# Shapley Value for Model Interpretability

**Yufang Liu**

*East China Normal University*

# Background

- Shapley Value
  - originally defined for cooperative game
  - an equitable way of sharing the group reward
  - evaluate feature importance √
- More ?
  - data importance
  - source task importance in transfer learning
  - neuron importance

# Evaluate Data Importance

**Data Shapley: Equitable Valuation of Data for Machine Learning**

**Amirata Ghorbani** [1] **James Zou** [2]

[1]Department of Electrical Engineering, Stanford University

[2]Department of Biomedical Data Science, Stanford University

*ICML 2019*

# Motivation

- Desired Characteristics
  - equitable measure of the value of each data point
  - compute efficiently
- Notation
  - $D = \{(x_i, y_i)\}_1^n$ , $S \subseteq D$
  - $\mathcal{A}$ denote the learning algorithm
  - $V$ performance score -> $V(S, \mathcal{A})$
  - $\phi_i(D, \mathcal{A}, V) \in \mathbb{R}$ shapley value

# Motivation

- Equitable properties of data valuation
- Zero Contribution

$$\text{if } \forall S \subseteq N \setminus \{i\} : V(S \cup \{i\}) = V(S) \Rightarrow \phi_i = 0$$

- Symmetric Elements

$$\text{if } \forall S \subseteq N \setminus \{i, j\} : V(S \cup i) = V(S \cup j) \Rightarrow \phi_i = \phi_j$$

- Additivity in Performance Metric

$$V = V_1 + V_2 \Rightarrow \phi_i(V, N) = \phi_i(V_1, N) + \phi_i(V_2, N)$$

# Methods

$$\phi_i = C \sum_{S \subseteq D - \{i\}} \frac{V(S \cup \{i\}) - V(S)}{\binom{n-1}{|S|}}$$ $C$ *is an arbitrary constant.*

$S_\pi^i$ is the set of data points coming before datum i in permutation $\pi$

- Approximating Shapley Value
  - Monte-Carlo method (unbiased estimation, usually 3n sample)

$$C = 1/n \quad \Rightarrow \quad \phi_i = \mathbb{E}_{\pi \sim \Pi}[V(\boxed{S_\pi^i} \cup \{i\}) - V(S_\pi^i)]$$

    - marginal contribution to every subset and normalize
  - Truncation (set threshold)
    - adding only one more training point becomes smaller and small

# Methods

$$\frac{1}{n}\sum_{i=1}^{n}\frac{|\phi_i^t-\phi_i^{t-100}|}{|\phi_i^t|} < 0.05$$

randomly initialized classifier

**Algorithm 1** **Truncated Monte Carlo Shapley**

**Input:** Train data $D = \{1, \ldots, n\}$, learning algorithm $\mathcal{A}$, performance score $V$

**Output:** Shapley value of training points: $\phi_1, \ldots, \phi_n$

Initialize $\phi_i = 0$ for $i = 1, \ldots, n$ and $t = 0$

**while** Convergence criteria not met **do**

  $t \leftarrow t + 1$

  $\pi^t$: Random permutation of train data points

  $\boxed{v_0^t \leftarrow V(\emptyset, \mathcal{A})}$

  **for** $j \in \{1, \ldots, n\}$ **do**

    **if** $|V(D) - v_{j-1}^t| <$ Performance Tolerance **then**

      $v_j^t = v_{j-1}^t$

    **else**

      $v_j^t \leftarrow V(\{\pi^t[1], \ldots, \pi^t[j]\}, \mathcal{A})$

    **end if**

    $\phi_{\pi^t[j]} \leftarrow \frac{t-1}{t}\phi_{\pi^{t-1}[j]} + \frac{1}{t}(v_j^t - v_{j-1}^t)$

  **end for**

**end for**

calculate marginal contributions and average

# Methods

- Approximating Performance Metric $V$
  - calculating $V(S)$ requires $\mathcal{A}$ to learn a new model
  - Gradient Shapley: train the model with only one pass through the training data

# Methods

**Algorithm 2 Gradient Shapley**

**Input:** Parametrized and differentiable loss function $\mathscr{L}(.;\theta)$, train data $D = \{1, \ldots, n\}$ , performance score function $V(\theta)$

**Output:** Shapley value of training points: $\phi_1, \ldots, \phi_n$

Initialize $\phi_i = 0$ for $i = 1, \ldots, n$ and $t = 0$

**while** Convergence criteria not met **do**

$t \leftarrow t + 1$

$\pi^t$: Random permutation of train data points

$\boxed{\theta_0^t \leftarrow \text{Random parameters}}$

$v_0^t \leftarrow V(\theta_0^t)$

**for** $j \in \{1, \ldots, n\}$ **do**

$\boxed{\theta_j^t \leftarrow \theta_{j-1}^t - \alpha \nabla_\theta \mathscr{L}(\pi^t[j]; \theta_{j-1})}$

$v_j^t \leftarrow V(\theta_j^t)$

$\phi_{\pi^t[j]} \leftarrow \frac{t-1}{t} \phi_{\pi^{t-1}[j]} + \frac{1}{t}(v_j^t - v_{j-1}^t)$

