Interchange Interventions on Vision Models

Emily Bunnapradist
Department of Computer Science
Stanford University
embunna@stanford.edu

Abstract

Convolutional neural networks (CNNs) are experts at image classification, due to their ability to extract image features. These image features include low-level features such as edges and textures, or high-level features such as shapes and objects. To what extent does a convolutional neural network rely on each of these features to classify a model? Our research broadly aims to contribute to the stated need for more interpretability in vision models by showcasing the effectiveness of Distributed Alignment Search (DAS) in enhancing the interpretability of common convolutional neural network (CNN) architectures. We evaluate three high-level causal variables (object color, object shape, and background color) utilizing DAS, and preliminarily determine that in our custom dataset, a ResNet causally depends the most on background color, shedding light on the inner workings of the model and creating a mechanism for evaluating the causal dependence of image features with future datasets and models. [0]

1. Introduction

Recent advances in computer vision systems have enabled image classification performance to surpass human capabilities. At the heart of these systems are convolutional neural networks (CNNs). Despite their success, our understanding of the underlying mechanisms of these networks remains limited. In vision models, features such as color, texture, background, and position are crucial for classification. Although we have some insights into how humans weigh these features in image classification [1], the internal workings of neural networks in this context have not been thoroughly explored.

Distributed Alignment Search (DAS) is an interpretability method that provides insights into how neural networks process these features [5]. This method allows for interchange interventions, enabling detailed examinations of a model's internal response to changes in specific features. In this paper, we trained a ResNet on a custom dataset of shapes and performed interchange interventions on key image features, including color, background, and shape.

Our approach aims to deepen our understanding of the influence these features have on the network's decision-making process, as well as demonstrate the usefulness of DAS in the computer vision field. Furthermore, we also aim to contribute to the stated need for more interpretability in vision models [16] by showcasing the effectiveness of DAS in enhancing the interpretability of common convolutional neural network (CNN) architectures.

2. Related Work

Image Classification. The physical world is extremely diverse, with objects composed of various textures, colors, shapes, and backgrounds. Humans have an exceptional ability to classify objects into different categories, even at a quick glance or with lossy information. This capability stems from our highly developed visual system, which has evolved to efficiently process and interpret the vast array of visual stimuli we encounter in our environment.

Human visual perception is adept at recognizing objects despite variations in size, orientation, lighting, and occlusion [1]. The human brain detects and processes low-level features such as edges, lines, and curves. These features are then combined to form more complex representations of objects [13]. The human visual cortex has been highly studied, with a deep understanding of the inner mechanisms of the visual system. Early stages of visual processing occur in the retina and primary visual cortex (V1), where basic visual information is extracted. Higher-level areas of the visual cortex, such as V2 and V4, process increasingly complex features, leading to the perception of entire objects and scenes in the inferotemporal cortex [6].

Replicating human-like object classification in computer vision systems presents several challenges, including variability in object appearance, background clutter and noise, and generalization across categories. Recent advances in

[0] All code is made available publicly at github.com/emilybunn/cs231nDAS.

computer vision, particularly through the use of deep learning and convolutional neural networks (CNNs), have significantly improved the ability of machines to classify and recognize objects, even beyond human performance [8][9]. However, even with these recent advances, we are seemingly no closer to being able to understand the stages of visual processing or the features that are the most relevant to image classification in computer vision models [16].

Interpretability for Vision Models. Prior interpretability methods for deep computer vision models have been roughly outlined into the following five categories: (1) Visualization of CNN representations in intermediate network layers. (2) Diagnosis of CNN representations. (3) Disentanglement of the 'mixture of patterns' encoded in each filter of CNNs. (4) Building explainable models. (5) Semantic-level middle-to-end learning [12]. These methods all share a common goal, where each aims to develop a deeper understanding of the inner mechanisms of a vision model at a higher level. However, each of these models does so at a highly granular level, utilizing individualized and small-scale approaches to qualitatively interpret the representations of the CNN.

Causal abstraction. Causal abstraction is a conceptual and mathematical framework used to simplify complex systems by representing them at a higher level of abstraction while preserving their causal relationships [3]. Prior attempts at causal abstraction techniques require a *brute-force* search process [2][4], to find an alignment between the states of a low-level model and the variables of a high-level model. Distributed Alignment Search (DAS) finds the alignment between high-level and low-level models by learning an abstraction via gradient descent, rather than *brute-force* [5]. The details of this mechanism are further explored in the Methods section of this paper.

