RAVEL: Evaluating Interpretability Methods on Disentangling Language Model Representations

Jing Huang
Stanford University
hij@stanford.edu

Zhengxuan Wu
Stanford University
wuzhengx@stanford.edu

Christopher Potts
Stanford University
cgpotts@stanford.edu

Mor Geva
Tel Aviv University
morgeva@tauex.tau.ac.il

Atticus Geiger
Pr(Ai)^{2}R Group
atticusg@gmail.com

Abstract

Individual neurons participate in the representation of multiple high-level concepts. To what extent can different interpretability methods successfully disentangle these roles? To help address this question, we introduce **RAVEL** (Resolving Attribute–Value Entanglements in Language Models), a dataset that enables tightly controlled, quantitative comparisons between a variety of existing interpretability methods. We use the resulting conceptual framework to define the new method of Multi-task Distributed Alignment Search (MDAS), which allows us to find distributed representations satisfying multiple causal criteria. With Llama2-7B as the target language model, MDAS achieves state-of-the-art results on RAVEL, demonstrating the importance of going beyond neuron-level analyses to identify features distributed across activations. We release our benchmark at https://github.com/explanare/ravel.

1 Introduction

A central goal of interpretability is to localize an abstract concept to a component of a deep learning model that is used during inference. However, this is not as simple as identifying a neuron for each concept, because neurons are *polysemantic* – they represent multiple high-level concepts (Smolensky, 1988; Rumelhart et al., 1986; McClelland et al., 1986; Olah et al., 2020; Cammarata et al., 2020; Bolukbasi et al., 2021; Gurnee et al., 2023).

Several recent interpretability works (Bricken et al., 2023; Cunningham et al., 2024; Geiger et al., 2023b; Wu et al., 2023) tackle this problem using a *featurizer* that disentangles the activations of polysemantic neurons by mapping to a space of *monosemantic* features that each represent a distinct concept. Intuitively, these methods should have a significant advantage over approaches that identify concepts with sets of neurons. However, these methods have not been benchmarked.

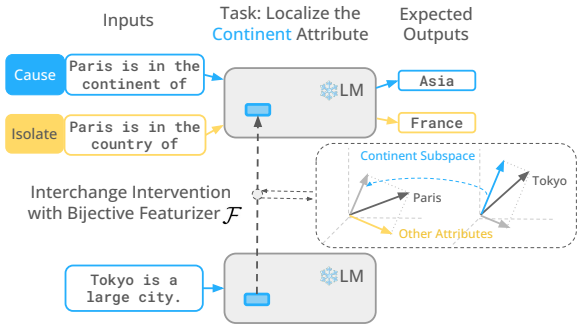


Figure 1: An overview of the RAVEL benchmark, which evaluates how well an interpretability method can find features that isolate the causal effect of individual attributes of an entity.

To facilitate these method comparisons, we introduce a diagnostic benchmark, **RAVEL** (Resolving Attribute–Value Entanglements in Language Models). RAVEL evaluates interpretability methods on their ability to localize and disentangle the attributes of different types of entities encoded as text inputs to language models (LMs). For example, the entity type "city" has instances such as "Paris" or "Tokyo", which each have attributes for "continent", namely "Europe" and "Asia". An interpretability method must localize this attribute to a group of neurons $\mathbf{N}$, learn a featurizer $\mathcal{F}$ (e.g., a rotation matrix or sparse autoencoder), and identify a feature F (e.g., a linear subspace of the residual stream in a Transformer) for the attribute. RAVEL contains five types of entities (cities, people names, verbs, physical objects, and occupations), each with at least 500 instances, at least 4 attributes, and at least 50 prompt templates per entity type.

The metric we use to assess interpretability methods is based on interchange interventions (also known as activation patching). This operation has emerged as a workhorse in interpretability, with a wide swath of research applying the technique to test if a high-level concept is stored in a model representation and used during inference (Geiger

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 8669–8687

et al., 2020; Vig et al., 2020; Geiger et al., 2021; Li et al., 2021; Finlayson et al., 2021; Meng et al., 2022; Chan et al., 2022; Geva et al., 2023; Wang et al., 2023; Hanna et al., 2023; Conmy et al., 2023; Goldowsky-Dill et al., 2023; Hase et al., 2023; Todd et al., 2024; Feng and Steinhardt, 2024; Cunningham et al., 2024; Huang et al., 2023; Tigges et al., 2023; Lieberum et al., 2023; Davies et al., 2023; Hendel et al., 2023; Ghandeharioun et al., 2024).

Specifically, we use the LM to process a prompt like "`Paris is in the continent of`" and then intervene on the neurons $\mathbf{N}$ to fix the feature F to be the value it would have if the LM were given a prompt like "`Tokyo is a large city.`" If this leads the LM to output "`Asia`" instead of "`Europe`", then we have evidence that the feature F encodes the attribute "continent". Then, we perform the same intervention when the LM processes a prompt like "`People in Paris speak`". If the LM outputs "`French`" rather than "`Japanese`', then we have evidence that the feature F has disentangled the attributes "continent" and "language".

A variety of existing interpretability methods are easily cast in the terms needed for RAVEL evaluations, including supervised probes (Peters et al., 2018; Hupkes et al., 2018; Tenney et al., 2019; Clark et al., 2019), Principal Component Analysis (Tigges et al., 2023; Marks and Tegmark, 2023), Differential Binary Masking (DBM: Cao et al. 2020; Csordás et al. 2021; Cao et al. 2022; Davies et al. 2023), sparse autoencoders (Bricken et al., 2023; Cunningham et al., 2024), and Distributed Alignment Search (DAS: Geiger et al. 2023b; Wu et al. 2023). Our apples-to-apples comparisons reveal conceptual similarities between the methods.

In addition, we propose multi-task training objectives for DBM and DAS. These objectives allow us to find representations satisfying multiple causal criteria, and we show that Multi-task DAS is the most effective of all the methods we evaluate at identifying disentangled features. This contributes to the growing body of evidence that interpretability methods need to identify features that are distributed across neurons.

2 The RAVEL Dataset

The design of RAVEL is motivated by four high-level desiderata for interpretability methods:

1. **Faithful**: Interpretability methods should accurately represent the model to be explained.
2. **Causal**: Interpretability methods should analyze the causal effects of model components on model input–output behaviors.
3. **Generalizable**: The causal effects of the identified components should generalize to similar inputs that the underlying model makes correct predictions for.
4. **Isolating individual concepts**: Interpretability methods should isolate causal effects of individual concepts involved in model behaviors.

Entity Type	Attributes	# Entities	# Prompt Templates
City	Country, Language, Latitude, Longitude, Timezone, Continent	3552	150
Nobel Laureate	Award Year, Birth Year, Country of Birth, Field, Gender	928	100
Verb	Definition, Past Tense, Pronunciation, Singular	986	60
Physical Object	Biological Category, Color, Size, Texture	563	60
Occupation	Duty, Gender Bias, Industry, Work Location	799	50

Table 1: Types of entities and attributes in RAVEL.

The goal of RAVEL is to assess the ability of methods to isolate individual explanatory factors in model representations (desideratum 4), and do so in a way that is faithful to how the target models work (desideratum 1). The dataset train/test structure seeks to ensure that methods are evaluated by how well their explanations generalize to new cases (desideratum 3), and RAVEL is designed to support intervention-based metrics that assess the extent to which methods have found representations that causally affect the model behavior (desideratum 2).

RAVEL is carefully curated as a diagnostic dataset for the attribute disentanglement problem. RAVEL has five types of entity, where every instance has every attribute associated with its type. Table 1 provides an overview of RAVEL's structure.

The Attribute Disentanglement Task We begin with a set of entities $\mathcal{E} = \{E_1, \ldots, E_n\}$, each with attributes $\mathcal{A} = \{A_1, \ldots, A_k\}$, where the correct value of A for E is given by A_E. Our interpretability task asks whether we can find a feature F that encodes the attribute A separately from the other attributes $\mathcal{A} \setminus \{A\}$. For Transformer-based models (Vaswani et al., 2017), a feature might be a dimension in a hidden representation of an MLP or a linear subspace of the residual stream.

We do not know a priori the degree to which it is

possible to disentangle a model's representations. However, our benchmark evaluates interpretability methods according to the desiderata given above and so methods will need to be faithful to the model's underlying structure to succeed. In other words, assuming methods are faithful, we can favor methods that achieve more disentanglement.

2.1 Data Generation

Selecting Entity Types and Attributes We first identify entity types from existing datasets that potentially have thousands of instances (see Appendix A.1), such as cities or famous people. Moreover, each entity type has multiple attributes with different degrees and types of associations. For example, for attributes related to city, "country" entails "continent", but not the reverse; "country" is predictable from "timezone" but non-entailed; and "latitude" and "longitude" are the least correlated compared with the previous two pairs, but have identical output spaces. These entity types together cover a diverse set of attributes such that predicting the value of the attribute uses factual, linguistic, or commonsense knowledge.

