Adaptive wavelet distillation from neural networks through interpretations

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recent deep-learning models have achieved impressive prediction performance, but often sacrifice interpretability and computational efficiency. Interpretability is crucial in many disciplines, such as science and medicine, where models must be carefully vetted or where interpretation is the goal itself. Moreover, interpretable models are concise and often yield computational efficiency. Here, we propose adaptive wavelet distillation (AWD), a method which aims to distill information from a trained neural network into a wavelet transform. Specifically, AWD penalizes feature attributions of a neural network in the wavelet domain to learn an effective multi-resolution wavelet transform. The resulting model is highly predictive, concise, computationally efficient, and has properties (such as a multi-scale structure) which make it easy to interpret. In close collaboration with domain experts, we showcase how AWD addresses challenges in two real-world settings: cosmological parameter inference and molecular-partner prediction. In both cases, AWD yields a scientifically interpretable and concise model which gives predictive performance better than state-of-the-art neural networks. Moreover, AWD identifies predictive features that are scientifically meaningful in the context of respective domains. All code and models are released in a full-fledged package available on Github.[1]

1 Introduction

Recent advancements in deep learning have led to impressive increases in predictive performance. However, the inability to interpret deep neural networks (DNNs) has led them to be characterized as black boxes. It is often critical that models are inherently interpretable [1–3], particularly in high-stakes applications such as medicine, biology, and policy-making. In these cases, interpretations which are relevant to a particular domain/audience [3] can ensure that models behave reasonably, identify when models will make errors, and make the models more amenable to inspection and improvement by domain experts. Moreover, interpretable models tend to be faster and more computationally efficient than large neural networks.

One promising approach to constructing interpretable models without sacrificing prediction performance is model distillation. Model distillation [4] transfers the knowledge in one model (i.e., the teacher), into another model (i.e., the student), where the student model often has desirable properties, such as being more interpretable than the teacher model. Recent works have considered distilling a DNN into inherently interpretable models such as a decision tree [5–7] or a global additive model [8], with some success. Here, we consider distilling a DNN into a learnable wavelet transform, which is a powerful tool to describe signals both in time (spatial) and frequency domains that has found numerous successful applications in physical and biomedical sciences.

[1] Github link anonymized.

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

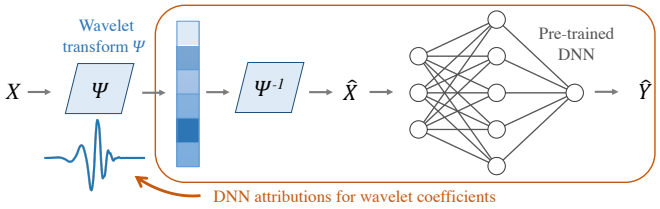

Figure 1: Adaptive wavelet distillation uses attributions from a trained DNN to improve its wavelet transform, while satisfying constraints for reconstruction error and wavelet constraints.

Wavelets have many properties amenable to interpretation: they can form an orthogonal basis, identify a sparse representation of a signal, and tile different frequencies and spatial locations (and sometimes rotations), allowing for multiresolution analysis. Most previous work has focused on hand-designed wavelets for different scenarios rather than wavelets which adapt to given data. Recent work has explored wavelets which adapt to an input data distribution, under the name optimized wavelets or adaptive wavelets [9–16]. Moreover, some work has used wavelets as part of the underlying structure of a neural network, as in wavelet networks [17], wavelet neural networks [18, 19], or the scattering transform [20, 21]. However, none of them utilize wavelets for model interpretability.

Fig 1 outlines Adaptive Wavelet Distillation (AWD), our approach for distilling a wavelet transform from a trained DNN. A key novelty of AWD is that it uses *attributions from a trained DNN* to improve the learned wavelets;[2] this incorporates information not just about the input signals, as is done in previous work, but also about the target variable and the inductive biases present in the DNN.[3]

This paper deviates significantly from a typical NeurIPS paper. While there has been an explosion of work in "interpretable machine learning" [22], there has been very limited development and grounding of these methods *in the context of a particular problem and audience*. This has led to much confusion about how to develop and evaluate interpretation methods [23, 24]; in fact, a major part of the issue is that interpretability cannot be properly defined without the context of a particular problem and audience [3]. As interpretability and scientific machine learning enter a new era, researchers must ground themselves in real-world problems and work closely with domain experts.

This paper focuses on scientific machine learning—providing insight for a particular scientific audience into a chosen scientific problem— and from its outset, was designed to solve a particularly challenging cosmology problem in close collaboration with cosmologists. We showcase how AWD can inform relevant features in a fundamental problem in cosmology: inferring cosmological parameters from weak gravitational lensing convergence maps.[4] In this case, AWD identifies high-intensity peaks in the convergence maps and yields an easily interpretable model which outperforms state-of-the-art neural networks in terms of prediction performance. We next find that AWD successfully provides prediction improvements in another scientific application (now in collaboration with cell-biology experts): molecular-partner prediction. In this case, AWD allows us to vet that the model's use of clathrin corresponds to our domain knowledge about how clathrin must build up slowly then fall in order to predict a successful event. In both cases, the wavelet models from AWD further extract compressed representations of the input in comparison to the standard wavelet model while concisely explaining model behavior. We hope that the depth and grounding of the scientific problems in this work can spur further interpretability research in real-world problems, where interpretability can be evaluated by and enrich domain knowledge, beyond benchmark data contexts such as MNIST [25] where the need for interpretability is less cogent.

2 Background on wavelet transform and TRIM

2.1 Wavelet transform

Wavelets are a class of functions that are localized both in the time and frequency domains. In the classical setting, each wavelet is a variation of a single wavelet ψ, called the *mother wavelet*. A

[2] By attributions, we mean feature importance scores given input data and a pre-trained DNN.

[3] Though we focus on DNNs, AWD works for any black-box models for which we can attain attributions.