Past approaches to utilizing DAS as an interpretability method have focused on language models. Research has successfully utilized DAS to identify causal mechanisms in alpaca [15], or to find distributed representations in Llama2-7B [10]. Here, DAS exhibits itself as a useful and flexible way to quantitatively discover internal structures that are distributed across bases in language models. However, the application of DAS to computer vision models has not yet been fully investigated, likely due to its recent release.

3. Methods

Baseline Model. We finetune a pre-trained deep neural network named Microsoft Resnet-18 as a baseline model, aiming to achieve a loss of zero [7]. A Residual Network (ResNet) is a type of deep neural network architecture, which addresses the problem of training very deep networks, which previously faced issues such as vanishing and exploding gradients, making it difficult for the training process to converge. The most notable features of a ResNet are the introduction of "skip connections" and bottleneck blocks.

Skip connections, also known as shortcut connections, allow the network to bypass one or more layers by feeding the input of a layer directly to a subsequent layer. This effectively helps in preserving the gradient during backpropagation, which is essential for training deeper networks. The concept of residual learning through skip connections helps the network to learn identity mappings, which simplifies the optimization process and leads to improved performance.

Layer	Output Size	Layer Config	# Repeat
Conv1	112x112	7x7, 64, stride 2	1
MaxPool	56x56	3x3, stride 2	1
Conv2_x	56x56	1x1, 64 3x3, 64 1x1, 256	3
Conv3_x	28x28	1x1, 128 3x3, 128 1x1, 512	4
Conv4_x	14x14	1x1, 256 3x3, 256 1x1, 1024	6
Conv5_x	7x7	1x1, 512 3x3, 512 1x1, 2048	3
AvgPool	1x1	7x7	1
FC	1x1	1000-d	1

Table 1: Resnet-18 Architecture

The Microsoft Resnet-18, in particular, is a ResNet that consists of 18 layers [7]. The architecture includes a series of convolutional layers, batch normalization, ReLU activations, and max pooling, followed by fully connected layers. The model is made available publicly with pretrained weights on the ImageNet-1k dataset, which includes 1,000 different classes and over a million images. For our specific task, we modify the architecture by adjusting the final fully connected layer to output the desired number of classes (`num_classes = 3`).

In our experiments, we fine-tune the Microsoft Resnet-18 on our custom dataset, applying various data augmentation techniques such as random cropping, horizontal flipping, and normalization to enhance the robustness of the model. The training process involves optimizing the model parameters using cross-entropy loss and Adam [11]. The model architecture can be examined in Table 1.

Distributed Interchange Interventions. Distributed interchange interventions are a technique used in the context of model interpretability to systematically modify specific features of the input data and observe how these modifications affect the model's output. This method helps in understanding the role and importance of different features in the model's decision-making process.

Consider a causal model $\mathcal{M}$ with input variables $\mathbf{S}$ and *source* input settings $\{\mathbf{s}_j\}_{j=1}^k$. Furthermore, allow $\mathbf{N}$ to be our target variables in $\mathcal{M}$, where $\mathbf{N} \subseteq \mathbf{S}$. Allow $\mathbf{Y}$ be a vector space with subspaces $\{\mathbf{Y}_j\}_0^k$ that form an orthogonal decomposition and $\mathbf{R}$ to be an invertible function that maps our target variables $\mathbf{N}$ to our vector space $\mathbf{Y}$. Finally, let $\mathsf{Proj}_{\mathbf{Y}_j}$ denote the orthogonal projection of a vector in $\mathbf{Y}$ to our subspace $\mathbf{Y}_j$. [1]

A distributed interchange intervention aims to produce an intervened model DII, which is similar to our original causal model $\mathcal{M}$ with the replacement of the original mechanisms $F_{\mathbf{N}}$ to the intervened mechanisms as follows.

$$F^*_{\mathbf{N}}(\mathbf{v}) = \mathbf{R}^{-1}\Big(\mathsf{Proj}_{\mathbf{Y}_0}\Big(\mathbf{R}(F_{\mathbf{N}}(\mathbf{v}))\Big) + \sum_{j=1}^{k} \mathsf{Proj}_{\mathbf{Y}_j}\Big(\mathbf{R}(F_{\mathbf{N}}(\mathcal{M}(\mathbf{s_j})))\Big)\Big)$$

When performing distributed interchange interventions with neural networks, we assume that $\mathbf{R}$ are rotation operators. The difficulty therein lies in locating the best rotation operator, which can be accomplished using DAS.