Constructing Prompts We consider two types of prompts: attribute prompts and entity prompts. Attribute prompts $\mathcal{P}_E^A$ contain mentions of E and instruct the model to output the attribute value A_E. For example, E = `Paris` is an instance of the type "city", which has an attribute A = `Continent` that can be queried with prompts "`Paris is in the continent of`". Prompts can also be JSON-format, e.g., "`{"city": "Paris", "continent":"`", which reflects how entity–attribute association might be encoded in training data. For each format, we do zero- and few-shot prompting. In addition to attribute prompts, entity prompts $\mathcal{W}_E$ contain mentions of the E, but does not query any $A \in \mathcal{A}$. For example, "`Tokyo is a large city`". We sample entity prompts from the Wikipedia corpus.[1]

For a set of entities $\mathcal{E}$ and a set of attributes to disentangle $\mathcal{A}$, the full set of prompts is

$$\mathcal{D} = \{x : x \in \mathcal{P}_E^A \cup \mathcal{W}_E, E \in \mathcal{E}, A \in \mathcal{A}\}$$

Generating Splits RAVEL offers two settings, `Entity` and `Context`, to evaluate the *generalizability* (desideratum 3) of an interpretability method across unseen entities and contexts. Each setting has a predefined train/dev/test structure. In `Entity`, for each entity type, we randomly split the entities into 50%/25%/25% for train/dev/test, but use the same set of prompt templates across the three splits. In `Context`, for each attribute, we randomly split the prompt templates into 50%/25%/25%, but use the same set of entities across the three splits.

Filtering for a Specific Model When evaluating interpretability methods that analyze a model $\mathcal{M}$, we generally focus on a subset of the instances where $\mathcal{M}$ correctly predicts the values of the attributes (see Appendix A.2). This allows us to focus on understanding why models succeed, and it means that we don't have to worry about how methods might have different biases for incorrect predictions.

2.2 Interpretability Evaluation

Interchange Interventions A central goal of RAVEL is to assess methods by the extent to which they provide causal explanations of model behaviors (desideratum 2). To build such analyses, we need to put models into counterfactual states that allow us to isolate the causal effects of interest.

The fundamental operation for achieving this is the intervention (Spirtes et al., 2000; Pearl, 2001, 2009): we change the value of a model-internal state and study the effects this has on the model's input–output behavior. In more detail: let $\mathcal{M}(x)$ be the entire state of the model when $\mathcal{M}$ receives input x, i.e., the set of all input, hidden, and output representations created during inference. Let $\mathcal{M}_{\mathbf{N}\leftarrow\mathbf{n}}$ be the model where neurons $\mathbf{N}$ are intervened upon and fixed to take on the value $\mathbf{n} \in \mathsf{Values}(\mathbf{N})$.

Geiger et al. (2023b) generalize this operation to intervene upon features that are distributed across neurons using a bijective featurizer $\mathcal{F}$. Let $\mathcal{M}_{F\leftarrow f}$ be the model where neurons $\mathbf{N}$ are projected into a feature space using $\mathcal{F}$, the feature F is fixed to take on value f, and then the features are projected back into the space of neural activations using $\mathcal{F}^{-1}$. If we let $\tau(\mathcal{M}(x))$ be the token that a model predicts for a given prompt $x \in \mathcal{D}$, then comparisons between $\tau(\mathcal{M}(x))$ and $\tau(\mathcal{M}_{F\leftarrow f}(x))$ yield insights into the causal role that F plays in model behavior.

However, most conceivable interventions fix model representations to be values that are never realized by any input. To characterize the high-level conceptual role of a model representation, we need a data-driven intervention that sets a representation

[1] We use the `20220301.en` version pre-processed by HuggingFace at https://huggingface.co/datasets/wikipedia

to values it could actually take on. This is achieved by the *interchange intervention*, which fixes a feature F to the value it would take if a different input x' were provided:

$$\texttt{II}(\mathcal{M}, F, x, x') \stackrel{\text{def}}{=} \tau\Big(\mathcal{M}_{F \leftarrow \texttt{GetFeature}(\mathcal{M}(x'), F)}(x)\Big) \quad (1)$$

where $\texttt{GetFeature}(\mathcal{M}(x'), F)$ is the value of F in $\mathcal{M}(x')$. Interchange interventions represent a very general technique for identifying abstract causal processes that occur in complex black-box systems (Beckers and Halpern, 2019; Beckers et al., 2020; Geiger et al., 2023a).

Evaluation Data For evaluation, each intervention example consists of a tuple: an input $x \in \mathcal{P}_E^A$, an input $x' \in \mathcal{P}_{E'}^{A'} \cup \mathcal{W}_{E'}$, a target attribute A^*, and an intervention label y. If $A^* = A$, then y is $A_{E'}$ and otherwise y is A_E. For example, if the set of "city" entities to evaluate on is $\{\texttt{"Paris"}, \texttt{"Tokyo"}\}$ and the goal is to disentangle the "country" attribute from the "continent" attribute, then the set of test examples becomes the one shown in Figure 1.

Metrics If $\mathcal{M}$ achieves behavioral success on a dataset, we can use that dataset to evaluate an interpretability method on its ability to identify a collection of neurons $\mathbf{N}$, a featurizer $\mathcal{F}$ for those neurons, and a feature F that represents an attribute A separately from all others attributes $\mathcal{A} \setminus \{A\}$.

If F encodes A, then interventions on F should change the value of A. When $\mathcal{M}$ is given a prompt $x \in \mathcal{P}_E^A$, we can intervene on F to set the value to what it would be if a second prompt $x' \in \mathcal{P}_{E'}^{A'} \cup \mathcal{W}_{E'}$ were provided. The token predicted by $\mathcal{M}$ should change from A_E to $A_{E'}$:

$$\texttt{Cause}(A, F, \mathcal{M}, \mathcal{D}) \stackrel{\text{def}}{=} \mathbb{E}_{\mathcal{D}}\left[\texttt{II}(\mathcal{M}, F, x, x') = A_{E'}\right]$$

If F isolates A, then interventions on F should not cause the values of other attributes $A^* \in \mathcal{A} \setminus \{A\}$ to change. When $\mathcal{M}$ is given a prompt $x^* \in \mathcal{P}_E^{A^*}$, we can again intervene on F to set the value to what it would be if a second prompt $x' \in \mathcal{P}_{E'}^{A'} \cup \mathcal{W}_{E'}$ were provided. The token predicted by $\mathcal{M}$ should remain A_E^*:

$$\texttt{Iso}(A, F, \mathcal{M}, \mathcal{D}) \stackrel{\text{def}}{=} \frac{1}{|\mathcal{A} \setminus \{A\}|} \sum_{A^* \in \mathcal{A} \setminus \{A\}} \mathbb{E}_{\mathcal{D}}\left[\texttt{II}(\mathcal{M}, F, x^*, x') = A_E^*\right]$$

To balance these two objectives, we define the `Disentangle` score as a weighted average between `Cause` and `Iso`.

$$\texttt{Disentangle}(A, F, \mathcal{M}, \mathcal{D}) = \frac{1}{2}\Big[\texttt{Cause}(A, F, \mathcal{M}, \mathcal{D}) + \texttt{Iso}(A, F, \mathcal{M}, \mathcal{D})\Big]$$

The score on RAVEL for an entity type is its average `Disentangle` score over all attributes.

In practice, two attributes might not be fully disentanglable in the model $\mathcal{M}$ so there is no guarantee that it is possible to find a feature F that achieves $\texttt{Cause} = 1$ and $\texttt{Iso} = 1$ at the same time. However, evidence that two attributes might not be separable is an insight into how knowledge is structured in the model.

3 Interpretability Methods

We use RAVEL to evaluate a variety of interpretability methods on their ability to disentangle attributes while generalizing to novel templates and entities. Each method uses data from the training split to find a set of neurons $\mathbf{N}$, learn a featurizer $\mathcal{F}$, and find a feature F_A that captures an attribute $A \in \mathcal{A}$ independent from the other attributes. In Section 4, we describe the baseline procedure we use for considering different sets of neurons. In this section, we define methods for learning a featurizer and identifying a feature given a set of neurons. For each method, the core intervention for A is given by $\texttt{II}(\mathcal{M}, F_A, x, x')$ where F_A is defined by the method. In this section, we use $\texttt{GetVals}(\mathcal{M}(x), \mathbf{N})$ to mean the activations of neurons $\mathbf{N}$ when $\mathcal{M}$ processes input x.

3.1 PCA

Principal Component Analysis (PCA) is a dimensionality reduction method that minimizes information loss. In particular, given a set of real valued vectors $\mathcal{V} \subset \mathbb{R}^n$, $|\mathcal{V}| > n$, the principal components are n orthogonal unit vectors $\mathbf{p}_1, \ldots, \mathbf{p}_n$ that form an $n \times n$ matrix:

$$\text{PCA}(\mathcal{V}) = \begin{bmatrix}\mathbf{p}_1 & \ldots & \mathbf{p}_n\end{bmatrix}$$

For our purposes, the orthogonal matrix formed by the principal components serves as a featurizer that maps neurons $\mathbf{N}$ into a more interpretable space (Chormai et al., 2022; Marks and Tegmark, 2023; Tigges et al., 2023). Given an attribute A, a training

dataset $\mathcal{D}$ from RAVEL for a particular entity type, a model $\mathcal{M}$, and a set of neurons $\mathbf{N}$, we define

$$\mathcal{F}_A(\mathbf{n}) = \mathbf{n}^T \mathsf{PCA}(\{\mathtt{GetVals}(\mathcal{M}(x), \mathbf{N}) : x \in \mathcal{D}\})$$

PCA is an unsupervised method, so there is no easy way to tell what information is encoded in each principal component. To solve this issue, for each attribute $A \in \mathcal{A}$ we train a linear classifier with L1 regularization to predict the value of A from the featurized neural representations. Then, we define the feature F_A to be the set of dimensions assigned a weight by the classifier that is greater than a hyperparameter ϵ.

