Discovering Variable Binding Circuitry with Desiderata

Anonymous Authors[1]

Abstract

Recent work has shown that computation in language models may be human-understandable, with successful efforts to localize and intervene on both single-unit features and input-output circuits. Here, we introduce an approach which extends causal mediation experiments to automatically identify model components responsible for performing a specific subtask by solely specifying a set of *desiderata*, or causal attributes of the model components executing that subtask. As a proof of concept, we apply our method to automatically discover shared *variable binding circuitry* in LLaMA-13B, which retrieves variable values for multiple arithmetic tasks. Our method successfully localizes variable binding to only 9 attention heads (of the 1.6k) and one MLP in the final token's residual stream.

1. Introduction

Deploying powerful generative AI systems requires confidence in the reliability of their outputs, especially with respect to certain high stakes behaviors like manipulation or truthfulness (Carroll et al., 2023; Perez et al., 2022). The emerging field of *mechanistic interpretability* seeks to make model computation human-understandable by explaining the function of particular model components and locating groups of model components responsible for performing certain language tasks. Recent work has had success in localizing computation and intervening on model computation (Li et al., 2022; Burns et al., 2022; Wang et al., 2022; Conmy et al., 2023).

Here, we introduce an automated approach which extends activation patching (Meng et al., 2022; Vig et al., 2020) to localize which components within neural networks (*e.g.* attention heads, MLP layers) are responsible for a specific subtask of model computation. Our method allows for quickly and automatically localizing computation, while only needing to specify *desiderata*, or causal attributes of the target computation. As a proof of concept, we apply our method to automatically discover shared *variable binding circuitry* in LLaMA-13B (Touvron et al., 2023), which retrieves variable values for several arithmetic operations.

Contributions. In this ongoing work, we:

1. Describe a methodology for localizing computation by enumerating desiderata and learning a binary mask by performing causal interventions (Section 3, Figure 1).
2. Present initial results in applying this methodology to localize shared *variable binding* circuitry (Section 4, Figure 2).

2. Background

Circuit analysis. A deep neural network can be represented as a directed acyclic graph with specific nodes to accept inputs, generate outputs, and perform various operations to transform inputs into outputs. Circuit analysis involves localizing and understanding subgraphs within the computational graph of a model that are responsible for specific behaviors, and has had success in both language and vision models (Olah et al., 2020; Wang et al., 2022; Räukur et al., 2022; Chan et al., 2022).

Activation Patching. As introduced in (Meng et al., 2022), activation patching is a technique which uses causal intervention to identify which submodules' activations matter for producing some model output. To measure the effect of a certain activation (*e.g.*, a certain head's output), one may run all the layers of a model until the explored unit with an original input A and an alternative input B. The activations of this submodule with input B are then patched into that of imput A forward pass, and are then feed-forward through the rest of the model. This enables one to assess the role of the specific submodel in the functionality of the whole model, by quantifying how much this intervention shifts the model's output from its original answer on A.

Variable Binding. Variable binding is the process of associating a variable with a specific value, and is a fundamen-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