Distributed Alignment Search. Distributed Alignment Search (DAS) is a causal abstraction method that is able to learn an alignment between variables and sub-spaces of a large neural representation, utilizing a distributed interchange intervention objective which is optimized with stochastic gradient descent. [2]

Consider a low-level neural network $\mathcal{L}$ with *source* input settings $\mathbf{Inputs}_L$, a high-level algorithm $\mathcal{H}$ with high-level output settings $\mathbf{Out}_H$, and an alignment τ between the input and output variables. We can utilize a distributed interchange intervention objective to minimize the distance between two total high-level settings as follows, where DII is our distributed interchange intervention model and A is our attribute represented by a targetted feature.

$$\sum_{\mathbf{b},\mathbf{s}_1,\ldots,\mathbf{s}_k \in \mathbf{Inputs}_L} \mathsf{Loss}(DII, A)$$

In our intervention, we utilize a Cross-Entropy loss in order to find the best alignment. I also utilize the Pyvene library [14] to facilitate customized interventions in our experiments. Pyvene provides a flexible framework for encoding and implementing interventions, allowing us to tailor our approach to specific model architectures and datasets.

4. Dataset and Features

We developed two custom datasets, a baseline and an intervention dataset, consisting of three classes: 'dax', 'wug', and 'blicket.' We ensure that there are no repeated images between the baseline and intervention datasets using rejection sampling. The full dataset can be found at github.com/emilybunn/cs231nDAS.

Class	Shapes	Background Color
blicket	circle, square	(250, 250, 100)
dax	circle, triangle	(250, 200, 250)
wug	square, triangle	(150, 200, 250)

Table 2: Classes described by shapes and background color.

Baseline dataset. The baseline dataset consists of 400 PIL images of size (244, 244) per class with an 80/20 train/test split. Examples of the dataset can be seen in Figure 1.

Each class consists of two shapes and an associated background color, as shown in Table 2. Furthermore, each shape has a fixed size consistent across shapes and has a fixed color consistent amongst shape type.

Shape Type	Color	Size
circle	(170, 100, 255)	rad=15
square	(100, 200, 0)	len=30
triangle	(255, 50, 50)	len=15

Table 3: Shapes described by their colors and sizes.

Each shape type has an associated color and size range in the training dataset, as shown in Table 3. The positions of each shape are randomized, while ensuring that shapes do not overlap and are entirely present in each image.

Intervention dataset. The intervention dataset consists of 60 PIL image pairs of size (244, 244) per intervention type split evenly across class pairs. Examples of an image pair can be seen in Figure 2.

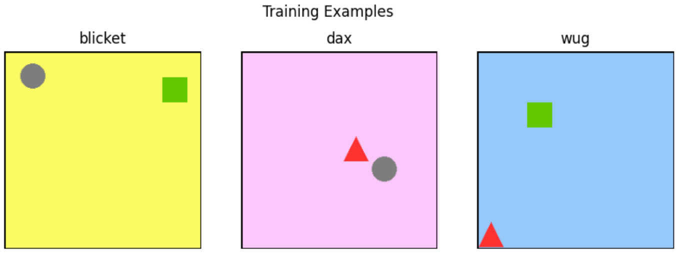

Figure 1: An example image for each class in the baseline dataset.

[1] All of the notation and formulas are taken from the original Distributed Alignment Search (DAS) paper. [5]

[2] All of the notation and formulas are taken from the original Distributed Alignment Search (DAS) paper. [5]

Intervention	Feature Swap Examples
Shape Color	(100, 200, 0) -> (255, 50, 50)
Background Color	(250, 250, 100) -> (250, 200, 250)
Shape Type	square -> triangle

Table 4: The intervention types described by their swaps.