[4] For the purpose of this work, we work with simulated lensing maps.

family of discrete wavelets can be created by scaling and translating the mother wavelet in discrete increments:

$$\left\{\psi_{j,n}(t) = \frac{1}{\sqrt{2^j}}\psi\left(\frac{t-2^j n}{2^j}\right)\right\}_{(j,n)\in\mathbb{Z}^2}, \tag{1}$$

where each wavelet in the family $\psi_{j,n}(t)$ represents a unique scale and translation of ψ. With a carefully constructed wavelet ψ (see Appendix A.2), the family of wavelets (1) forms an orthonormal basis of $L^2(\mathbb{R})$. Namely, any signal $x \in L_2(\mathbb{R})$ can be decomposed into

$$x = \sum_n \sum_j d_j[n]\psi_{j,n}, \tag{2}$$

where the wavelet (or detail) coefficients $d_j[n]$ at scale 2^j are computed by taking the inner product with the basis functions, $d_j[n] = \langle x, \psi_{j,n}\rangle = \int x(t)\psi_{j,n}(t)dt$. The decomposition (2) requires an infinite number of scalings to calculate the discrete wavelet transform. To make this decomposition computable, the *scaling function* ϕ is introduced so that

$$x = \sum_n a_J[n]\phi_{J,n} + \sum_n \sum_j^J d_j[n]\psi_{j,n}, \tag{3}$$

where $\phi_{J,n}(t) = 2^{-J/2}\phi(2^{-J}(t-n))$ represent different translations of ϕ at scale 2^J and $a_J[n] = \langle x, \phi_{J,n}\rangle$ are the corresponding approximation coefficients. Conceptually, the $\phi_{J,n}$ form an orthogonal basis of functions that are smoother at the given scale 2^J, and therefore can be used to decompose the smooth residuals not captured by the wavelets [26].

A fundamental property of the discrete wavelet transform is that the approximation and detail coefficients at scale 2^{j+1} can be computed from the approximation coefficients of the previous scale at 2^j [27, 28]. To see this, let us define the two discrete filters, lowpass filter h and highpass filter g

$$h[n] = \langle \frac{1}{\sqrt{2}}\phi(t/2), \phi(t-n)\rangle \quad \text{and} \quad g[n] = \langle \frac{1}{\sqrt{2}}\psi(t/2), \phi(t-n)\rangle. \tag{4}$$

Then the following recursive relations hold between the approximation and detail coefficients at two consecutive resolutions:

$$\begin{cases} a_{j+1}[p] = \sum_n h[n-2p]a_j[n] = a_j \star \bar{h}[2p]; \\ d_{j+1}[p] = \sum_n g[n-2p]a_j[n] = a_j \star \bar{g}[2p], \end{cases} \tag{5}$$

where we denote $\bar{h}[n] = h[-n]$ and $\bar{g}[n] = g[-n]$. Conversely, the approximation coefficients at scale 2^j can be recovered from the coarser-scale approximation and detail coefficients using

$$a_j[p] = \sum_n h[p-2n]a_{j+1}[n] + \sum_n g[p-2n]d_{j+1}[n]. \tag{6}$$

Together, these recursive relations lead to the filter bank algorithm, the cascade of discrete convolution and downsampling, which can be efficiently implemented in time $\mathcal{O}$(Signal length). The discrete wavelet transform can be extended to two dimensions, using a separable (row-column) implementation of 1D wavelet transform along each axis (see Appendix A.1).

2.2 Transformation Importance (TRIM)

The work here requires the ability to compute attributions which identify important features given input data and a trained DNN. Most work on interpreting DNNs has focused on attributing importance to features in the input space of a model, such as pixels in an image or words in a document [29–32]. Instead, here we rely on TRIM (Transformation Importance) [33], an approach which attributes importance to features in a transformed domain (here, the wavelet domain) via a straightforward model reparameterization.

Formally, let f be a pre-trained model that we desire to interpret. If $\Psi : \mathcal{X} \to \mathcal{W}$ is a bijective mapping that maps an input x to a new domain $w = \Psi(x) \in \mathcal{W}$, TRIM reparameterizes the model as $f' = f \circ \Psi^{-1}$, where Ψ^{-1} denotes the inverse of Ψ. In the case that Ψ is not exactly invertible, TRIM adds the residuals to the output of Ψ^{-1}, i.e., f' is reparameterized by $f'(w) = f(\Psi^{-1}w + r)$

where $w = \Psi(x)$ and $r = x - \Psi^{-1}(w)$. If S indexes a subset of features in the transformed space indicating which part of the transformed input to interpret, we then define

$$\text{TRIM}_{\Psi,f}(w_S) = \textit{attr}(f'; w_S), \tag{7}$$

where $\textit{attr}(; w)$ is an attribution method that is evaluated at w and outputs an importance value, and where w_S denotes the subvector of w indexed by S. The choice of the attribution method $\textit{attr}()$ can be any local interpretation technique (e.g. LIME [29] or Integrated Gradients (IG) [31]); here we focus mainly on the saliency map [34], which simply calculates the gradient of the model's output with respect to its transformed input to define feature attribution. We leave more complex attribution methods such as IG or ACD [32] to future work.

3 Adaptive wavelet distillation through interpretations

Adaptive wavelet distillation (AWD) aims to learn a wavelet transform which effectively represents the input data as well as captures information about a model trained to predict a response using the input data. Depending on a problem's context, the resulting wavelet model may or not be sufficiently interpretable for use, or may or not provide similar or better prediction performance as the pre-trained model f. However, we provide two scientific data problems in section 4 where we can do both. In fact, the AWD method has been developed in the context of solving the cosmology problem.

