HYPERDAS: TOWARDS AUTOMATING CAUSAL INTERPRETABILITY WITH HYPERNETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Mechanistic interpretability has made great strides in identifying neural network features (e.g., directions in hidden activation space) that mediate concepts (e.g., *the birth year of a Nobel laureate*) and enable predictable manipulation. Distributed alignment search (DAS) leverages supervision from counterfactual data to learn concept features within hidden states, but DAS assumes we can afford to conduct a brute force search over potential feature locations. To address this, we present HyperDAS, a transformer-based hypernetwork architecture that (1) automatically locates the token-positions of the residual stream that a concept is realized in and (2) learns features of those residual stream vectors for the concept. In experiments with Llama3-8B, HyperDAS achieves state-of-the-art performance on the RAVEL benchmark for disentangling concepts in hidden states. In addition, we review the design decisions we made to mitigate the concern that HyperDAS (like all powerful interpretabilty methods) might inject new information into the target model rather than faithfully interpreting it.

1 INTRODUCTION

Causal interpretability methods promise to demystify the internal workings of black-box language models (LMs), thereby helping us to more accurately control these models and predict how they will behave. Automating such efforts is critical for interpreting our largest and most performant models, and great strides toward this goal have been made for circuit discovery (Conmy et al., 2023; Rajaram et al., 2024) and neuron / feature labeling (Bills et al., 2023; Huang et al., 2023; Schwettmann et al., 2023; Shaham et al., 2024). In the present paper, we complement these efforts by taking the first steps toward automating interpretability for identifying features of hidden representations (e.g., directions in activation space) that mediate concepts (Mueller et al., 2024; Geiger et al., 2024a).

Interventions on model-internal states are the building blocks of causal interpretability. To establish that features of a hidden representation are mediators of a concept, a large number of *interchange intervention* (Vig et al., 2020; Geiger et al., 2020; Finlayson et al., 2021) experiments are performed on the LM. Interchange interventions change features to values they would take on if a counterfactual input were processed. For example, if the concept is C = *the birth year of a person*, we can fix the features F of an LM processing the input *Albert Einstein was born in* to the value they take for *Marie Curie was a chemist*. If the output changes from *1879* to *1934*, we have a piece of evidence that F mediates C. The field has developed a variety of methods for learning such interventions, but all of them require a brute-force search through all potential hidden representations.

To address this significant bottlebeck for causal interpretability, we propose HyperDAS, a method to automate this search process via a hypernetwork, i.e., a network trained to manipulate a target model. In the HyperDAS architecture, a transformer-based hypernetwork localizes a concept within the residual stream of a fixed layer in a target LM by:

1. Encoding a language description (e.g., *the birth year of a person*) of a concept using a transformer that can attend to the target LM processing a *base prompt* (e.g., *Albert Einstein was born in*) and a *counterfactual prompt* (e.g., *Marie Curie was a chemist*).
2. Pairing tokens in the base and counterfactual prompts (e.g., align "Cur" with "Ein") with an attention mechanism using the encoding from (1) as a query and token-pairs as keys/values.

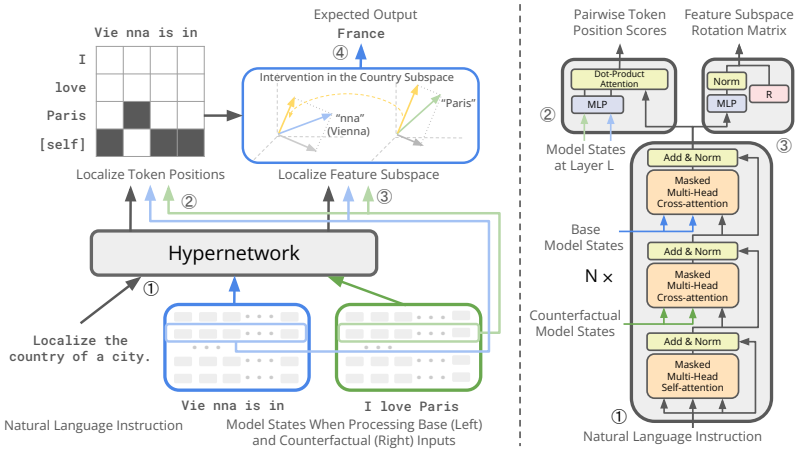


Figure 1: **The HyperDAS framework.** To find the features that mediate the concept of "country", **(1) Encoding Intervention** A natural language description of the concept, "Localize the country of a city," is provided to the Hypernetwork. This Hypernetwork is a decoder-only transformer with two additional cross-attention blocks attending to the hidden states from the target LM when prompted by the base sentence "Vienna is in" and the counterfactual sentence "I love Paris." **(2) Generating Location** With the representation from step 1 as query, HyperDAS performs attention to select the counterfactual/base token pair "nna" and "Paris" as the localization the concept of "country" **(3) Generating Feature Subspace** With the representation from step 1 as encoding, HyperDAS generates an orthogonal matrix that spawns the feature for "country". **(4) Interchange Intervention** With the intervention location from step 2 and feature subspace from step 3, HyperDAS performs causal intervention by patching the features from "Paris" into "nna" in the country subspace, resulting in the model to predict "nna" from the base prompt "Vienna is in".

3. Selecting features of the residual stream via a fixed orthogonal matrix that undergoes a Householder transformation (Householder, 1958) using the encoding from (1).
4. Patching the selected residual stream features of aligned tokens from the base prompt to the values they take on in the residual stream of aligned tokens from the counterfactual prompt.

We benchmark HyperDAS on the RAVEL interpretability benchmark (Huang et al., 2024), in which concepts related to a type of entity are disentangled. For example, we might seek to separate features for the *birth year* and *field of study* of a Nobel laureate. HyperDAS achieves state-of-the-art performance on RAVEL with a single model. Greater gains are achieved when a separate HyperDAS is trained for each entity type (e.g., *Nobel laureates*).

Finally, we address the issue of whether HyperDAS is faithful to the target model. As we use more complex machine learning tools for interpretability, there is an increasing concern that we are not uncovering latent causal structure, but instead injecting new information to steer or edit a model (Meng et al., 2022; Ghandeharioun et al., 2024). If we allow our supervised interpretability models too much power, we run the risk of false-positive signals. Thus, we conclude with a discussion of how our decisions about architecture, training, and evaluation were made in order to mitigate these concerns for HyperDAS.

2 Background

Automating Interpretability Workflows The growing size and complexity of language models demands scalable techniques for interpretability. Two major directions include localizing task-specific information flow to connected model components (Conmy et al., 2023; Marks et al., 2024; Rajaram et al., 2024; Ferrando & Voita, 2024) and finding feature subspaces that capture human-interpretable concepts (Geiger et al., 2024b; Wu et al., 2024; Huben et al., 2024; Braun et al., 2024). Depending on how the feature space is discovered, some methods also require an additional step of automating feature labeling with natural language descriptions (Mu & Andreas, 2021; Hernandez et al., 2022; Bills et al., 2023; Huang et al., 2023; Shaham et al., 2024). In this work, we focus on the

second direction, taking a step towards automating the workflow of localizing human-interpretable concepts in the feature subspaces of LLMs.

