CS 360: Machine Learning

Sara Mathieson, Sorelle Friedler

Spring 2024

HAVERFORD
COLLEGE

Admin

- **Lab 8** due Thursday!
  - Sorelle office hours Wednesday 3-4pm
  - Check-in during lab today (hopefully through Part 2)
- **Midterm April 25** in class (next Thursday)
  - Study guide out this Thursday
  - Can still do handout videos for extra credit! Up to 24 hours before exam
- **Project presentations**: last week of classes
- **Writeup** due by the end of finals period
  - May 11 for seniors (AND groups involving seniors)
  - May 17 for non-seniors

Lab 8 notes

- For generating text you can make your own string of at least “window” length, then encode and convert as before
- You don’t need “y”, only “x”

- The transformer for text generation part (end of Part 3) will be worth a small amount of credit – try it out and include ideas in your README even if it doesn’t quite work
- Text generated from both models may not be amazing, that’s okay!

Lab 8 notes

```python
def to_dataset(sequence, length, shuffle=False, seed=None, batch_size=32):
    ds = tf.data.Dataset.from_tensor_slices(sequence)
    ds = ds.window(length + 1, shift=1, drop_remainder=True)
    ds = ds.flat_map(lambda window_ds: window_ds.batch(length + 1))
    if shuffle:
        ds = ds.shuffle(10_000, seed=seed)
    ds = ds.batch(batch_size)
    return ds.map(lambda window: (window[:, :-1], window[:, 1:])).prefetch(1)
```

X Y

Outline for April 16

- Finish GAN (CNN generators)
- Interpretability (LIME paper)
- Interpretability (saliency maps)

Outline for April 16

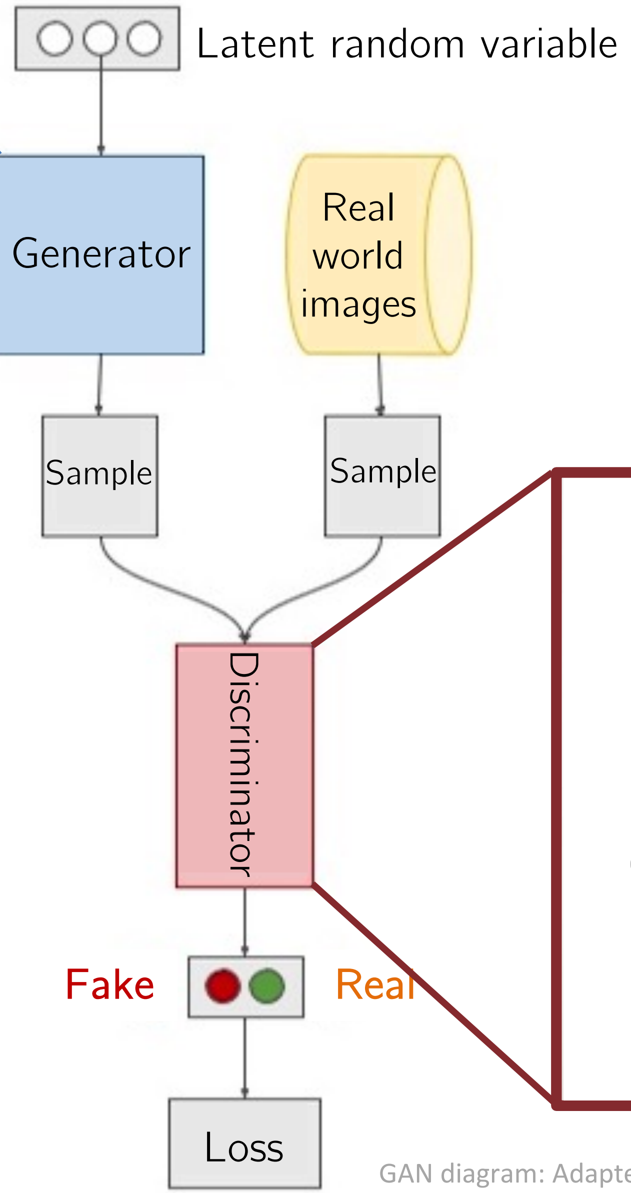
- Finish GAN (CNN generators)

- Interpretability (LIME paper)

- Interpretability (saliency maps)

Typical architecture of an image GAN

For images: generator is often a CNN



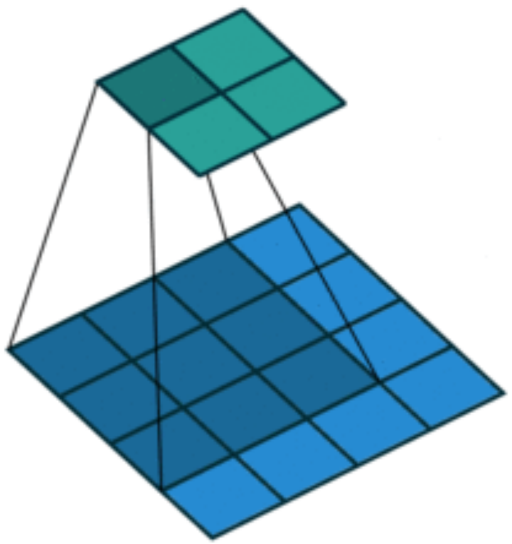
Discriminator is also often a CNN
(Note: goal is 50% accuracy)

