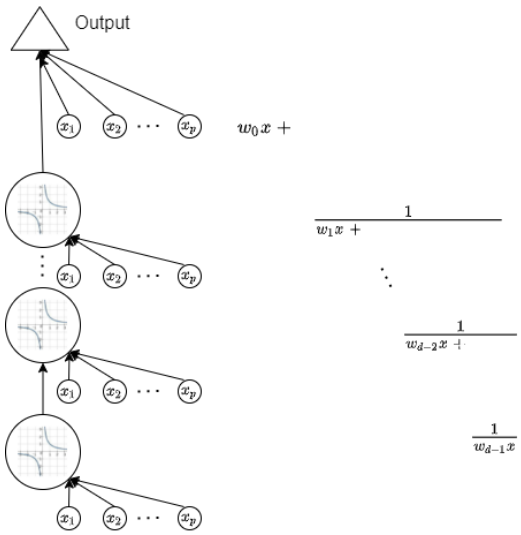


Figure 1: Single ladder (depth d) CoFrNet architecture on the left. On the right we see the corresponding function computed at each stage.

Simply being able to represent a rich class of functions does not imply effective learnability. (After all, unlike Ramanujan, the answer will not simply come to the machine's mind.) However, we empirically demonstrate that learning this function class is indeed possible. We propose variants of the base architecture catered toward ease of interpretation and efficiency of training, while still minimizing generalization error. We apply CoFrNets to tabular, text, and image data, and show they are either competitive or significantly better than other interpretable models and MLPs. In addition, we are not only able to model synthetic data generated from complicated nonlinear functions accurately, but also obtain feature attributions and recover the functional form reasonably well by leveraging properties of the architecture.

In summary, the main contribution of this paper is a new architecture covering an interesting and rich function class. By taking advantage of properties studied from the very beginnings of formal mathematics, we have stumbled upon a simple, yet powerful new idea in neural architecture design with much promise for accuracy, interpretability, and even efficiency. In this initial paper, we believe we have only scratched the surface of the CoFrNet architecture. Different training strategies and architecture variants as discussed briefly in Sections 4 and 7, among other enhancements, may lead to even better performance in future work.

2 Related Work

There are numerous types of explainable machine learning methods. Given our proposal of an interpretable neural architecture, we focus our discussion of related work on methods that yield global explanations or interpretable models or neural architectures, even though the latter may be opaque.

[1] A constant term is assumed to be absorbed in x for clearer exposition.

[2] See the supplement for the exact quantification of the number of weights.

[3] Skip connections such as those seen in residual network-type architectures may end up passing the input to upper layers, but it is unlikely that the input would be consistently passed undisturbed to all upper layers.

Black-Box Neural Architectures. MLP is a standard neural architecture typically composed of a fully-connected network [14]. However, MLPs have limitations in their representation and performance, leading to many modern architectures. Convolutional neural networks (CNNs) [15] have been very successful in modeling image data. Further improvements were seen with residual network (ResNet) [18] type of architectures which employ ideas such as skip connections. This idea is now used in many other architectures such as DenseNet [21] and MobileNetV2 [43]. Transformers [47] have seen immense success for text data and recently were also shown to perform well for images [13]. Π-nets [7] were recently shown to perform well for images where they learn a high degree polynomial using tensor decomposition to reduce dimensionality of the search space. Neural networks with activation functions other than the standard ReLU or sigmoid such as those based on Padé approximation [32] have also been suggested. Other architectures such as pi-sigma networks [46] and capsule networks [42] have also found wide appeal.

Globally Interpretable Models. Standard machine learning models such as logistic regression, decision trees [5], and rule based models [41, 10] are globally interpretable. Generalized additive models (GAMs) [6] and their more recent variants such as neural additive models (NAMs) [1] and explainable boosted machines (EBMs) [35] also belong to this category. LassoNet [25] is one of the most recently proposed architectures that can be considered interpretable. However, it is restrictive in the sense that if a feature is not selected by itself, it will not appear in any interaction terms. This precludes accurate estimation of functions which have only interaction terms including the very simple bivariate function x_1x_2.

Local to Global Post Hoc Methods. There are post hoc explanation methods which take local explanations and create global ones. TreeShap [27] creates a global SHAP explanation for tree based models. While, model agnostic multilevel explanation (MAME) [39] can create global LIME explanations. Alternatively, the global Boolean feature learning (GBFL) [36] method leverages local constrastive explanations [12] to create globally interpretable rule-based models.

Self-Explaining Models. Another category of models may not be globally interpretable, but provides local explanations without post hoc mechanisms. Reference [2] is suited for tabular and image data, whereas [52] is suitable for text data. Reference [19] is a framework that provides explanations for new examples if explanations are available for training examples. All of these methods, however, do not readily expose the global behavior of the model.