Identifying Features that Mediate Concepts Interchange interventions (Vig et al., 2020; Geiger et al., 2020) is a method to identify neural representations that are causal mediators of high-level concepts. Geiger et al. (2024b); Wu et al. (2024) further extend interchange interventions to localizing concepts in rotated feature subspaces. However, these methods require an exhaustive search over all layers and tokens to measure causal effects at each position. In practice, the lack of an effective search method leads to heuristics in token selection. For examples, in knowledge editing and model inspection, a widely held assumption is that the entity information is localized to the last entity token (Meng et al., 2022; 2023; Hernandez et al., 2024; Geva et al., 2023; Ghandeharioun et al., 2024), which does not hold for all entities (Meng et al., 2022). Our purposed method directly addresses this problem by using an end-to-end optimization to automatically learn to select the intervention site across all tokens, conditioned on the entity and concept to localize.

The RAVEL Dataset The RAVEL benchmark evaluates how well an interpretability method can localize and disentangle entity attributes through causal interventions. An example consists of a base prompt that queries a specific attribute of an entity (e.g., *Albert Einstein studied the field*), a counterfactual prompt containing a different entity of the same type (e.g., *Poland declared 2011 the Year of Marie Curie*), an attribute targeted for intervention (e.g., *fields of study* or *birth year*), and a counterfactual label for the base prompt. The label would be *physics* if the targeted attribute is *birth year*, i.e., the intervention should not affect *the field of study* attribute, and it would be *chemistry* if the targeted attribute is *field of study*.

Distributed Interchange Interventions Counterfactual inputs in the RAVEL dataset exist to support evaluations with *distributed interchange interventions* on the features of a hidden representation $\mathbf{H}$ that encode an attribute in the original model $\mathcal{M}$. In our experiments, features will be lines in activation space encoded in a low-rank orthogonal matrix $\mathbf{R}$. We perform an intervention that fixes the linear subspace spanned by $\mathbf{R}$ to the value it takes for counterfactual input $\hat{\mathbf{x}}$:

$$\mathbf{H} \leftarrow \bar{\mathbf{h}} + \mathbf{R}^{-1}\big(\mathbf{R}(\hat{\mathbf{h}}) - \mathbf{R}(\bar{\mathbf{h}})\big) \tag{1}$$

where $\bar{\mathbf{h}}$ and $\hat{\mathbf{h}}$ are the values that variable $\mathbf{H}$ has when the model $\mathcal{M}$ is run on $\bar{\mathbf{x}}$ and $\hat{\mathbf{x}}$, respectively.

RAVEL Metrics The metric from the RAVEL dataset has two components. The Cause score is the proportion of interchange interventions that successfully change the attribute that was targeted, and the Iso score is the proportion of interchange interventions that successfully do not change an attribute that was not targeted. The Disentangle score is the average of these two.

Distributed Alignment Search The RAVEL evaluations use distributed alignment search (DAS; Geiger et al. 2024b) as a baseline for learning the features of a hidden representation that mediate an attribute. A rotation matrix is optimized on RAVEL examples with base input $\bar{\mathbf{x}}$, counterfactual input $\hat{\mathbf{x}}$, and counterfactual label y using the following loss:

$$\mathcal{L}_{\mathsf{DAS}} = \sum \mathsf{CE}\big(\mathcal{M}_{\mathbf{H}\leftarrow\bar{\mathbf{h}}+\mathbf{R}^{-1}\left(\mathbf{R}(\hat{\mathbf{h}})-\mathbf{R}(\bar{\mathbf{h}})\right)}(\bar{\mathbf{x}}), y\big) \tag{2}$$

where $\mathcal{M}_{\gamma}(\bar{\mathbf{x}})$ is the output of the model $\mathcal{M}$ run on input $\bar{\mathbf{x}}$ with an intervention γ. Only the parameters $\mathbf{R}$ are updated.

3 HyperDAS

To localize a concept in a LM $\mathcal{M}$, a HyperDAS architecture consists of a hypernetwork $\mathcal{H}$ that takes in a text specification $\mathbf{x}$ of the target concept and dynamically selects the hidden representations $\bar{\mathbf{h}} = \mathcal{M}(\bar{\mathbf{x}})$ and $\hat{\mathbf{h}} = \mathcal{M}(\hat{\mathbf{x}})$ for base input $\bar{\mathbf{x}}$ and counterfactual input $\hat{\mathbf{x}}$ and learns linear features $\mathbf{R}$, at the l-th layer, that mediate the target concept. Our specific model is as follows.

Target Concept Embedding A token sequence $\mathbf{x}$ of length K that specifies the concept to localize, e.g., *the country a city is in*, is encoded with the embeddings of the target model $\mathcal{M}$ to form the zeroth layer of the residual stream for the hypernetwork $\mathbf{h}^0 = \mathrm{Emb}(\mathbf{x}) \in \mathbb{R}^{K \times d}$.

Cross-attention Decoder Layers After embedding the target concept, we run a transformer with N decoder layers. Besides the standard multi-headed self-attention (MHA) and feed-forward layers (MLP), each decoder block has two additional cross-attention modules to incorporate information from the base and counterfactual runs.

Let $\bar{\mathbf{h}} \in \mathbb{R}^{B \times L \times d}$ and $\hat{\mathbf{h}} \in \mathbb{R}^{S \times L \times d}$ be the stacks of base and counterfactual hidden states from the base and the counterfactual input, where L is the total number of sublayers of $\mathcal{M}$, d is the hidden dimension, and B and S are the sequence length of the base and source example, respectively. Two multi-headed cross-attention modules $\bar{\mathsf{MHA}}$ and $\hat{\mathsf{MHA}}$ allow $\mathcal{H}$ to attend to $\bar{\mathbf{h}}$ and $\hat{\mathbf{h}}$. Each layer of the hypernetwork $\mathcal{H}$ can attend to the residual stream at every layer of the target model.

For the p-th decoder layer, of the hypernetwork $\mathcal{H}$, the three attention mechanisms are as follows

$$\mathbf{h}'_p = \mathsf{MHA}\big(\mathbf{Q}(\mathbf{h}_p), \mathbf{K}(\mathbf{h}_p), \mathbf{V}(\mathbf{h}_p)\big) \tag{3}$$

$$\mathbf{h}''_p = \bar{\mathsf{MHA}}\big(\bar{\mathbf{Q}}(\mathbf{h}'_p), \bar{\mathbf{K}}(\bar{\mathbf{h}}), \bar{\mathbf{V}}(\bar{\mathbf{h}})\big) \tag{4}$$

$$\mathbf{h}_{p+1} = \hat{\mathsf{MHA}}\big(\hat{\mathbf{Q}}(\mathbf{h}''_p), \hat{\mathbf{K}}(\hat{\mathbf{h}}), \hat{\mathbf{V}}(\hat{\mathbf{h}})\big) \tag{5}$$

After the final transformer block is applied, the residual stream vector for the at the last token position $\mathbf{h}_K^{(N)} \in \mathbb{R}^d$, encodes information about the concept targeted for intervention and the target model's base and counterfactual runs. This representation is used to generate pairwise token position scores and feature subspace rotation matrix.

Pairwise Token Position Scores To localize the position of the intervention, we compute an intervention score matrix G, which quantifies the extent of intervention for each pair of base-token and counterfactual-token. The values in G range from 0 to 1, where 0 indicates no intervention and 1 signifies a full intervention. As above, B and S are the sequence lengths of the base and counterfactual inputs, respectively, making G a matrix of dimensions $(B, S+1)$. The element G_s^b denotes the score for replacing the b-th base token with the s-th counterfactual token. The additional row G_{S+1}^b corresponds to the score for retaining the b-th base token without any intervention. Each column G^b forms a probability distribution that adds up to 1, ensuring that each base token is influenced by at most one source in sum. In Figure 1, only the token "nna" (the last token of the entity "Vienna") in the base input receives an intervention score of 1 when paired with the token "Paris" from the counterfactual input. All other base tokens are paired with themselves at the extra row [self], indicating no intervention. This demonstrates that the concept of "country" is localized to the last token of city entities in this example.