We now detail how AWD wavelets can be built upon to form an extremely simple model in various contexts (see Sec 4). We first require that the wavelet transform is invertible, allowing for reconstruction of the original data. This ensures that the transform does not lose any information in the input data. We next assure that the learned wavelet is a valid wavelet: the wavelet function ψ and the corresponding scaling function ϕ span a sequence of subspaces satisfying the multiresolution axioms [35]. Finally, we add the distillation part of AWD. We calculate the attribution scores of a given model f for each coefficient in the wavelet representation, and try to find a wavelet function ψ that makes these attributions sparse. Intuitively, this ensures that the learned wavelet should find a representation which can concisely explain a model's prediction. Writing the discrete wavelet transform using the discrete filters h and g (see Eq. 4), we now give a final optimization problem for AWD:

$$\underset{h,g}{\text{minimize}}\ \mathcal{L}(h,g) = \underbrace{\frac{1}{m}\sum_i \|x_i - \widehat{x}_i\|_2^2}_{\text{Reconstruction loss}} + \underbrace{\frac{1}{m}\sum_i W(h,g,x_i;\lambda)}_{\text{Wavelet loss}} + \underbrace{\gamma \sum_i \|\text{TRIM}_{\Psi,f}(\Psi x_i)\|_1}_{\text{Interpretation loss}}, \tag{8}$$

where Ψ denotes a wavelet transform operator induced by ψ, and $\widehat{x}_i$ denotes the reconstruction of the data point x_i. Here $\lambda, \gamma > 0$ represent hyperparameters that are tuned by users. The only parameters optimized are the lowpass filter h and the highpass filter g. The corresponding scaling and wavelet functions can be obtained from (h, g) via the following mapping [26]: $\widehat{\phi}(w) = \prod_{p=1}^{\infty} \frac{\widehat{h}(2^{-p}w)}{\sqrt{2}}$ and $\widehat{\psi}(w) = \frac{1}{\sqrt{2}}\widehat{g}(w/2)\widehat{\phi}(w/2)$, where $\widehat{\phi}$ and $\widehat{\psi}$ represent the Fourier transforms of ϕ and ψ respectively.

Wavelet loss The wavelet loss ensures that the learned filters yield a valid wavelet transform. In contrast to the wavelet constraints used in [9], our formulation introduces additional terms that ensure almost sufficient and necessary conditions on the filters (h, g) to build an orthogonal wavelet basis. Specifically, [26, Theorem 7.2] states the following sufficient conditions on the lowpass filter: if h satisfies

$$\sum_n h[n] = \sqrt{2} \quad \text{and} \quad |\widehat{h}(w)|^2 + |\widehat{h}(w+\pi)|^2 = 2 \ \text{ for all } w, \tag{9}$$

as well as some mild conditions, it can generate a scaling function such that the scaled and translated family of the scaling function forms an orthonormal basis of the space of multiresolution approximations of $L^2(\mathbb{R})$. [36, Theorem 3] further shows that the orthogonality of translates of the scaling function implies that the lowpass filter is orthogonal after translates by 2, i.e.,

$$\sum_n h[n]h[n-2k] = \begin{cases} 1 & \text{if } k = 0 \\ 0 & \text{otherwise} \end{cases}, \quad \text{and as a result, } \|h\|_2 = 1. \tag{10}$$

Hence the conditions (9), (10) characterize the almost sufficient and necessary conditions on the lowpass filter. Moreoever, [26, Theorem 7.3] shows that the valid highpass filter can be constructed from the lowpass filter: in the time domain, it can be written as

$$g[n] = (-1)^n h[N-1-n], \tag{11}$$

where N is the support size of h. Together with (10), we can also deduce that the highpass filter has mean zero, i.e., $\sum_n g[n] = 0$ which is necessary for the filter g. See Appendix A.2 for further details.

Finally, we want the learned wavelet to provide sparse representations so we add the ℓ_1 norm penalty on the wavelet coefficients. Combining all these constraints via regularization terms, we define the wavelet loss at the data point x_i as

$$\begin{aligned} W(h, g, x_i; \lambda) = \lambda \|\Psi x_i\|_1 + (\sum_n h[n] - \sqrt{2})^2 + (\sum_n g[n])^2 + (\|h\|_2^2 - 1)^2 \\ + \sum_w (|\widehat{h}(w)|^2 + |\widehat{h}(w+\pi)|^2 - 2)^2 + \sum_k (\sum_n h[n]h[n-2k] - \mathbf{1}_{k=0})^2, \end{aligned}$$

where g is set as in (11) and $\lambda > 0$ controls strength of the sparsity of the wavelet representations. We enforce the penalty $(|\widehat{h}(w)|^2 + |\widehat{h}(w+\pi)|^2 - 2)^2$, only at the discrete values of $w \in \{\frac{2\pi k}{N}, k = 1, \ldots, N\}$ through the discrete Fourier transform. Notice that the wavelet loss does not introduce any additional hyperparameters besides λ. In fact, we empirically observe that the sum of penalty terms, except the sparsity penalty, remains very close to zero as long as the filters (h, g) are initialized using known wavelet filters and the interpretation loss is not enforced too strongly.

Interpretation loss The interpretation loss enables the distillation of knowledge from the pre-trained model f into the wavelet model. It ensures that attributions in the space of wavelet coefficients Ψx_i are sparse, where the attributions of wavelet coefficients is calculated by TRIM [33], as described in Sec 2.2. This forces the wavelet transform to produce representations that concisely explain the model's predictions at different scales and locations. The hyperparameter γ controls the overall contribution of the interpretation loss; large values of γ can result in large numerical differences from satisfying the conditions of the mathematical wavelet filters. To our knowledge, this is the first method which uses interpretations from a pre-trained model to improve a wavelet representation. This enables the wavelets to not only adapt to the distribution of the inputs, but also gain information about the predicted outputs through the lens of the model f.