3.2 Sparse Autoencoder

A recent approach to featurization is to train an autoencoder to project neural activations into a higher dimensional sparse feature space and then reconstruct the neural activations from the features (Bricken et al., 2023; Cunningham et al., 2024). We train a sparse autoencoder on the loss

$$\sum_{x \in \mathcal{D}} ||\mathtt{GetVals}(\mathcal{M}(x), \mathbf{N}) - \Big(W_2\mathbf{f} + b_2\Big)||_2 + ||\mathbf{f}||_1$$

$$\mathbf{f} = \mathsf{ReLU}(W_1(\mathtt{GetVals}(\mathcal{M}(x), \mathbf{N}) - b_2) + b_1)$$

with $W_1 \in \mathbb{R}^{n \times m}$, $W_2 \in \mathbb{R}^{m \times n}$, $b_1 \in \mathbb{R}^m$, and $b_2 \in \mathbb{R}^n$. To construct a training datset, we sample 100k sentences from the Wikipedia corpus for each entity type, each containing a mention of an entity in the training set. We extract the 4096-dimension hidden states of Llama2-7B at the target intervention site as the input for training a sparse autoencoder with 16384 features.

We use the autoencoder to define a featurizer

$$\mathcal{F}_A(\mathbf{n}) = \mathsf{ReLU}(W_1(\mathbf{n} - b_2) + b_1)$$

and an inverse $\mathcal{F}_A^{-1}(\mathbf{n}) = W_2\mathbf{n} + b_2$.

An important caveat to this method is that the featurizer is only truly invertible if the autoencoder has a reconstruction loss of 0. The larger the loss is, the more unfaithful this interpretability method is to the model being analyzed. All other methods considered use an orthogonal matrix, which is truly invertible up to floating point precision.

Similar to PCA, sparse autoencoders are an unsupervised method that does not produce features with obvious meanings. Again, to solve this issue, for each attribute $A \in \mathcal{A}$ we train a linear classifier with L1 regularization and define the feature F_A to be the set of dimensions assigned a weight that is greater than a hyperparameter ϵ.

3.3 Relaxed Linear Adversarial Probe

Supervised probes are a popular interpretability technique for analyzing how neural activations correlate with high-level concepts (Peters et al., 2018; Hupkes et al., 2018; Tenney et al., 2019; Clark et al., 2019). When probes are arbitrarily powerful, this method is equivalent to measuring the mutual information between the neurons and the concept (Pimentel et al., 2020; Hewitt et al., 2021). However, probes are typically simple linear models in order to capture how easily the information about a concept can be extracted. Probes have also been used to great effect on the task of concept erasure (Ravfogel et al., 2020; Elazar et al., 2021; Ravfogel et al., 2022).

Following the method of Ravfogel et al. (2022), we train a relaxed linear adversarial probe (RLAP) to learn a linear subspace parameterized by a set of k orthonormal vectors $W \in \mathbb{R}^{k \times n}$ that captures an attribute A, using the following loss objective:

$$\min_{\theta} \max_{W} \sum_{x \in \mathcal{D}} \mathsf{CE}\big(\theta^T \mathbf{f}, A_{E_x}\big)$$

$$\mathbf{f} = (I - W^T W)\big(\mathtt{GetVals}(\mathcal{M}(x), \mathbf{N})\big)$$

where $\mathbf{f}$ is the representation of the entity with the attribute information erased, and θ is a linear classifier that tries to predict the attribute value A_{E_x} from the erased entity representation.

We define the $\mathcal{F}$ using the set of k orthonormal vectors that span the row space of W and the set of $n - k$ orthonormal vectors that span the null space:

$$\mathcal{F}_A(\mathbf{n}) = \mathbf{n}\begin{bmatrix}\mathbf{r}_1 & \dots & \mathbf{r}_k & \mathbf{u}_{k+1} & \dots & \mathbf{u}_n\end{bmatrix}$$

Our feature F_A is the first k dimensions of the feature space, i.e. the row space of W. Intuitively, since the linear probe was trained to extract the attribute A, the rowspace is the linear subspace of neural activations that the probe is "looking at" to make predictions.

3.4 Differential Binary Masking

Differential Binary Masking (DBM) learns a binary mask to select a set of neurons that causally represents a concept (Cao et al., 2020; Csordás et al., 2021; Cao et al., 2022; Davies et al., 2023). The loss objective used to train the mask is a combination of matching the counterfactual behavior and

Method	Supervision	Entity	Context
Full Rep.	None	40.5	39.5
PCA	None	39.5	39.1
SAE	None	48.6	46.8
RLAP	Attribute	48.8	50.9
DBM	Counterfactual	52.2	49.8
DAS	Counterfactual	56.5	57.3
MDBM	Counterfactual	53.7	53.9
MDAS	Counterfactual	**60.1**	**65.6**

Table 2: The disentanglement score on RAVEL for each interpretability method. Numbers are represented in %.

forcing the mask to be sparse with coefficient λ:

$$\mathcal{L}_{\texttt{Cause}} = \texttt{CE}(\tau(\mathcal{M}_{\mathbf{N}\leftarrow\mathbf{n}}(x)), A_{E'}) + \lambda||\mathbf{m}||_1$$
$$\mathbf{n} = \big(\mathbf{1} - \sigma(\mathbf{m}/T)\big) \circ \texttt{GetVals}(\mathcal{M}(x), \mathbf{N}) + \sigma(\mathbf{m}/T) \circ \texttt{GetVals}(\mathcal{M}(x'), \mathbf{N})$$

where the intervention is determined by inputs x, x' and learnable parameter $\mathbf{m} \in \mathbb{R}^n$, where $\circ$ is element wise multiplication and $T \in \mathbb{R}$ is a temperature annealed throughout training.

The feature space is the original space of neural activations, i.e., featurizer $\mathcal{F}_A(\mathbf{n}) = \mathbf{n}$. The feature F_A is the set of dimensions i where $1-\sigma(\mathbf{m}_i/T) < \epsilon$ for a (small) hyperparameter ϵ.

3.5 Distributed Alignment Search

Distributed Alignment Search (DAS) (Geiger et al., 2023b) learns a linear subspace of a model representation with a training objective defined using interchange interventions. In the original work, the linear subspace learned by DAS is parameterized as an $n \times n$ orthogonal matrix $Q = [\mathbf{u}_1 \dots \mathbf{u}_n]$, which rotates the representation into a new coordinate system, i.e., $\mathcal{F}_A(\mathbf{n}) = Q^\top \mathbf{n}$. The set of feature F_A is the first k dimensions of the rotated subspace, where k is a hyperparameter. The matrix Q is learned by minimizing the following loss:

$$\mathcal{L}_{\texttt{Cause}}(A, F_A, \mathcal{M}) = \texttt{CE}(\texttt{II}(\mathcal{M}, F_A, x, x'), A_{E'})$$

Computing Q is expensive, as it requires computing n orthogonal vectors. To avoid instantiating the full rotation matrix, we use an alternative parameterization where we only learn the $k \ll n$ orthogonal vectors that form the feature F_A (see Appendix B.4).

3.6 Multi-task DBM and DAS

To address the disentanglement problem, we propose a multitask extension of DBM (MDBM) and DAS (MDAS). The original training objective of DBM and DAS only optimizes for the Cause score,

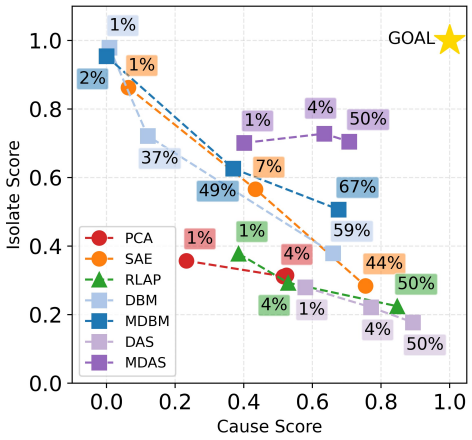


Figure 2: Cause and Iso scores for each method when using different feature sizes, shown as the ratio (%) between the dimension of F_A and the dimension of the output space of $\mathcal{F}$. Each method has three data points that vary from using very few (≈1%) to half (≈50%) of the dimensions. Increasing feature dimensions generally leads to higher Cause score, but lower Iso score. Figure best viewed in color.

without considering the impact on the Iso score. We introduce the Iso aspect into the training objective through multitask learning. For each attribute $A^* \in \mathcal{A} \setminus \{A\}$, we define the Iso objective as

$$\mathcal{L}_{\texttt{Iso}}(A, F_A, \mathcal{M}) = \texttt{CE}(\texttt{II}(\mathcal{M}(x), F_A, x'), A^*_E)$$

We minimize a linear combination of losses from each task:

$$\mathcal{L}_{\texttt{Disentangle}}(\mathcal{A}, F_A, \mathcal{M}) = \mathcal{L}_{\texttt{Cause}}(A, F_A, \mathcal{M}) + \sum_{A^* \in \mathcal{A}\setminus\{A\}} \frac{\mathcal{L}_{\texttt{Iso}}(A^*, F_A, \mathcal{M})}{|\mathcal{A} \setminus \{A\}|}$$

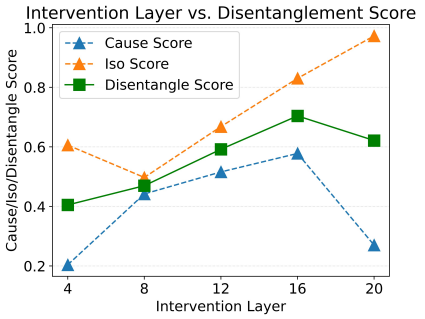
4 Experiments

We evaluate the methods described in Section 3 on RAVEL with Llama2-7B (Touvron et al., 2023), a 32-layer decoder-only Transformer model, as the target LM. Implementation details of each method are provided in Appendix B.