To encode each base-counterfactual token pair in G at the layer l of the target LM, we define $\bar{\mathbf{h}}_b^{(l)}$ and $\hat{\mathbf{h}}_s^{(l)}$ to be the l-th layer residual stream representation of the target model at b-th token and s-th token of the base and counterfactual input, respectively. We represent the token pair by:

$$\tilde{\mathbf{h}}_{(b,s)} = F([\bar{\mathbf{h}}_b^{(l)}; \hat{\mathbf{h}}_s^{(l)}]) \tag{6}$$

where $F(.) : \mathbb{R}^{2d} \rightarrow \mathbb{R}^d$ is a linear projection that condenses the concatenated representation into the original dimension d. For the extra row representing retaining the base token, the representation is simply the original base token representation:

$$\tilde{\mathbf{h}}_{(b,S+1)} = \bar{\mathbf{h}}_b^{(l)} \tag{7}$$

Using $\mathbf{h}_K^{(N)} \in \mathbb{R}^d$ from the previous step as the argument to the query Q and the token pair representations $\tilde{\mathbf{h}}$ as the argument to the key K, a multi-head attention weight, softmaxing at every column of the matrix, is calculated to be the pairwise token position scores after averaging across all N attention heads:

$$G = \mathsf{ColumnSoftmax}\left(\frac{\sum Q(\mathbf{h}_K^{(N)})K(\tilde{\mathbf{h}})^T}{N\sqrt{d}}\right) \tag{8}$$

Feature Subspace Rotation Matrix In addition to pairing token positions for token-level localization, HyperDAS constructs linear features that mediate the target concept within a subspace. We learn an orthogonal matrix $\mathbf{R}$ in conjunction with a multi-layer perceptron layer MLP that projects the intervention representation $\mathbf{h}_K^{(N)} \in \mathbb{R}^d$ into a new vector $\mathbf{v} = \mathsf{MLP}(\mathbf{h}_K^{(N)}) \in \mathbb{R}^d$ used to perform a Householder transformation.

Given a non-zero vector $\mathbf{v} \in \mathbb{R}^d$, the Householder transformation $\mathbf{H}$ is defined as:

$$\mathbf{H} = \mathbf{I} - 2\frac{\mathbf{v}\mathbf{v}^\top}{\mathbf{v}^\top\mathbf{v}} \tag{9}$$

where $\mathbf{I}$ is the $d \times d$ identity matrix.

The Householder transformation $\mathbf{H}$ is an orthogonal matrix, which means $\mathbf{RH}$ is orthogonal if the matrix $\mathbf{R}$ is orthogonal. Utilizing this property, we can dynamically select the orthogonal feature subspace $\mathbf{R}'$ based on the intervention representation $\mathbf{h}_N^K$ by computing $\mathbf{R}' = \mathbf{RH}$.

Interchange Intervention With the pairwise token position scores and feature subspace rotation matrix, HyperDAS could perform a interchange intervention that adheres to the principles of causal mediation analysis. For the b-th token position base hidden states $\bar{\mathbf{h}}_b^l$, the source hidden states is:

$$\tilde{\mathbf{h}}_b^{(l)} = G_{(b,S+1)}\bar{\mathbf{h}}_b^{(l)} + \sum_{s=1}^{S} G_{(b,s)}\hat{\mathbf{h}}_s^{(l)} \tag{10}$$

With each column G_b of the matrix being a probability distribution sums up to 1, the source hidden states remain identical to the base hidden states when $G_{(b,S+1)} = 1$. This condition indicates that no intervention occurs at the b-th token. Conversely, If $G_{(b,s)} = 1$ for a specific position s, the source hidden states are exactly those from the s-th counterfactual token.

For the hidden states at l-th layer, we construct a differentiable weighted interchange intervention with the source hidden states $\tilde{\mathbf{h}}^{(l)}$ and low-rank orthogonal matrix $\mathbf{R}'$. At each token position b:

$$\mathbf{H}_b^{(l)} \leftarrow \bar{\mathbf{h}}_b^{(l)} + \mathbf{R}'^{-1}\big(\mathbf{R}'(\tilde{\mathbf{h}}_b^{(l)}) - \mathbf{R}'(\bar{\mathbf{h}}_b^{(l)})\big) \tag{11}$$

3.1 Training

We train HyperDAS on the RAVEL dataset with a two-component loss: a training loss which measures success on the intervention task, and a sparsity loss which incentivizes the model to select unique token-pairings.

Training Loss A RAVEL example consists of a base input $\bar{\mathbf{x}}$, counterfactual input $\hat{\mathbf{x}}$, target concept input $\mathbf{x}$, and a counterfactual label $\mathbf{y}$. When the target concept matches the attribute queried in the base input, the label $\mathbf{y}$ is the attribute of $\hat{\mathbf{x}}$. Otherwise, $\mathbf{y}$ is $\bar{\mathbf{x}}$. The loss is given as:

$$\mathcal{L}_{\text{DAS}} = \sum \text{CE}\big(\ \mathcal{M}_{\mathbf{H}_b^{(l)} \leftarrow \bar{\mathbf{h}}_b^{(l)} + \mathbf{R}^{-1}\left(\mathbf{R}(\bar{\mathbf{h}}_b^{(l)}) - \mathbf{R}(\tilde{\mathbf{h}}_b^{(l)})\right)}(\bar{\mathbf{x}}), \mathbf{y}\big) \tag{12}$$

Sparse Attention Loss Note that the columnwise softmax from Equation 8 only constrains the number of tokens paired with each base token, which allows the each counterfactual token to be paired with multiple base tokens. Thus, we include a sparse attention loss that penalizes cases where one counterfactual token attends strongly to multiple base tokens in each row of matrix G:

$$\mathcal{L}_{\text{sparse}} = \frac{1}{S}\sum_{s=1}^{S}\begin{cases}\text{Sum}(G_s) & \text{if Sum}(G_s) > 1\\ 0 & \text{if Sum}(G_s) \leq 1\end{cases} \tag{13}$$

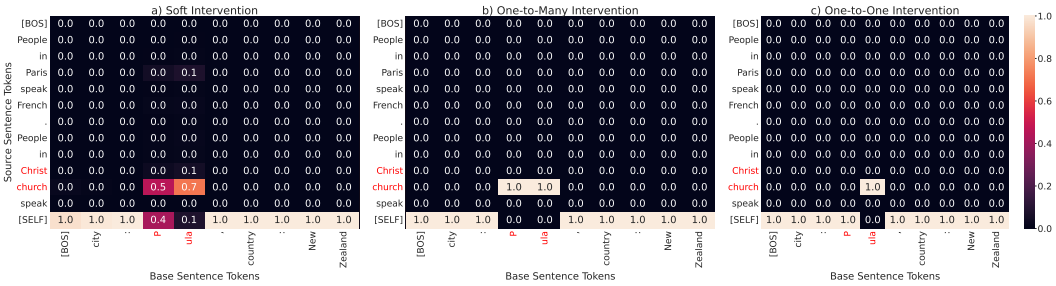


Figure 2: The intervention location while intervening from the counterfactual sentence *"People in Paris speak French. People in Christchurch speak"* to *"city: Pula, country: "*. The attribute targeted for intervention is *country*, so the output should be *New Zealand*. A weighted intervention (left) that constructs a counterfactual representation with a weighted sum, a one-to-many intervention (middle) that snaps each base token to a counterfactual token, and a one-to-one intervention (right) that snaps each base token to a unique counterfactual token.