3 Preliminaries

We now introduce some notation and discuss equivalent forms for representing continued fractions. We also discuss some of their properties. As mentioned in the introduction, the generalized form for a continued fraction is $a_0 + \frac{b_1}{a_1+\frac{b_2}{a_2+\cdots}}$, where a_ks and b_ks can be complex numbers. If none of the a_k or b_k are zero $\forall k \in \mathbb{N}$, then using equivalence transformations [22], one can create simpler equivalent forms where either the $b_k = 1$ or the $a_k = 1\ \forall k \in \mathbb{N}$, with $a_0 = 0$ in the latter form. A more concise way to write these two forms is as follows: i) $a_0 + \frac{1}{a_1+\frac{1}{a_2+\cdots}} \equiv a_0 + \frac{1}{a_1+}\frac{1}{a_2+\cdots}$ and ii) $\frac{b_1}{1+\frac{b_2}{1+\cdots}} \equiv \frac{b_1}{1+}\frac{b_2}{1+\cdots}$. Form i) is known as *canonical form*. We will interchangeably use the different forms in the paper based on convenience. One of the nice properties of continued fractions is that in representing any real number with natural number parameters $a_k, b_k \in \mathbb{N}$, the rational approximations formed by any of its finite truncations (termed *convergents*) are closer to the true value than any other rational number with the same or smaller denominator. A continued fraction is therefore the "best" possible rational approximation in this precise sense [22, 30].

In the case where a_k and/or b_k are linear *functions* of a variable $x \in \mathbb{R}^p$, these can be written as functions expressible by power series expansions around $x = 0$, where there is a one-to-one correspondence between the coefficients of both [22, 30]. Hence, given parameter vectors $w_k \in \mathbb{R}^p$, we can write the following equality as functions of x:

$$w_0x + \frac{1}{w_1x+}\frac{1}{w_2x+\cdots} = \sum_{i_1,\ldots,i_p=0}^{\infty} c_{i_1,\ldots,i_p} \prod_{j=1}^{p} x_j^{i_j} \tag{1}$$

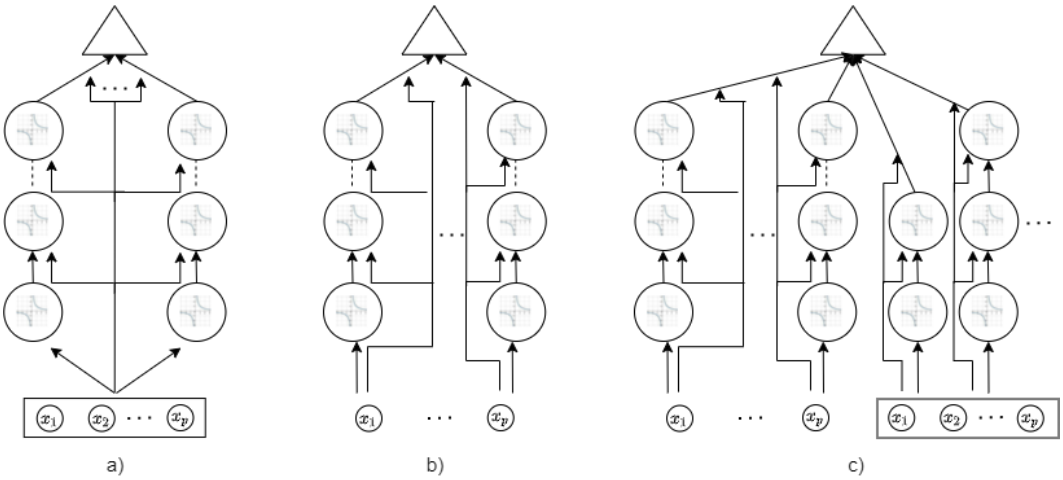


Figure 2: Three variants of the CoFrNet architecture. In all three variants, the output (top triangle) is a linear combination of the ladders below it. a) CoFrNet-F is the full-fledged variant where each ladder receives the whole input x at every stage. b) CoFrNet-D is a diagonalized variant where each ladder only receives one of the input dimensions x_j and hence is an additive model. c) CoFrNet-DL is a combination of the diagonalized variant and the full variant. The full ladders are of increasing depth and can be understood to capture the respective order of interactions. Code for all three variants is provided in the supplement.

for tuples of powers $i_1, ..., i_p$ and complex numbers $c_{i_1,...,i_p}$. We will leverage this relationship in Section 4 as one of the strategies to interpret CoFrNets, since it allows us to express a CF as a power series and hence derive feature attributions for individual features as well as their interactions.

4 CoFrNet Architecture

In this section, we present the proposed architecture based on CFs with three variants. We then discuss aspects of effectively training such architectures and how to obtain feature attributions for interpretation.

Proposed Architecture. We focus on continued fractions in canonical form, with unit numerators $b_k = 1$. As in (1), we let the denominators $a_k = w_k^T x$ be linear functions of the input x. Then with d denoting the depth, the basic continued-fractional function that we work with is

$$f(x; w) = a_0 + \frac{1}{a_1 +} \cdots \frac{1}{a_{d-1} +} \frac{1}{a_d}, \quad a_k = w_k^T x. \tag{2}$$

Each such function corresponds to the diagram in Figure 1. We refer to such a function as a "ladder" due to this pictorial representation with a rail and rungs that carry the input to each node.

We propose three variants of the architecture, shown in Figure 2, where each variant is a linear combination of functions f in (2), i.e., a combination of ladders. We propose CoFrNet-F as a full-fledged variant in which all ladders receive the full input x at each layer. We propose a diagonalized variant CoFrNet-D, in which each ladder operates on a single input dimension x_j, i.e., $f_j(x_j; w^{(j)})$. The linear combination of these ladders is therefore an additive model and directly interpretable [17]. Finally, we propose CoFrNet-DL, which contains both single-feature ladders and full ladders of increasing depth starting at depth two and hence capturing interactions to that order. This variant combines the benefits of the other two architecture variants.

