Removal-based explanations

CSEP 590B: Explainable AI

Ian Covert & Su-In Lee
University of Washington

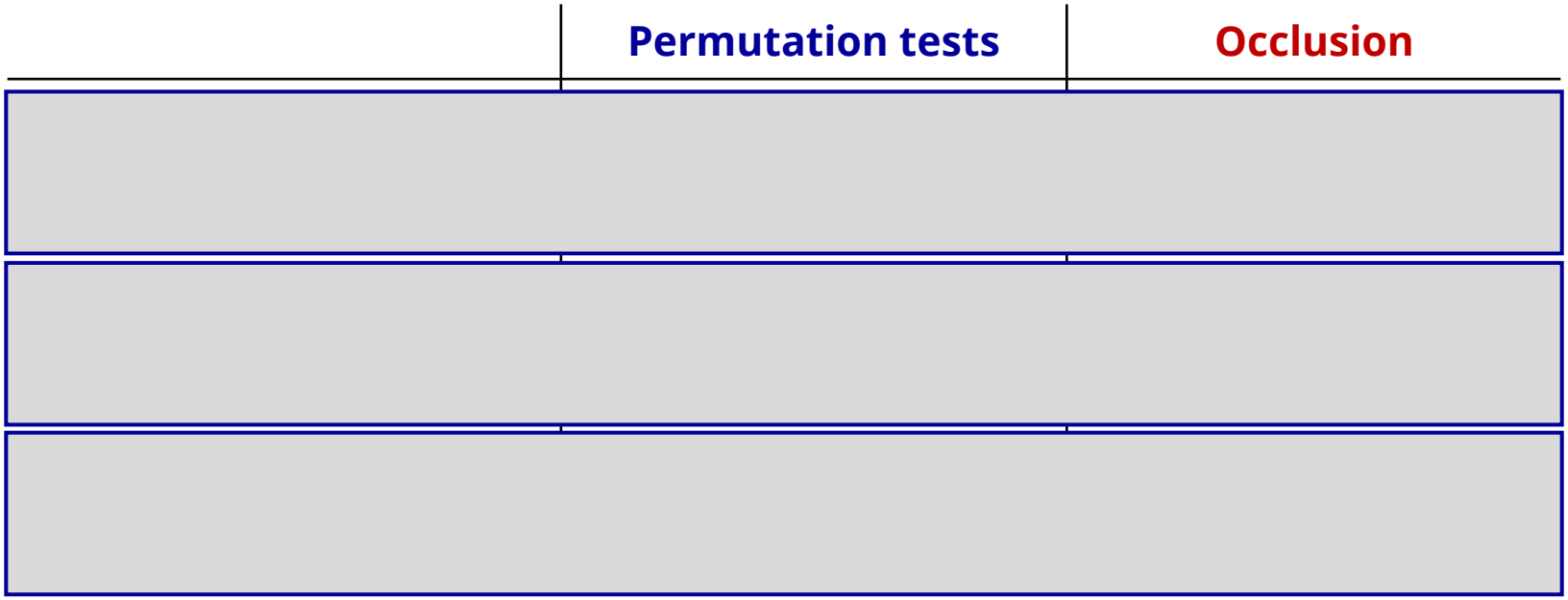
Today

- Section 1
  - Notation and definitions
  - Removal-based explanations
  - Example methods
- Section 2
  - Unified framework for removal-based methods
  - (Bonus) Meaningful perturbations

Perm. tests vs. occlusion

- Seemingly different methods
  - Designed for random forests vs. CNNs
  - Global vs. local explanations
- However, some notable parallels

Perm. tests vs. occlusion (cont.)



Perm. tests vs. occlusion (cont.)

	Permutation tests	Occlusion
Corrupt input	Randomize features	Set to zero
Observe model change	Change in accuracy	Change in prediction
Calculate impact	Remove single feat.	Remove single feat.