Each image pair consists of one base input and one source input, where there is one feature swap between the pair. An example image pair for each intervention type between a blicket and dax would involve the swapping various features, shown in Table 4. A shape type intervention between a blicket and dax is visualized in Figure 2.

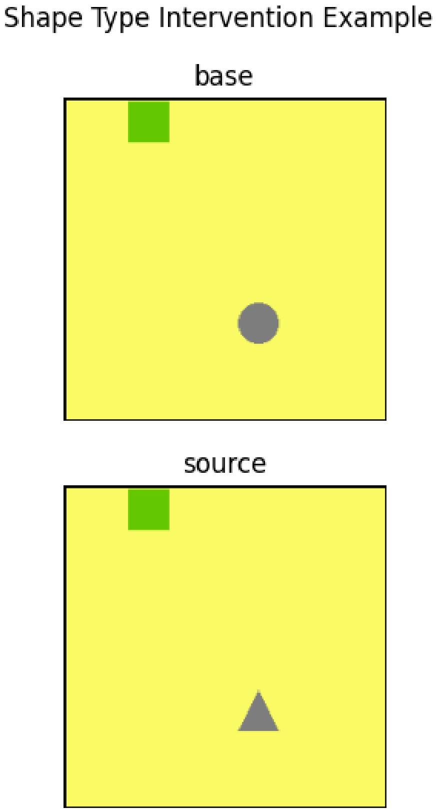

Figure 2: An example image pair for a shape type intervention in the intervention dataset.

5. Experiments/Results/Discussion

5.1. Model

In finetuning the pretrained Resnet-18 model, we utilized the same hyperparameters as the Microsoft Resnet-18 [7] while adjusting the number of classes to match our custom dataset, with which we were able to achieve perfect classification accuracy and a close to zero loss after ten epochs of training on 960 images. [3]

5.2. Experimental Setup

Training a Neural Network. We finetune the pretrained Microsoft Resnet-18 [7] on our custom training set, reaching perfect classification accuracy on both our training and test dataset.

Creating Intervention Dataset. We create an intervention dataset, where each example consists of a base input, a singular source input, a high-level causal variable targetted for intervention, and a counterfactual gold label that we hope the network will output if the interchange intervention works as desired.

As elaborated on in our Dataset section, we create three interventions each targetting a high-level causal variable: shape color, background color, and shape type.

Shape Color Intervention: Swap out the typical shape color for the base image class' different shape to the typical shape color for the source image class' different shape. For instance, consider the following procedure for a shape color intervention between a base class 'blicket' to a source class 'wug': take an image of our base class 'blicket' and modify the differing shape of square to have the same color as a triangle. The counterfactual gold label would be 'wug'.

Background Color Intervention: Swap out the background shape color for the base image class to the typical background color for the source image class'. For instance, consider the following procedure for a background color intervention between a base class 'blicket' to a source class 'wug': take an image of our base class 'blicket' and modify the background color of yellow to be blue. The counterfactual gold label would be 'wug'.

Shape Type Intervention: Swap out the typical shape for the base image class' different shape to the typical shape for the source image class' different shape. For instance, consider the following procedure for a shape intervention between a base class 'blicket' to a source class 'wug': take an image of our base class 'blicket' and modify the differing shape of square to be a triangle. The counterfactual gold label would be 'wug'.

Note that we keep all other factors that we are not intervening on the same, and importantly, ensure that our intervention dataset does not overlap with our training or test dataset. [4] An example of a shape type intervention between

[3] Given that the focus of our paper is more closely aligned to the impact of interventions, our main objective in determining hyperparameters was to ensure that our model would reach 100% classification accuracy on our training and test dataset. The small size and simplicity of our dataset made this easily attainable.

[4] This is somewhat trivial, given that our source image dataset necessarily cannot overlap with our training or test dataset.

a blicket and wug can be seen in Figure 2. More examples can be seen in the Appendix in Figures 6, 7, and 8.

Learn a Distributed Alignment. Utilizing our intervention datasets, we can now learn a distributed alignment between our low-level model (i.e. our ResNet) and our high-level models (i.e. our ResNet intervened on with a high-level model ignoring shape color, ignoring background color, and ignoring shape type). Based on these distributed alignments, the high-level model that is most aligned with the low-level model is, in theory, the high-level causal variable that the low-level model is the most reliant on.