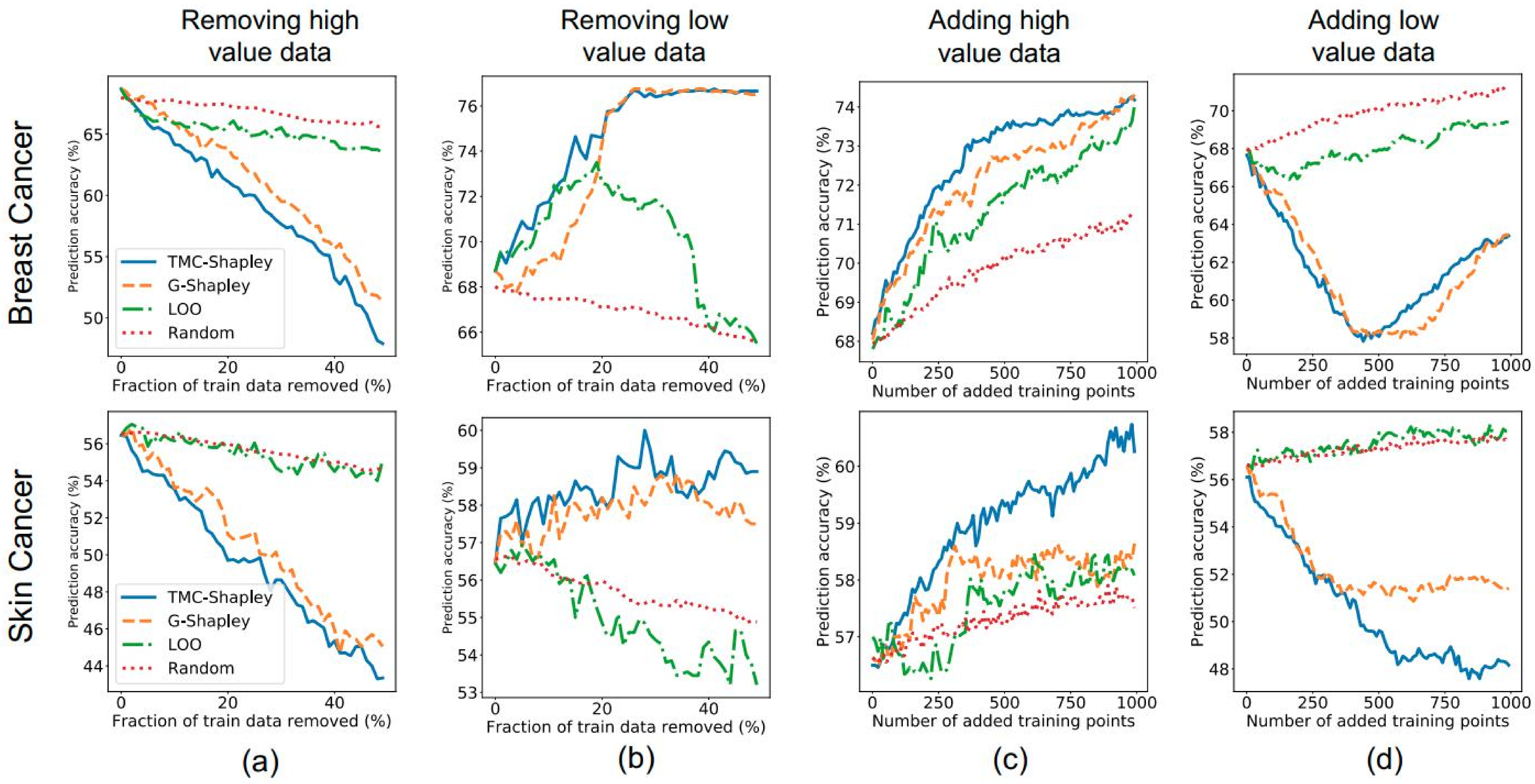
**end for**

**end for**

# Experiment

- Data Shapley for Disease Prediction
  - UK Biobank data set (1k training set)
  - Malignant neoplasm of breast and skin , binary classification
  - using logistic regression with 285 features
- Compare with Leave-one-out (LOO)
  - do not consider the combinations of the sources
  - eg. two points are helpful when they are both present or absent, otherwise harmful
- Acquiring new data with calculated data value
  - learn a Random Forest regression model to predict data value
  - from 2000 candidates

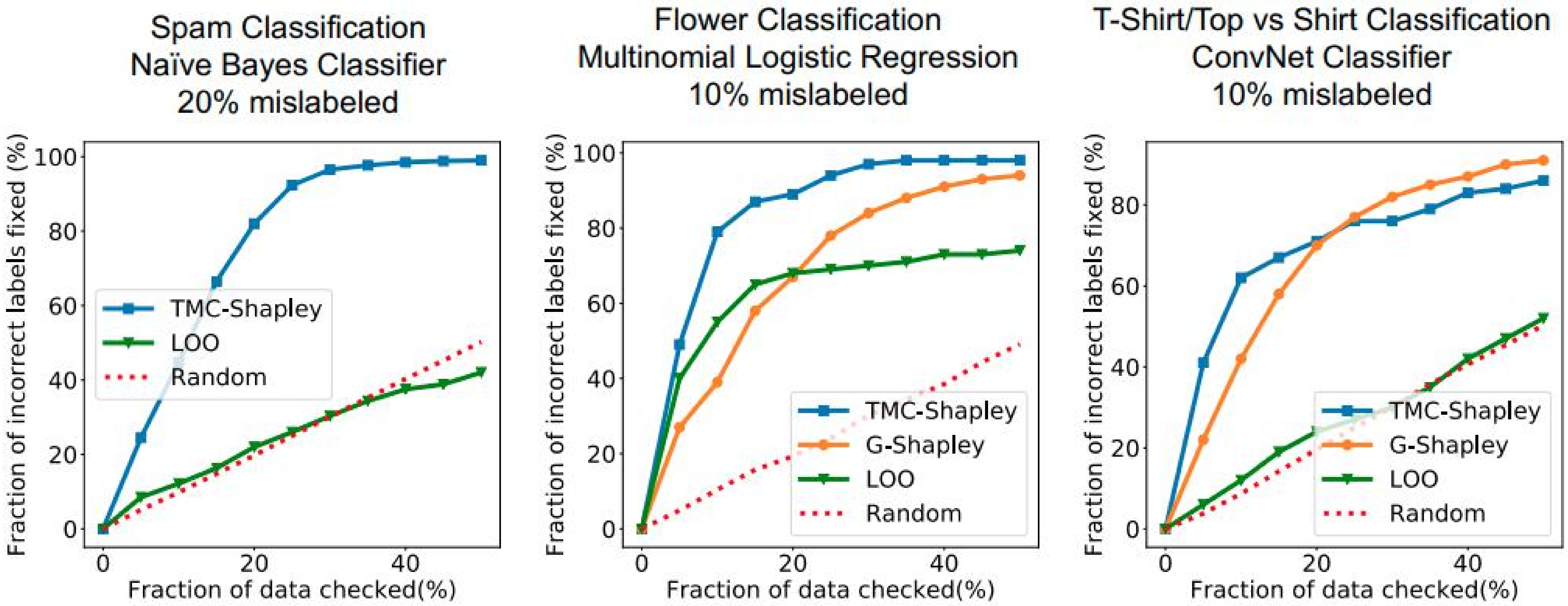
# Experiment



Removing high value data
Removing low value data
Adding high value data
Adding low value data
Breast Cancer
Skin Cancer
Prediction accuracy (%)
Fraction of train data removed (%)
Number of added training points
TMC-Shapley
G-Shapley
LOO
Random
(a)
(b)
(c)
(d)