4 AWD improves interpretability, prediction performance, and compression in two scientific problems

Fig 3 shows a visual schematic of the distillation and prediction setup for one synthetic and two scientific data problems in this section, whose details will be discussed in the following subsections.[5]

4.1 Synthetic data

We begin our evaluation using simulations to verify whether AWD can recover groundtruth wavelets from noisy data. In these simulations, the inputs x_i are generated i.i.d. from a standard Gaussian distribution $\mathcal{N}(0, 1)$. To generate the response variable, the inputs are transformed into the wavelet domain using Daubechies (DB) 5 wavelets [39], and the response is generated from a linear model $y_i = \langle \Psi x_i, \beta \rangle + \epsilon_i$, where the true regression coefficients are 2 for a few selected locations at a particular scale and 0 otherwise; the noise ϵ_i is generated i.i.d. from a Gaussian distribution $\mathcal{N}(0, 0.1^2)$. Then, a 3-layer fully connected neural network with ReLU activations is trained on the pairs of x_i, y_i to accurately predict this response. Note that for any non-singular matrix A, the mapping $x \mapsto \langle A^{-1}\Psi x, A^\top \beta \rangle$ predicts the response equally well, but the representations in the groundtruth wavelet explain the model's prediction most concisely. The challenge is then to accurately distill the groundtruth wavelet (DB5) from this DNN. This task is fairly difficult: AWD must not only select which scale and locations are important, it must also precisely match the shape of h and g to the groundtruth wavelet.

[5] In all experiments, the wavelet function is computed from the corresponding lowpass filter using the *PyWavelets* package [37] and building on the *Pytorch Wavelets* [38, Chapter 3] package.

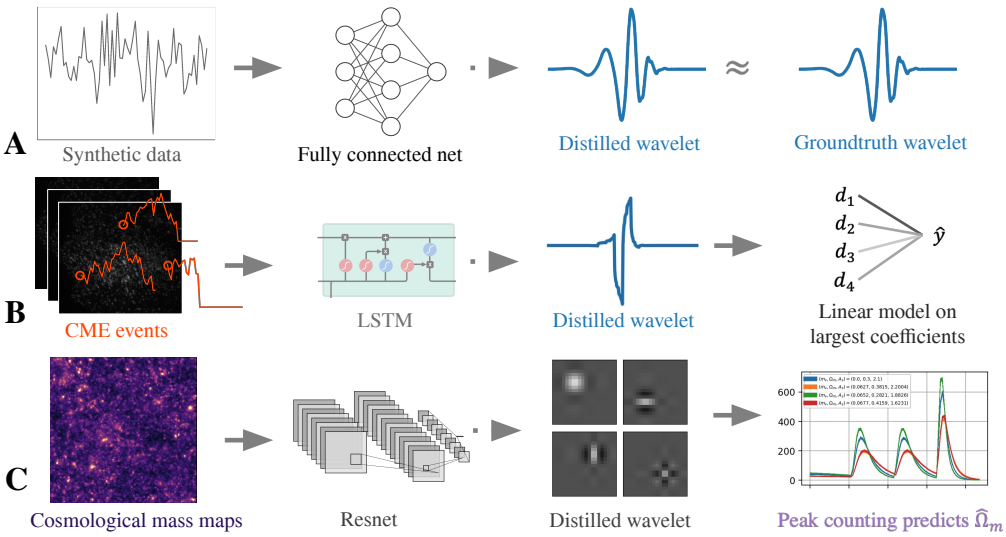

Figure 2: Distillation and prediction setup for the three scenarios in Sec 4. (**A**) In synthetic simulations, AWD is able to recover groundtruth wavelet (DB5) that are linked to a response variable (Sec 4.1). (**B**) Wavelets distilled by AWD from an LSTM trained to predict molecular partners capture biologically meaningful properties of a large build up in clathrin fluorescence followed by a sharp drop and enable prediction using only a few key coefficients (Sec 4.2). (**C**) AWD finds wavelets that are efficient at capturing cosmological information in weak lensing convergence maps and can improve state-of-the-art performance of cosmological parameter inference using an AWD-based simple peak-counting algorithm (Sec 4.3).

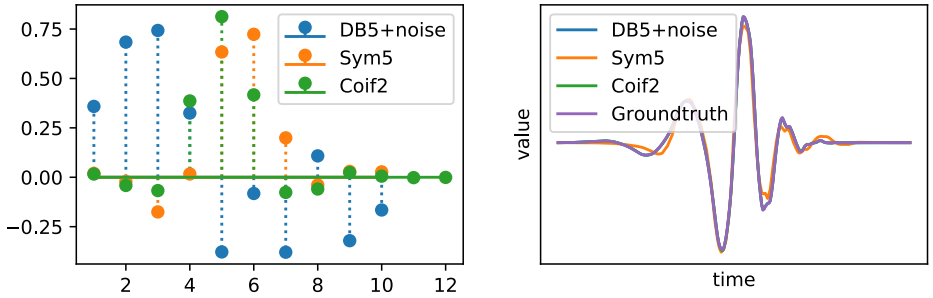

Figure 3: AWD accurately identifies the groundtruth important wavelet in simulated data. (**A**) Plots of the initial lowpass filters. (**B**) Final wavelets extracted by AWD.

Fig 3 shows the performance of AWD on this task. We initialize the AWD lowpass filter to different known lowpass filters corresponding to DB5 (and add noise), Symlet 5, and Coiflet 2 (shown in Fig 3(**A**)) and then optimize the objective given in Eq. 8. For the purpose of recovering the groundtruth, we select hyperparameters λ and γ that minimize the distance to the groundtruth wavelet $\psi^\star$. Distance is measured via $\mathrm{d}(\psi, \psi^\star) = \min\{\min_k \|\psi^k - \psi^\star\|_2, \min_k \|\widetilde{\psi}^k - \psi^\star\|_2\}$, where ψ^k is the wavelet ψ circular shifted by k and $\widetilde{\psi}$ is the wavelet ψ flipped in the left/right direction. That is, d calculates the minimum ℓ_2 distance between two wavelets under circular shifts and left/right flip. When the two wavelets have different size of support, the shorter wavelet is zero-padded to the length of the longer [9]. Fig 3(**B**) shows that for each different initialization, we find that the distilled wavelet gets considerably closer to the groundtruth wavelet. In particular, the results for DB5+noise and Coiflet 2 are nearly identical to the groundtruth and cannot be distinguished in the plot. This is particularly impressive since the support size of Coiflet 2 differs from that of the groundtruth wavelet, making the task more difficult. Overall, these results demonstrate the ability of AWD to distill key information out of a pre-trained neural network.