The final loss is given as $\mathcal{L} = \mathcal{L}_{\text{RAVEL}} + \lambda\mathcal{L}_{\text{sparse}}$. λ is a hyperparameter scheduled from 0 to 1.5 throughout the training.

3.2 Evaluation

HyperDAS is end-to-end differentiable because discrete operations like aligning base and counterfactual tokens are "softened" using softmax operators and sparsity loss constraints. During evaluation, we force these discrete decisions. As the left-most figure of Figure 2 shows, the matrix G contains weights for multiple tokens in the counterfactual sentence. First, each column of the intervention ratio matrix is argmaxed to obtain the counterfactual-base tokens pair with the most weight:

$$G_{(b,s)} = \begin{cases} 1 & \text{if } G_{(b,s)} = \max(G_b) \\ 0 & \text{if } G_{(b,s)} \neq \max(G_b) \end{cases} \tag{14}$$

This is the *One-to-Many Intervention* setting, because multiple base tokens can be aligned with a single counterfactual token. For *One-to-One Intervention* setting, each counterfactual token is only and fully aligned with the base token with the highest weight on top of the *One-to-Many* setting.

4 Experiments

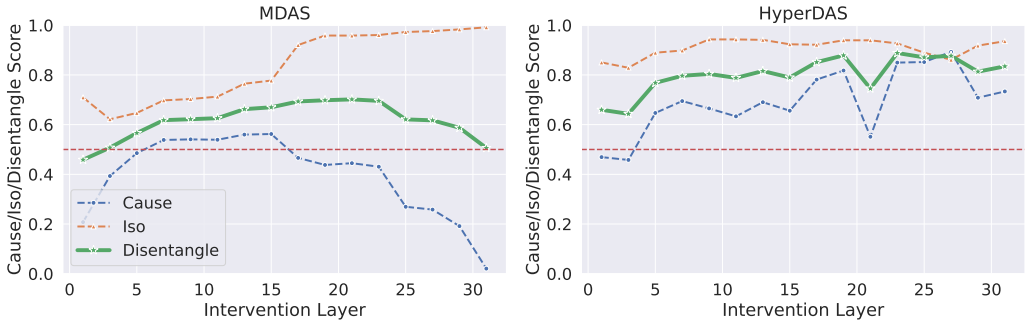
We benchmark HyperDAS on RAVEL with Llama3-8B (Meta, 2024) as a target model. We train a separate HyperDAS model for each of the five entity domains in the RAVEL benchmark, i.e., *cities*, *Nobel laureates*, *occupations*, *physical objects*, and *verbs*. We experimented with initializing the transformer hypernetwork from pre-trained parameters, but found there were no significant advantage in preliminary evaluations.

Multi-task DAS (MDAS) Baseline The current state-of-the-art method on RAVEL is MDAS, which uses a multi-task learning objective to satisfy multiple high-level causal criteria. MDAS, however, relies on manually selected layer and token position for interventions, and it also requires supervision on the attribute. We follow Huang et al. (2024) to intervene on the last entity token of the middle layer, i.e., layer 15 in Llama3-8B.

Masking of the Base Prompt As the hypernetwork has access to the target attribute information from the instruction and the base attribute information from the base model states, a trivial solution the hypernetwork can learn is to condition the intervention location on whether the target attribute matches the base attribute, namely if the two attributes match, attending to a location that has causal effect on the output, otherwise, attending to the extra `[self]` row (See Appendix A.3 for an example). This solution, however, does not find the actual concept subspace. To prevent the hypernetwork from learning this trivial solution, we apply attention mask on the base prompt to mask

Methods	City		Nobel Laureate		Occupation		Physical Object		Verb		Average
	Causal	Iso	Causal	Iso	Causal	Iso	Causal	Iso	Causal	Iso	Disentangle
MDAS	55.8	77.9	56.0	93.5	50.7	88.1	85.0	97.9	74.3	79.6	76.0
HyperDAS	70.8	93.9	55.4	95.1	50.4	99.1	92.7	97.2	93.0	98.9	**84.7**
- All Domains	58.8	90.5	47.6	92.0	75.7	82.1	92.9	94.5	86.9	95.8	80.7

(a) Main results of HyperDAS on five domains of RAVEL with Llama3-8B. HyperDAS achieves the state-of-the-art attribute disentangling performance across the board.



(b) The causal/iso/disentangle score of the baseline method and HyperDAS for the entity type of "city" across the layers of Llama3-8B. For the MDAS baseline, the highest Cause Score also happens at L19.

Figure 3: RAVEL benchmark results. HyperDAS establishes a new state-of-the-art.

out the attribute information. With the masking, the hypernetwork no longer has access to the base attribute information, hence the localization prediction is only conditioned on the target attribute in the natural language instruction.

Crucial Hyperparameters We use 8 decoder block for the hypernetwork and 32 attention heads for computing the pairwise token position attention. The sparsity loss weight is scheduled to linearly scheduled = from 0 to 1.5, starting at 50% of the total steps. A learning rate between 2×10^4 to 2×10^5 is chosen depends on the dataset. Discussion of choices about the sparsity loss is in Sec 4.3. For the feature subspace, we experiment with dimension from 32 up to 2048 (out of 4096 dimensions) and use a subspace of dimension 128 for both the HyperDAS and the MDAS baseline.

Results In Table 3a, we show results on RAVEL for layer 15 of Llama3-8B. In Figure 3b, we also run HyperDAS targeting every 2 layers in Llama3-8Bstarting from the embedding layer. The peak performance of attribute disentanglement for both MDAS and the HyperDAS is around layer 15.

4.1 HyperDAS Variants

We experiment with a variety of architecture design choices, reporting results in Table 5.

Symmetric We enforce symmetry between base and counterfactual inputs during token selection by randomly flipping the order of the concatenation between base and counterfactual hidden representations in Equation 6.

No Hypernetwork Instead of encoding concepts with a transformer and using the resulting vector encoding, we simply learn a vector representation for each target concept in a look-up table.

No DAS We no longer use a rotation matrix at all, and intervene on the entire hidden representation of the selected tokens.

No Cross Attention We remove attention heads, cutting the hypernetwork's access to the original's model hidden states on the base and counterfactual inputs.

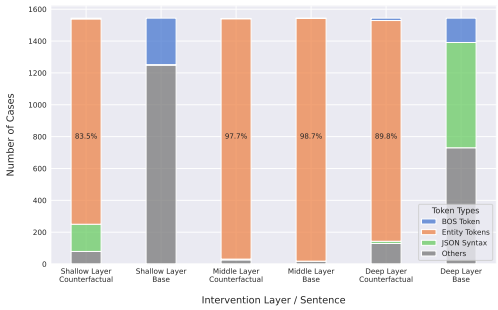


Figure 4: The intervention location, in counterfactual and base sentence, picked by HyperDAS when targeting shallow (7), middle (15) and deep (29) decoder layers.

Ablation	Causal	Iso	Disentangle
HyperDAS	70.8	93.9	82.4
-Symmetric	60.2	98.2	79.2
-No Cross Attention	68.2	83.9	76.1
-No DAS	0.8	97.4	49.1
-No Hypernetwork	15.1	46.9	31.0

Figure 5: Ablation results for HyperDAS. **No DAS** has no rotation matrix and intervenes on entire hidden representations. **Symmetric** randomizes the concatenation order of base and counterfactual hidden representations. **No Hypernetwork** replaces concept encoding via transformer with a vector lookup. **No Cross Attention** removes attention head submodules connecting the hypernetwork and target model.