# Experiment

- Label Noise Detection



# Evaluate Source Task Importance

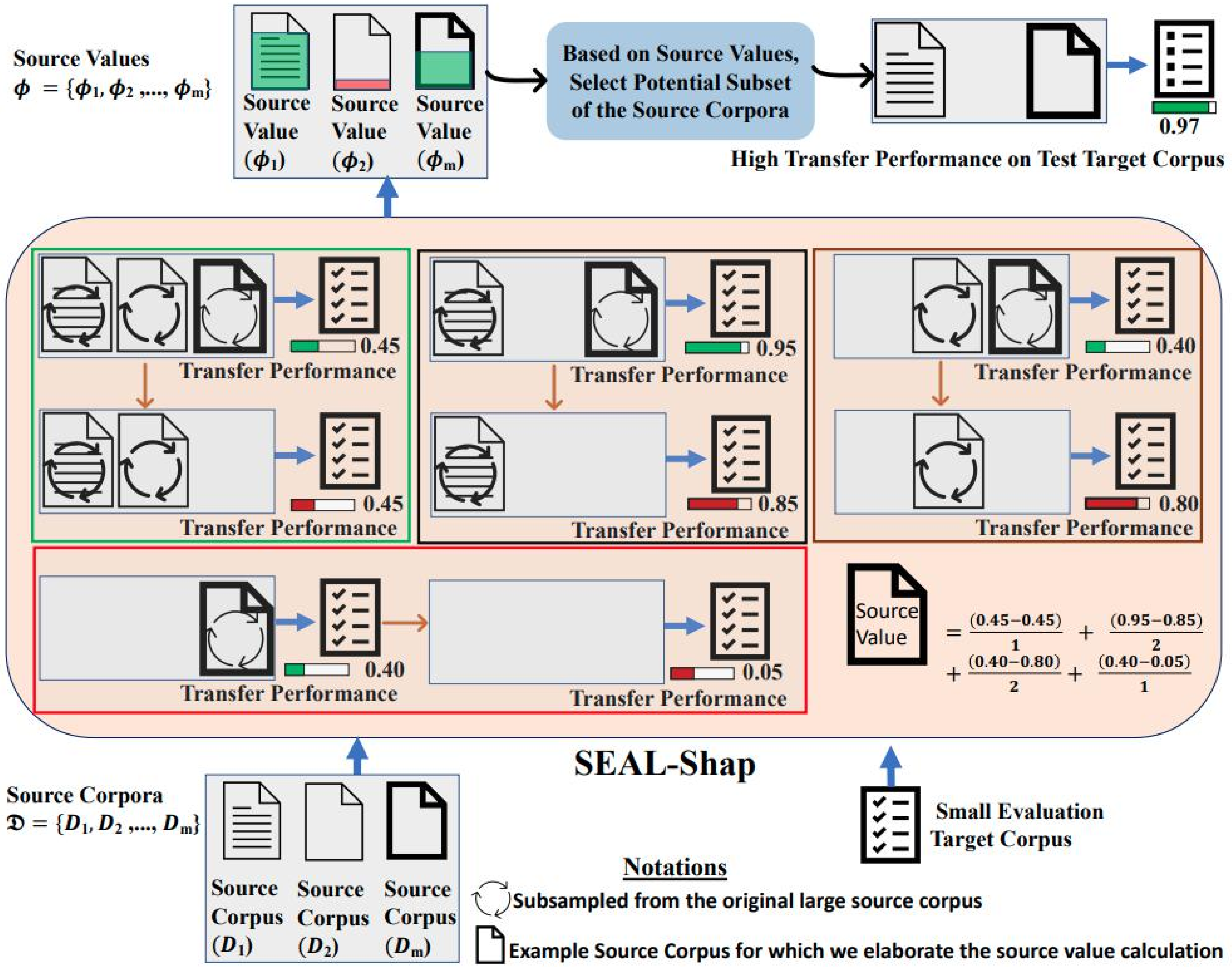
**Evaluating the Values of Sources in Transfer Learning**

**Md Rizwan Parvez**
University of California Los Angeles
rizwan@cs.ucla.edu

**Kai-Wei Chang**
University of California Los Angeles
kwchang@cs.ucla.edu

*NAACL 2021*

# Methods



Source Values, $\phi = \{\phi_1, \phi_2, ..., \phi_m\}$
Source Value ($\phi_1$)
Source Value ($\phi_2$)
Source Value ($\phi_m$)
Based on Source Values, Select Potential Subset of the Source Corpora
0.97
High Transfer Performance on Test Target Corpus
0.45
Transfer Performance
0.45
Transfer Performance
0.95
Transfer Performance
0.85
Transfer Performance
0.40
Transfer Performance
0.80
Transfer Performance
0.40
Transfer Performance
0.05
Transfer Performance
Source Value
$= \frac{(0.45-0.45)}{1} + \frac{(0.95-0.85)}{2} + \frac{(0.40-0.80)}{2} + \frac{(0.40-0.05)}{1}$
SEAL-Shap
Source Corpora $\mathfrak{D} = \{D_1, D_2, ..., D_m\}$
Source Corpus ($D_1$)
Source Corpus ($D_2$)
Source Corpus ($D_m$)
Small Evaluation Target Corpus
Notations
Subsampled from the original large source corpus
Example Source Corpus for which we elaborate the source value calculation

# Methods

- zero-shot cross-lingual and cross-domain
- POS tagging, sentiment analysis, and NLI
- using BERT and BiLSTM
- two settings
  - a small target corpus is available
  - only the linguistic or statistical features of the target
    - train a source ranker based on SEAL-Shap and the available features
    - consider $D_j$ as target and the rest as sources

# Methods

- Stratified Sampling
  - sampling training instances from each source corpus
- Truncation for early steps too
  - restrict the variance of the marginal contributions
- Caching
  - improve the computation time by about 2x
  - cache calculated results

# Methods

**Algorithm 1:** **SEAL-Shap**

**Input:** Source corpora $\mathcal{D} = \{D_1, \cdots, D_m\}$, target corpus $V$, Random sampler $\mathcal{S}$, sample size $\eta$, num of epochs $nepoch$, and Classifier $C$

**Output:** Source-corpora Shapley values $\{\Phi_1 ..., \Phi_m\}$

**Initialize:** Score cache $S \leftarrow \{\}$, source Shapley values $\Phi_x \leftarrow 0$ for $x = 1 \dots m$, and epoch $t \leftarrow 0$

$\mathcal{D}_{samp} \leftarrow \{\mathcal{S}(D_x, \eta), \forall D_x \in \mathcal{D}\}$

$C_{\mathcal{D}_{samp}} \leftarrow$ Train $C$ on $\mathcal{D}_{samp}$

**while** *Converge or* $t < nepoch$ **do**

    $t \leftarrow t + 1$

    $\pi$ : Random permutation of $\mathcal{D}$

    $v_0 \leftarrow \rho$

    **for** $j \in \{1, \cdots m\}$ **do**

        $\Omega \leftarrow \{\pi_1, \cdots, \pi_j\}$

        **if** $|$ *Score*$(C_{\mathcal{D}_{samp}}, V)$ - $v_{j-1}| <$ *Tolerance* **then**

            $v_j \leftarrow v_{j-1}$

        **else**

            **if** $\Omega \notin S$ **then**

                $\mathcal{T} \leftarrow \{\mathcal{S}(\Omega_x, \eta), \forall \Omega_x \in \Omega\}$

                $C_j \leftarrow$ Train $C$ on $\mathcal{T}$

                Insert $\Omega$ into $S$ with $S_\Omega \leftarrow$ Score$(C_j, V)$

            $v_j \leftarrow S_\Omega$

        $\Phi_{\pi_j} \leftarrow \frac{t-1}{t} \Phi_{\pi_j} + \frac{1}{t}(v_j - v_{j-1})$