4.2 Molecular partner-prediction for a central cell biology process

We now turn our attention to a crucial question in cell biology: understanding clathrin-mediated endocytosis (CME) [40]. CME is the primary pathway for entry into the cell, making it essential to eukaryotic life [41]. Many questions about this process remain unanswered, prompting a line of research aiming to better understand this process [42]. One major challenge with analysis of CME is the ability to readily distinguish whether or not a buildup of certain molecules will allow for endocytosis, i.e., successfully transporting an object into a cell. Previous approaches have largely relied on relatively simple thresholds of fluorescence images, based on duration and intensity [43, 44]. However, recent studies have suggested that these features alone may be insufficient to discern successful endocytosis [45, 46].

Here, we address this challenge using a recently collected dataset which tags two molecules: clathrin, which is used as the predictor variable, and auxilin, the target variable. This setup is feasible, as a buildup of clathrin molecules usually precedes endocytosis and a buildup of auxilin molecules follows only when endocytosis successfully occurs (see data details in Appendix C). Time-series of fluorescence amplitudes are extracted from raw cell videos for clathrin (see Fig 2B) and used to predict the mean amplitude of the auxilin signal, a measure of whether an event was successful or not. The dataset is randomly split into a training set of 2,936 data units of dimension 40 and a test set of 1,005 data units. This is a challenging problem where deep learning has recently been shown to outperform classical methods. We train a DNN (an LSTM [47], see architecture in the Supplement) to predict the auxilin response from the clathrin signal. The model predicts well, but is extremely limited in its interpretability, so we aim here to distill it into a wavelet model through AWD.

Fig 4 shows qualitatively how the learned wavelet function ψ changes as a function of the interpretation penalty γ (increasing to the right) and the sparsity penalty λ (increasing downwards). In the initial stage of training, we initialize the lowpass filter to that corresponding to the Daubechies (DB) 5 wavelet. Different combinations of the penalties lead to vastly different learned wavelets, though they all tend to reveal edge-detecting characteristics for a reasonable range of hyperparameter values.

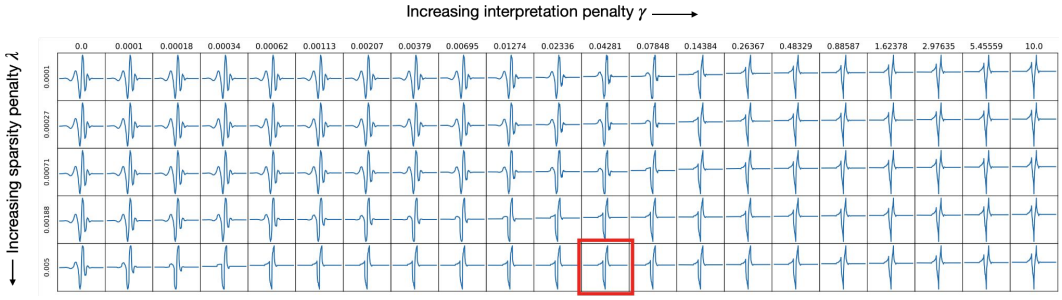

Figure 4: Varying sparsity and interpretation penalty yields different valid wavelets.

We now test the distilled wavelets for their predictive power. To create an extremely transparent model, we extract only the maximum 6 wavelet coefficients at each scale. By taking the maximum coefficients, these features are expected to be invariant to the specific locations where a CME event occurs in the input data. This results in a final model with 30 coefficients (6 wavelet coefficients at 5 scales). These wavelet coefficients are used to train a linear model, and the best hyperparameters are selected via cross-validation on the training set. Fig 2 shows the best learned wavelet (for one particular run) extracted by AWD corresponding to the setting of hyperparameters $\lambda = 0.005$ and $\gamma = 0.043$. Table 1 compares the results for AWD and the original LSTM/the initialized, non-adaptive DB5 wavelet model, where the performance is measured via a standard R^2 score, a proportion of variance in the response that is explained by the model. The AWD model not only closes the gap between the standard wavelet model (DB5) and the neural network, it considerably improves the LSTM's performance (a 10% increase of R^2 score). Moreover, we calculate the compression rates of the AWD wavelet and DB5—these rates measure the proportion of wavelet coefficients in the test set, in which the magnitude and the attributions are both above 10^{-3}. The AWD wavelet exhibits much better compression than DB5 (an 18% reduction), showing the ability of AWD to simultaneously provide sparse representations and explain the LSTM's predictions concisely. The AWD model

also dramatically decreases the computation time at test time, a more than 200-fold reudction when compared to LSTM.

In addition to improving prediction accuracy, AWD enables domain experts to vet their experimental pipelines by making an AWD model transparent. Interestingly, the learned wavelet resembles a canonical pattern which is present in successful events but not in abortive events (see Fig B1); this translates to the biological build-up and dissipation of clathrin molecules. Overall, AWD allows for checking what clathrin signatures signal a CME event and indicates that the AWD model concentrates on identifying a large buildup in clathrin fluorescence (corresponding to the building of a clathrin-coated pit) followed by sharp drop in clathrin fluorescence (corresponding to the rapid deconstruction of the pit). This domain knowledge is extracted from the pre-trained LSTM model by AWD using only the saliency interpretations in the wavelet space.

Table 1: Performance comparisons for different models: AWD wavelet, DB5, LSTM. AWD substantially improves predictive accuracy, compression rate, and computation time on the test set. The higher R^2 score, and lower compression rate and computation time indicate better results. For AWD, the values are averaged over 5 different runs of random seed.