Training Strategies. As discussed in the introduction, we regard the proposed architectures as neural architectures in which the input is passed to all layers, linear functions with weights w_k are computed, and the nonlinear activation function is the reciprocal $z \mapsto 1/z$. All variants are differentiable and can thus be trained using standard techniques such as ADMM and backpropagation and popular frameworks such as TensorFlow or PyTorch. Other commonly used ideas such as dropout [44] could also be leveraged for better generalization.

The most natural choice is to jointly train all the ladders; however, one could envision other training strategies. For example, one could use boosting where one or a group of ladders are first (jointly) trained and where we successively train new ladders on the residuals with appropriate example weighting. One could also incrementally fit each layer within a ladder. Another strategy might be to collapse the linear combination of ladders into a single rational function. However, it may be a challenge to tie the coefficients of this rational function to the weights w_k; not constraining the rational function coefficients in this way would result in an exponential number of coefficients to estimate (exponential in the depth of the ladders).

Handling Poles. A key issue that arises from the $\frac{1}{z}$ reciprocal nonlinearity is that the denominator may go to zero during training, leading to the function being undefined at that point (a *pole* in the context of rational functions). To tackle this issue, we slightly alter the activation function to $\mathsf{sgn}(z)\frac{1}{\max(|z|,\epsilon)}$ for some (small) $\epsilon > 0$, where $|\cdot|$ denotes absolute value. The ϵ can be fixed to a small positive value or tuned during training. Other solutions to this problem involve either restricting each of the denominators to be positive [32] or the final denominator of the rational function to be positive [3]. Both of these constraints can be restrictive as well as computationally challenging.

Interpretability. We now discuss two strategies to interpret the full-fledged version of our architecture CoFrNet-F. Both strategies exploit its functional form. As mentioned, CoFrNet-D is an additive model and can be interpreted by visualizing the univariate functions $f_j(x_j; w^{(j)})$ that compose it.

i) Interpretation using Continuants (IC): It is well-known from the theory of continued fractions [22] that $f(x; w)$ in (2) can be expressed as the following ratio of polynomials,

$$f(x;w) = a_0 + \frac{1}{a_1+} \cdots \frac{1}{a_{d-1}+} \frac{1}{a_d} = \frac{K_{d+1}(a_0, \ldots, a_d)}{K_d(a_1, \ldots, a_d)}, \tag{3}$$

where the polynomials K_k, known as *continuants*, satisfy the recursion

$$K_0 = 1, \qquad K_1(a_d) = a_d, \tag{4}$$

$$K_k(a_{d-k+1}, \ldots, a_d) = a_{d-k+1} K_{k-1}(a_{d-k+2}, \ldots, a_d) + K_{k-2}(a_{d-k+3}, \ldots, a_d). \tag{5}$$

The following result (proven in the supplement) provides a compact expression for the gradient of $f(x; w)$ with respect to the inputs x_j, $j = 1, \ldots, p$.

Proposition 1. *The partial derivative of $f(x; w)$ with respect to x_j is given by*

$$\frac{\partial f(x;w)}{\partial x_j} = \sum_{k=0}^{d} (-1)^k \left(\frac{K_{d-k}(a_{k+1}, \ldots, a_d)}{K_d(a_1, \ldots, a_d)} \right)^2 w_{jk}.$$

Proposition 1 provides a computationally efficient means to compute the gradient of a ladder with respect to its inputs, which is useful for multiple feature-based methods of interpretation [33]. Given an input x, and assuming that the linear functions $a_k = w_k^T x$ have already been computed in evaluating $f(x; w)$, the continuants K_k can be computed using (5) in $O(d)$ operations. Then for each input x_j, the sum in Proposition 1 also requires $O(d)$ operations, for a total of $O(dp)$ operations for all x_j. Proposition 1 additionally suggests an interpretation of the coefficients w_{jk} as contributions to the partial derivative for x_j, weighted by ratios of continuants and with alternating signs.

For linear combinations of ladders f as in Figure 2, the above result extends straightforwardly since differentiation is a linear operation. This yields feature attributions in $O(Ldp)$ time for L ladders.

ii) Interpretation using Power Series (IPS): The above method using continuants gives only first-order attributions. To obtain higher-order terms as well as first-order, we turn to the representation of a ladder in (1) as a multivariate power series, where as mentioned before there is a one-to-one mapping between the coefficients of the two forms. A linear combination of ladders, which our architecture entails, can also be represented by a multivariate power series by summing the coefficients $c_{i_1,\ldots,i_p}$ for each monomial term. These coefficient sums thus provide attributions for individual features x_j as well as higher-order interactions, up to the depth of the ladders.

For low-depth ladders, it is possible to manually equate and find the appropriate coefficients based on a linear recurrence relation [38]. However in general, manual computation can be too laborious. For such cases we recommend using symbolic manipulation tools such as Mathematica [40]. For a function g that is a linear combination of ladders f of depth d, one can obtain the power series expansion up to order dp by applying the following set of Mathematica operations:

$$\text{N}\,[\text{Normal}\,[\text{Series}\,[g, \{x_1, 0, d\}, \cdots, \{x_p, 0, d\}]]] \tag{6}$$

where "Series" produces a Taylor series expansion, "Normal" implies normalized form, and "N" represents fractional coefficients as decimals. The appropriate coefficients can then be picked off to determine feature attributions or attributions for interactions.