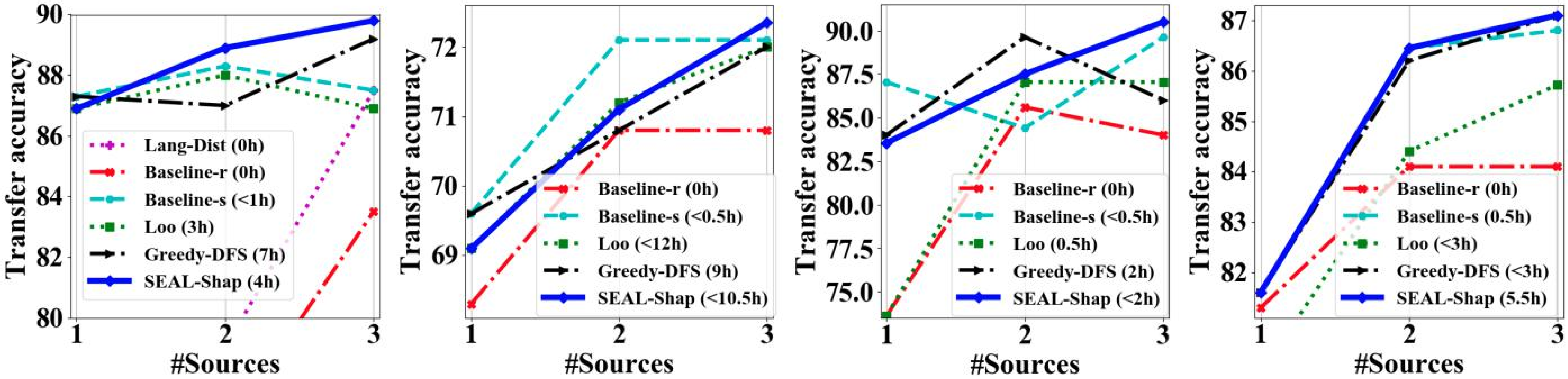
# Experiment

- Cross-lingual Datasets -> multi-lingual BERT
  - universal POS tagging on UD, 31 languages of 13 different language families
  - NLI from XNLI dataset, 15 different languages
- Cross-domain Datasets -> Bert
  - POS tagging, SANCL 2012 shared task datasets, 6 domains -> BiLSTM
  - Sentiment analysis, multi-domain sentiment datasets, 14 domains
  - NLI, modified binary classification dataset, 4 domains

# Experiment

- Evaluating Source Valuation
  - Baseline-s: source values are single source transfer performance
  - LOO
  - Baseline-r: random value
  - Greedy DFS: greedily select sources
  - Lang-Dist: reverse order of target-source language distance

# Source Corpora Selection




(a) UD Treebank, target: en (b) XNLI, target: vi (c) mtl-dom-senti, target: E (d) mGLUE, target:MNLI-mm

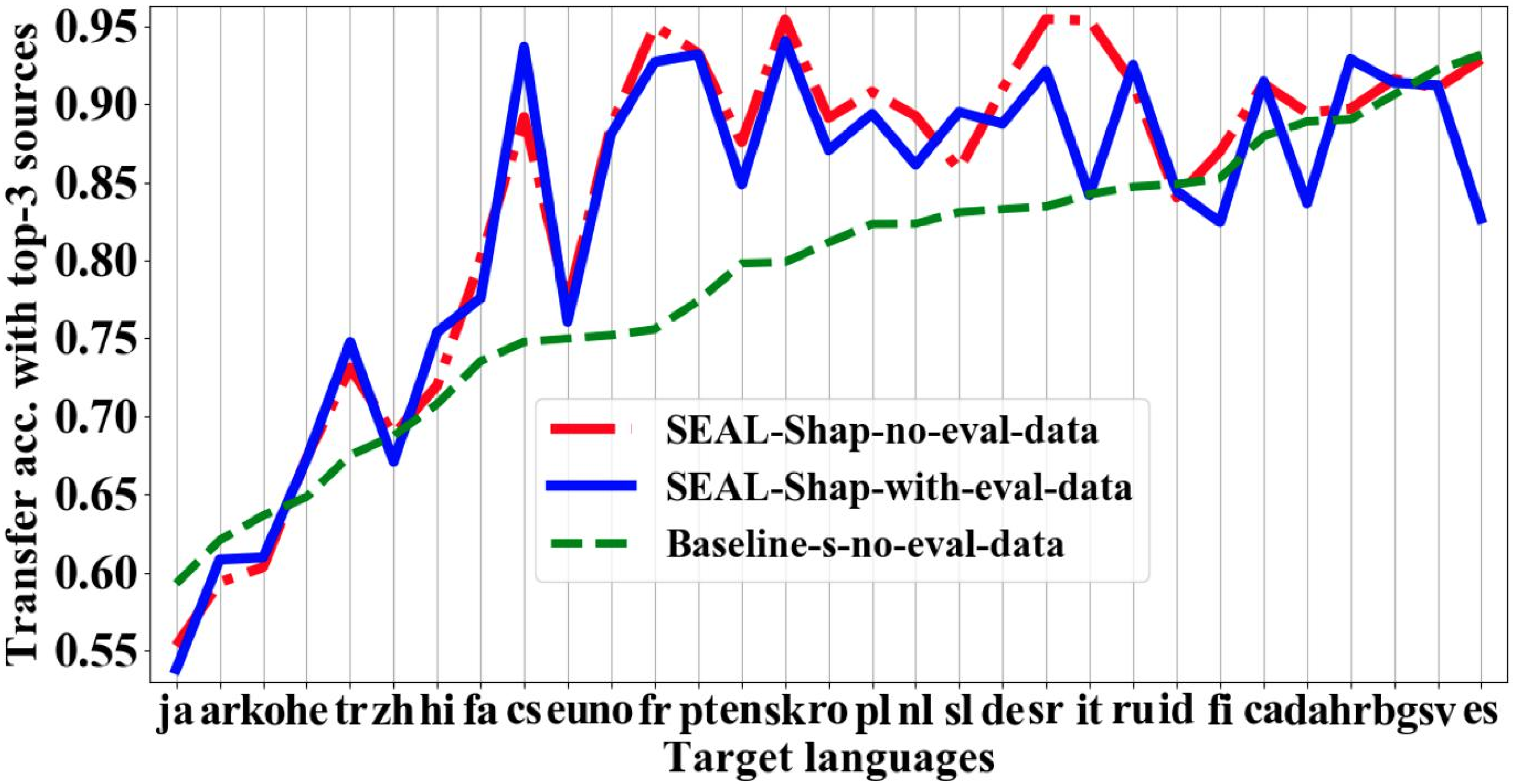
Figure 2: Performance, and run time with up to top-3 sources ranked by different approaches. (a), (b) denotes cross-lingual and (c), (d) denotes cross-domain transfer. All models have same training configurations (e.g., sample size). All the run times are final except for *Greedy DFS* where it increases linearly with top-$k$. Adding top-2 and top-3 ranked sources, other methods drop their accuracy across the tasks while ours shows a consistent gain in all tasks and achieves the best results with top-3 sources.