4.1 Setup

We consider the residual stream representations at the last token of the entity as our potential intervention sites. For autoregressive LMs, the last token of the entity t_E (e.g., the token "is" in the case of "Paris") likely aggregates information of the entity (Meng et al., 2022; Geva et al., 2023). For Transformer-based LMs like Llama2, an activation vector $\mathbf{N}^L_t$ in the residual stream is created for each token t at each Transformer layer

(a) Cause score for all attributes when intervening on the attribute features identified by DAS (left) and MDAS (right). A Cause score of 0.62 for column Continent, row Timezone (bottom left corner), means that, when intervening on the Continent feature, the same subspace changes Timezone 62% of the time.



(b) The Cause, Iso, and Disentangle score on the Entity split for the "country" feature found by MDAS. The attributes of cities become more disentangled across layers.

Figure 3: Additional results for the MDAS method.

L. As the contributions of the MLP and attention heads must pass through the residual stream, it serves as a bottleneck. Therefore, we will limit our methods to examining the set of representations $\mathcal{N} = \{\mathbf{N}^L_{t_E} : L \in \{1, \ldots, 32\}\}$.

This simplification is only to establish baseline results on the RAVEL benchmark. We expect the best methods will consider other token representations, such as the remainder of the token sequence that realizes the entity.

4.2 Results

We evaluate each method on every representation $\mathbf{N}^L_{t_E}$ and report the highest disentanglement score on test splits in Table 2. We additionally include a baseline that simply replaces the full representation $\mathbf{N}^L_{t_E}$ regardless of what attribute is being targetted (see Full Rep. in Table 2). A breakdown of the results with per-attribute Cause and Iso is in Appendix C.

In Figure 2, we show for each method, how the Iso and Cause scores vary as we change the dimensionality of F_A, the feature targeted for intervention. For RLAP, DAS, and MDAS, the dimensionality of F_A is a hyperparameter we vary directly. For other methods, we vary the coefficient of L1 penalty to vary the size of F_A. Details are given in Appendix B.

In Figure 3, we focus on using MDAS, the best performing method, to understand how attributes are disentangled in Llama2-7B. Figure 3a shows two heat maps summarizing the performance of DAS and MDAS on the entity type "city". These heat maps also show how attributes have different levels of disentanglement. Figure 3b shows how the Cause, Iso, and Disentangle scores change for the "country" attribute across model layers.

Methods with counterfactual supervision achieve strong results while methods with unsupervised featurizers struggle. MDAS is the state-of-the-art method on RAVEL, being able to achieve high Disentangle scores while only intervening on a feature F_A with a dimensionality that is 4% of $|\mathbf{N}|$ where $\mathbf{N}$ are the neurons the feature is distributed across (Figure 2). DBM, MDBM, and DAS, the other methods that are trained with interventions using counterfactual labels as supervision, achieve the next best performance. PCA and Sparse Autoencoder achieve the lowest Disentangle scores, which aligns with the prior finding that disentangled representations are difficult to learn without supervision (Locatello et al., 2018). Unsurprisingly, more supervision results in higher performance.

Multi-task supervision is better at isolating attributes. Adding multitask objectives to DBM and DAS increases the overall disentanglement score by 1.5%/4.1% and 3.6%/8.3% on the Entity/Context split respectively. To further illustrate the differences, we compare DAS with MDAS in Figure 3a. On the left, attributes such as "continent" and "timezone" are naturally entangled with all other attributes; intervening on the feature learned by DAS for any city attribute will also change these two attributes. In contrast, in Figure 3a right, MDAS is far more successful at disentangling these attributes, having small Cause

scores in all off-diagonal entries.

Some groups of attributes are more difficult to disentangle than others. As show in Figure 3a, the attribute pairs "country–language" and "latitude–longitude" are difficult to disentangle. When we train DAS to find a feature for either of these attributes (Figure 3a left), the same feature also has causal effects on the other attribute. Even with the additional supervision (Figure 3a right), MDAS cannot isolate these attributes. Changing one of these entangled attributes has seemingly unavoidable ripple effects (Cohen et al., 2024) that change the other. In contrast, the attribute pair "language–continent" can be disentangled. Moreover, the pairs that are difficult to disentangle are consistent across all five supervised methods in our experiment, despite these methods using different training objectives. We include additional visualizations in Appendix C.2.

Attributes are gradually disentangled across layers. The representations of different attributes gradually disentangle as we move towards later layers, as shown in Figure 3b. Early layer features identified by MDAS fail to generalize to unseen entities, hence low `Cause` score. While MDAS is able to identify a feature with relatively high `Cause` starting at layer 8, the `Iso` increases from 0.5 to 0.8 from layer 8 to layer 16. It is not until layer 16 that the highest `Disentangle` score is achieved.

5 Related Work

Intervention-based Interpretability Methods Intervention-based techniques, branching off from interchange intervention (Vig et al., 2020; Geiger et al., 2020) or activation patching (Meng et al., 2022), have shown promising results in uncovering causal mechanisms of LMs. They play important roles in recent interpretability research of LMs such as causal abstraction (Geiger et al., 2021, 2023b), causal tracing to locate factual knowledge (Meng et al., 2022; Geva et al., 2023), path patching or causal scrubbing to find causal circuits (Chan et al., 2022; Conmy et al., 2023; Goldowsky-Dill et al., 2023), and Distributed Alignment Search (Geiger et al., 2023b). Previous works suggest that activation interventions that result in systematic counterfactual behaviors provide clear causal insights into model components.

Isolating Individual Concepts LMs learn highly distributed representations that encode multiple concepts in a overlapping sets of neurons (Smolensky, 1988; Olah et al., 2020; Elhage et al., 2022). Various methods have been proposed to find components that capture a concept, such as finding a linear subspace that modifies a concept (Ravfogel et al., 2020, 2022; Belrose et al., 2023; Cao et al., 2020; Geiger et al., 2023b) and generating a sparse feature space where each direction captures a word sense or is more interpretable (Arora et al., 2018; Bricken et al., 2023; Cunningham et al., 2024; Tamkin et al., 2023). However, these methods have not been evaluated against each other on their ability to isolate concepts. Isolating an individual concept is also related to the goal of "disentanglement" in representation learning (Schölkopf et al., 2021), where each direction captures a single generative factor. In this work, we focus on isolating the causal effect of a representation.

Knowledge Representation in LMs Understanding knowledge representation in LMs starts with probing structured linguistic knowledge (Conneau et al., 2018; Tenney et al., 2019; Manning et al., 2020). Recent work expands to factual knowledge stored in Transformer MLP layers (Geva et al., 2021; Dai et al., 2022; Meng et al., 2022), associations represented in linear structures (Merullo et al., 2023; Hernandez et al., 2024; Park et al., 2023), and deeper study of the semantic enrichment of subject representation (Geva et al., 2023). These findings suggest LMs store knowledge modularly, motivating the disentanglement objective in our work.

Benchmarking Interpretability Methods Testing the faithfulness of interpretability method relies on counterfactuals. Existing counterfactual benchmarks use behavioral testing (Atanasova et al., 2023; Schwettmann et al., 2023; Mills et al., 2023), interventions (Abraham et al., 2022), or a combination of both (Huang et al., 2023). Recent model editing benchmarks (Meng et al., 2022; Zhong et al., 2023; Cohen et al., 2024) also provide counterfactuals that have potential for evaluating interpretability methods. MQuAKE (Zhong et al., 2023) and RippleEdits (Cohen et al., 2024), in particular, consider entailment relationships of attributes, while we focus on disentanglement.

6 Conclusion

We present RAVEL a benchmark for evaluating the ability of interpretability methods to localize and disentangle entity attributes in LMs in a causal,

generalizable manner. We show how RAVEL can be used to evaluate five different families of interpretability methods that are commonly used in the community. We benchmark several strong interpretability methods on RAVEL with Llama2-7B model as baselines, and we introduce a multi-task objective that improves the performance of Differential Binary Masking (DBM) and Distributed Alignment Search (DAS). Multi-task DAS achieves the best results in our experiments. Results on our attribute disentanglement task also offer insights into the different levels of entanglement between attributes and the emergence of disentangled representations across layers in the Llama2-7B model.

The community has seen an outpouring of innovative new interpretability methods. However, these methods have not been systematically evaluated for whether they are *faithful*, *generalizable*, *causally effective*, and able to *isolate individual concepts*. We release RAVEL[2] to the community and hope it will help drive the assessment and development of interpretability methods that satisfy these criteria.