4.2 Layer-Specific Intervention Behaviors of HyperDAS

HyperDAS searched for an optimal location to intervene within the target hidden state in one layer. We evaluate MDAS and HyperDAS on 16 layers across the model (Figure 3b) and chose an early layer, middle layer, and deep layer for detailed study: Layer 7, Layer 15, and Layer 29. Layer 7 is the earliest at which HyperDAS achieves peak performance with the weighted intervention. Layer 15 is recognized by both methods as offering the optimal performance, while Layer 29 demonstrates effectiveness in one-to-one interventions.

Analysis presented in Figure 4 reveals that HyperDAS consistently targets the entity token in the counterfactual input across all layers, suggesting robust detection of attribute information in the entity token's residual stream from an early stage. However, the choice of intervention location within the base input shows significant variation. For each example in the "city" entity split, we categorize the base and counterfactual token pair that gets the *largest* intervention weight, and classify them into the following categories: (1) **BOS Token** represents the beginning-of-sentence token. (2) **Entity Token** refers to tokens representing entities. (3) **JSON Syntax** includes special characters and syntactic tokens typical of JSON formatted text (e.g., opening curly brace "{"). (4) **Others** comprises all tokens irrelevant to the current analysis, with "is" following the entity token being a common example in both shallow (36%) and deep (29%) layer bases.

At very early layers, HyperDAS displays turbulent behavior, targeting random or even beginning-of-sentence tokens in the base sentence. By the middle layers, the model consistently favors the entity token for intervention, aligning with findings from Huang et al. (2024); Geva et al. (2023). In contrast, at deeper layers, the hypernetwork learns to intervene on unintuitive positions such as syntax tokens within a JSON-formatted prompt, which were previously unknown to store attributes.

4.3 Discussion

HyperDAS establishes a new state-of-the-art performance on RAVEL Our results show that HyperDAS outperforms MDAS, the previous state-of-the-art, across all entity splits at layer 15 in Llama3-8B and across all layers of Llama3-8B for the "cities" entity split.

Householder vectors analysis provides a window into attribute features. To analyze the Householder vectors generated by the model, we collected vectors from each test example and categorized them according to their respective attributes. For each attribute category, a subset of 1,000 samples was randomly selected. We then computed the similarity scores between pairs of attributes by calculating the average cosine similarity across these 1,000 pairs of selected Householder vectors.

We analyze the geometry of the learned householder vectors, with the PCA projection shown in Figure 6. We also compute the average pairwise cosine similarity of householder vectors sampled from within the same attribute or cross two different attributes, as shown in Figure 7. Despite an overall high cosine similarity among all householder vectors associated with the same entity type,

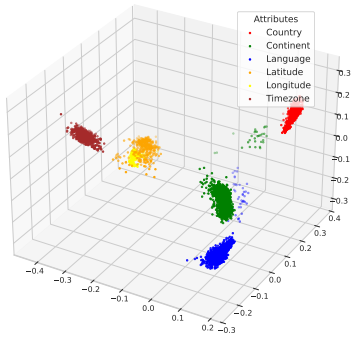


Figure 6: The relative position between the Householder vector (after PCA) of attributes for all the correct predictions in city domain. The clustering indicates that HyperDAS learns different feature subspace for each attribute.

Figure 7: The cosine similarity between the Householder vectors of different attributes in the city domain, computed using 100,000 samples from each attribute. Notably, HyperDAS effectively learns a highly similar feature subspace for the attributes 'Longitude' and 'Latitude'.

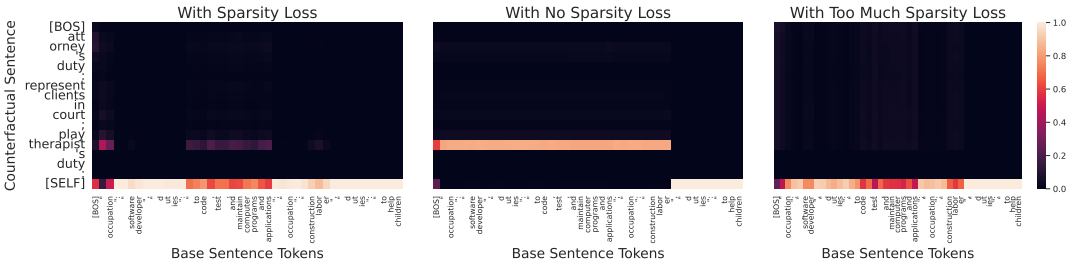


Figure 8: Intervention locations for a base/counterfactual sentences pair with *Occupation* entity-type selected by HyperDAS trained with different amounts of sparsity loss. This comparison illustrates the intervention locations generated by HyperDAS when trained under three different sparsity loss conditions. All three models achieved a Disentangle Score ≈ **94.0**% using weighted interventions. With no sparsity loss (middle), HyperDAS tends to intervene from the last subject token in the counterfactual sentence to most tokens in the base sentence, which yields adequate performance under many-to-one constraints but not under strict one-to-one constraints. With too much sparsity loss (right), the pairwise token selection attention within HyperDAS fails, resulting in interventions that blend all hidden states. Although this approach achieves a near-perfect disentangle score with weighted intervention, the model's does not have interpretable intervention patterns and fails entirely during test time when masks are snapped to align base and source tokens one-to-one.

the household vectors associated with the same attribute form a tighter cluster, with a higher cosine similarity score than pairs of vectors associated with two different attributes. These per-attribute clusters might explain why the learned feature subspace can disentangle different attributes of the same entity, as different attributes are localized into different subspaces of the entity representation.

How do we know HyperDAS uncovers actual causal structures faithful to the target model? On one hand, we should leverage the power of supervised machine learning to develop increasingly sophisticated interpretability methods. On the other hand, such methods are incentivized to "hack" evaluations without uncovering actual causal structure in the target model. We have taken several steps to maintain fidelity to the underlying model structures when training and evaluating HyperDAS, by constraining optimization flexibility to prevent inadvertently steering or editing the model with out-of-distribution interventions.

The weighted interchange interventions used in training hacks the objective without soft constraints via loss terms. The loss term $\mathcal{L}_{\text{sparse}}$ is crucial for ensuring that HyperDAS learns a one-to-one alignment between base tokens and counterfactual tokens (Fig 8). When no sparsity loss is applied, the model aligns the final entity token (e.g., "nna" from Figure 1) to many tokens in the

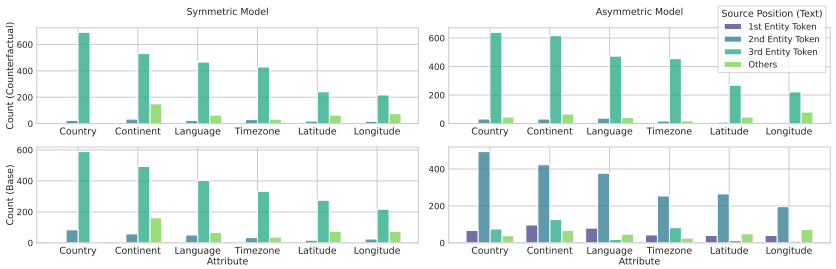


Figure 9: The count of intervention location picked by HyperDAS at the counterfactual prompt (upper) v.s. at the base prompt (bottom) across all the attributes in the city domain on entities with three tokens. The asymmetric variant (right) of HyperDAS favors getting the attribute information from the **last entity token** for the majority of the counterfactual prompts ($\geq 95\%$), and intervene on the **second last entity token**. The symmetric variant (left) favors **last entity token** consistently for both base and counterfactual prompt.

base sentence. These solutions fail during evaluations where token alignments are snapped to be one-to-one. Conversely, with excessive sparsity loss, the model constructs a counterfactual hidden representation that is the linear combination of many hidden states, resulting in a high flexibility optimization scheme that is closer to model steering or editing. This also fails during one-to-one evaluations. See Figure 8 for an example of these pathological settings.