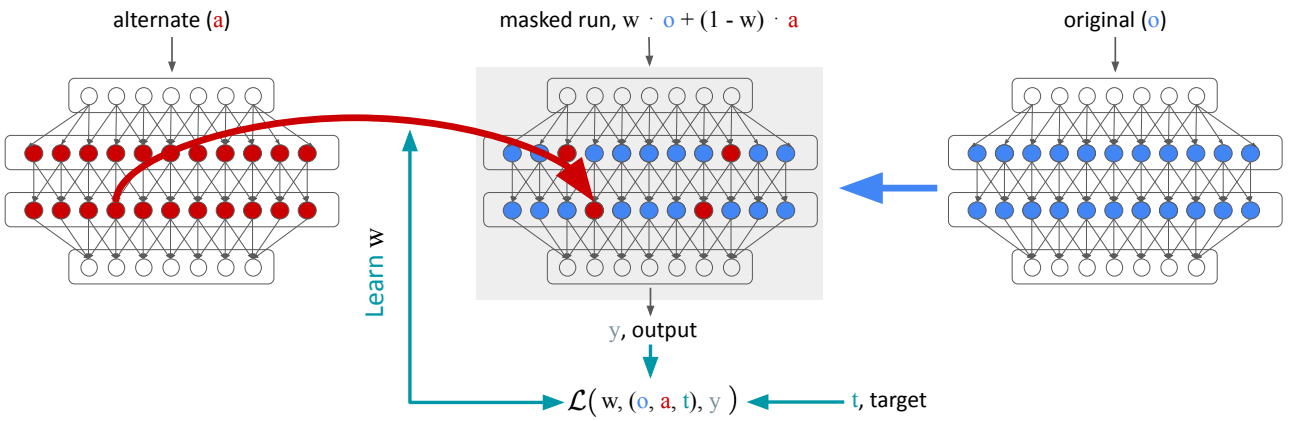

Figure 1. **Localizing computation with desiderata.** The figure depicts training with a single (original, alternate, target) tuple within a desideratum. We learn a mask that combines activations from a into the computation of $\mathcal{M}$ on input o such that the output y moves towards the target t.

tal concept in symbolic reasoning considered essential for solving tasks such as natural language understanding and reasoning (Marcus, 2001). However, it is still a mystery if and how generative AI systems, especially highly capable contemporary language models, implement this process.

See Appendix B for related work.

3. Using Desiderata to Localize Computation

We discover circuitry by enumerating properties of that circuitry, and then learning a binary mask which accords with those properties (Figure 1). We specify properties (or desiderata) in terms of causal interventions with target effects, and combine various interventions into a single objective function. We then learn a sparse mask on the targeted model components, such that applying causal interventions on the masked components alters model behavior to satisfy the objective function.

Model components. As a first step, we specify our set of *model components*. Models can be represented at various levels of granularity. More granular components is more computational expensive, but allows for more specific localization of a model behavior. In Section 4, we decompose LLaMA-13B into a set of attention heads and MLPs, as opposed to more granular (e.g. splitting by Query, Key, and Value matrices) or less granular (e.g. grouping into layers) representations.

Desiderata. We specify the computational circuitry we are attempting to localize by assuming we have successfully localized the circuitry, and then enumerating the effect of various causal interventions on that circuitry. Each desideratum d corresponds to a set of 3-tuples, each of which consists of an original sequence (o), an alternate sequence (a), and a target (t). When the activations of the sought-after circuitry on the input o are replaced with its activations on the input a, the model should output t. The target t could be changing the model's output from $\mathcal{M}(o)$ to $\mathcal{M}(a)$ ($t = \mathcal{M}(a)$), having no effect on the model's output ($t = \mathcal{M}(o)$), or moving the output to some third value.[1]

Each desideratum d contributes to a loss term which measures how well performing activation patching on components $\{c_i\}$ achieves t for all n 3-tuples,

$$\mathcal{L}_d(\{c_i\}) = \frac{1}{n} \sum_{(o,a,t)\in d} \mathcal{L}(\{c_i\}, (o, a, t), y) \qquad (1)$$

Note that some measure of proximity between the induced model output y and the target t is needed.[2] Furthermore, one can combine multiple desiderata into a single objective function $\mathcal{L}_D(\{c_i\}) = \sum_{d\in D} \mathcal{L}_d(\{c_i\})$. Desiderata for the specific case of the value-copying components involved in variable binding are presented in Figure 2 and discussed in Section 4.1.

Learning a Binary Mask. In order to find a set of $\{c_i\}$ that achieve a low value of $\mathcal{L}_D(\{c_i\})$, we use a continuous relaxation of Equation 1. We assign learnable weight $w_c \in [0, 1]$ to each model component, where $w_c = 0$ corresponds

[1]The target could be some third value as well. For example, the target could be a behavior *break*, approximated by distance from a uniform output, or some specified new target value t defined by performing part of a given computation with different values.

[2]Different metrics can be used to measure the proximity between model output and t, such as a change in target logit, or a change in a difference of target logits.

