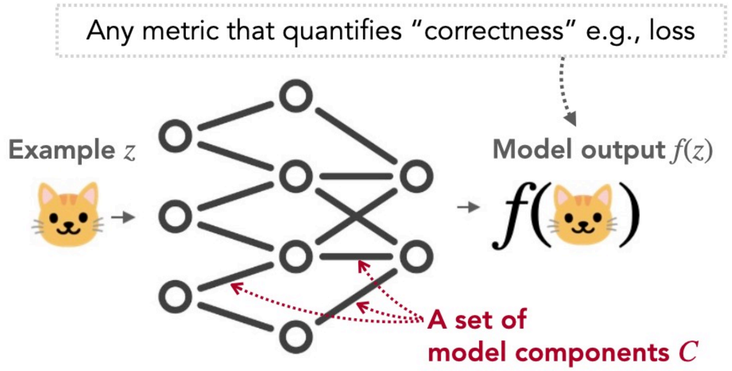
Decomposing and Editing Predictions by Modeling Model Computation

Harshay Shah, Andrew Ilyas, Aleksander Mądry

Models as computation graphs

Models ~ computation graphs over components



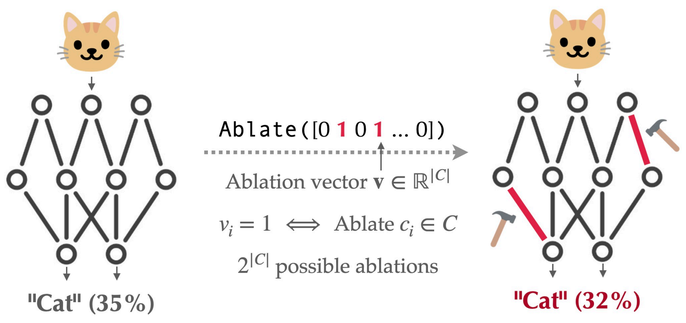
Some examples

- Convolution filters in CNNs
- Attention heads and MLPs in Transformers

Q: How do individual model components $c \in C$ *collectively* turn examples z into predictions $f(z)$?

Intervening via component ablations

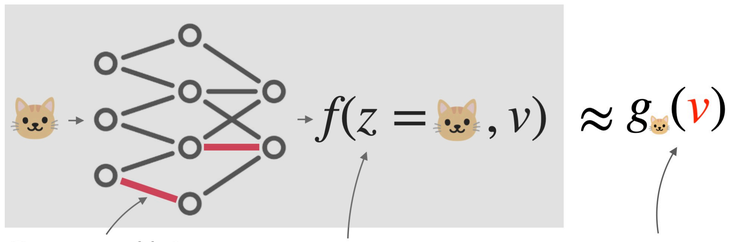
Component ablation intervenes on the *parameters* of a subset of model components $C \subseteq C$.



Ablation is user-specified (e.g., zero-ing out parameters)

Component attribution

Component attribution $g^{(z)}(\cdot)$ for example z maps ablation vector v to its effect on prediction $f(z)$



We focus on **linear** component attributions

$$f(z = \text{🐱}, v) \approx g_{\text{🐱}}(v) = \langle w^{(z)}, v \rangle + b^{(z)}$$

Component attribution g predicts ablated model output $f($ 🐱 $, v)$

Attribution scores $w^{(z)} \in \mathbb{R}^{|C|}$ encode component-wise "contributions"

Intuitively: $w_i^{(z)}$ ~ "importance" (*additive contribution*) of component $c_i \in C$ to the final prediction $f(z)$

Component Attrib via Regression (COAR)

Key idea: Component attribution via linear regression

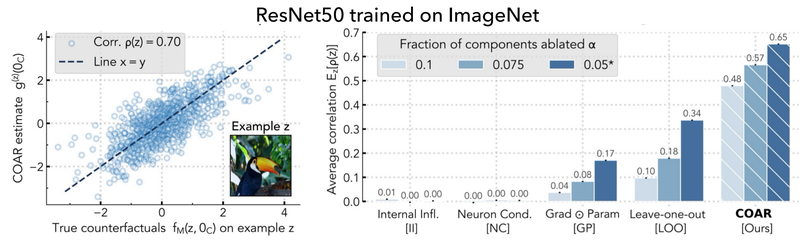
$$(w^{(z)}, b^{(z)}) = \arg\min_{w,b} \sum_{D^{(z)}} \left(f(z, v_i) - v_i^\top w - b \right)^2$$

Ground-truth output of ablated model

Dataset of component ablations

Attribution-based estimate

COAR is accurate on large-scale vision and language models



Model editing using COAR-Edit

Goal: Construct model intervention that:

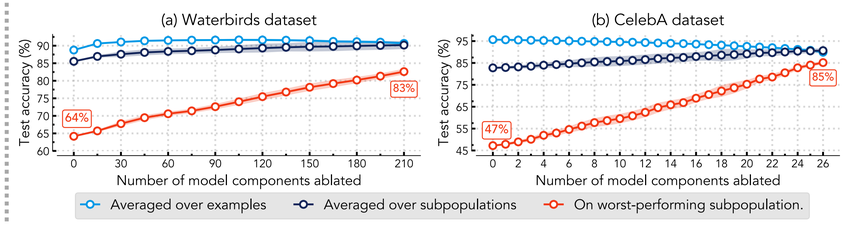
- Changes model behavior on target distribution $\mathcal{D}_T$
- Maintains behavior on reference distribution $\mathcal{D}_R$
- Uses only a few samples S from $\mathcal{D}_R \cup \mathcal{D}_T$

COAR-Edit: Ablate components $C' \subseteq C$ that are:

- **Important** (large $w_i^{(z)}$)) for target examples $z \sim \mathcal{D}_T$
- **Unimportant** (small $w_i^{(z)}$) for reference examples $z \sim \mathcal{D}_R$

Case study: Boosting subpopulation robustness

- **Model**: ResNet fine-tuned on Waterbirds/CelebA
- **Target** $\mathcal{D}_T$: Under-performing minority groups
- **Reference** $\mathcal{D}_R$: Majority groups



Result: Significant boost in worst-group accuracy by ablating a targeted subset of components (no retraining!)

Additional case studies (Section 5 in our paper)

- Improving CLIP robustness to typographic attacks
- Selectively "forgetting" subpopulations
- Localizing and removing a backdoor attack
- Correcting misclassified examples