# Cross-Lingual POS Tagging

- “en” refers to the only source (“en”)
- ’*’, ‘$’, ‘†’ denote SEALShap model is statistically significantly outperforms AllSources, Baseline-r and Baseline-s

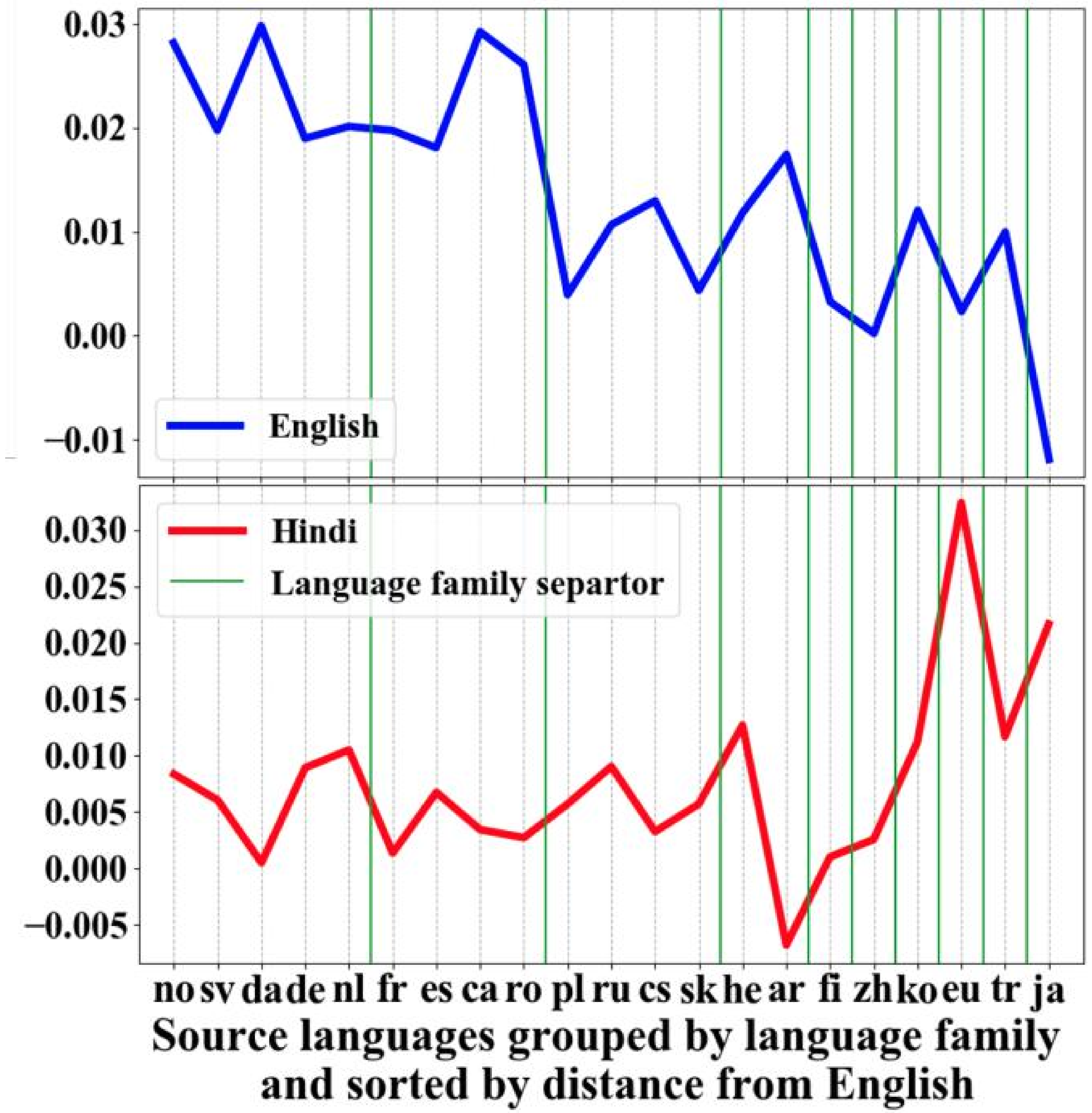
| Lang | en | All Source | Baseline-r | Baseline-s | SEAL-Shap |
|---|---|---|---|---|---|
| en | - | 82.71 | 86.32 | 86.39 | **88.55**$^{*\$\dagger}$ |
| fr | - | 94.60 | 94.63 | **94.83** | 94.79 |
| da | 88.3 | 88.94 | 89.30 | 89.23 | **89.47**$^{*}$ |
| es | 85.2 | 93.15 | 93.00 | 93.04 | **93.21**$^{\$}$ |
| it | 84.7 | 96.58 | 96.43 | **96.71** | 96.67 |
| ca | - | 91.54 | 91.64 | 90.78 | **92.08**$^{*\$\dagger}$ |
| sl | 84.2 | 93.28 | 93.50 | 92.89 | **93.52**$^{*\dagger}$ |
| nl | 75.9 | 90.10 | 90.19 | 90.14 | **90.26** |
| ru | - | 92.98 | 92.91 | 92.71 | **93.13**$^{*\$\dagger}$ |
| de | 89.8 | 90.79 | 91.07 | **91.44** | 91.06 |
| he | - | 76.67 | 75.75 | 75.43 | **76.73**$^{\$\dagger}$ |
| cs | - | 93.89 | 93.04 | 93.94 | **94.81**$^{*\$\dagger}$ |
| sk | 83.6 | 95.68 | 95.62 | 95.53 | **95.81**$^{\dagger}$ |
| sr | - | 97.55 | 97.47 | 97.43 | **97.58**$^{\dagger}$ |
| id | - | 84.10 | 85.23 | 85.50 | **85.97**$^{*\$}$ |
| fi | - | **87.13** | 86.89 | 86.86 | 87.05 |
| ko | - | 63.59 | **64.27** | 63.77 | 64.19 |
| hi | - | 81.49 | 80.27 | 79.94 | **82.41**$^{*\$\dagger}$ |
| ja | - | 66.86 | 65.99 | 67.71 | **67.81**$^{*\$}$ |
| fa | 72.8 | 81.03 | 80.69 | **82.37** | 81.79 |
| Average | - | 82.98 | 83.05 | 83.15 | **83.66** |

# without an Evaluation Corpus



# Interpret Source Value by SEAL-Shap

- The value gradually decreases when the word order distance increase
- As for the target language Hindi, the trend is opposite