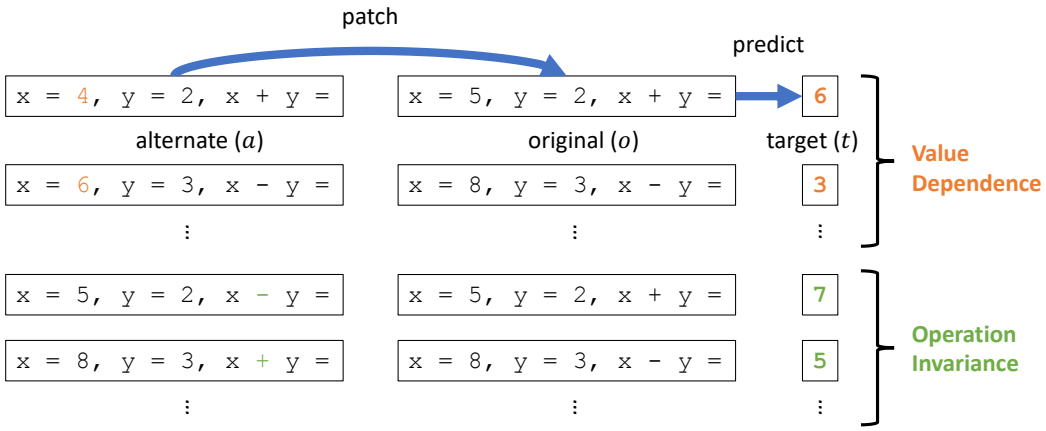

Figure 2. **Variable Binding Desiderata.** Each desideratum is a set of original (o), alternate (a), and target (t) 3-tuples. In the *Value Dependence* desideratum, patching should change the output to the alternate's output; in the *Operation Invariance* desideratum, patching should have no effect.

to fully patching component c with its value v_a from the sequence a, $w_c = 1$ corresponds to not patching c, and $0 < w_c < 1$ corresponds to taking a convex combination of the component's activation value v and the value v_a,

$$w_c \cdot v + (1 - w_c) \cdot v_a \tag{2}$$

Note that when we patch multiple components of the model, the value v of a later-layer component will be influenced by the patching of earlier layers before it itself is combined with v_a as above.

We optimize the continuous mask according to $\mathcal{L}_D$, which measures how well the patching intervention defined by the mask meets our desiderata. We use L0.5 regularization with tunable strength λ over the mask entries to encourage patching only a sparse set of model components (Louizos et al., 2018). Throughout learning, we clamp values between 0 and 1. After training, we round weights to either 0 or 1 to form a binary mask. Empirically, we find that rounding the mask to become binary typically has little effect on its ability to satisfy the desiderata. We speculate this is due to the regularization for sparsity during training.

4. Variable Binding

We apply our method (Section 3) to locate circuitry responsible for retrieving variable values when computing simple arithmetic expressions like those in Figure 2. We use LLaMA-13B, a 40-layer, decoder-only transformer language model, trained on a diverse data set (Touvron et al., 2023).

We hypothesize that there exist subcomponents of LLaMA-13B that, in order to complete sequences like those appearing in Figure 2, copy the value previously assigned to the variable x into the final token's residual stream. We further hypothesize that the x's value is then combined with y's value to compute the desired expression.

We design desiderata to search specifically for this value-copying circuitry. Throughout, we only evaluate accuracy of models based on whether their prediction of the first digit of the answer value is correct; we ensure a diverse set of targets to avoid degenerate solutions. Code to replicate our results is available in a public repository.[3]

4.1. Variable Binding Desiderata

We propose two desiderata to isolate this hypothesized value-copying circuitry:[4]

1. *Value Dependence* (VD; Figure 2, top). Patching our target circuitry with its activations from alternate sequences containing different x values should control which value is copied into the final residual stream. Accordingly, patching the target circuitry should alter the model's output to match the output of the alternate sequences.

2. *Operation Invariance* (OI; Figure 2, bottom). Since we are looking for circuitry shared across arithmetic operations, the specific operation being performed in the expression should not affect the behavior of our target circuitry; it should copy the same value regardless of the operation. Accordingly, we form pairs of sequences and equivalent sequences but with a flipped operation (either addition or subtraction), and enforce that the model's predicted value of the expression should not change when components are patched.

[3] https://anonymous.4open.science/r/anima_submission-D770/README.md

[4] We note that additional desiderata are possible, and would likely improve performance.