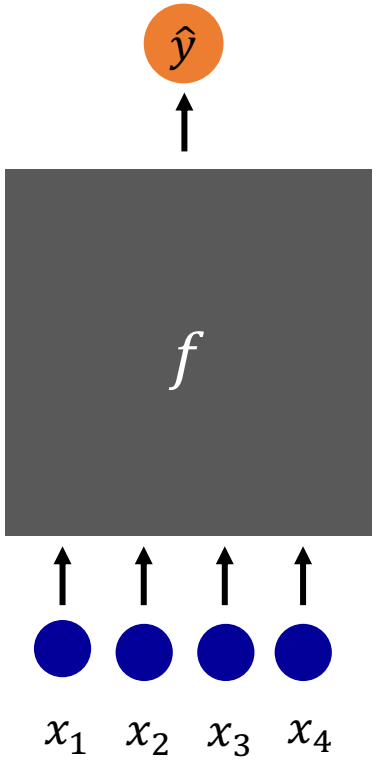
Framework

- **Idea:** create new explanation methods by changing these implementation choices

- Three design choices:
  1. **Feature removal approach**
  2. **Model behavior**
  3. **Summarization technique**

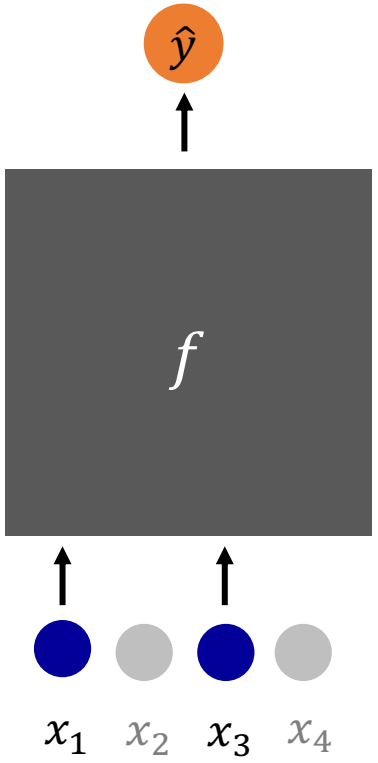
Choice 1: Feature removal

- Models must make predictions given all the features
- We want to remove information from certain features



Choice 1: Feature removal

- Models must make predictions given all the features
- We want to remove information from certain features
- Most models don't support this, but we can *simulate feature removal*



Feature removal (cont.)

- Several options include:
  - Replace with default values (zero)
  - Replace with random values (more on this next time)
  - Train separate models with each feature set
  - Use a model that accepts missing features
  - Blurring (for images)

Choice 2: Model behavior

- We can remove features and observe impact on the model
- But we must choose a specific quantity to focus on
- Can use any measurable model behavior

Prediction

$\hat{y}$

Prediction loss

$\ell(\hat{y}, y)$

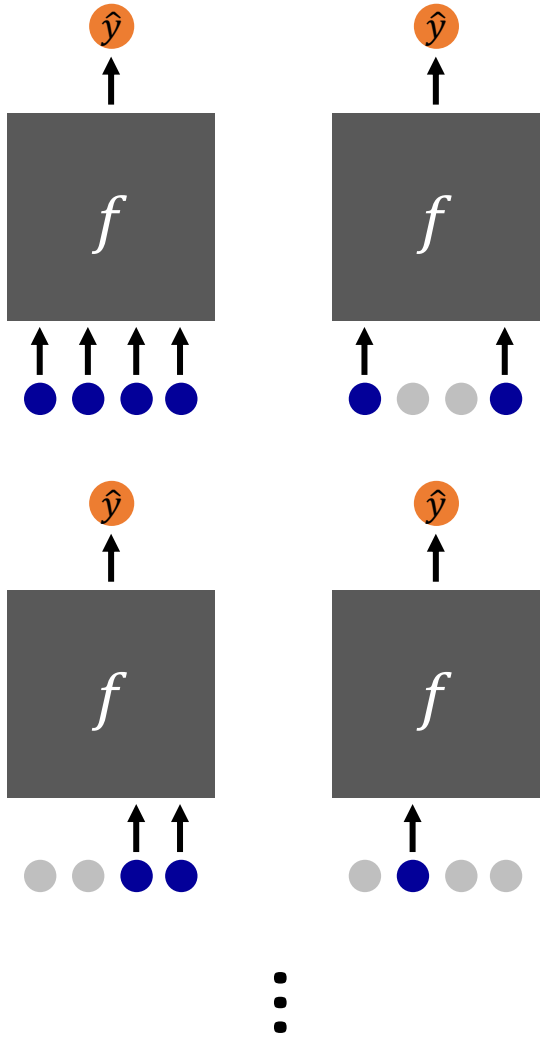
Dataset loss

$\mathbb{E}[\ell(\hat{y}, y)]$

Etc.

Choice 3: Summarization

- We can remove observe model behavior with any feature subset
- But given d features, there are 2^d subsets to consider
- Too much information – how to communicate to a human?