Therefore, we optimize an orthogonal matrix for each high-level model to learn a distributed alignment using our source inputs. From here, we can run our base inputs through the model to output the classification label, which we measure to see if it matches the counterfactual gold label. If so, we expect that the interchange intervention has a strong effect on the model's behavior (i.e. by replacing the background color, the model outputs a different classification label corresponding to the class associated with that background color).

Details on IIA can be found below and details on the algorithm can be found in the Methods section of the paper.

5.3. Metrics

Interchange Intervention Accuracy (IIA). Interchange intervention accuracy (IIA) measures the number of successful interchange interventions, or how many classification outputs match our counterfactual gold label after intervention. When interchange intervention accuracy (IIA) is 100%, the high-level model is a perfect abstraction of the low-level model. When IIA is less than 100%, this still gives us an approximation of the average ”limiting” causal effect of the feature on the causal structure. The formula is as follows. [5]

$$\text{IIA} = \sum_{\mathbf{b},\mathbf{s_1},\ldots,\mathbf{s_k}\in\mathbf{Inputs}_L} \frac{1}{|\mathbf{Inputs}_L^{k+1}|}(DII = A)$$

5.4. Results

We perform an interchange intervention with each intervention type, and derive interchange intervention accuracies (IIAs) by causal variable shown in Table 5. Furthermore, we can qualitatively observe examples of interchange interventions that failed in Figures 3, 4, and 5.

5.5. Discussion

The high interchange intervention accuracy for background color indicates that this feature is a significant causal variable for a high-level model, serving as an almost perfect abstraction of the ResNet. This also suggests that while shape type and shape color are also important causal variables, the fine-tuned ResNet does not rely on them as heavily as it does on background color.

Examples of failure cases reveal that, seemingly independent of position, interventions on shape color and shape type do not strongly affect the model's classification. This implies that each image in the dataset is predominantly in-

Intervention Type	**IIA (%)**
Shape Color	76%
Background Color	95%
Shape Type	68%

Table 5: IIA by Intervention Type Averaged across Class

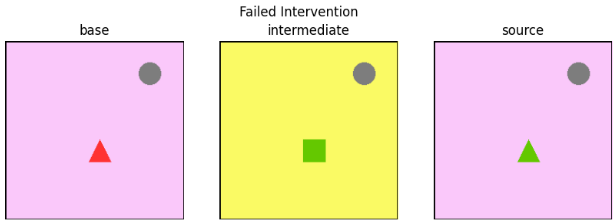

Figure 3: An example failed image pair for a shape color intervention between a dax and a blicket. The base is a dax, the intermediate is a blicket, and the source is a dax where the triangle color is substituted by a square color.

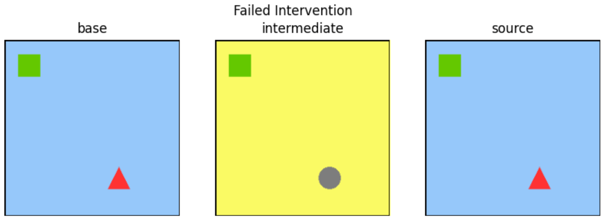

Figure 4: An example failed image pair for a shape color intervention between a dax and a blicket. The base is a wug, the intermediate is a blicket, and the source is a wug where the triangle color is substituted by a square color.

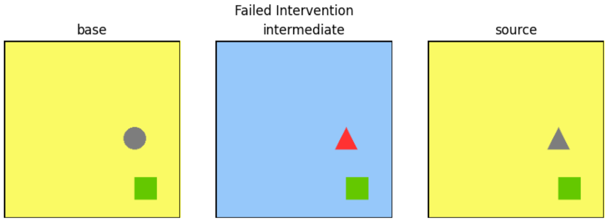

Figure 5: An example failed image pair for a shape type intervention between a blicket and a wug. The base is a blicket, the intermediate is a wug, and the source is a blicket where the circle is substituted by a triangle.

[5] All of the notation and formulas are taken from the original Distributed Alignment Search (DAS) paper. [5]

fluenced by background color. Given the simplicity of the task, the model does not need to depend on shape color or shape type to classify the images correctly.