	VD Acc. (+, -)	TI Acc. (+, -)	VD Acc. (*)	TI Acc. (+, *)	# Patched
Original Model	18%	91%	11%	93%	0
Incomplete Desiderata (VD)	93%	11%	82%	13%	10
Full Desiderata (VD & TI)	84%	82%	84%	91%	10

Table 1. **Accuracies of patching experiments.** Learning the patching mask according to an incomplete set of desiderata (in particular, only using the Value Dependence desideratum) fails to localize our target computation (Operation Invariance accuracy suffers). Using both desiderata (Full Desiderata) successfully causes Value Dependence behavior while maintaining Operation Invariance. All accuracies are calculated on held-out problems.

4.2. Binary Mask Details

We take our set of model components to be all MLPs and attnetion heads, forming a total of 1640 components, and learn a binary mask as described in Section 3. We only consider patches to each component's contribution to the final-token residual stream, as that is where we expect our value-copying circuitry to be active. For both the VD and OI sequences, we use two-digit variable values and addition and subtraction expressions.

As our proximity measure, when evaluating the model on the VD task, we calculate the difference between the logit of the original answer and the logit of the alternative answer, and see how much this changes due to the masking intervention. For OI, we calculate this same logit difference, but search for masks that produce minimal alteration to this logit difference.

We optimize with Adam (Kingma & Ba, 2017), using a learning rate of 0.01. We alternate between taking gradient steps from the VD loss and the OI loss to save memory. For the our primary mask result described below, we use a weight of $\lambda = 0.03$ for the L0.5 sparsity regularization, though in Appendix A we vary this value to find masks with varying numbers of heads.

5. Results

We find a set of ten components, nine of which are attention heads and one of which is an MLP, such that masking those ten scores well according to the two desiderata as described above. We patch heads according to this mask and evaluate the model's accuracy on a held-out set of VD and OI problems. On VD scenarios, accuracy measures how often the model outputs the answer from the Alternative sequence. On OI scenarios, accuracy measures how often the model outputs the answer from the Original sequence. We expect a mask that finds heads corresponding to the value copying subtask of variable binding to achieve high accuracy in both VD and OI.

We find that the heads located by our method indeed achieve high accuracy in both these tasks (see Table 1, first and second columns). We test the mask further on VD problems involving a multiplication equation instead of addition or subtraction, and on OI problems involving swapping between addition and multiplication, and see high accuracy here as well despite not using multiplication during localization (see Table 1, third and fourth columns).

This is suggestive evidence that these modules are the hypothesized circuitry that moves variable values to the final residual stream before the model operates on these values; these components successfully cause the model's output to change in the case that one of the bound values changes, but not in the case that the operation in the equation changes.

We also find that including both desiderata is necessary to find these heads; with only the VD desideratum, the heads that are found successfully alter model behavior in the VD scenario, but also affect the model's output in the OI scenario (see Table 1, second row). When both desiderata are included in the loss, the modules masked are mostly attention heads in the middle of the model;[5] when only using the first desideratum, the modules masked are a cluster of late-layer MLPs.[6] A possible explanation for this difference is that using only the Value Dependence desideratum permits the mask to find the MLPs that write the computed final value of the expression to the residual stream, whereas adding the second desideratum encourages the mask to find the value-copying circuitry in particular.

6. Conclusion

In this paper, we propose a new approach to localizing internal subtasks of model computation, by searching using a set of causal behavior desiderata. We use the method to localize 10 components responsible for copying variable values in LLaMA-13B. We encourage future work further examining variable binding, extending our desiderata methodology, and applying it to increasingly complex and safety-relevant tasks.

[5]Heads 11.11, 12.0, 12.7, 15.11, 15.25, 17.17, 18.11, 18.18, 19.20, and MLP 27.

[6]MLPs 18, 27, 28, 29, 30, 31, 32, 33, 35, and 36.

References

Burns, C., Ye, H., Klein, D., and Steinhardt, J. Discovering latent knowledge in language models without supervision, 2022.

Carroll, M., Chan, A., Ashton, H., and Krueger, D. Characterizing manipulation from ai systems, 2023.

Chan, L., Garriga-Alonso, A., Goldowsky-Dill, N., Greenblatt, R., Nitishinskaya, J., Radhakrishnan, A., Shlegeris, B., and Thomas, N. Causal Scrubbing: a method for rigorously testing interpretability hypotheses [Redwood Research], December 2022. URL https://www.alignmentforum.org/posts/JvZhhzycHu2Yd57RN/causal-scrubbing-a-method-for-rigorously-testing.