Limitations

Our attribute disentanglement results in Section 4 are based on the Llama2-7B model. While Llama2-7B uses the widely adopted decoder-only Transformer architecture, different model architectures or training paradigms could produce LMs that favor different interpretability methods. Hence, when deciding which interpretability method is the best to apply to a new model, we encourage people to instantiate RAVEL on the new model.

When choosing intervention sites, we limit our search to the residual stream above the last entity token. However, representations of attributes can be distributed across multiple tokens or layers. We encourage future work to explore different intervention sites when using this benchmark.

Ethics Statement

In this paper, we present an interpretability benchmark that aims to assess the faithfulness, generalizability, causal effects, and the ability to isolate individual concepts in language models. While an interpretability method that satisfies these criteria could be useful for assessing model bias or steering model behaviors, the same method might also be used for manipulating models in undesirable applications such as triggering toxic outputs. These interpretability methods should be studied and used in a responsible manner.

[2] https://github.com/explanare/ravel

Acknowledgements

This research is supported in part by grants from Open Philanthropy and the Stanford Institute for Human-Centered Artificial Intelligence (HAI).

References

Eldar David Abraham, Karel D'Oosterlinck, Amir Feder, Yair Gat, Atticus Geiger, Christopher Potts, Roi Reichart, and Zhengxuan Wu. 2022. CEBaB: Estimating the causal effects of real-world concepts on NLP model behavior. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2018. Linear algebraic structure of word senses, with applications to polysemy. In *Transactions of the Association of Computational Linguistics (TACL)*.

Pepa Atanasova, Oana-Maria Camburu, Christina Lioma, Thomas Lukasiewicz, Jakob Grue Simonsen, and Isabelle Augenstein. 2023. Faithfulness tests for natural language explanations. In *Association for Computational Linguistics (ACL)*.

Sander Beckers, Frederick Eberhardt, and Joseph Y. Halpern. 2020. Approximate causal abstractions. In *Uncertainty in Artificial Intelligence Conference (UAI)*.

Sander Beckers and Joseph Y. Halpern. 2019. Abstracting causal models. In *Conference on Artificial Intelligence (AAAI)*.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. 2023. LEACE: Perfect linear concept erasure in closed form. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Tolga Bolukbasi, Adam Pearce, Ann Yuan, Andy Coenen, Emily Reif, Fernanda B. Viégas, and Martin Wattenberg. 2021. An interpretability illusion for BERT. In *arXiv preprint arXiv:2104.07143*.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. In *Transformer Circuits Thread*.

Nick Cammarata, Shan Carter, Gabriel Goh, Chris Olah, Michael Petrov, Ludwig Schubert, Chelsea Voss, Ben Egan, and Swee Kiat Lim. 2020. Thread: Circuits. In *Distill*.

Nicola De Cao, Michael Sejr Schlichtkrull, Wilker Aziz, and Ivan Titov. 2020. How do decisions emerge across layers in neural models? Interpretation with differentiable masking. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Nicola De Cao, Leon Schmid, Dieuwke Hupkes, and Ivan Titov. 2022. Sparse interventions in language models with differentiable masking. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*.

Lawrence Chan, Adrià Garriga-Alonso, Nicholas Goldowsky-Dill, Ryan Greenblatt, Jenny Nitishinskaya, Ansh Radhakrishnan, Buck Shlegeris, and Nate Thomas. 2022. Causal scrubbing: a method for rigorously testing interpretability hypotheses. In *Alignment Forum Blog post*.

Pattarawat Chormai, Jan Herrmann, Klaus-Robert Müller, and Grégoire Montavon. 2022. Disentangled explanations of neural network predictions by finding relevant subspaces. In *arXiv preprint arXiv:2212.14855*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? An analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2024. Evaluating the ripple effects of knowledge editing in language models. In *Transactions of the Association of Computational Linguistics (TACL)*.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Association for Computational Linguistics (ACL)*.

Róbert Csordás, Sjoerd van Steenkiste, and Jürgen Schmidhuber. 2021. Are neural nets modular? Inspecting functional modularity through differentiable weight masks. In *International Conference on Learning Representations (ICLR)*.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations (ICLR)*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Association for Computational Linguistics (ACL)*.

Xander Davies, Max Nadeau, Nikhil Prakash, Tamar Rott Shaham, and David Bau. 2023. Discovering variable binding circuitry with desiderata. In *arXiv preprint arXiv:2307.03637*.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. Amnesic Probing: Behavioral Explanation with Amnesic Counterfactuals. In *Transactions of the Association of Computational Linguistics (TACL)*.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy models of superposition. In *arXiv preprint arXiv:2209.10652*.

Jiahai Feng and Jacob Steinhardt. 2024. How do language models bind entities in context? In *International Conference on Learning Representations (ICLR)*.

Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen, and Yonatan Belinkov. 2021. Causal analysis of syntactic agreement mechanisms in neural language models. In *Association for Computational Linguistics and International Joint Conference on Natural Language Processing (ACL-IJCNLP)*.

Atticus Geiger, Hanson Lu, Thomas F Icard, and Christopher Potts. 2021. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Atticus Geiger, Christopher Potts, and Thomas Icard. 2023a. Causal abstraction for faithful model interpretation. Ms., Stanford University.

Atticus Geiger, Kyle Richardson, and Chris Potts. 2020. Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the 2020 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah D. Goodman. 2023b. Finding alignments between interpretable causal variables and distributed neural representations. In *Causal Learning and Reasoning (CLeaR)*.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. 2024. Patchscopes: A unifying framework for inspecting hidden representations of language models. In *arXiv preprint arXiv:2401.06102*.

Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. 2023. Localizing model behavior with path patching. In *arXiv preprint arXiv:2304.05969*.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. 2023. Finding neurons in a haystack: Case studies with sparse probing. In *Transactions on Machine Learning Research (TMLR)*.

Michael Hanna, Ollie Liu, and Alexandre Variengien. 2023. How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. 2023. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Roee Hendel, Mor Geva, and Amir Globerson. 2023. In-context learning creates task vectors. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. 2024. Linearity of relation decoding in transformer language models. In *International Conference on Learning Representations (ICLR)*.

John Hewitt, Kawin Ethayarajh, Percy Liang, and Christopher Manning. 2021. Conditional probing: measuring usable information beyond a baseline. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Jing Huang, Atticus Geiger, Karel D'Oosterlinck, Zhengxuan Wu, and Christopher Potts. 2023. Rigorously assessing natural language explanations of neurons. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and "diagnostic classifiers" reveal how recurrent and recursive neural networks process hierarchical structure. In *Journal of Artificial Intelligence Research (JAIR)*.

Belinda Z. Li, Maxwell I. Nye, and Jacob Andreas. 2021. Implicit representations of meaning in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 1813–1827. Association for Computational Linguistics.

Tom Lieberum, Matthew Rahtz, János Kramár, Neel Nanda, Geoffrey Irving, Rohin Shah, and Vladimir Mikulik. 2023. Does circuit analysis interpretability scale? Evidence from multiple choice capabilities in chinchilla. In *arXiv preprint arXiv:2307.09458*.

Francesco Locatello, Stefan Bauer, Mario Lucic, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2018. Challenging common assumptions in the unsupervised learning of disentangled representations. *CoRR*, abs/1811.12359.

Christopher D. Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. 2020. Emergent linguistic structure in artificial neural networks trained by self-supervision. In *Proceedings of the National Academy of Sciences (PNAS)*.

Samuel Marks and Max Tegmark. 2023. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *arXiv preprint arXiv:2310.06824*.

J. L. McClelland, D. E. Rumelhart, and PDP Research Group, editors. 1986. *Parallel Distributed Processing. Volume 2: Psychological and Biological Models*. MIT Press, Cambridge, MA.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. 2023. A mechanism for solving relational tasks in transformer language models. In *arXiv preprint arXiv:2305.16130*.

Edmund Mills, Shiye Su, Stuart Russell, and Scott Emmons. 2023. Almanacs: A simulatability benchmark for language model explainability. In *arXiv preprint arXiv:2312.12747*.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. In *Distill*.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2023. The linear representation hypothesis and the geometry of large language models. In *arXiv preprint arXiv:2311.03658*.

Judea Pearl. 2001. Direct and indirect effects. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, UAI'01, pages 411–420, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Judea Pearl. 2009. *Causality*. Cambridge University Press.

Matthew E. Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018. Dissecting contextual word embeddings: Architecture and representation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020. Information-theoretic probing for linguistic structure. In *Association for Computational Linguistics (ACL)*.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null it out: Guarding protected attributes by iterative nullspace projection. In *Association for Computational Linguistics (ACL)*.

Shauli Ravfogel, Michael Twiton, Yoav Goldberg, and Ryan D Cotterell. 2022. Linear adversarial concept erasure. In *International Conference on Machine Learning (ICML)*.

D. E. Rumelhart, J. L. McClelland, and PDP Research Group, editors. 1986. *Parallel Distributed Processing. Volume 1: Foundations*. MIT Press, Cambridge, MA.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. 2021. Toward causal representation learning. *Proc. IEEE*, 109(5):612–634.

Sarah Schwettmann, Tamar Rott Shaham, Joanna Materzynska, Neil Chowdhury, Shuang Li, Jacob Andreas, David Bau, and Antonio Torralba. 2023. Find: A function description benchmark for evaluating interpretability methods. In *Advances in Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*.