Summarization (cont.)

- Two common summary types:
  - **Feature selection** (subset of important features)
  - **Feature attribution** (assign feature scores)
    - E.g., permutation tests and occlusion

Framework recap

	Permutation tests	Occlusion
1. Feature removal	Sample new values	Set to zero
2. Model behavior	Dataset loss	Individual prediction
3. Summarization	Remove single feat.	Remove single feat.

A recipe for many methods

- Framework introduced in a recent paper
  - At least 26 published papers follow this recipe
  - Example methods include SHAP, LIME, etc.
  - Suggested the term **removal-based explanations**

Covert et al., "Explaining by removing: a unified framework for model explanation" (2021)

A recipe for many methods

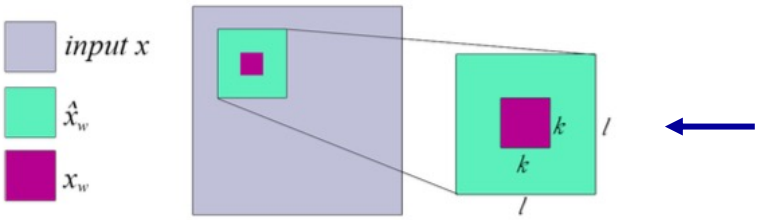
Method	Removal	Behavior	Summary
IME (2009)	Separate models	Prediction	Shapley value
IME (2010)	Marginalize (uniform)	Prediction	Shapley value
QII	Marginalize (marginals product)	Prediction	Shapley value
SHAP	Marginalize (conditional/marginal)	Prediction	Shapley value
KernelSHAP	Marginalize (marginal)	Prediction	Shapley value
TreeSHAP	Tree distribution	Prediction	Shapley value
LossSHAP	Marginalize (conditional)	Prediction loss	Shapley value
SAGE	Marginalize (conditional)	Dataset loss (label)	Shapley value
Shapley Net Effects	Separate models (linear)	Dataset loss (label)	Shapley value
SPVIM	Separate models	Dataset loss (label)	Shapley value
Shapley Effects	Marginalize (conditional)	Dataset loss (output)	Shapley value
Permutation Test	Marginalize (marginal)	Dataset loss (label)	Remove individual
Conditional Perm. Test	Marginalize (conditional)	Dataset loss (label)	Remove individual
Feature Ablation (LOCO)	Separate models	Dataset loss (label)	Remove individual
Univariate Predictors	Separate models	Dataset loss (label)	Include individual
L2X	Surrogate	Prediction loss (output)	High-value subset
REAL-X	Surrogate	Prediction loss (output)	High-value subset
INVASE	Missingness during training	Prediction mean loss	High-value subset
LIME (Images)	Default values	Prediction	Linear model
LIME (Tabular)	Marginalize (replacement dist.)	Prediction	Linear model
PredDiff	Marginalize (conditional)	Prediction	Remove individual
Occlusion	Zeros	Prediction	Remove individual
CXPlain	Zeros	Prediction loss	Remove individual
RISE	Zeros	Prediction	Mean when included
MM	Default values	Prediction	Partitioned subsets
MIR	Extend pixel values	Prediction	High-value subset
MP	Blurring	Prediction	Low-value subset
EP	Blurring	Prediction	High-value subset
FIDO-CA	Generative model	Prediction	High-value subset

Quick comparisons

- Alternative **feature removal** choices
  - PredDiff
  - Meaningful Perturbations
- Alternative **summarization** choices
  - RISE
  - LIME

PredDiff

- Removes information using a conditional in-painting model



Remove small regions, in-paint using neighboring pixels

Zintgraf et al., "Visualizing deep neural network decisions: Prediction difference analysis" (2017)

Meaningful perturbations

- Removes information via blurring

Fong & Vedaldi, “Interpretable explanations of black boxes by meaningful perturbation” (2017)

RISE

- Samples many subsets of missing features
- Calculates mean prediction when x_i included

$$a_i = \mathbb{E}\left[F_y(x_S) \mid i \in S \right]$$

Prediction given features x_S only

Petsiuk et al., "RISE: Randomized input sampling for explanation of black-box models" (2018)

LIME

- Has a weighting kernel $\pi(S)$ on feature subsets
- Fits linear/additive proxy model