Conmy, A., Mavor-Parker, A. N., Lynch, A., Heimersheim, S., and Garriga-Alonso, A. Towards automated circuit discovery for mechanistic interpretability, 2023.

Geiger, A., Wu, Z., Lu, H., Rozner, J., Kreiss, E., Icard, T., Goodman, N., and Potts, C. Inducing causal structure for interpretable neural networks. In *International Conference on Machine Learning*, pp. 7324–7338. PMLR, 2022.

Geiger, A., Wu, Z., Potts, C., Icard, T., and Goodman, N. D. Finding alignments between interpretable causal variables and distributed neural representations. *arXiv preprint arXiv:2303.02536*, 2023.

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

Li, K., Hopkins, A. K., Bau, D., Viégas, F., Pfister, H., and Wattenberg, M. Emergent world representations: Exploring a sequence model trained on a synthetic task, 2022. URL https://arxiv.org/abs/2210.13382.

Louizos, C., Welling, M., and Kingma, D. P. Learning Sparse Neural Networks through L_0 Regularization, June 2018. URL http://arxiv.org/abs/1712.01312. arXiv:1712.01312 [cs, stat].

Marcus, G. F. Relations between Variables. In *The Algebraic Mind: Integrating Connectionism and Cognitive Science*. The MIT Press, 04 2001. ISBN 9780262279086. doi: 10.7551/mitpress/1187.003.0005. URL https://doi.org/10.7551/mitpress/1187.003.0005.

Meng, K., Bau, D., Andonian, A., and Belinkov, Y. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 36, 2022.

Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., and Carter, S. Zoom in: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001. https://distill.pub/2020/circuits/zoom-in.

Perez, E., Ringer, S., Lukošiūtė, K., Nguyen, K., Chen, E., Heiner, S., Pettit, C., Olsson, C., Kundu, S., Kadavath, S., Jones, A., Chen, A., Mann, B., Israel, B., Seethor, B., McKinnon, C., Olah, C., Yan, D., Amodei, D., Amodei, D., Drain, D., Li, D., Tran-Johnson, E., Khundadze, G., Kernion, J., Landis, J., Kerr, J., Mueller, J., Hyun, J., Landau, J., Ndousse, K., Goldberg, L., Lovitt, L., Lucas, M., Sellitto, M., Zhang, M., Kingsland, N., Elhage, N., Joseph, N., Mercado, N., DasSarma, N., Rausch, O., Larson, R., McCandlish, S., Johnston, S., Kravec, S., Showk, S. E., Lanham, T., Telleen-Lawton, T., Brown, T., Henighan, T., Hume, T., Bai, Y., Hatfield-Dodds, Z., Clark, J., Bowman, S. R., Askell, A., Grosse, R., Hernandez, D., Ganguli, D., Hubinger, E., Schiefer, N., and Kaplan, J. Discovering language model behaviors with model-written evaluations, 2022.

Räukur, T., Ho, A., Casper, S., and Hadfield-Menell, D. Toward transparent ai: A survey on interpreting the inner structures of deep neural networks. *arXiv preprint arXiv:2207.13243*, 2022.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Vig, J., Gehrmann, S., Belinkov, Y., Qian, S., Nevo, D., Singer, Y., and Shieber, S. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33: 12388–12401, 2020.

Wang, K., Variengien, A., Conmy, A., Shlegeris, B., and Steinhardt, J. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small, 2022.

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

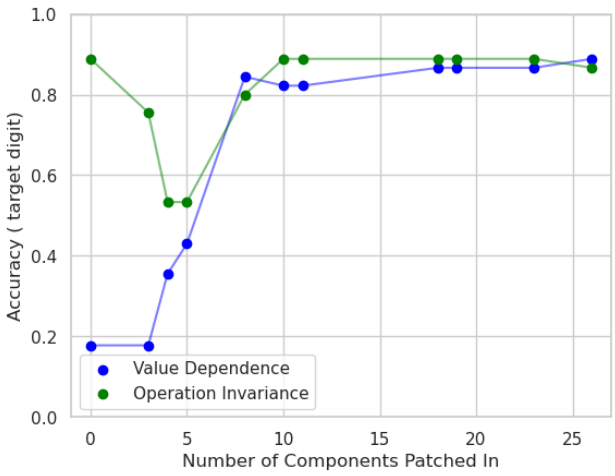

Figure 3. **Evaluating masks of various numbers of heads on held-out VD and OI problems.** Each vertical pair of datapoints corresponds to a mask learned by a training run with a different value of λ, the sparsity regularization weight. With too few components patched, the model does not score well at Value Dependence. We interpret this as indicating that not enough of the value-copying heads have been patched.