Often only one token is aligned between base and counterfactual inputs. The MDAS baseline performes well on the RAVEL benchmark by one token in the base and one token in the source. However, our new state-of-the-art HyperDAS model will select multiple tokens 53% of the time.

Asymmetric HyperDAS targets different tokens for base and counterfactual examples. Intuitively, if we have localized a concept, then "get" operations that retrieve the concept and "set" operations that fix the concept should both target the same features and hidden representations. For this reason, we consider a variant of HyperDAS that enforces symmetry in the localization of base and counterfactual prompts, which is introduced as **Symmetric** in Sec 4.1. Figure 9 shows the tokens selected by the symmetric and asymmetric variants of HyperDAS. When allowed asymmetric parametrization, networks break symmetry in positional assignments; for a single input prompt, HyperDAS will select different tokens depending on whether that input is the base or counterfactual.

5 Conclusion

In this work, we introduced HyperDAS, a novel hypernetwork-based approach for automating causal interpretability methods. HyperDAS achieves state-of-the-art performance on the RAVEL benchmark, demonstrating its effectiveness in localizing and disentangling entity attributes through causal interventions. Our method's ability to dynamically select hidden representations and learn linear features that mediate target concepts represents a significant advancement in interpretability techniques for language models. We are optimistic that HyperDAS will open new avenues for understanding and interpreting the internal workings of complex language models.

Limitations HyperDAS will only be successful if the target concept is mediated by linear features, however there is emerging evidence that non-linear mediators are a possibility (Csordás et al., 2024; Engels et al., 2024). As discussed extensively in the main text, applying supervised machine learning to interpretability has the potential to lead to false positive results. While we have taken steps to maintain fidelity to underlying model structures, future work should continue to explore the delicate balance between uncovering latent causal relationships and the risk of model steering.

Reproducibility Statement Detailed settings and hyperparameters are provided in Section 4. For transparency and ease of replication, we have included all relevant codes and experimental scripts as supplementary material with the paper.

REFERENCES

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. *URL https://openaipublic. blob. core. windows. net/neuron-explainer/paper/index. html.(Date accessed: 14.05. 2023)*, 2, 2023.

Dan Braun, Jordan Taylor, Nicholas Goldowsky-Dill, and Lee Sharkey. Identifying functionally important features with end-to-end sparse dictionary learning, 2024. URL https://arxiv.org/abs/2405.12241.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/34e1dbe95d34d7ebaf99b9bcaeb5b2be-Abstract-Conference.html.

Róbert Csordás, Christopher Potts, Christopher D. Manning, and Atticus Geiger. Recurrent neural networks learn to store and generate sequences using non-linear representations, 2024. URL https://arxiv.org/abs/2408.10920.

Joshua Engels, Isaac Liao, Eric J. Michaud, Wes Gurnee, and Max Tegmark. Not all language model features are linear, 2024. URL https://arxiv.org/abs/2405.14860.

Javier Ferrando and Elena Voita. Information flow routes: Automatically interpreting language models at scale, 2024. URL https://arxiv.org/abs/2403.00824.

Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen, and Yonatan Belinkov. Causal analysis of syntactic agreement mechanisms in neural language models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1828–1843, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.144. URL https://aclanthology.org/2021.acl-long.144.

Atticus Geiger, Kyle Richardson, and Christopher Potts. Neural natural language inference models partially embed theories of lexical entailment and negation. In Afra Alishahi, Yonatan Belinkov, Grzegorz Chrupała, Dieuwke Hupkes, Yuval Pinter, and Hassan Sajjad (eds.), *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 163–173, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.16. URL https://aclanthology.org/2020.blackboxnlp-1.16.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, and Thomas Icard. Causal abstraction: A theoretical foundation for mechanistic interpretability, 2024a. URL https://arxiv.org/abs/2301.04709.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In *Causal Learning and Reasoning*, pp. 160–187. PMLR, 2024b.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12216–12235, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.751. URL https://aclanthology.org/2023.emnlp-main.751.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes: A unifying framework for inspecting hidden representations of language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=5uwBzcn885.

Evan Hernandez, Sarah Schwettmann, David Bau, Teona Bagashvili, Antonio Torralba, and Jacob Andreas. Natural language descriptions of deep visual features. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2201.11114.

Evan Hernandez, Belinda Z. Li, and Jacob Andreas. Inspecting and editing knowledge representations in language models. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=ADtL6fgNRv.

Alston S Householder. Unitary triangularization of a nonsymmetric matrix. *Journal of the ACM (JACM)*, 5(4):339–342, 1958.

Jing Huang, Atticus Geiger, Karel D'Oosterlinck, Zhengxuan Wu, and Christopher Potts. Rigorously assessing natural language explanations of neurons. In Yonatan Belinkov, Sophie Hao, Jaap Jumelet, Najoung Kim, Arya McCarthy, and Hosein Mohebbi (eds.), *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2023, Singapore, December 7, 2023*, pp. 317–331. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.BLACKBOXNLP-1.24. URL https://doi.org/10.18653/v1/2023.blackboxnlp-1.24.

Jing Huang, Zhengxuan Wu, Christopher Potts, Mor Geva, and Atticus Geiger. RAVEL: Evaluating interpretability methods on disentangling language model representations. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8669–8687, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.470. URL https://aclanthology.org/2024.acl-long.470.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=F76bwRSLeK.

Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *Computing Research Repository*, arXiv:2403.19647, 2024. URL https://arxiv.org/abs/2403.19647.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=MkbcAHIYgyS.

AI Meta. Introducing meta llama 3: The most capable openly available llm to date. *Meta AI*, 2024.

Jesse Mu and Jacob Andreas. Compositional explanations of neurons, 2021. URL https://arxiv.org/abs/2006.14032.

Aaron Mueller, Jannik Brinkmann, Millicent Li, Samuel Marks, Koyena Pal, Nikhil Prakash, Can Rager, Aruna Sankaranarayanan, Arnab Sen Sharma, Jiuding Sun, Eric Todd, David Bau, and Yonatan Belinkov. The quest for the right mediator: A history, survey, and theoretical grounding of causal interpretability, 2024. URL https://arxiv.org/abs/2408.01416.

Achyuta Rajaram, Neil Chowdhury, Antonio Torralba, Jacob Andreas, and Sarah Schwettmann. Automatic discovery of visual circuits. *CoRR*, abs/2404.14349, 2024. doi: 10.48550/ARXIV.2404.14349. URL https://doi.org/10.48550/arXiv.2404.14349.

Sarah Schwettmann, Tamar Rott Shaham, Joanna Materzynska, Neil Chowdhury, Shuang Li, Jacob Andreas, David Bau, and Antonio Torralba. FIND: A function description benchmark for evaluating interpretability methods. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/ef0164c1112f56246224af540857348f-Abstract-Datasets_and_Benchmarks.html.

Tamar Rott Shaham, Sarah Schwettmann, Franklin Wang, Achyuta Rajaram, Evan Hernandez, Jacob Andreas, and Antonio Torralba. A multimodal automated interpretability agent. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=mDw42ZanmE.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 12388–12401. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/92650b2e92217715fe312e6fa7b90d82-Paper.pdf.