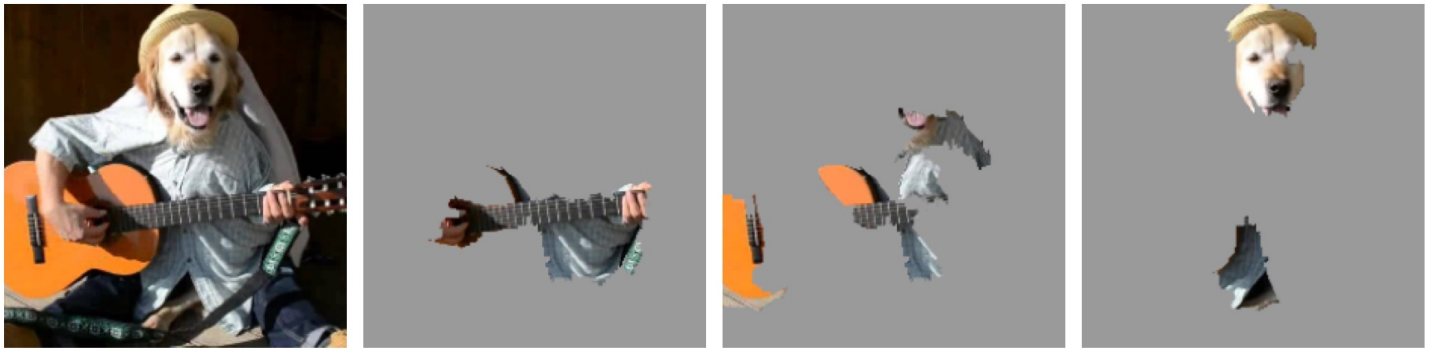
$$\min_{a_0,\ldots,a_d} \sum_{S \subseteq D} \pi(S) \left(a_0 + \sum_{i \in S} a_i - F_y(x_S) \right)^2 + \Omega(a_1, \ldots, a_d)$$

Additive approximation

Optional regularization (e.g., lasso)

Ribeiro et al., "Why should I trust you? Explaining the predictions of any classifier" (2016)

LIME (cont.)



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Figure 4: Explaining an image classification prediction made by Google's Inception neural network. The top 3 classes predicted are "Electric Guitar" ($p = 0.32$), "Acoustic guitar" ($p = 0.24$) and "Labrador" ($p = 0.21$)

Many more removal-based explanations

- A framework that helps understand many methods
- All based on simulating feature removal, and each method is specified by three choices

Next time: SHAP

- A popular method, and also a removal-based explanation
- Requires additional background on cooperative game theory

Today

- Section 1
  - Notation and definitions
  - Removal-based explanations
  - Example methods
- Section 2
  - Unified framework for removal-based methods
  - (Bonus) Meaningful perturbations

Meaningful perturbations

- **Idea:**
  - Take an image that's correctly classified by the model
  - Blur the image to alter the prediction

Fong & Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation" (2017)

Notation

- Let $x \in \mathbb{R}^{w \times h}$ be the image
- Let $m \in [0,1]^{w \times h}$ be a mask
- Let $\Phi(x, m) \in \mathbb{R}^{w \times h}$ be the masked image

Q: how to perform masking?

Masking options

- Replace with constant value μ

$$\Phi(x,m)_{ij} = m_{ij} \cdot x_{ij} + \left(1 - m_{ij}\right) \cdot \mu$$

- Replace with noise $\epsilon \sim N(0,\sigma^2)$

$$\Phi(x,m)_{ij} = m_{ij} \cdot x_{ij} + \left(1 - m_{ij}\right) \cdot \epsilon_{ij}$$

- Blur with Gaussian kernel

$$\Phi(x,m)_{ij} = \textbf{[blur with kernel } g_{\sigma \cdot m_{ij}} \textbf{]}$$

Learn optimal blur

- Initially, target class y has $f_y\big(\Phi(x, \mathbf{1})\big) \approx 1$
- **Goal:** learn m such that $f_y\big(\Phi(x, m)\big) \approx 0$

- Minimize the following loss:

$$\min_m f_y\big(\Phi(x, m)\big)$$

Other considerations

1. Blur should be minimal
2. Mask should be smooth
3. Optimization should be robust against adversarial perturbations

Actual loss function:

$$\min_m \mathbb{E}_\tau\left[f_y\left(\Phi(x(\cdot-\tau), m)\right)\right] + \lambda_1 \|1-m\|_1 + \lambda_2 \|\nabla m\|_\beta^\beta$$