# SEAL-Shap on Similar Targets

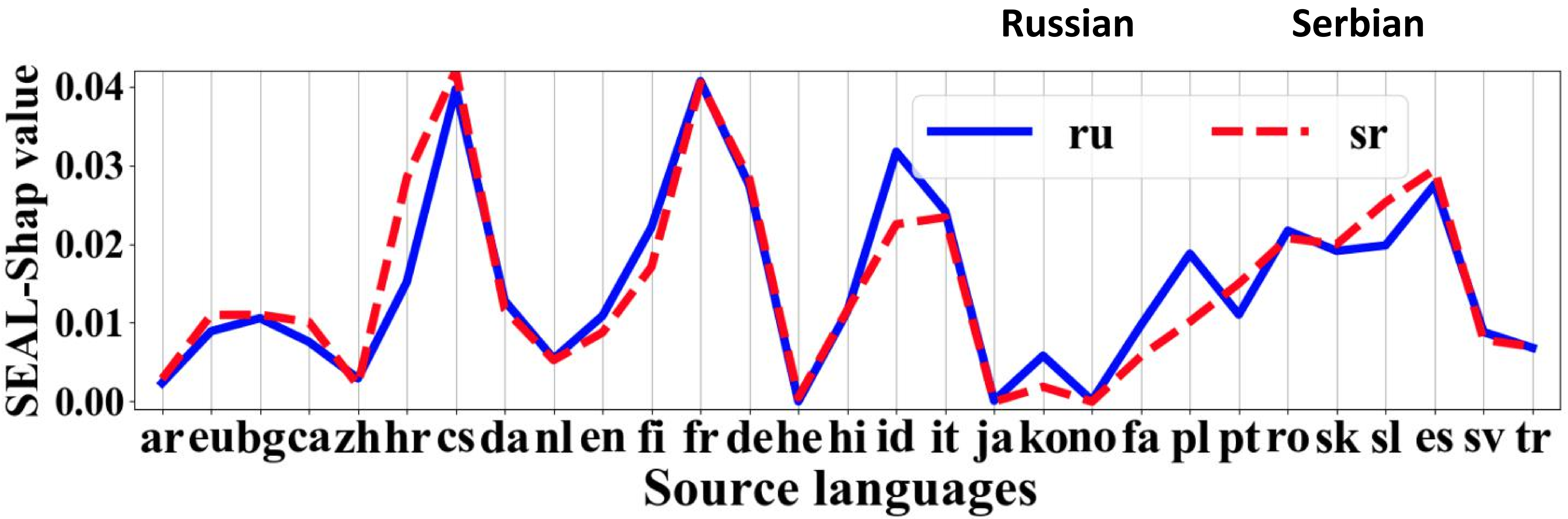



Figure 6: Similar SEAL-Shap value curves for two closely related target languages in cross-lingual POS tagging.

# Evaluate Neuron Importance

## Neuron Shapley: Discovering the Responsible Neurons

**Amirata Ghorbani**
Department of Electrical Engineering
Stanford University
Stanford, CA 94025
amiratag@stanford.edu

**James Zou***
Department of Biomedical Data Science
Stanford University
Stanford, CA 94025
jamesz@stanford.edu

*NeurIPS 2020*

# Background

- Interpret and visualize specific neurons
  - showing which  training data leads to the most positive or negative activation of this neuron
  - activation maximization
- Drawbacks
  - not clear which ones to investigate
  - a neruon's relevance to the overall function of network is unknown
  - its interactions with all other neurons is not considered

# Methods

- Adaptive Sampling
  - only a sparse number of influential neurons
  - finding the subset of bounded random variables with the largest expected value
  - formulated as a multi-armed-bandit (MAB) problem
- Intuition
  - keep tracking a lower and upper confidence bound (CB) on $\phi_i$
  - only sample k'th largest neurons between their bounds
- By "zero out" elements
  - In convnet, fixing the output of a filter(neuron) as mean output for dev set
  - kill the information flow while keeping the mean statistics

# Methods

**Algorithm 1 Truncated Multi Armed Bandit Shapley**

**Input:** Network's elements $N = \{1, \ldots, n\}$; performance metric $V(.)$; failure probability $\delta$, tolerance $\epsilon$, number of important elements $k$, Early truncation performance $v_T$
**Output:** Shapley value of elements: $\{\phi_i\}_{i=1}^n$
**Initializations:** $\{\phi_i\}_{i=1}^n = 0$, $\{\sigma_i\}_{i=1}^n = 0$, $\mathscr{U} = N$, $t = 0$
**while** $\mathscr{U} \neq \emptyset$ **do**
  $t \leftarrow t + 1$
  Random permutation of network's elements: $\pi^t = \{\pi^t[1], \ldots, \pi^t[n]\}$
  $v_0^t \leftarrow V(N)$
  **for** $j \in \{1, \ldots, N\}$ **do**
    **if** $j \in \mathscr{U}$ **then**
      **if** $v_{j-1}^t < v_T$ **then**
        $v_j^t \leftarrow v_{j-1}^t$
      **else**
        $v_j^t \leftarrow v(\{\pi^t[j+1], \ldots, \pi^t[n]\})$
      $\phi_{\pi^t[j]}, \sigma_{\pi^t[j]} \leftarrow$ Moving Average$(v_{j-1}^t - v_j^t, \phi_{\pi^t[j]})$, Moving Variance$(v_{j-1}^t - v_j^t, \phi_{\pi^t[j]})$
      $\phi_{\pi^t[j]}^{ub}, \phi_{\pi^t[j]}^{lb} \leftarrow$ Confidence Bounds$(\phi_{\pi^t[j]}, \sigma_{\pi^t[j]}, t)$
  $\mathscr{U} \leftarrow \{i : \phi_i^{lb} + \epsilon < k\text{'th largest } \{\phi_i\}_i = 1^n < \phi_i^{ub} - \epsilon\}$

# Methods

$$16: \quad \mathscr{U} \leftarrow \{i : \phi_i^{lb} + \epsilon < k\text{'th largest } \{\phi_i\}_i = 1^n < \phi_i^{ub} - \epsilon\}$$

- Empirical Bernstein error bound

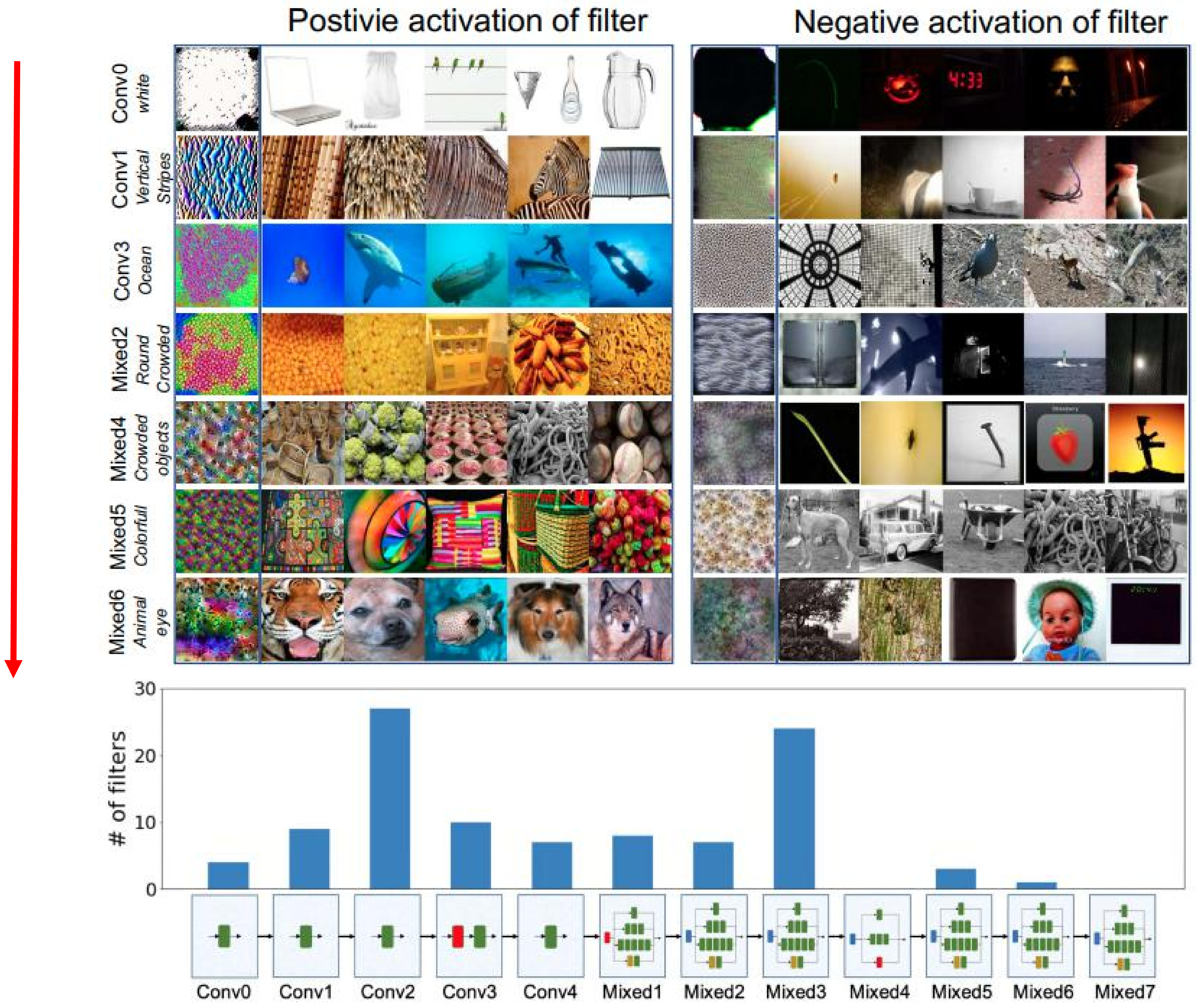
$$|\phi_i - \text{Empirical AVG}(\phi_i)| \leq \sqrt{\frac{2ln(2/\delta)\text{Empirical VAR}(\phi_i)}{t_i}} + \frac{7R}{3}\frac{ln(2/\delta)}{t_i - 1}$$