5 Universal Approximation

We now prove (Theorem 2) that a linear combination of continued fractions has the property of universal approximation. More precisely, we show this for the family of functions that are linear combinations of a finite number of continued fractions, each with finite depth and linear layers. Our strategy essentially comprises three steps: i) showing polynomials of linear functionals ($\mathcal{PL}$) are a unital subalgebra [4] and separating on the domain, ii) applying the Stone-Weierstrass theorem [45] to show that they are thus dense in the space of bounded continuous functions, and iii) showing that $\mathcal{PL}$ are a subset of the aforementioned class of functions, i.e. finite number of finite-depth continued fractions where each layer is a linear function of the input. This implies the latter class is also dense in the space of bounded continuous functions, a.k.a. universal approximators.

Without loss of generality we consider the domain $\chi = [0,1]^p$ along with the usual Euclidean metric $d(x,y) = \|x-y\|_2,\ x,y \in \chi$. Since χ is bounded and closed, it is a compact metric space. The space of bounded continuous functions $C(\chi,\mathbb{R}) = \{f : \chi \mapsto \mathbb{R} : f \text{ is continuous},\ \exists M \text{ s.t. } |f(x)| \leq M\ \forall x \in \chi\}$. Let $\|f(x)\|_\infty = \max_{x\in\chi}|f(x)|$. Also for the proof we explicitly mention the constant term for each linear function which was subsumed in x in previous sections.

Definition 1. *(Identity Function) Define* $\mathrm{Id}(x)$ *to be the identity function where* $\mathrm{Id}(x) = 1\ \forall x \in \chi$.

Definition 2. $\mathcal{P} \subset C(\chi,\mathbb{R})$ *is a unital subalgebra if a)* $\mathrm{Id}(x) \in \mathcal{P}$*, b)* $\forall f,g \in \mathcal{P}, f * g \in \mathcal{P}$*, c)* $\forall f,g \in \mathcal{P}$ *and* $\alpha,\beta \in \mathbb{R}$*,* $\alpha f + \beta g \in \mathcal{P}$.

Definition 3. $\mathcal{P} \subset C(\chi,\mathbb{R})$ *is separating on the domain* χ *if* $\forall x,y \in \chi,\ x \neq y, \exists f \in \mathcal{P} : f(x) \neq f(y)$.

Theorem 1. *(Stone-Weierstrass Theorem [45]) If* χ *is a compact metric space, and if* $\mathcal{P} \subset C(\chi,\mathbb{R})$ *is a unital subalgebra and separating on the domain* χ*, then* $\mathcal{P}$ *is dense in* $C(\chi,\mathbb{R})$ *with respect to the* ℓ_∞ *metric.*

Definition 4. *(Polynomials of Linear Functions) Define*

$$\mathcal{PL} = \left\{ c_0 + \sum_{S\in\mathcal{S}} c_S \prod_{k\in S} (u_k^T x) : c_0, c_S \in \mathbb{R},\ u_k \in \mathbb{R}^p\ \forall k \in [m],\ \mathcal{S} \subset 2^{[m]},\ m \in \mathbb{N} \right\} \tag{7}$$

to be the set of polynomials on linear functions $u_k^T x$ *of* x.

In the above definition, note that we may have $u_l = u_k$ for $l \neq k$ to obtain higher powers of $u_k^T x$.

Lemma 1. $\mathcal{PL}$ *is a unital subalgebra and is separating on* χ*.* $\mathcal{PL}$ *is dense in* $C(\chi,\mathbb{R})$.

Proof. Let $f(x), g(x) \in \mathcal{PL}$. It is easy to see that $f(x) * g(x) \in \mathcal{PL}$ and $\alpha f(x) + \beta g(x) \in \mathcal{PL}$ for all $\alpha,\beta \in \mathbb{R}$. Further, setting $c_0 = 1$ and $c_S = 0$ in the definition of $\mathcal{PL}$ yields the identity function. Therefore, $\mathcal{PL}$ is a unital subalgebra.

For any two $x \neq y,\ x,y \in \chi$, let $u \in \mathbb{R}^p,\ u \neq 0$ be such that it does not belong to the null space of $x - y$. Such a u can always be found by the rank-nullity theorem applied to the subspace $\mathrm{span}\{x-y\}$ in the vector space $\mathbb{R}^p$. Now, consider $f(s) = c_0 + u^T s \in \mathcal{PL}$. Then, $f(x) \neq f(y)$ since $u^T(x-y) \neq 0$. This shows that $\mathcal{PL}$ is separating in the domain χ. Hence, by Theorem 1, $\mathcal{PL}$ is dense in $C(\chi,\mathbb{R})$. □

Definition 5. *(Continued Fractions with Linear Functions) Let*

$$\mathcal{CFL} = \left\{ \frac{v_0^T x + \alpha_0}{1+} \frac{v_1^T x + \alpha_1}{1 + w_1^T x + \beta_1 +} \frac{v_2^T x + \alpha_2}{1 + w_2^T x + \beta_2 +} \cdots \frac{v_d^T x + \alpha_d}{1 + w_d^T x + \beta_d} : \right.$$
$$\left. v_0, v_k, w_k \in \mathbb{R}^p,\ \alpha_0, \alpha_k, \beta_k \in \mathbb{R},\ 1 \leq k \leq d,\ d \in \mathbb{N} \right\} \tag{8}$$

be the set of finite-depth continued fractions with affine functions $v_k^T x + \alpha_k$ *and* $w_k^T x + \beta_k$ *as numerators and denominators.*

In the above definition, we are explicitly writing the constant terms α_k, β_k for clarity.