Spatially jittered input — Minimal mask — Smooth mask

Optimization

- **Actual loss function:**

$$\mathcal{L}(m) = \mathbb{E}_{\tau}\left[f_y\left(\Phi(x(\cdot - \tau), m)\right)\right] + \lambda_1 \|1 - m\|_1 + \lambda_2 \|\nabla m\|_{\beta}^{\beta}$$

- Determine optimal mask using SGD:

$$m^{(+)} = m - \alpha \cdot \frac{\partial \mathcal{L}}{\partial m}(m)$$

Results

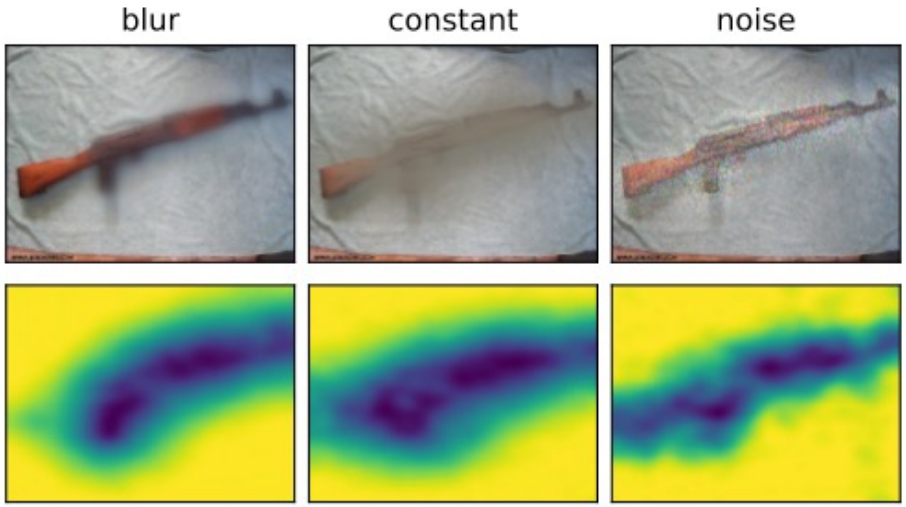
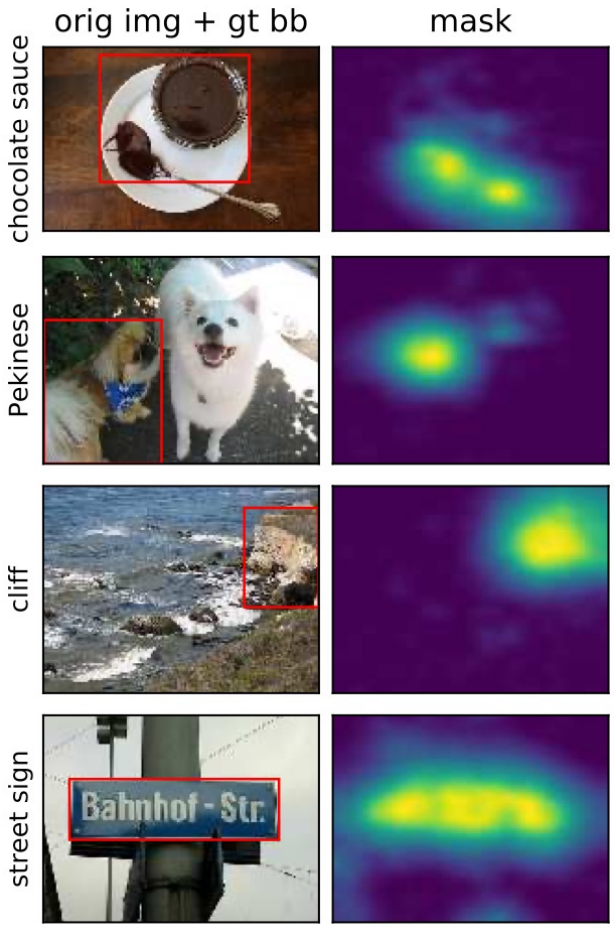


Figure 4. Perturbation types. Bottom: perturbation mask; top: effect of blur, constant, and noise perturbations.

Results (cont.)



Placing it in the framework

- We can concisely describe MP by how it fits into the framework

- Its three choices are:
  - **(Removal)** Blurring
  - **(Model behavior)** Prediction probability
  - **(Summarization)** Remove small feature subset to change prediction