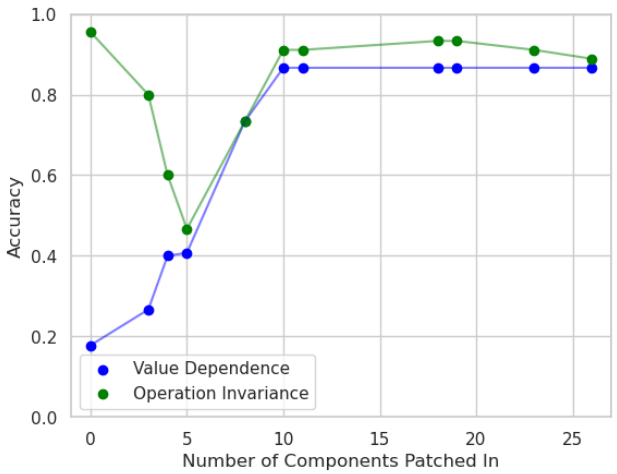

Figure 4. **Transfer to accuracy on multiplication problems.** This graph depicts the same masks as Figure 3 (which were trained on sequences involving only addition and subtraction), but evaluated on all-multiplication Value Dependence problems, and addition-to-multiplication (and vice versa) Operation Invariance problems. Similarly to Figure 3, VD accuracy is low with too few heads patched.

A. Varying regularization strength

We vary the regularization strength λ in order to learn masks with varying numbers of heads. We find that setting the regularization so that the masks learns approximately 10 model subcomponents is the approximate minimum number that can score highly for held-out Value Dependence accuracy and Operation Invariance accuracy, and so we use that setting for the main results of our paper. We also show further that removing the Operation Invariance desideratum causes the mask to score poorly on that criterion.

B. Related Work

Previous work has developed automated approaches to localizing computation (Conmy et al., 2023; Geiger et al., 2022; Wu et al., 2023). Our work varies from Conmy et al. (2023) in learning a mask and considering a broader class of ablations (patches to change behavior, instead of just preserve). Our work shares features with recent work from Geiger et al. (2023) and Wu et al. (2023), but differs in attempting to isolate shared computation common in multiple input-output circuits as opposed to understanding full input-output circuits.

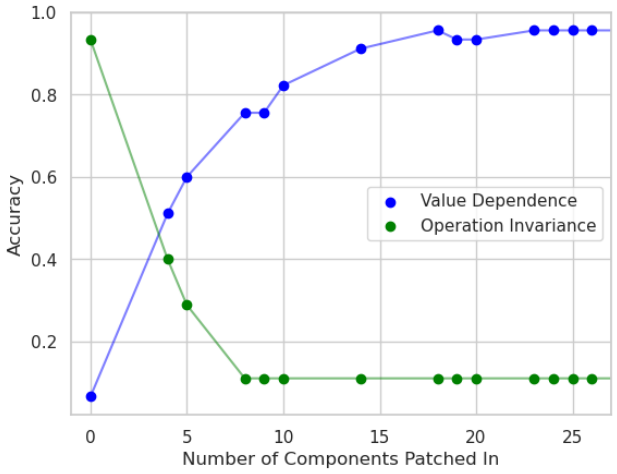

Figure 5. **Varying regularization strength with incomplete desiderata.** This graph demonstrates learning a mask with only the Value Dependence desideratum. Again, each vertical pair of datapoints corresponds to a mask learned by a training run with a different value of λ, the sparsity regularization weight. Unlike when the mask is optimized according to both desiderata, these masks fail to achieve high accuracies on both Operation Invariance and Value dependence at the same time, as discussed in Section 4.