Given a set A of functions on χ, let $A \bigoplus A = \{\alpha a + \beta b : a, b \in A, \alpha, \beta \in \mathbb{R}\}$ be the set of linear combinations of two functions from A, and $\bigoplus^L A$ the set of linear combinations of L functions from A.

Theorem 2. *(Representation Theorem)* $\mathcal{PL} \subset \bigcup_{L=1}^{\infty} \bigoplus^L \mathcal{CFL}$. *Also,* $\bigcup_{L=1}^{\infty} \bigoplus^L \mathcal{CFL}$ *is dense in* $C(\chi, \mathbb{R})$.

Proof. By Euler's formula for continued fractions we have:

$$\frac{a_0}{1+}\frac{-a_1}{1+a_1+}\frac{-a_2}{1+a_2+}\cdots\frac{-a_d}{1+a_d} = a_0 + a_0a_1 + \ldots a_0a_1\ldots a_d. \tag{9}$$

By applying the above formula twice to two nested sums, we have:

$$a_0a_1\ldots a_d = \left[\frac{a_0}{1+}\frac{-a_1}{1+a_1+}\frac{-a_2}{1+a_2+}\cdots\frac{-a_d}{1+a_d}\right] - \left[\frac{a_0}{1+}\frac{-a_1}{1+a_1+}\frac{-a_2}{1+a_2+}\cdots\frac{-a_{d-1}}{1+a_{d-1}}\right]. \tag{10}$$

Now to represent a monomial $c\prod_{k\in[d]}(u_k^T x),\ c \in \mathbb{R}$, we observe that we need to set $a_0 = c,\ a_k = (u_k^T x)\ \forall k \in \{1, ..., d\}$ in (10). This in turn can be realized as a member of $\mathcal{CFL}$ by setting $v_0 = 0$, $\alpha_0 = c$, $w_k = -v_k = u_k$, and $\alpha_k = \beta_k = 0$ for $k = 1, \ldots, d$. Hence we have $c\prod_{k\in[d]}(u_k^T x) \in \mathcal{CFL} \bigoplus \mathcal{CFL}$.

Now consider $f(x) = c_0 + \sum_{S\in\mathcal{S}} c_S \prod_{k\in S}(u_k^T x) \in \mathcal{PL}$. Then we have $f \in \bigoplus^{2|\mathcal{S}|+1} \mathcal{CFL}$ by doing a term by term expansion in terms of $\mathcal{CFL} \bigoplus \mathcal{CFL}$. This implies that $\mathcal{PL} \subset \bigcup_{L=1}^{\infty} \bigoplus^L \mathcal{CFL}$. Combining with Lemma 1 implies that $\bigcup_{L=1}^{\infty} \bigoplus^L \mathcal{CFL}$ is dense in $C(\chi, \mathbb{R})$. □

Remark. Note that $\bigcup_{L=1}^{\infty} \bigoplus^L \mathcal{CFL}$ does contain functions with singularities in $\mathbb{R}^p$. This implies that it contains functions that are not in $C(\chi, \mathbb{R})$. But nevertheless, it is dense in $C(\chi, \mathbb{R})$. The presence of singularities is the reason why it is not possible to directly apply proof techniques using the Hahn-Banach theorem [9], which are used for proving representation theorems for two-layer neural networks. It is noteworthy that one only needs linear functions in every layer of a continued fraction and linear combinations of these to represent any bounded function on a compact set.

Compactness of Representation for Learning Sparse Polynomials. If one wants to learn a sparse polynomial in the variables x_j where the number of non-zero monomials $|\mathcal{S}|$ is a constant and degree bounded, i.e. $|S| \leq d,\ \forall S \in \mathcal{S}$, Lasso-based techniques would require a representation which is p^d in size (although sample complexity may be polynomial in $d \log p$) [34]. However, our representation would require only $2|\mathcal{S}| + 1$ parameterized ladders each of depth at most d. Efficient learning of sparse polynomials using such compact representations is an interesting direction for future work.

6 Experiments

We now conduct synthetic and real data experiments. The goal of the synthetic experiments is two fold: i) to show that we can accurately model well known nonlinear functions such as the Matyas function, Rosenbrock function, and Ackley function (list provided here [50]) and more importantly ii) to show that our architecture lends itself to global interpretation using the two strategies described in Section 4. Performance comparisons with MLP of the same depth and with similar number of parameters where 300 points are sampled for each function are in the supplement, as our main intent here is to showcase interpretability.