Zhengxuan Wu, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah Goodman. Interpretability at scale: Identifying causal mechanisms in alpaca. *Advances in Neural Information Processing Systems*, 36, 2024.

A APPENDIX

A.1 HYPERDAS OVER ALL DOMAINS

Our results at Table 3a show that we can train HyperDAS to achieve state of the art performance on the RAVEL benchmark by training a separate model for each entity type split, which is the set up used to train the previous state of the art MDAS. To test the scalibility and generalizability of HyperDAS, we train a single model across all the entity type splits and evaluate its performance.

Experiment Set-up We aggregate the training split of the dataset from all 5 domains and train HyperDAS for 5 epochs. We adjuste the learning rate from 2×10^{-5} to 5×10^{-4} and schedule the sparsity weight λ ranging from 0.75 to 1.5 starting after 50% of the total steps. This set-up allows the model to first find a stable solution across all domains with soft intervention before forcing it to converge to a single token selection.

Result We report the performance of HyperDAS trained over all entity type split in Table 3a. The model performs better than MDAS but **slightly worse** than HyperDAS trained on individual entity type split by $\mathbf{4.0}\%$. Specifically, HyperDAS-All-Domain performs worse over **city** and **nobel laureate** split, better over **occupation** split, and on-par over **physical object** and **verb** split.

A.2 DATASET SPECIFICATION

Domain/Attribute	# of Cause Example	# of IsolateExample	# of Entity
City	34899/7016	49500/9930	3552/3374
Country	7925/1544	8250/1655	3528/2411
Language	6207/1252	8250/1655	3471/2221
Continent	8254/1658	8250/1655	3543/2567
Timezone	5371/1144	8250/1655	3414/1900
Latitude	3813/743	8250/1655	3107/1519
Longitude	3329/675	8250/1655	2989/1357
Nobel Laureate	39771/6754	44628/7600	928/928
Country of Birth	7218/1356	8908/1520	928/909
Award Year	11037/1904	8930/1520	928/926
Gender	854/96	8930/1520	592/149
Field	9518/1558	8930/1520	928/922
Birth Year	11144/1840	8930/1520	928/927
Occupation	54444/1582	29052/864	799/785
Work Location	24216/724	9684/288	799/708
Duty	12090/371	9684/288	785/522
Industry	18138/487	9684/288	799/600
Physical Object	49114/4659	35285/3636	563/563
Color	14707/1518	8825/909	563/563
Category	13540/1273	8820/909	563/562
Texture	14666/1265	8821/909	563/561
Size	6201/603	8819/909	563/528
Verb	70003/3806	14396/782	986/984
Past Tense	34043/1848	7188/391	986/975
Singular	35960/1958	7208/391	986/978

Table 1: The details of the dataset used for the experiment, in the format of train/test splits. For every model in each setting. Methods are trained on the full dataset of that setting with 5 epochs. The prompts used by the train/test splits are completely disjoint.

A.3 Dataset Preprocessing

HyperDASuses attention mechanism to gather information from the hidden states of the target model $\mathcal{M}$ when running the base and counterfactual sentences. This makes HyperDAS overly powerful as it needs in some situation. Consider the following input:

$$\begin{cases} \text{Base} & \bar{\mathbf{x}} = \text{Vienna, known for its Imperial palaces, is a city in the country of} \\ \text{Counterfactual} & \hat{\mathbf{x}} = \text{I love Paris} \\ \text{Instruction} & \mathbf{x} = \text{Localize the latitude of the city} \end{cases} \tag{15}$$

If the model works as intended, it will intervene on the 'Latitude' subspace, which will leave the 'Country' features intact and therefore the target model will predict Austria.

However, since the model can access the hidden states $\mathcal{M}(\bar{\mathbf{x}})$, it knows that the queried attribute in the sentence is 'Country', which is different than the targeted attribute 'Latitude'. Through training, HyperDAS learns a shortcut to a trivial solution—not doing anything when the target attribute is different from the one mentioned in the sentence. With this shortcut, the **Isolate** objective no longer works and the HyperDASfails to learn disentangled feature subspaces for different attributes.

Figure 10 shows how the HyperDAS may learn a trivial solution to the RAVEL benchmark if the relevant information (base prompt attribute) can be accessed by the model.

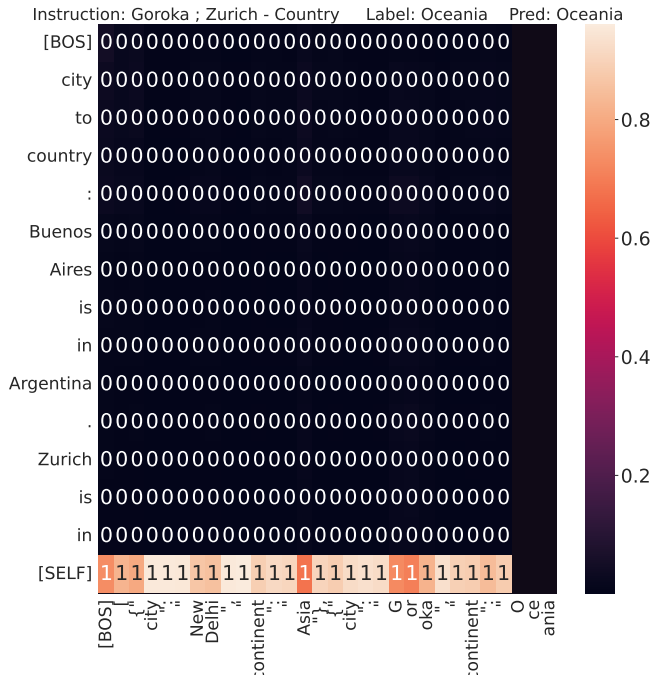


Figure 10: The trivial solution learnt by the HyperDAS on isolate examples when no mask is applied on the attribute token in the prompt. HyperDAS learns to do no intervention at all if it sees the base prompt attribute to be different than the attribute in the instruction.

Therefore, for each pair of prompts $\bar{\mathbf{x}}, \hat{\mathbf{x}}$ at training, we apply an intervention mask to all the tokens starting from the attribute mention. The hidden states from token with intervention mask is not visible to HyperDAS and therefore cannot be selected for intervention. The example becomes:

$$\begin{cases} \text{Base} & \bar{\mathbf{x}} = \text{Vienna, known for its Imperial palaces, is a city in the country of} \\ \text{Counterfactual} & \hat{\mathbf{x}} = \text{I love Paris} \\ \text{Instruction} & \mathbf{x} = \text{Localize the latitude of the city} \end{cases} \tag{16}$$

where the hidden states of the red text is masked from the HyperDAS .

A.4 Loading HyperDAS with Pre-trained Parameters

We have also explored initializing the HyperDAS from a pretrained LM instead of initializing it from scratch. With Llama3-8b (Meta, 2024) as the target LM, we initialize the modules of HyperDAS , besides the multi-head cross-attention heads and pairwise token position scores attention heads, as the copy of the parameters from the target model. We then evaluate the performance of this variation of the model on the city dataset of RAVEL (Huang et al., 2024).