	AWD (Ours)	Standard Wavelet (DB5)	LSTM
Regression (R^2 score)	**0.262 (0.001)**	0.197	0.237
Compression rate	**0.574 (0.010)**	0.704	N/A
Computation time	**0.0002s**	**0.0002s**	0.0449s

4.3 Estimating a fundamental parameter surrounding the origin of the universe

We now focus on a cosmology problem, where AWD helps replace DNNs with a more interpretable alternative. Specifically, we consider weak gravitational lensing convergence maps, i.e., maps of the mass distribution in the universe integrated up to a certain distance from the observer. In a cosmological experiment (e.g. a galaxy survey), these mass maps are obtained by measuring the distortion of distant galaxies caused by the deflection of light by the mass between the galaxy and the observer [48]. These maps contain a wealth of physical information of interest, such as the total matter density in the universe, Ω_m. Current cosmology research aims to identify the most informative features in these maps for inferring the cosmological parameters such as Ω_m. The traditional summary statistic for lensing maps is the power spectrum which is known to be sub-optimal for parameter inference. Tighter parameter constraints can be obtained by including higher-order statistics, such as the bispectrum [49] and peak counts [50]. However, DNN-based inference methods claim to improve on constraints based on these traditional summaries [51–53].

Here, we aim to improve the predictive power of DNN-based methods while gaining interpretability by distilling a predictive AWD model. In this context, it is critically important to obtain interpretability, as it provides deeper understanding into what information is most important to infer Ω_m and can be used to design optimal experiments or analysis methods. Moreover, because these models are trained on numerical simulations (realizations of the Universe with different cosmological parameters), it is important to validate that the model uses reasonable features rather than latching on to numerical artifacts in the simulations. We start by training a model to predict Ω_m from simulated weak gravitational lensing convergence maps. We train a DNN[6] to predict Ω_m from 100,000 mass maps simulated with 10 different sets of cosmological parameter values at the universe origin from the `MassiveNuS` simulations [55] (full simulation details given in Appendix D), achieving an R^2 value of 0.92 on the test set (10,000 mass maps); Fig 2C shows an example mass map.

We again construct an interpretable model using the wavelets distilled by AWD from the trained DNN. To make predictions, we use the simple peak counting algorithm developed in a previous work [52], which convolves a peak-finding filter with the input images. Then, these peaks are used to regress on the outcome. In contrast to the fixed filters such as Laplace or Roberts cross used in previous works [52], here we use the wavelets distilled by AWD, which result in three 2D wavelet filters (LL, LH, HL) and the 2D approximation filter (LL). The size of the distilled AWD filters is 12×12 and inspection of these filters shows a majority of the mass is concentrated on 3×3 subfilters (see Fig 2C). Then we extract those subfilters to use for peak-finding filters—by doing so, the size of the filters

[6]The model's architecture is Resnet 18 [54], modified to take only one input channel.

match with those used in [52] (additional details given in Appendix D.1). The hyperparameters for AWD are selected by evaluating the predictive model's performance on a held-out validation set.

Table 2 shows the results of predicting using the peak-finding algorithm with various filters. The evaluation metric is the RMSE (Root mean square error). Its performance again outperforms the fully trained neural network (Resnet) model and the standard non-adaptive wavelet (DB5) model, as well as other baseline methods using Laplace filter and Roberts cross filter (see Appendix D.1 for details on how these filters are defined). Moreover, as can be seen in the compression rate, the AWD wavelet provides more efficient representations for the mass maps as well as concise explanation for the DNN's predictions compared to the DB 5 wavelet.

Table 2: Performance comparisons for different models in cosmological parameter prediction. The lower RMSE and compression rate indicate better results. For RMSE, standard deviations are estimated from $10,000$ bootstrap samples.

	AWD (Ours)	Roberts-Cross	Laplace	DB5 Wavelet	Resnet
Regression (RMSE $\times 10^{-2}$)	**1.029** **(0.033)**	1.259 (0.039)	1.369 (0.047)	1.569 (0.048)	1.156 (0.024)
Compression rate	**0.610**	N/A	N/A	0.620	N/A

Fig 2C shows the learned AWD filters corresponding to the best distilled wavelet. The learned wavelet filters are symmetric and resemble the "matched filters" which have been used in the past to identify peaks on convergence maps in the cosmology literature [56, 57]. We expect from cosmology knowledge that much information is contained in the peaks of the convergence maps (their amplitude, shape, and numbers), so this indeed matches our expectations based on physics. The high predictive performance further demonstrates that the AWD filters are more efficient at capturing cosmological information and better adapted to the shape of the peaks, than standard wavelets could do.

Moreover, the adaptive wavelet distillation allows us to look at "wavelet activation maps" (see Fig C2) to localize on locations in the convergence maps where important information is concentrated. In other words, we can indeed see that the AWD wavelet concentrates on identifying high intensity peaks, which is where most of the "localized" information is expected from theory.

5 Discussion

In this work, we introduce AWD, a method to distill adapative wavelets from a pre-trained supervised model such as DNNs for interpretation. Doing so enables AWD to automatically detect and adapt to aspects of data that are important for prediction in an interpretable manner. The benefits of distilling relevant predictive information captured in a DNN are demonstrated through applications to synthetic and real data in two scientific settings. Overall, AWD allows us to interpret DNNs in terms of conventional wavelets, bringing interpretability with domain insights while simultaneously improving compression and computational costs, all while preserving or improving predictive power.

Future work Here, we test our method with the saliency attribution method; many advanced interpretation techniques have been developed in the past years and the comparison between different interpretation techniques must be carefully explored in the context of a particular problem and audience. When optimizing the objective Eq. 8 via gradient descent, it requires the gradient of the gradient, which is computationally expensive especially for large data and network sizes. A wavelet-based distillation approach that is computationally more amenable is an interesting direction for future research. The current work learns a single-layer wavelet transform, but the complex nature of modern datasets often require strong nonlinearities. Future work could extend AWD beyond a single-layer wavelet transform, e.g. by borrowing ideas from scattering transform [21]. This would allow for bridging closer to deep learning while keeping interpretability, which can be effectively applied to other areas, such as computer vision and natural-image classification. We hope to continue this line of research in order to improve the interpretability and computational efficiency of DNN models across many domains ranging from physical and biomedical sciences to computer vision and information technology.

References

[1] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019. (Cited on page 1.)