Paul Smolensky. 1988. On the proper treatment of connectionism. *Behavioral and Brain Sciences*, 11(1):1–23.

Peter Spirtes, Clark Glymour, and Richard Scheines. 2000. *Causation, Prediction, and Search*. MIT Press.

Alex Tamkin, Mohammad Taufeeque, and Noah D. Goodman. 2023. Codebook features: Sparse and discrete interpretability for neural networks. In *arXiv preprint arXiv:2310.17230*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Association for Computational Linguistics (ACL)*.

Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. 2023. Linear representations of sentiment in large language models. In *arXiv preprint arXiv:2310.15154*.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. 2024. Function vectors in large language models. In *International Conference on Learning Representations (ICLR)*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. In *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *International Conference on Learning Representations (ICLR)*.

Zhengxuan Wu, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah D. Goodman. 2023. Interpretability at scale: Identifying causal mechanisms in alpaca. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Zexuan Zhong, Zhengxuan Wu, Christopher Manning, Christopher Potts, and Danqi Chen. 2023. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Empirical Methods in Natural Language Processing (EMNLP)*.

A Dataset Details

Attributes	$\lvert A_E \rvert$	Sample Values	Sample Prompts
City			

Country	158	United States, China, Russia, Brazil, Australia	`city to country: Toronto is in Canada. {E} is in,`
			`[{"city": "Paris", "country": "France"}, {"city": "{E}",`
			`"country": "`

Continent	6	Asia, Europe, Africa, North America, South America	`{E} is a city in the continent of,` `[{"city": "{E}", "continent": "`
Latitude	122	41, 37, 47, 36, 35	`[{"city": "Rio de Janeiro", "lat": "23"},` `{"city": "{E}", "lat": "`,`[{"city": "{E}", "lat": "`
Longitude	317	30, 9, 10, 33, 11	`[{"city": "Rome", "long": "12.5"}, {"city": "{E}", "long": "`, `"long": "122.4"}, {"city": "{E}", "long": "`

Language	159	English, Spanish, Chinese, Russian, Portuguese	`[{"city": "Beijing", "lang": "Chinese"},`
			`{"city": "{E}", "lang": "`,
			`[{"city": "{E}", "official language": "`

Timezone	267	America/Chicago, Asia/Shanghai, Asia/Kolkata, Europe/Moscow, America/Sao_Paulo	`Time zone in Los Angeles is America/Santiago;`
			`Time zone in {E} is,`
			`[{"city": "New Delhi", "timezone": "UTC+5:30"},`
			`{"city": "{E}", "timezone": "UTC`

Nobel Laureate			

Field	7	Medicine, Physics, Chemistry, Literature, Peace	`Jules A. Hoffmann won the Nobel Prize in Medicine.`
			`{E} won the Nobel Prize in,`
			`name: {E}, award: Nobel Prize in`

Award Year	118	2001, 2019, 2009, 2011, 2000	`"name": {E}, "award": "Nobel Prize", "year": "`, `laureate: Frances H. Arnold, year: 2018, laureate: {E}, year:`
Birth Year	145	1918, 1940, 1943, 1911, 1941	`Alan Heeger was born in 1936. {E} was born in,` `laureate: {E}, date of birth (YYYY-MM-DD):`
Country of Birth	81	United States, United Kingdom, Germany, France, Sweden	`name: {E}, country:,` `Roderick MacKinnon was born in United States. {E} was born in`
Gender	4	his, male, female, her	`name: {E}, gender:,` `David M. Lee: for his contributions in physics. {E}: for`

Table 3: Attributes in RAVEL. $\lvert A_E \rvert$ is the number of unique attribute values. In sampled prompts, `{E}` is a placeholder for the entity.

A.1 Details of Entities and Attributes

We first identify entity types from existing datasets such as the Relations Dataset (Hernandez et al., 2024) and RIPPLEEDITS (Cohen et al., 2024), where each entity type potentially contains thousands of instances. We then source the entities and ground truth references for attribute values from online sources.[3] [4] [5] [6] [7] [8] These online sources are distributed under MIT, Apache-2.0, and CC-0 licenses. Compared with similar entity types in the Relations Dataset and RIPPLEEDITS, RAVEL has expanded the number of entities by a factor of at least 10 and included multiple attributes per entity.

We show the cardinality of the attributes, most frequent attribute values, and random samples of prompt templates in Table 3.

A.2 The RAVEL Llama2-7B Instance

The RAVEL Llama2-7B instance is used for benchmarking interpretability methods in Section 4. There are a total of 2800 entities in the Llama2-7B instance. Table 4 shows the number of entities, prompt

[3] https://github.com/kevinroberts/city-timezones
[4] https://github.com/open-dict-data/ipa-dict/blob/master/data/en_US.txt
[5] https://github.com/monolithpl/verb.forms.dictionary
[6] https://www.nobelprize.org/prizes/lists/all-nobel-prizes/
[7] https://huggingface.co/datasets/corypaik/coda
[8] https://www.bls.gov/ooh, https://www.bls.gov/cps

Attributes	$\lvert A_E \rvert$	Sample Values	Sample Prompts
Verb			
Definition	986	take hold of, make certain, show, express in words, make	`talk: communicate by speaking; win: achieve victory; {E}:, like: have a positive preference; walk: move on foot; {E}:`
Past Tense	986	expanded, sealed, terminated, escaped, answered	`present tense: {E}, past tense:, write: wrote; look: looked; {E}:`
Pronunciation	986	kənˈfjuːz, fɹˈnɪʃ, bɔɪl, mˈʃʊər, tɪp	`create: kriˈeɪt; become: bɪˈkʌm; {E}:, begin: bɪˈgɪn; change: tʃeɪndʒ; {E}:`
Singular	986	compensates, kicks, hunts, earns, accompanies	`tell: tells; create: creates; {E}:, present tense: {E}, 3rd person present:`
Physical Object			
Category	29	plant, non-living thing, animal, NO, fruit	`bird is a type of animal: YES; rock is a type of animal: NO; {E} is a type of animal:, Among the categories "plant", "animal", and "non-living thing", {E} belongs to "`
Color	12	green, white, yellow, brown, black	`The color of apple is usually red. The color of violet is usually purple. The color of {E} is usually, The color of apple is usually red. The color of turquoise is usually blue. The color of {E} is usually`
Size	4	cm, mm, m, km	`Among the units "mm", "cm", "m", and "km", the size of {E} is usually on the scale of ", Given the units "mm" "cm" "m" and "km", the size of {E} usually is in "`
Texture	2	soft, hard	` hard or soft: rock is hard; towel is soft; blackberry is soft; wood is hard; {E} is, Texture: rock: hard; towel: soft; blackberry: soft; charcoal: hard; {E}:`
Occupation			
Duty	650	treat patients, teach students, sell products, create art, serve food	`"occupation": "photographer", "duties": "to capture images using cameras"; "occupation": "{E}", "duties": "to, "occupation": "{E}", "primary duties": "to`
Gender Bias	9	he, male, his, female, she	`The {E} left early because, The newspaper praised the {E} for`
Industry	280	construction, automotive, education, health care, agriculture	`"occupation": "sales manager", "industry": "retail"; "occupation": "{E}", "industry": ", "occupation": "software developer", "industry": "technology"; "occupation": "{E}", "industry": "`
Work Location	128	office, factory, hospital, construction site, studio	`"occupation": "software developer", "environment": "office"; "occupation": "{E}", "environment": "`

Table 3: Attributes in RAVEL, continued.

Entity Type	# Entities	# Prompts Templates	# Test Examples in `Entity/Context`	Accuracy (%)
City	800	90	15K/33K	97.1
Nobel Laureate	600	60	9K/23K	94.3
Verb	600	40	12K/20K	95.1
Physical Object	400	40	4K/6K	94.3
Occupation	400	30	10K/16K	96.4

Table 4: Stats of RAVEL in its Llama2-7B instance, created by sampling a subset of examples where Llama2-7B has a high accuracy in predicting attribute values.

templates, and test examples, i.e., the number of base–source input pairs for interchange intervention in the Llama2-7B instance.

The RAVEL Llama2-7B instance is created by filtering examples where the pre-trained Llama2-7B has a high accuracy in predicting attribute values. For each entity type, we take the k entities with the highest accuracy over all prompt templates and the n prompt templates with the highest accuracy over all entities, with the average accuracy over all prompts shown in Table 4. For most attributes, we directly compare

model outputs against the ground truth attribute values. For "latitude" and "longitude" of a city, we relax the match to be ± 2 within the ground truth value. For "pronunciation" of a verb, we relax the match to allow variations in the transcription. For attributes with more open-ended outputs, including "definition" of a verb and "duty" of an occupation, we manually verify if the outputs are sensible. For "gender bias" of an occupation, we check for the consistency of gender bias over a set of prompts that instruct the model to output gender pronouns.

B Method Details

B.1 PCA

For PCA, we extract the 4096-dimension hidden state representations at the target intervention site as the inputs. The representations are first normalized to zero-mean and unit-variance using mean and variance estimated from the training set. We use the `sklearn` implementation[9] to compute the principal components. We then apply L1-based feature selection[10] to identify a set of dimensions that most likely encode the target attribute A. We undo the normalization after projecting back to the original space.