We then experiment on six real public datasets covering tabular, text and image data. In addition to MLP we compare against four well known interpretable models namely, GAMs (`github.com/dswah/pyGAM`), NAMs (`github.com/nickfrosst/neural_additive_models`),

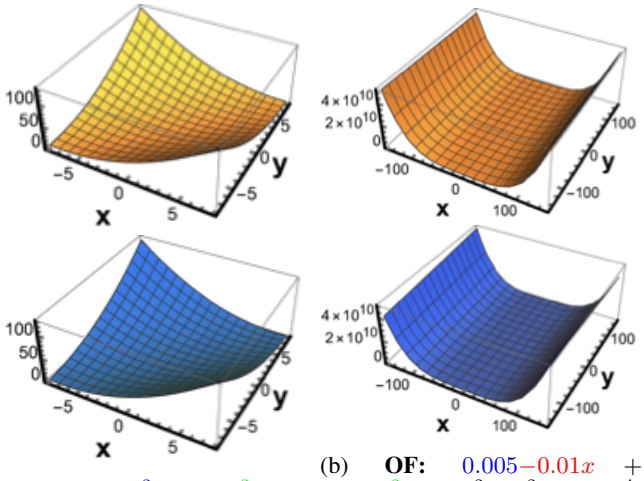
EBMs (`github.com/interpretml/interpret`), and CART decision trees with the main intent being to compare test accuracies. For the text data we used Glove embeddings [37]. We report feature attributions for some of them in the main paper.

We average results over five random train/validation/test splits (65%/5%/30%) for all datasets except those that come with their own pre-specified test set. A two P100 GPU system with 120 GB RAM was used to run the experiments. In the activation function, ϵ was set to 0.1 to mitigate poles.

6.1 Synthetic Experiments

In these experiments we use the CoFrNet-F variant to model the different synthetic functions, and compute feature attributions using the continuants strategy, and the entire function using the power series strategy mentioned in Section 4. A single full ladder is used in each case and the depth is equal to the degree of the function we are approximating. For non-polynomial functions we set the depth to be six. We choose this variant since it is the hardest to interpret; we show that it can be interpreted.

Figure 3 shows two well-known non-linear functions: the Matyas function and the Rosenbrock function. CoFrNet-F is able to accurately approximate them both (7.31% and 13.08% error respectively). Importantly, the IPS interpretation leveraging the one-to-one correspondence to power series is able to replicate the functions quite closely. The constructed interpretation is not only close in prediction, but also in (univariate) feature attribution. For Matyas, we even recover higher order coefficients accurately (possibly due to a better fit). The linear and constant terms have very small coefficients in the Matyas function approximation, making the IPS and original function very similar. Moreover, the feature attributions for the linear terms are also recovered by IC, the closed-form formula involving continuants.



(a) **OF:** $0.54x^2 + 0.54y^2 - xy$, **IPS:** $0.56x^2 + 0.44y^2 - xy + 0.06x - 0.07y - 0.01$, **IC:** $0.06 \to x$, $-0.07 \to y$.

(b) **OF:** $0.005 - 0.01x + 0.005x^2 + 0.5y^2 - x^2y + 0.5x^4$, **IPS:** $0.002 - 0.01x + 0.008x^2 - 0.01x^2y + x^4 - 0.09x^3(3.1 - y)$, **IC:** $0.01 \to x$, $0 \to y$.

Figure 3: Original function (OF; in yellow) and the corresponding IPS approximation (in blue) for the Matyas function (left) and Rosenbrock function (right). The subfigure caption lists the OF, IPS, and feature attributions from IC. The equations for OF and IPS are normalized by maximum coefficient. Coefficients of the same order terms are color-coded for ease of comparison. CoFrNet-F is able to approximate the shape of the functions well. The (univariate) feature attributions for IPS and IC are consistent with the OF.

6.2 Real Data Experiments

We experiment with the Credit Card, Magic and Waveform tabular datasets [11], the sentiment analysis [28] and Quora Insincere Questions [23] text datasets, and the CIFAR10 [24] image dataset. The dataset characteristics are in the supplement. We report performance of the CoFrNet-DL architecture, which was the best performing among the three variants and also CoFrNet-D to show how well our simplest form performs. The depth of the ladders is varied up to 250. For CoFrNet-DL, besides the p univariate ladders, we consider up to 50 full ladders with increasing depth as per the architecture in Figure 2. However, the best performance for most datasets using CoFrNet-DL was obtained using ladders of depth 12 or less.

We observe in Table 1, that the CoFrNet-DL model is competitive with other interpretable models on the tabular datasets (i.e. within few percent) and within 6% of the state-of-the-art (SOTA) black-box models for these datasets (gradient boosted trees) [36]. On images and text, although we are farther off from SOTA black-box models ([16] for CIFAR10 and [51] for text), we are significantly better than other interpretable models, and similar or better than (uninterpretable) MLP. We believe this is

Table 1: Test accuracies of different methods on six real datasets. For datasets without pre-specified test sets, a paired t-test was conducted to determine statistical significance. DNC denotes 'did not converge.' Best results (excluding SOTA) are in bold.