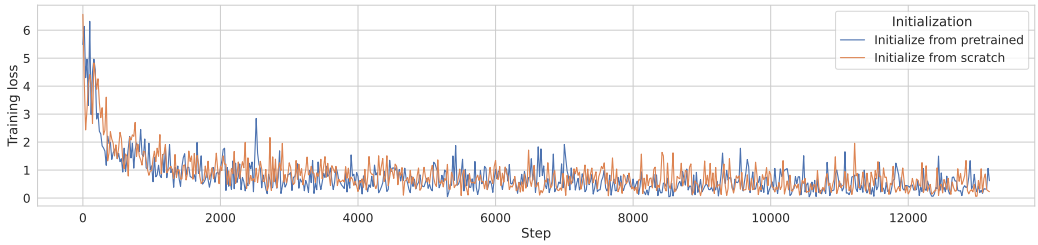


Figure 11: The loss drop of HyperDAS initialized from scratch or from pretrained LM while training on the city dataset of RAVEL.

In Figure 11, we observe that there is no significant difference between the model initialized from scratch and the model initialized from Llama3-8b parameters. However, it remains unknown how would this difference change as the training of HyperDAS scales.

A.5 Subspace Dimension

We experiment with different feature subspace dimension, as shown in Figure 12. We add an trained **sparse autoencoder** as another baseline. Following the exact same setting in (Huang et al., 2024), we train sparse autoencoder that projects the target hidden states into a higher-dimensional sparse feature space and then reconstruct the original hidden states.

A.6 Computation Overhead

HyperDAS's transformer and its access to the model hidden states makes it a much more powerful interpretability method than MDAS, which also makes it more computationally expensive. To capture how much, we trained both HyperDAS and MDAS over the RAVEL-city domain and reported the cost, training speed, and convergence speed (Figure 13). Our HyperDAS model has 10x memory cost compared to MDAS, i.e. more parameters are loaded in, to reach the same training speed per token. However, HyperDAS converges much faster than MDAS after a first few steps.

A.7 Intervention Patterns

Here we include a few demonstrations of the intervention pattern that HyperDAS generates on RAVEL, as shown in Figure 14.

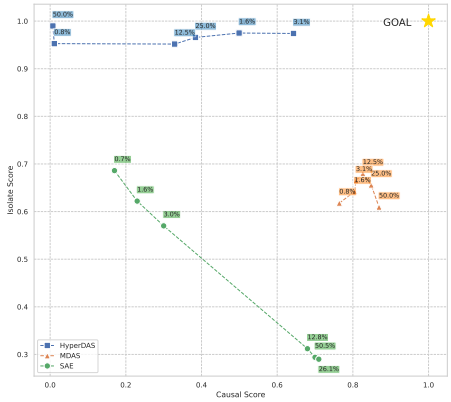


Figure 12: Cause (x-axis) and Iso (y-axis) scores trade-off for HyperDAS, MDAS, and SEA when using different feature size shown as the ratio %. GOAL $(1, 1)$ indicates the score with which the method is able to disentangle the feature subspace perfectly.

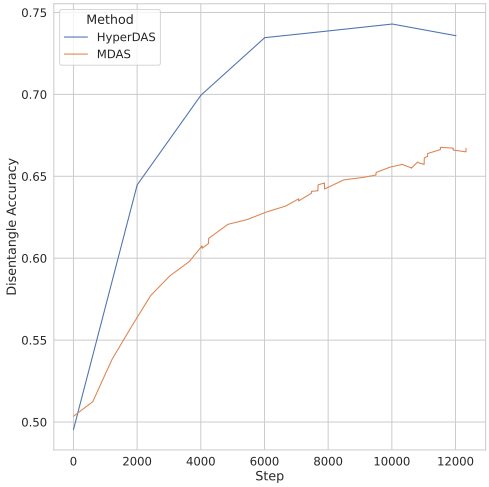
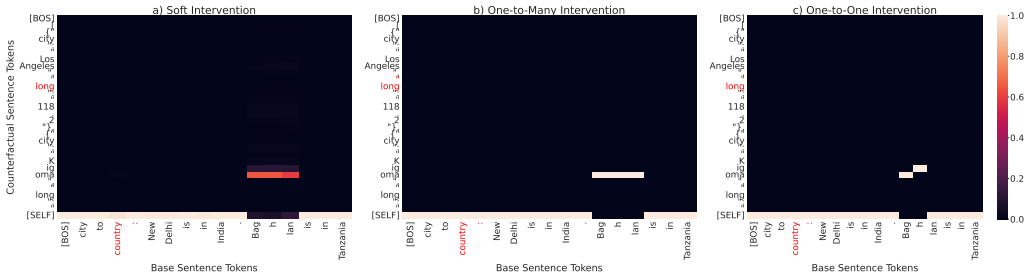
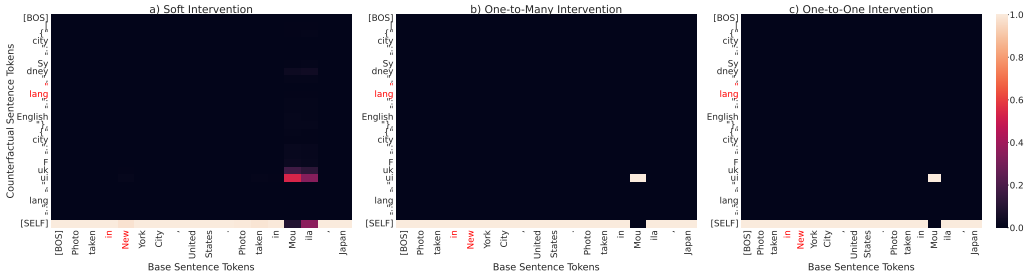


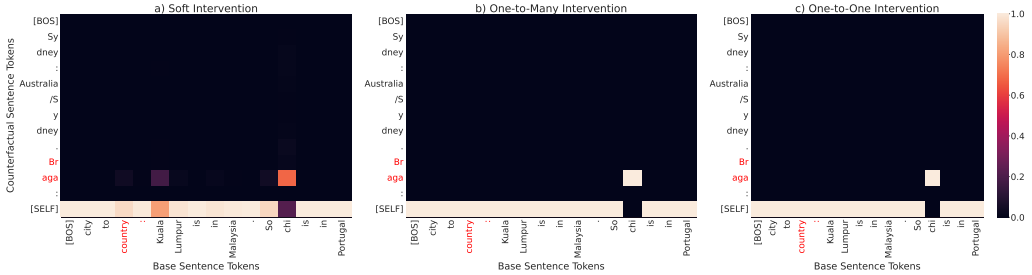
Figure 13: Training process of HyperDAS and MDAS on the city split of the RAVEL dataset with a single A100-80G GPU. HyperDAScosts 68G memory and MDAS costs 6.4G memory.



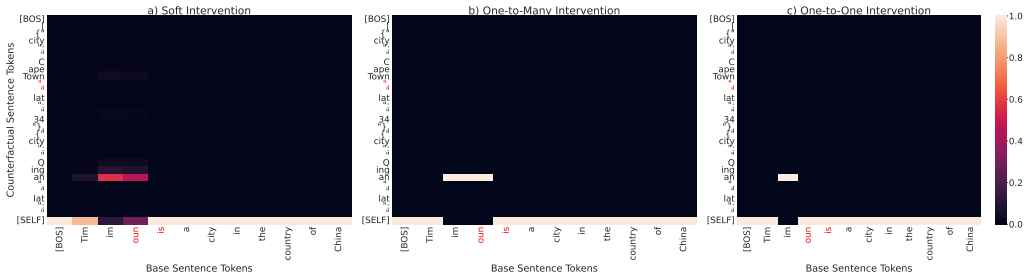
(a)



(b)



(c)



(d)

Figure 14: Four types of intervention patterns.