- R is the size of the range of i'th filter's marginal contributions, set R = 1

# Experiment

- Inception-v3 architecture, ImageNet dataset, 17216 filters
- Select top-100 important filters
  - origin Inception-v3 -> 74%, remove top 10 -> 38%, top 20 -> 8%, random 20 -> 74%

# Critical Neurons for Overall Acc

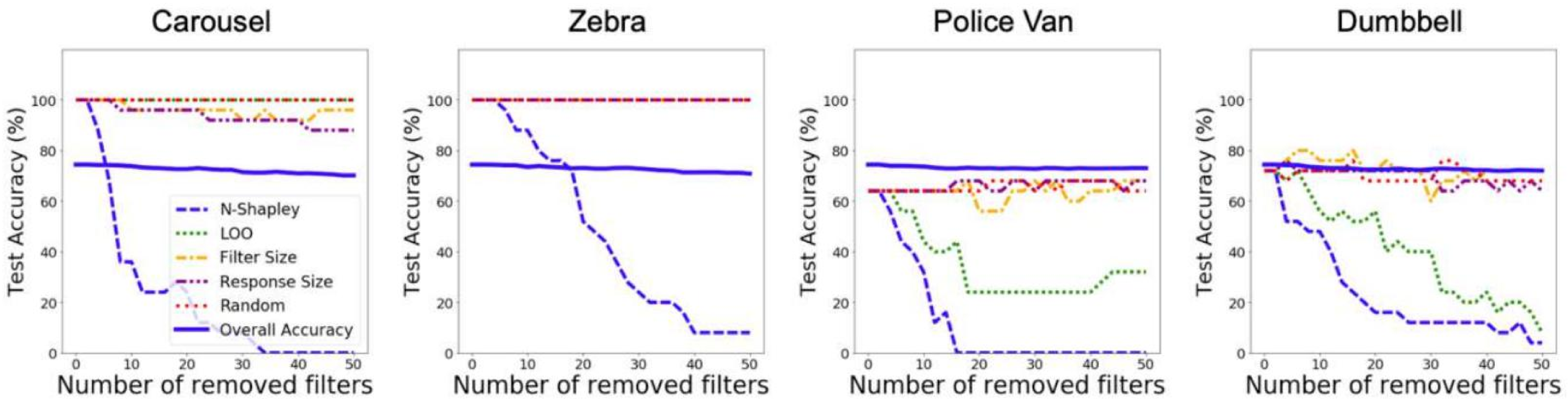


# Class Specific Critical Neurons

- Use class recall as V
- Excluding the top-20% neurons that contributed mostly to the overall accuracy
- Compare with
  - filter size -> $L_2$ norm of the weights
  - Response size -> $L_2$ norm of the filter's response
  - leave-one-out impact

# Class Specific Critical Neurons

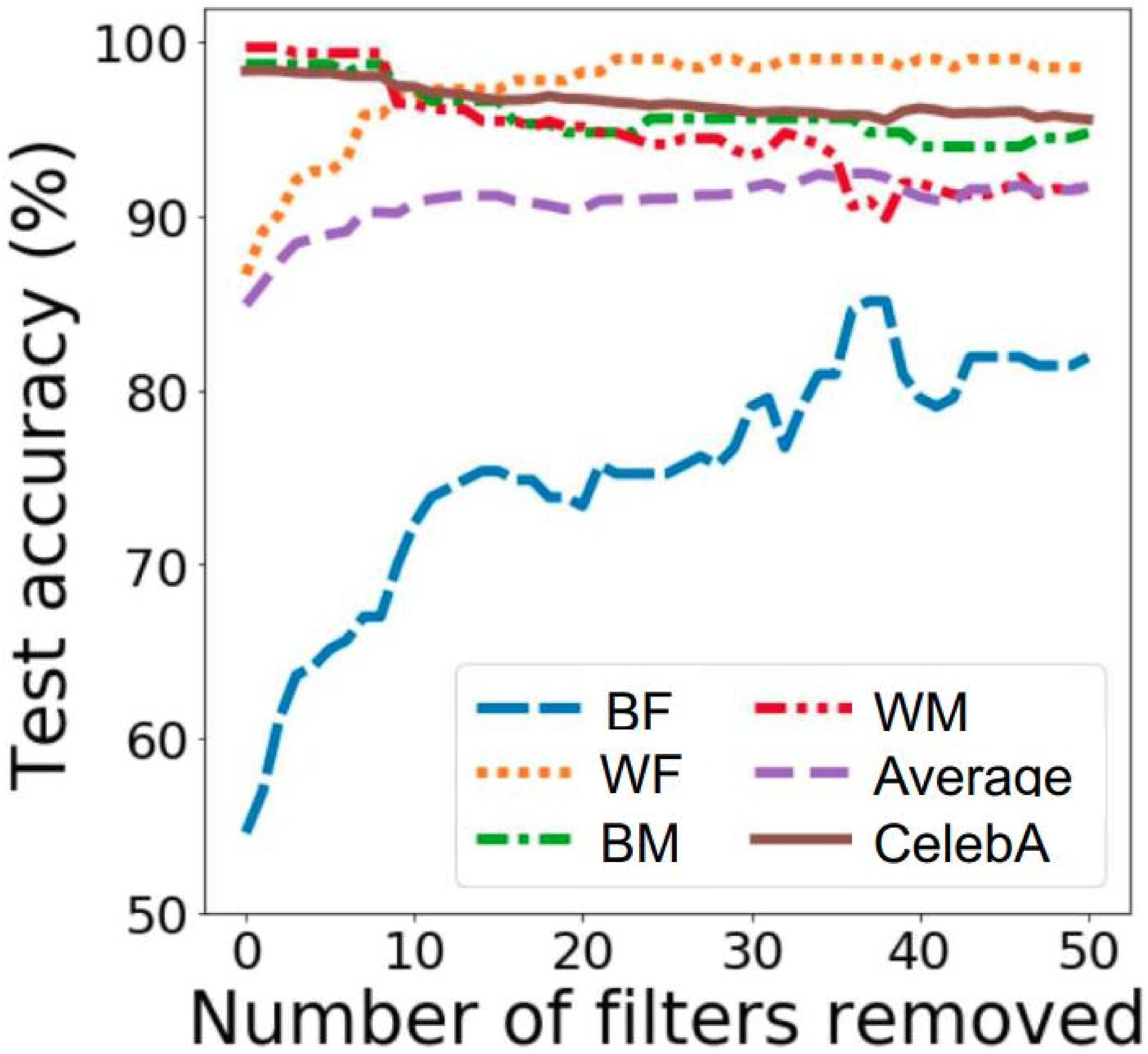
- Class-specific filters are more common in the deeper layers
- Removing class-specific critical neurons does not affect the overall performance