Methods	Interpretable	Waveform	Magic	Credit Card	CIFAR10	Sentiment	Quora
CoFrNet-DL	Yes	0.81	**0.84**	0.69	**0.73**	**0.84**	**0.88**
CoFrNet-D	Yes	0.69	0.76	0.66	0.38	0.80	0.75
GAM	Yes	**0.85**	**0.85**	**0.72**	DNC	0.51	DNC
NAM	Yes	**0.86**	0.81	0.69	0.38	0.50	0.49
EBM	Yes	**0.85**	**0.85**	**0.72**	0.40	0.59	0.49
CART	Yes	0.75	0.79	0.69	0.29	0.52	0.73
MLP	No	0.34	0.65	0.50	0.35	**0.83**	0.85
SOTA	No	0.86	0.86	0.75	0.99	0.96	0.94

because CoFrNets can compactly represent a rich class of functions in high-dimensional space where properties of CFs such as fast convergence are likely witnessed.

We also report the top three feature attributions from CoFrNet-DL for some of the datasets (others in supplement). For Credit Card, these were "amount of bill statement in April, 2005", "the repayment status in September, 2005" and "amount of bill statement in September, 2005", which make sense as the task is to predict defaulting on payments; if someone has not repaid their previous balance, they are likely to default. For the Sentiment dataset, our method provides the most important word positions (more details in supplement). After filtering out articles, prepositions and auxiliary verbs, and noting the most common words in those positions, we obtained the following top three attributions: "good", "like" and "bad". These also make sense as "good" and "like" correspond to positive sentiment and "bad" corresponds to negative sentiment. For the Magic dataset, which categorizes telescope measurements as gamma radiation, the top attributions are: fM3Long (3rd root of 3rd moment along major axis), fSize (sum of content of all pixels) and fLength (major axis of ellipse). Although interpretation requires some knowledge of astrophysics, it seems reasonable that the average intensity of the signal (fSize) and the primary recorded direction (fM3Long and fLength) are the most important features when detecting gamma radiation.

7 Discussion

In this paper, we have proposed CoFrNets, a new neural architecture inspired by continued fractions. We have theoretically shown its universal approximation ability, empirically shown its competence on real-world datasets where we are either competitive or much better than other interpretable models as well as MLPs, and analytically shown how to tease interpretability out from it. The training optimization relies on specific properties of CFs: while CFs are rational functions, optimizing rational functions directly leads to either exponentially-many coefficients or constraints that are difficult to enforce, but these pitfalls do not happen when the CF representation is used. Similarly, interpretation is obtained efficiently and naturally by leveraging the theory of CFs.

We hypothesize that CoFrNets have favorable adversarial robustness properties due to their functional simplicity and we intend to test this in future work. The hypothesis arises from the following argument. Ladders of small depth, which we find to have good accuracy, yield smooth low-degree rational functions that make it difficult for small input perturbations to produce large changes in output. We may even be able to analytically compute adversarial robustness metrics akin to the CLEVER score for CoFrNets [48].

Beyond this initial proposal of a new architecture, we admit there is room for improvement to achieve empirical accuracies that match or outdo state-of-the-art black-box models across modalities. Since CoFrNet can be viewed as a neural architecture, we have been able to exploit the well-developed tools available to train neural architectures. It is possible however that different training strategies such as those mentioned in Section 4 could be advantageous: each ladder could be built rung by rung, or a linear combination of ladders could be built incrementally. Similarly, while the ϵ modification of the $1/z$ function is a practical solution to avoid singularities, it is possible that it also limits the expressiveness of the function, and more advanced ways to defeat poles [3] could be less restrictive. In addition, known tactics from other neural architectures (convolutional blocks, pooling, etc.) or even something new may be warranted. It is possible that convolutional blocks may have a natural implementation based on older filter design literature in signal processing that builds upon CFs [31].

References

[1] R. Agarwal, N. Frosst, X. Zhang, R. Caruana, and G. E. Hinton. Neural additive models: Interpretable machine learning with neural nets. In *arXiv:2004.13912*, 2020.

[2] D. Alvarez-Melis and T. Jaakkola. Towards robust interpretability with self-explaining neural networks. In *Advances in Neural Information Processing Systems*, pages 7775–7784. 2018.

[3] A. P. Austin, M. Krishnamoorthy, S. Leyffer, S. Mrenna, J. Müller, and H. Schulz. Practical algorithms for multivariate rational approximation. *Computer Physics Communications*, 261, 2021.

[4] N. Bourbaki. *Elements of mathematics*. Springer-Verlag, 1989.

[5] L. Breiman, J. Friedman, C. J. Stone, and R. Olshen. *Classification and Regression Trees*. Chapman and Hall, 1984.

[6] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 1721–1730, New York, NY, USA, 2015. ACM.

[7] G. Chrysos, S. Moschoglou, G. Bouritsas, J. Deng, Y. Panagakis, and S. Zafeiriou. Deep polynomial neural networks. In *Intl. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[8] Churchill, Brown, and Verhey. *Complex Variables and Applications*. Springer-Verlag, 1989.

[9] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[10] S. Dash, O. Günlük, and D. Wei. Boolean decision rules via column generation. In *Advances in Neural Information Processing Systems*, pages 4655–4665, 2018.

[11] D. Dheeru and E. Karra Taniskidou. UCI machine learning repository, 2017.

[12] A. Dhurandhar, P.-Y. Chen, R. Luss, C.-C. Tu, P. Ting, K. Shanmugam, and P. Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *Advances in Neural Information Processing Systems*, pages 592–603, 2018.