We vary the coefficient of the L1 penalty, i.e., the parameter "C" in the `sklearn` implementation, to experiment with different intervention dimensions. We experiment with C $\in \{0.1, 1, 10, 1000\}$. We observe that regardless of the intervention dimension, the features selected have a high overlap with the first k principal components. For most attributes, the highest `Disentangle` score is achieved when using the largest intervention dimension.

B.2 Sparse Autoencoder

For the sparse autoencoder, we use a single-layer encoder-decoder model.[11] The autoencoder is trained on Wikipedia data as described below.

Model Encoder: Fully connected layer with ReLU activations, dimensions 4096×16384. Decoder: Fully connected layer, dimensions 16384×4096. Latent dimension: 4×4096. The model is trained to optimize a combination of an L2 loss to reconstruct the representation and an L1 loss to enforce sparsity.

Training Data For each entity type, we sample 100k sentences from the Wikipedia corpus, each containing a mention of an entity in the training set. We extract the 4096-dimension hidden states at the target intervention site as the input for training the sparse autoencoder.

Similar to PCA, we apply L1-based feature selection on the latent representation to identify a set of dimensions that most likely encode the target attribute A. We vary the coefficient C of the L1 penalty to experiment with different intervention dimension. The optimal C varies across attributes.

B.3 RLAP

RLAP learns a set of linear probes to find the feature F. Each linear probe aims to predict the attribute value from the entity representations. Similar to PCA and sparse autoencoders, we use the 4096-dimension hidden state representations at the target intervention site as the initial inputs and the corresponding attribute value as labels. In the case of attributes with extremely large output spaces, e.g., numerical outputs, we approximate the output with the first token. Table 5 shows the linear classifier accuracy on each attribute classification task.

We use the official R-LACE implementation[12] and extract the rank-k orthogonal matrix W from the final null projection[13] as F_A. For each attribute, we experiment with rank $k \in \{32, 128, 512, 2048\}$. We run the algorithm for 100 iterations and select the rank with the highest `Disentangle` score on the dev set. The optimal intervention dimension is usually small, i.e., 32 or 128, for attributes that have a high accuracy linear classifier.

[9] https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

[10] https://scikit-learn.org/stable/modules/feature_selection.html#l1-based-feature-selection

[11] https://colab.research.google.com/drive/1u8larhpxy8w4mMsJiSBddNOzFGj7_RTn?usp=sharing#scrollTo=Kn1E_44gCa-Z

[12] https://github.com/shauli-ravfogel/rlace-icml

[13] https://github.com/shauli-ravfogel/rlace-icml/blob/master/rlace.py#L90

Attribute	Entity	Context
City		
Country	0.78	1.00
Continent	0.96	1.00
Latitude	0.18	1.00
Longitude	0.13	1.00
Language	0.60	1.00
Timezone	0.68	1.00
Nobel Laureate		
Field	0.82	1.00
Award Year	0.08	1.00
Birth Year	0.01	1.00
Country of Birth	0.63	1.00
Gender	0.93	1.00
Verb		
Definition	0.03	1.00
Past Tense	0.00	1.00
Pronunciation	0.00	1.00
Singular	0.00	1.00
Physical Object		
Category	0.90	1.00
Color	0.49	1.00
Size	0.86	1.00
Texture	0.75	1.00
Occupation		
Duty	0.06	1.00
Gender Bias	0.17	0.99
Industry	0.43	1.00
Work Location	0.44	1.00

Table 5: Accuracy of linear probes on dev splits using the Llama2-7B residual stream representations extracted from layer 7 above the last entity token. For most attribute, there exists a linear classifier with significant higher accuracy than random baseline on the entity dev split. For all attributes, there exists a linear classifier with close to perfect accuracy on the context dev split.

B.4 DBM-based and DAS-based Methods

For DBM- and DAS-based methods, we use the implementation from the pyvene library.[14] For training data, both methods are trained on base–source pairs with interchange interventions.

For DBM and MDBM, we use a starting temperature of $1e{-}2$ and gradually reducing it to $1e{-}7$. The feature dimension is controlled by the coefficient of the L1 loss. The optimal coefficient for the DBM penalty is around 0.001, while no penalty generally works better for MDBM, as the multi-task objective naturally encourages the methods to select as few dimensions as possible.

For DAS and MDAS, we do not instantiate the full rotation matrix, but only parameterize the k orthogonal vectors that form the feature F_A. The interchange intervention is defined as

$$\texttt{II}(\mathcal{M}, F_A, x, x') = (I - W^\top W)(\texttt{GetVals}(\mathcal{M}(x), \mathbf{N})) + W^\top W(\texttt{GetVals}(\mathcal{M}(x'), \mathbf{N}))$$

where the rows of W are the k orthogonal vectors. We experiment with $k \in \{32, 128, 512, 2048\}$ and select the dimension with the highest Disentangle score on the dev set. For most attributes, a larger intervention dimension, e.g., 512 or 2048, leads to a higher Disentangle score.

B.5 Computational Cost

All models are trained and evaluated on a single NVIDIA RTX A6000 GPU.

For training, the computational cost of sparse autoencoders is the lowest, as training sparse autoencoders does not involve backpropagating through the original Llama2-7B model or computing orthogonal factorization of weight matrices. Each epoch of the sparse autoencoder training, i.e., iterating over 100k examples, takes about 100 seconds with Llama2-7B features extracted offline. The computational cost of RLAP- and DAS-based method largely depends on the rank of the nullspace or the intervention dimension, i.e., the number of orthogonal vectors. For RLAP, it takes 1 hour per 100 iterations with

[14] https://github.com/stanfordnlp/pyvene

Method	Continent		Country		Language		Latitude		Longitude		Timezone		Iso	Cause	Disentangle
Entity															
PCA	32.7	45.2	36.3	58.6	34.2	33.3	32.7	44.2	39.3	35.4	36.0	36.6	35.2	42.2	38.7
SAE	82.5	15.2	40.4	70.0	91.8	5.0	92.1	17.4	93.3	21.2	91.1	13.6	81.9	23.7	52.8
RLAP	89.4	21.0	38.2	55.8	44.6	48.0	58.1	48.2	38.2	54.0	41.1	50.0	51.6	46.2	48.9
DBM	65.9	70.0	44.8	70.6	42.9	54.3	45.1	59.8	44.9	57.0	72.2	54.0	52.6	61.0	56.8
DAS	67.3	86.4	30.1	83.8	36.3	74.0	52.7	63.2	50.3	56.6	71.0	74.0	51.3	73.0	62.1
MDBM	72.6	68.2	58.6	73.0	56.7	52.3	59.1	55.2	59.9	54.4	75.7	56.4	63.8	59.9	61.8
MDAS	92.1	69.2	82.7	65.6	86.4	51.7	91.4	47.6	93.1	46.0	92.9	62.4	89.8	57.1	73.4
Context															
PCA	27.9	46.1	31.4	52.5	29.2	19.0	26.8	40.0	27.5	53.0	28.8	47.5	28.6	43.0	35.8
SAE	65.6	28.9	29.3	75.4	88.6	4.5	87.0	18.0	88.4	26.5	65.8	27.0	70.8	30.0	50.4
RLAP	86.0	21.4	22.4	84.7	36.8	43.0	46.1	55.0	28.3	72.5	34.8	51.0	42.4	54.6	48.5
DBM	58.7	58.6	37.9	66.0	36.4	36.0	38.3	61.4	38.9	69.0	67.4	53.5	46.3	57.4	51.8
DAS	58.9	84.9	17.7	89.3	27.7	54.0	33.9	77.6	40.9	72.5	64.6	73.5	40.6	75.3	58.0
MDBM	65.4	56.4	50.7	67.6	52.1	32.0	51.9	58.2	53.3	66.5	70.0	55.5	57.2	56.0	56.6
MDAS	86.6	64.9	70.5	70.7	90.3	20.0	88.0	57.0	89.8	62.0	90.0	57.5	85.9	55.4	70.6

(a) Scores of city attributes.

Method	Award Year		Birth Year		Country of Birth		Field		Gender		Iso	Cause	Disentangle
Entity													
PCA	24.2	22.7	30.8	2.3	22.4	70.0	24.3	78.3	4.3	81.0	21.2	50.9	36.0
SAE	79.8	0.7	80.1	0.7	39.8	49.0	43.4	54.0	71.3	63.7	62.9	33.6	48.2
RLAP	87.3	0.3	90.3	1.0	68.0	8.7	82.5	54.0	95.3	71.0	84.7	27.0	55.8
DBM	91.8	0.7	98.6	0.3	61.5	32.0	71.3	57.7	92.6	71.7	83.2	32.5	57.8
DAS	57.1	5.0	72.7	2.3	80.9	25.3	80.1	72.7	80.8	77.7	74.3	36.6	55.5
MDBM	40.8	19.3	70.2	2.0	66.9	36.3	69.2	62.3	76.4	79.7	64.7	39.9	52.3
MDAS	83.6	4.0	85.2	2.0	88.8	28.0	86.9	58.0	93.4	78.0	87.6	34.0	60.8
Context													
PCA	19.2	25.4	22.6	3.3	18.4	73.2	23.6	76.0	3.0	67.0	17.4	49.0	33.2
SAE	74.9	1.0	73.8	1.0	38.1	38.3	65.1	28.0	64.8	35.0	63.3	20.7	42.0
RLAP	88.1	0.4	90.3	0.8	54.4	67.3	77.7	67.3	94.0	61.0	80.9	39.4	60.1
DBM	88.1	0.2	96.9	0.0	50.6	50.2	56.1	59.3	96.8	61.7	77.7	34.3	56.0
DAS	42.7	18.4	13.9	7.5	37.1	72.8	30.2	82.3	88.0	72.7	42.4	50.7	46.5
MDBM	38.6	20.6	69.5	2.2	65.8	54.2	66.7	65.7	91.6	72.0	66.4	42.9	54.7
MDAS	80.2	27.4	83.9	12.3	86.6	72.8	90.2	72.0	93.4	73.0	86.9	51.5	69.2

(b) Scores of Nobel laureate attributes.