# Class Specific Critical Neurons

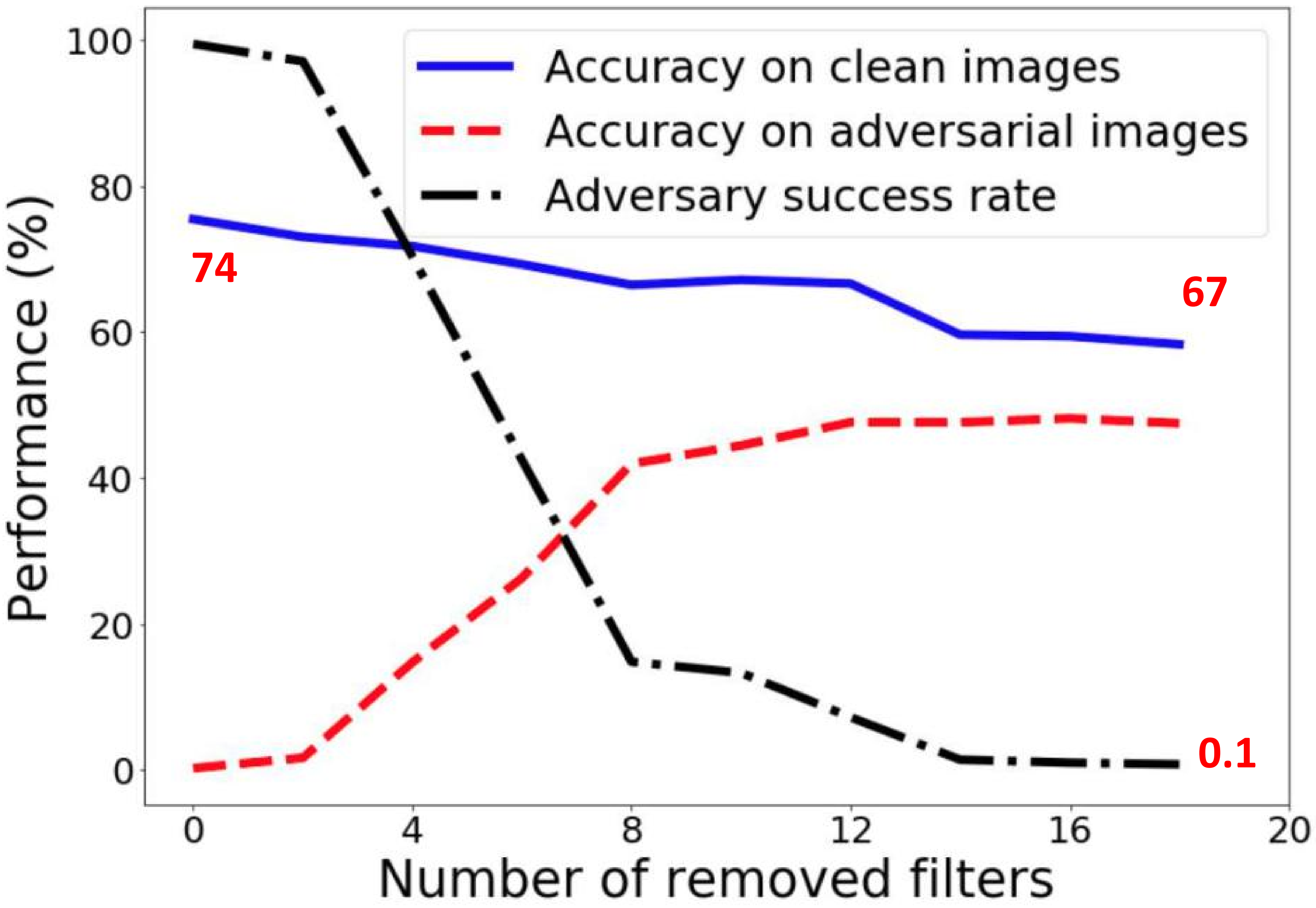
# Discovering Unfair Filters

- SqueezeNet, celebA dataset, 2976 filters, gender detection
- Accuracy on the PPB dataset as V, fairness
    - equal representations of four subgroups of gender-race
    - white female (WF), black female (BF), white male (WM) and black male (BM)
- Finding the most negative values and "zero out"
- PPB acc -> 84.9% to 91.7%
- CelebA acc drops a little



# Identifying Filters Vulnerable to Adversaries

- Use iterative PGD attack as goal
- $\ell_\infty$ perturbations with size $\epsilon = 16/255$
- V = Adversary's success rate - Accuracy on clean images
- Adversarial Shapley value & original value -> 0.3
  - filters interact differently on the adversarially perturbed images



# Experiment

- Compare with Gradient-based method
  - Neuron Conductance[1] based on Integrated Gradients

$$\text{Cond}_i^y(x) ::= (x_i - x_i') \cdot \int_{\alpha=0}^{1} \frac{\partial F(x'+\alpha(x-x'))}{\partial y} \cdot \frac{\partial y}{\partial x_i} \, d\alpha$$

  - removal twice as many filters to achieve similar reduction for overall acc
  - for removing unfair filters, NC 84.9% -> 88.7% by removing 105 neurons, NS -> 91.7%, 50 neurons
  - for vulnerable filters, NC 20 and NS 16 to achieve the same reduction in adversarial success rate

[1] How important is a neuron? Kedar Dhamdhere, John Wieting, Mukund Sundararajan, Qiqi Yan. ICLR 2019, Google AI.

# Conclusion

- Shapley Value can be useful in many tasks
- Usually needs sampling and truncation
- Does Shapley still satisfy equitable properties after approximation?
- Shapley Value vs Influence Function, can influence function performs better in some situation ?