[2] Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *arXiv preprint arXiv:2103.11251*, 2021. (Cited on page 1.)

[3] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019. (Cited on pages 1 and 2.)

[4] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. (Cited on page 1.)

[5] Nicholas Frosst and Geoffrey Hinton. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784*, 2017. (Cited on page 1.)

[6] Mark Craven and Jude W Shavlik. Extracting tree-structured representations of trained networks. In *Advances in neural information processing systems*, pages 24–30, 1996. (Cited on page 1.)

[7] Jiawei Li, Yiming Li, Xingchun Xiang, Shu-Tao Xia, Siyi Dong, and Yun Cai. Tnt: An interpretable tree-network-tree learning framework using knowledge distillation. *Entropy*, 22(11):1203, 2020. (Cited on page 1.)

[8] Sarah Tan, Rich Caruana, Giles Hooker, Paul Koch, and Albert Gordo. Learning global additive explanations for neural nets using model distillation. *arXiv preprint arXiv:1801.08640*, 2018. (Cited on page 1.)

[9] Daniel Recoskie and Richard Mann. Learning sparse wavelet representations. *arXiv preprint arXiv:1802.02961*, 2018. (Cited on pages 2, 4, and 6.)

[10] Phil Sallee and Bruno Olshausen. Learning sparse multiscale image representations. *Advances in neural information processing systems*, 15:1351–1358, 2002. (Cited on page 2.)

[11] Jing Lin and MJ Zuo. Gearbox fault diagnosis using adaptive wavelet filter. *Mechanical systems and signal processing*, 17(6):1259–1269, 2003. (Cited on page 2.)

[12] Laurent Brechet, Marie-Françoise Lucas, Christian Doncarli, and Dario Farina. Compression of biomedical signals with mother wavelet optimization and best-basis wavelet packet selection. *IEEE Transactions on Biomedical Engineering*, 54(12):2186–2192, 2007. (Cited on page 2.)

[13] James M Hereford, David W Roach, and Ryan Pigford. Image compression using parameterized wavelets with feedback. In *Independent Component Analyses, Wavelets, and Neural Networks*, volume 5102, pages 267–277. International Society for Optics and Photonics, 2003. (Cited on page 2.)

[14] Daniel Recoskie. Learning sparse orthogonal wavelet filters. 2018. (Cited on page 2.)

[15] Dhruv Jawali, Abhishek Kumar, and Chandra Sekhar Seelamantula. A learning approach for wavelet design. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5018–5022. IEEE, 2019. (Cited on page 2.)

[16] Cheng Tai and E Weinan. Multiscale adaptive representation of signals: I. the basic framework. *The Journal of Machine Learning Research*, 17(1):4875–4912, 2016. (Cited on page 2.)

[17] Qinghua Zhang and Albert Benveniste. Wavelet networks. *IEEE transactions on Neural Networks*, 3(6):889–898, 1992. (Cited on page 2.)

[18] Jun Zhang, Gilbert G Walter, Yubo Miao, and Wan Ngai Wayne Lee. Wavelet neural networks for function learning. *IEEE transactions on Signal Processing*, 43(6):1485–1497, 1995. (Cited on page 2.)

[19] J. Dheeba, N. Albert Singh, and S. Tamil Selvi. Computer-aided detection of breast cancer on mammograms: A swarm intelligence optimized wavelet neural network approach. *Journal of Biomedical Informatics*, 49:45–52, 2014. (Cited on page 2.)

[20] Stéphane Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012. (Cited on page 2.)

[21] Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1872–1886, 2013. (Cited on pages 2 and 9.)

[22] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2019. (Cited on page 2.)

[23] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, pages 9505–9515, 2018. (Cited on page 2.)

[24] Finale Doshi-Velez and Been Kim. A roadmap for a rigorous science of interpretability. *arXiv preprint arXiv:1702.08608*, 2017. (Cited on page 2.)

[25] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998. (Cited on page 2.)

[26] Stéphane Mallat. *A wavelet tour of signal processing, Third edition: The sparse way*. Academic Press, 2008. (Cited on pages 3, 4, and 5.)

[27] Stephane Mallat. Multiresolution approximations and wavelet orthonormal bases of $L^2(R)$. *Transactions of the American mathematical society*, 315(1):69–87, 1989. (Cited on page 3.)

[28] Yves Meyer. *Wavelets and Operators: Volume 1*. Number 37. Cambridge university press, 1992. (Cited on page 3.)

[29] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016. (Cited on pages 3 and 4.)

[30] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4768–4777, 2017. (Cited on page 3.)

[31] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *ICML*, 2017. (Cited on pages 3 and 4.)

[32] Chandan Singh, W. James Murdoch, and Bin Yu. Hierarchical interpretations for neural network predictions. In *International Conference on Learning Representations*, 2019. (Cited on pages 3 and 4.)

[33] Chandan Singh, Wooseok Ha, Francois Lanusse, Vanessa Boehm, Jia Liu, and Bin Yu. Transformation importance with applications to cosmology. *arXiv preprint arXiv:2003.01926*, 2020. (Cited on pages 3 and 5.)

[34] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. (Cited on page 4.)

[35] Stephane Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693, 1989. (Cited on page 4.)

[36] Sidney Burrus, Ramesh Gopinath, and Haitao Guo. Introduction to wavelets and wavelet transforms: a primer. *Englewood Cliffs*, 1997. (Cited on page 4.)

[37] Gregory Lee, Ralf Gommers, Filip Waselewski, Kai Wohlfahrt, and Aaron O'Leary. Pywavelets: A python package for wavelet analysis. *Journal of Open Source Software*, 4(36):1237, 2019. (Cited on page 5.)

[38] Fergal Cotter. *Uses of Complex Wavelets in Deep Convolutional Neural Networks*. PhD thesis, University of Cambridge, 2020. (Cited on page 5.)

[39] I DAUBECHIES. Orthonormal bases of compactly supported wavelets. *Commun. Pure Appl. Math.*, 41:909–996, 1988. (Cited on page 5.)