In interpreting the model's performance, the fact that it achieved 100% classification accuracy in just 2 epochs suggests that it may have learned to rely heavily on one feature exceptionally well. This rapid convergence indicates that the model found a shortcut to solve the classification task, which, in this case, is the background color.

This points to a limitation in the custom dataset, which fails to balance different causal variables properly in a heuristic manner. The model's reliance on background color over other features highlights a potential dataset bias that may lead to overfitting on a single feature. Consequently, the model's performance might degrade when exposed to more complex or varied data where background color is not a reliable predictor.

Interestingly, this reliance on background color aligns with human intuition about the most obvious and telling feature of each class. When humans categorize images, they often prioritize prominent and easily distinguishable features. In this dataset, background color appears to be the most salient feature, mirroring how humans might make quick judgments based on the most noticeable attribute.

Additionally, this scenario also underscores the utility of causal abstraction and Distributed Alignment Search (DAS) in determining the causal dependence of specific features. By systematically analyzing feature interventions, DAS helps in uncovering which features the model is genuinely relying on for its decisions. This insight is crucial for understanding model behavior, improving dataset design, and refining model training strategies.

6. Conclusion/Future Work

To address these findings, future work should focus on creating a more balanced dataset where multiple features contribute equally to the classification task. This approach would prevent the model from over-relying on a single feature and promote a more comprehensive understanding of the relationships between different causal variables. Specifically, this could entail:

- Creating a Range of Background Colors: Ensuring that each class has a range of similar background colors to prevent the model from associating a specific background color too strongly with a particular class.

- Adding Texture to Images: Introducing textures to both the shapes and the background to add another layer of complexity and relevance to the feature set.

- Adding Position to Images: To accomplish this, the dataset could be designed such that each class has a dedicated sector of the image where the shape(s) reside.

- Adjusting Shape Sizes: Making the shapes larger or smaller to see how the model adjusts its reliance on shape-related features.

- Varying the Number of Shapes: Switching to single-shape or multi-shape classes in the dataset to test the model's ability to generalize from different levels of complexity.

- Utilizing a Different Dataset: Using CIFAR-10 or an alternative dataset with real-world images.

As an aside, an additional research direction that could be interesting is utilizing oriented gradients, or some similar method, to glean which features in an image seem the most important, and intervene on that as a high-level feature. This could be an interesting interpretability method, that does not rely on explicitly knowing the potential high-level features of a dataset in advance.

Overall, while the current model's performance reveals certain limitations in the dataset, it also demonstrates the power of interpretability tools like DAS in dissecting and understanding the underlying mechanisms of deep learning models. This knowledge is essential for developing more robust and reliable models that align with human reasoning and perform well across varied and complex datasets. By leveraging DAS, researchers can pinpoint specific weaknesses in the dataset and model, guiding more informed and effective improvements.

7. Appendix

There are more examples of interventions below.

8. Contributions & Acknowledgments

Acknowledgements

We finetune the Microsoft ResNet-18 model trained on imagenet-1k, made publicly available on HuggingFace [7]. We also utilize the Pyvene library, an open-source Python library designed to support customizable interventions [14].

Contributions

Emily wrote all the code for creating the custom training and intervention datasets, finetuning a model, and crafting the interventions, and wrote the paper. A special thank you to Satchel Grant (satchelgrant@stanford.edu) for providing the initial idea and scoping and offering essential guidance throughout the project.

References

[1] C. I. Baker. Revealing the multidimensional mental representations of natural objects underlying human similarity judgements. 2020.

[2] A. Geiger, I. Cases, L. Karttunen, and C. Potts. Posing fair generalization tasks for natural language inference, 2019.

[3] A. Geiger, C. Potts, and T. Icard. Causal abstraction for faithful model interpretation, 2023.

[4] A. Geiger, K. Richardson, and C. Potts. Neural natural language inference models partially embed theories of lexical entailment and negation, 2020.

[5] A. Geiger, Z. Wu, C. Potts, T. Icard, and N. D. Goodman. Finding alignments between interpretable causal variables and distributed neural representations, 2024.

[6] K. Grill-Spector and R. Malach. The human visual cortex. *Annual Review of Neuroscience*, 27(Volume 27, 2004):649–677, 2004.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, 2015.

[9] T. Ho-Phuoc. Cifar10 to compare visual recognition performance between deep neural networks and humans, 2019.