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Intl. Conference on Learning Representations (ICLR)*, 2021.

[14] Fchollet, Matsuyamax, and Kemaswill. Keras mnist mlp implementation. In *https://github.com/fchollet/keras/blob/master/examples/mnist_mlp.py*, 2017.

[15] Fchollet, Matsuyamax, Smerity, and Kemaswill. Keras mnist cnn implementations. In *https://github.com/fchollet/keras/blob/master/examples/mnist_cnn.py*, 2017.

[16] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *Intl Conference on Learning Representations (ICLR)*, 2021.

[17] T. J. Hastie and R. J. Tibshirani. *Generalized Additive Models*. Chapman and Hall/CRC, 1990.

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Intl. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[19] M. Hind, D. Wei, M. Campbell, N. C. F. Codella, A. Dhurandhar, A. Mojsilovic, K. N. Ramamurthy, and K. R. Varshney. TED: Teaching AI to explain its decisions. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 123–129, 2019.

[20] K. Hornik. Some new results on neural network approximation. *Neural networks*, 6(8):1069–1072, 1993.

[21] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

[22] W. B. Jones and W. Thron. *Continued fractions. Analytic theory and applications*. Encyclopedia of Mathematics and its Applications. Addison-Wesley, 1980.

[23] Kaggle. Quora insincere question dataset, 2019.

[24] A. Krizhevsky. Learning multiple layers of features from tiny images, 2009.

[25] I. Lemhadri, L. A. Feng Ruan, and R. Tibshirani. Lassonet: A neural network with feature sparsity. In *AISTATS*, 2021.

[26] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang. The expressive power of neural networks: A view from the width. In *Advances in Neural Information Processing Systems*, 2017.

[27] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee. From local explanations to global understanding with explainable ai for trees. *Nature Mach. Intl.*, 2020.

[28] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *The 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

[29] Mathologer. How Ramanujan solved the Strand puzzle. https://www.youtube.com/watch?v=V2BybLCmUzs, 2020.

[30] K. Milton. Summation techniques, Padé approximants, and continued fractions. 2011.

[31] S. Mitra and R. Sherwood. Canonic realizations of digital filters using the continued fraction expansion. *IEEE Transactions on Audio and Electroacoustics*, 20(3):185–194, 1972.

[32] A. Molina, P. Schramowski, and K. Kersting. Padé activation units: End-to-end learning of flexible activation functions in deep networks. In *Intl. Conference on Learning Representations (ICLR)*, 2020.

[33] C. Molnar. *Interpretable Machine Learning*. https://christophm.github.io/interpretable-ml-book, 2019.

[34] S. Negahban and D. Shah. Learning sparse boolean polynomials. In *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 2032–2036. IEEE, 2012.

[35] H. Nori, S. Jenkins, P. Koch, and R. Caruana. Interpretml: A unified framework for machine learning interpretability. In *arXiv:1909.09223*, 2019.

[36] T. Pedapati, A. Balakrishnan, K. Shanmugam, and A. Dhurandhar. Learning global transparent models consistent with local contrastive explanations, 2020.

[37] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[38] M. W. Pownall. *Functions and Graphs: Calculus Preparatory Mathematics*. Prentice-Hall, 1983.

[39] K. N. Ramamurthy, B. Vinzamuri, Y. Zhang, and A. Dhurandhar. Model agnostic multilevel explanations, 2020.

[40] W. Research. Wolfram language function, 2020.

[41] C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Mach. Intell.*, 1(5):206–215, May 2019.

[42] S. Sabour, N. Frosst, and G. E. Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, 2017.

[43] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520, 2018.

[44] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[45] M. H. Stone. Applications of the theory of boolean rings to general topology. *Transactions of the American Mathematical Society*, 41(3):375–481, 1937.

[46] H. Swapna Rekha, J. Nayak, and H. S. Behera. Pi-sigma neural network: Survey of a decade progress. In A. K. Das, J. Nayak, B. Naik, S. Dutta, and D. Pelusi, editors, *Computational Intelligence in Pattern Recognition*, pages 429–441, Singapore, 2020. Springer Singapore.

[47] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[48] T.-W. Weng, H. Zhang, P.-Y. Chen, J. Yi, D. Su, Y. Gao, C.-J. Hsieh, and L. Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. In *Proceedings of the International Conference on Learning Representations*, 2018.

[49] Wikipedia. Coffer, 2021.

[50] Wikipedia. Test functions for optimization, 2021.

[51] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Inf. Processing Systems*, 2019.

[52] M. Yu, S. Chang, Y. Zhang, and T. S. Jaakkola. Rethinking cooperative rationalization: Introspective extraction and complement control. In *EMNLP*, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See sections 4, 5 and 6.
   (b) Did you describe the limitations of your work? [Yes] See section 7.
   (c) Did you discuss any potential negative societal impacts of your work? [No] This is a globally interpretable method with risks similar to other methods in this category.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] Main result on universal approximation is proved in section 5. Other result on closed form solution for continuants is in the supplement.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See section 6 and supplement.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See section 6 and supplement.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Statistically significant results based on paired t-test are reported where applicable.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See section 6.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No] All use Apache licence.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Code is provided in supplement.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] Datasets are publicly available.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]