Table 6: Per-task results.

a feature dimension 4096 and a target rank of 128. For DAS and MDAS with the reduced parameter formulation, the training time for an intervention dimension of 128 (out of a feature dimension of 4096) over 1k intervention examples is about 50 seconds. The computational cost of DBM-based method is about 35 seconds per 1k intervention examples.

For evaluation, the inference speed of our proposed framework is 20 seconds per 1k intervention examples.

C Results

For all methods, we conduct hyper-parameter search on the dev set. We report a single-run test set results using the set of hyper-parameters that achieves the highest score on the dev set. For intervention site, we choose layer 16 for city attributes and layer 7 for the rest attributes.

C.1 Breakdown of Benchmark Results

Table 6 shows the breakdown of benchmark results in Table 2. For each method, we report a breakdown of the highest `Disentangle` score per attribute, i.e., the pair of `Cause` score and `Iso` score that add up to the highest `Disentangle` score. The final score in Table 2 is an average of the `Disentangle` score over all five entity types. For example, for PCA, the `Disentangle` score under the `Entity` setting is $(38.7 + 36.0 + 41.3 + 43.3 + 38.1)/5 = 39.5$.

Method	Definition		Past Tense		Pronunciation		Singular		Iso Cause		Disentangle
Entity											
PCA	4.9	59.5	4.6	95.3	2.1	66.5	4.2	93.3	4.0	78.6	41.3
SAE	93.4	3.5	15.4	87.3	85.4	3.0	14.3	82.3	52.1	44.0	48.1
RLAP	22.1	42.0	15.8	87.3	23.9	45.5	13.5	85.3	18.8	65.0	41.9
DBM	22.0	51.0	16.3	88.7	10.2	58.0	14.2	87.0	15.7	71.2	43.4
DAS	90.3	12.0	11.9	92.0	89.4	19.5	13.6	85.8	51.3	52.3	51.8
MDBM	55.8	30.0	32.8	70.5	66.4	20.0	25.4	75.8	45.1	49.1	47.1
MDAS	97.6	6.5	88.4	1.2	89.5	25.0	85.4	2.5	90.2	8.8	49.5
Context											
PCA	9.6	57.0	8.3	84.3	4.3	44.0	9.2	78.3	7.9	65.9	36.9
SAE	84.3	10.5	16.8	77.3	74.1	5.5	16.2	73.7	47.9	41.8	44.8
RLAP	19.5	46.5	15.0	80.7	19.1	46.5	13.9	79.3	16.9	63.2	40.0
DBM	21.7	53.0	16.3	84.3	12.3	52.5	14.7	81.0	16.3	67.7	42.0
DAS	69.5	36.5	8.7	93.3	77.4	49.0	7.4	89.7	40.7	67.1	53.9
MDBM	64.4	29.5	28.4	70.0	62.9	28.0	27.5	68.0	45.8	48.9	47.3
MDAS	94.5	21.5	74.2	17.3	84.3	44.0	70.3	24.3	80.8	26.8	53.8

(c) Scores of verb attributes.

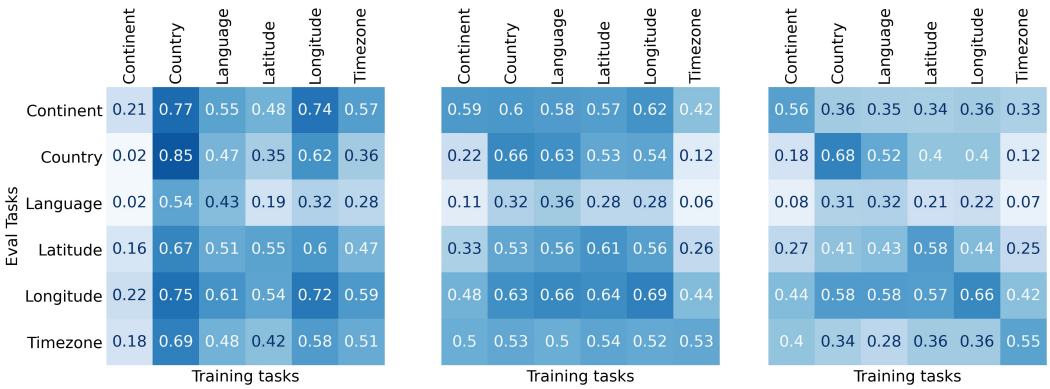
Method	Category		Color		Size		Texture		Iso Cause		Disentangle
Entity											
PCA	45.6	49.8	35.1	63.7	27.7	50.5	26.3	47.5	33.7	52.9	43.3
SAE	94.2	7.9	34.2	63.2	95.0	3.0	95.3	29.0	79.6	25.8	52.7
RLAP	85.6	30.6	83.9	8.0	62.0	28.5	58.7	47.5	72.5	28.7	50.6
DBM	70.1	35.6	62.0	40.0	98.0	2.0	97.7	30.0	81.9	26.9	54.4
DAS	77.3	52.0	79.7	28.7	87.2	24.0	92.0	47.5	84.0	38.1	61.1
MDBM	59.8	48.5	53.5	59.2	74.5	27.5	81.2	49.0	67.3	46.1	56.7
MDAS	85.1	49.8	87.0	19.8	88.5	19.5	91.5	46.5	88.0	33.9	60.9
Context											
PCA	43.1	66.8	40.3	63.3	30.8	46.5	25.4	68.0	34.9	61.1	48.0
SAE	39.9	70.0	43.8	62.2	91.4	6.0	90.9	34.5	66.5	43.2	54.9
RLAP	83.6	47.2	82.3	22.5	64.6	30.0	60.9	61.0	72.8	40.2	56.5
DBM	72.1	47.2	64.6	46.0	97.3	2.5	97.5	32.5	82.9	32.1	57.5
DAS	70.7	75.8	72.2	67.8	82.2	53.5	85.6	64.5	77.7	65.4	71.5
MDBM	64.3	59.0	60.6	59.7	78.6	33.0	83.2	59.5	71.7	52.8	62.2
MDAS	84.8	73.0	83.1	61.5	87.8	46.0	86.3	65.0	85.5	61.4	73.4

(d) Scores of physical object attributes.

Method	Duty		Gender Bias		Industry		Work Location		Iso Cause		Disentangle
Entity											
PCA	39.9	33.7	28.1	61.7	36.3	38.0	35.9	31.0	35.1	41.1	38.1
SAE	68.9	4.0	57.1	49.0	61.7	10.5	64.3	13.0	63.0	19.1	41.1
RLAP	62.1	17.7	93.8	44.0	58.9	18.5	62.0	18.0	69.2	24.5	46.9
DBM	59.3	23.3	93.2	42.7	67.2	18.3	66.4	16.0	71.5	25.1	48.3
DAS	59.8	23.0	83.7	75.7	57.9	29.3	57.9	27.0	64.9	38.7	51.8
MDBM	52.0	35.3	81.7	66.0	57.8	29.5	59.3	24.5	62.7	38.8	50.8
MDAS	82.5	12.0	85.0	70.0	82.5	17.5	83.7	14.5	83.4	28.5	56.0
Context											
PCA	39.2	45.0	21.9	68.0	33.8	42.7	38.3	44.5	33.3	50.0	41.7
SAE	66.7	7.7	47.7	61.0	58.9	14.3	65.1	14.5	59.6	24.4	42.0
RLAP	60.3	23.0	92.5	51.0	56.7	23.3	62.3	24.0	68.0	30.3	49.1
DBM	49.5	14.7	87.3	29.5	56.4	18.0	56.4	21.5	62.4	20.9	41.7
DAS	46.9	49.7	79.7	85.0	44.2	55.3	46.0	46.0	54.2	59.0	56.6
MDBM	43.6	22.7	77.7	70.5	54.2	31.3	60.9	27.0	59.1	37.9	48.5
MDAS	78.7	32.0	81.0	85.5	70.1	38.7	74.1	27.0	75.9	45.8	60.9

(e) Scores of occupation attributes.

Table 6: Per-task results, continued.



(a) Cause score from RLAP. (b) Cause score from DBM. (c) Cause score from MDBM.

Figure 4: Additional feature disentanglement results for RLAP, DBM, and MDBM methods.

C.2 Additional Attribute Disentanglement Results

In Figure 3, we show the feature entanglement results from DAS and MDAS. We provide additional results from all other supervised methods: RLAP, DBM, and MDBM in Figure 4. Though these methods are trained on different objectives and identify different features F_A, they show similar patterns in terms of entanglement between attribute representations. For all methods, representations of most attributes are entangled with "continent" (and "timezone", which for most cases starts with the continent name). Representations of attributes such as "county–language" are also highly entangled.