[10] J. Huang, Z. Wu, C. Potts, M. Geva, and A. Geiger. Ravel: Evaluating interpretability methods on disentangling language model representations, 2024.

[11] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017.

[12] Q. shi Zhang and S. chun Zhu. Visual interpretability for deep learning: a survey, 2018.

[13] L. Teichmann and A. N. Rich. The influence of object-color knowledge on emerging object representations in the brain, 2020.

[14] Z. Wu, A. Geiger, A. Arora, J. Huang, Z. Wang, N. D. Goodman, C. D. Manning, and C. Potts. pyvene: A library for understanding and improving PyTorch models via interventions. 2024.

[15] Z. Wu, A. Geiger, C. Potts, and N. Goodman. Interpretability at scale: Identifying causal mechanisms in alpaca. 2023.

[16] R. S. Zimmermann, T. Klein, and W. Brendel. Scale alone does not improve mechanistic interpretability in vision models, 2024.

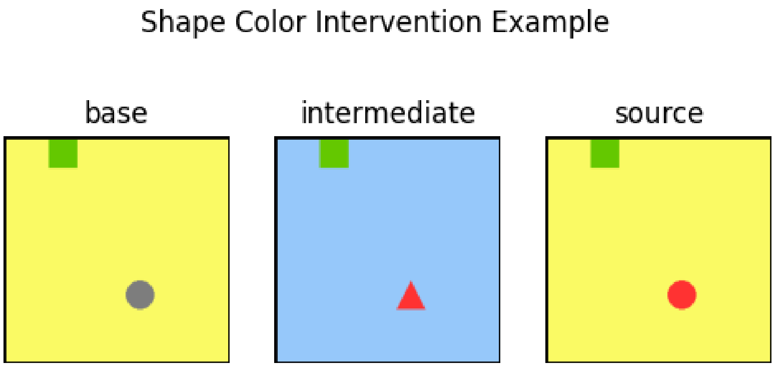

Figure 6: An example image pair for a shape color intervention between a blicket and a wug in the intervention dataset.

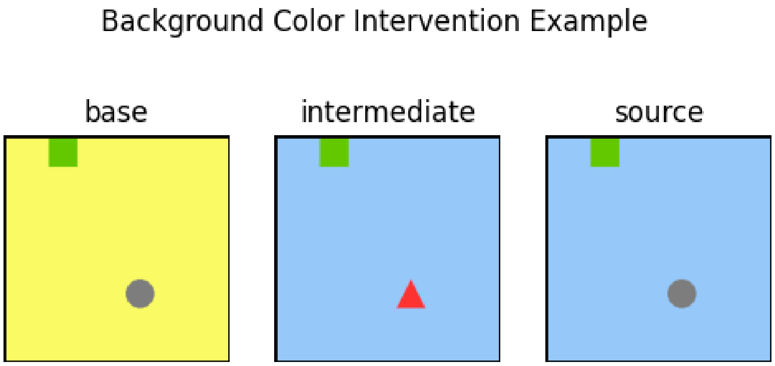

Figure 7: An example image pair for a background color intervention between a blicket and a wug in the intervention dataset.

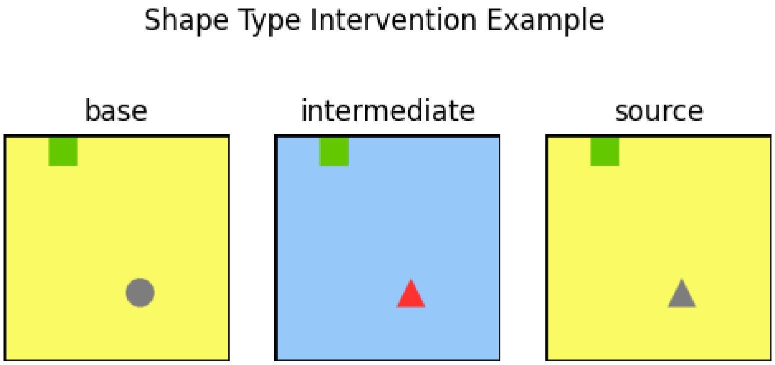

Figure 8: An example image pair for a shape type intervention between a blicket and a wug in the intervention dataset.