[40] Tom Kirchhausen, David Owen, and Stephen C Harrison. Molecular structure, function, and dynamics of clathrin-mediated membrane traffic. *Cold Spring Harbor perspectives in biology*, 6(5):a016725, 2014. (Cited on pages 7 and 15.)

[41] Harvey T McMahon and Emmanuel Boucrot. Molecular mechanism and physiological functions of clathrin-mediated endocytosis. *Nature reviews Molecular cell biology*, 12(8):517, 2011. (Cited on pages 7 and 15.)

[42] Marko Kaksonen and Aurélien Roux. Mechanisms of clathrin-mediated endocytosis. *Nature Reviews Molecular Cell Biology*, 19(5):313, 2018. (Cited on pages 7 and 15.)

[43] François Aguet, Costin N Antonescu, Marcel Mettlen, Sandra L Schmid, and Gaudenz Danuser. Advances in analysis of low signal-to-noise images link dynamin and ap2 to the functions of an endocytic checkpoint. *Developmental cell*, 26(3):279–291, 2013. (Cited on pages 7 and 15.)

[44] Zuzana Kadlecova, Stephanie J Spielman, Dinah Loerke, Aparna Mohanakrishnan, Dana Kim Reed, and Sandra L Schmid. Regulation of clathrin-mediated endocytosis by hierarchical allosteric activation of ap2. *Journal of Cell Biology*, 216(1):167–179, 2017. (Cited on pages 7 and 15.)

[45] Kangmin He, Eli Song, Srigokul Upadhyayula, Song Dang, Raphael Gaudin, Wesley Skillern, Kevin Bu, Benjamin R Capraro, Iris Rapoport, Ilja Kusters, et al. Dynamics of auxilin 1 and gak in clathrin-mediated traffic. *Journal of Cell Biology*, 219(3), 2020. (Cited on pages 7, 15, and 16.)

[46] Xinxin Wang, Zhiming Chen, Marcel Mettlen, Jungsik Noh, Sandra L Schmid, and Gaudenz Danuser. Dasc, a sensitive classifier for measuring discrete early stages in clathrin-mediated endocytosis. *eLife*, 9:e53686, 2020. (Cited on pages 7 and 15.)

[47] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. (Cited on pages 7 and 16.)

[48] Matthias Bartelmann and Peter Schneider. Weak gravitational lensing. *Physics Reports*, 340(4-5):291–472, 2001. (Cited on page 8.)

[49] William R Coulton, Jia Liu, Mathew S Madhavacheril, Vanessa Böhm, and David N Spergel. Constraining neutrino mass with the tomographic weak lensing bispectrum. *Journal of Cosmology and Astroparticle Physics*, 2019(05):043, 2019. (Cited on page 8.)

[50] Zack Li, Jia Liu, José Manuel Zorrilla Matilla, and William R Coulton. Constraining neutrino mass with tomographic weak lensing peak counts. *Physical Review D*, 99(6):063527, 2019. (Cited on page 8.)

[51] Dezső Ribli, Bálint Ármin Pataki, José Manuel Zorrilla Matilla, Daniel Hsu, Zoltán Haiman, and István Csabai. Weak lensing cosmology with convolutional neural networks on noisy data. *Monthly Notices of the Royal Astronomical Society*, 490(2):1843–1860, 2019. (Cited on page 8.)

[52] Dezső Ribli, Bálint Ármin Pataki, and István Csabai. An improved cosmological parameter inference scheme motivated by deep learning. *Nature Astronomy*, 3(1):93, 2019. (Cited on pages 8 and 9.)

[53] Janis Fluri, Tomasz Kacprzak, Aurelien Lucchi, Alexandre Refregier, Adam Amara, Thomas Hofmann, and Aurel Schneider. Cosmological constraints with deep learning from kids-450 weak lensing maps. *Physical Review D*, 100(6):063514, 2019. (Cited on page 8.)

[54] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. (Cited on page 8.)

[55] Jia Liu, Simeon Bird, José Manuel Zorrilla Matilla, J. Colin Hill, Zoltán Haiman, Mathew S. Madhavacheril, Andrea Petri, and David N. Spergel. MassiveNuS: cosmological massive neutrino simulations. *JCAP*, 2018(3):049, 2018. (Cited on pages 8 and 18.)

[56] Matteo Maturi, Massimo Meneghetti, Matthias Bartelmann, Klaus Dolag, and Lauro Moscardini. An optimal filter for the detection of galaxy clusters through weak lensing. *Astronomy & Astrophysics*, 442(3):851–860, 2005. (Cited on page 9.)

[57] Fabian Schmidt and Eduardo Rozo. Weak-lensing peak finding: Estimators, filters, and biases. *The Astrophysical Journal*, 735(2):119, 2011. (Cited on page 9.)

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] Our claim about interpretability and predictive performance of AWD is demonstrated through experimental results in Sec 4.
   (b) Did you describe the limitations of your work? [Yes] See Sec 5 Discussion: Limitations.

(c) Did you discuss any potential negative societal impacts of your work? [No] Our method does not have any negative societal impacts that we can foresee in the future.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] We have carefully read the ethics review guidelines and our paper conforms to them.

2. If you are including theoretical results...

   (a) Did you state the full set of assumptions of all theoretical results? [N/A]

   (b) Did you include complete proofs of all theoretical results? [N/A]

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] All code and results are released on Github (URL link anonymized) to reproduce the experimental results.

   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Sec 4 for a brief discussion about the training details; and see Appendix B, Appendix C, Appendix D for further details on synthedic data experiments, biology, and cosmology problems, respectively.

   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Sec 4; in Sec 4.2 the results are averaged over multiple runs with respect to the random seed; in Sec 4.3 standard error is estimated from $10,000$ bootstrap samples.

   (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 Appendix.

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] For our code, we used a few open-source python libraries developed by researchers and cited them in Sec 4 (see footnote).

   (b) Did you mention the license of the assets? [N/A]

   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

   (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]