GAN diagram: Adapted from Kevin McGuinness

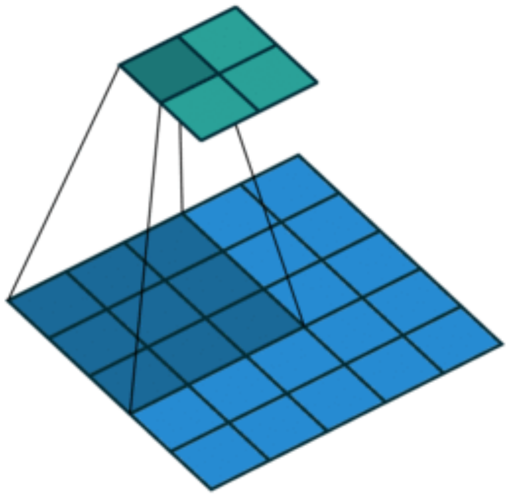
GAN discriminator is a “normal” CNN

Review of convolutions: typically output is smaller than the input
or we pool after to make it smaller

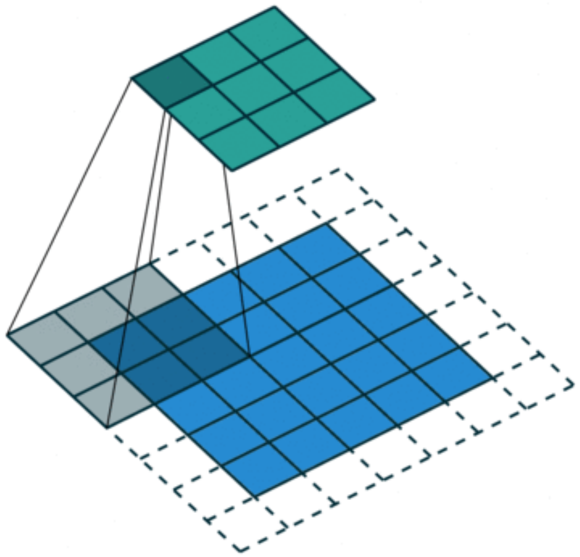
Blue maps are inputs, and cyan maps are outputs.



No padding, no strides



No padding, strides

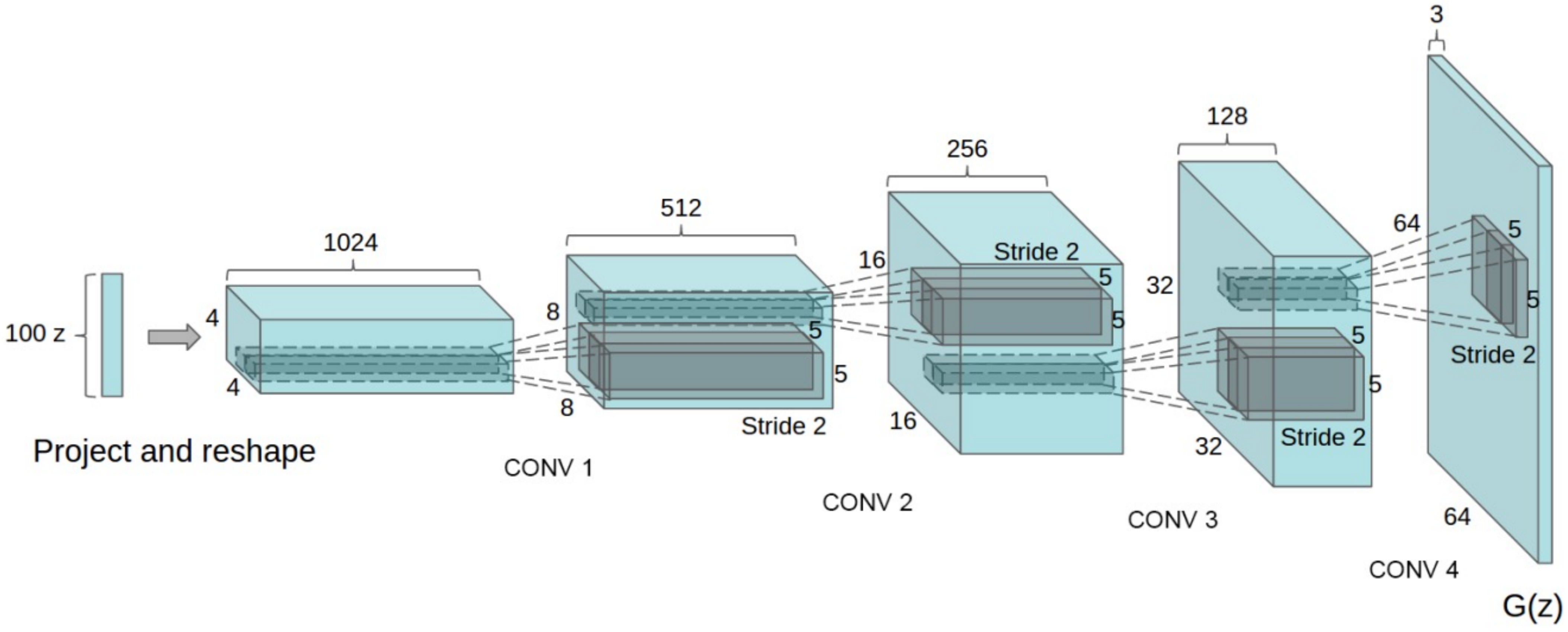


Padding, strides

Vincent Dumoulin, Francesco Visin - [A guide to convolution arithmetic for deep learning]

GAN generator uses "transposed" convolutions

- Often called "deconvolutions"
- Goal is to start from a small vector of noise and end with a 3D image

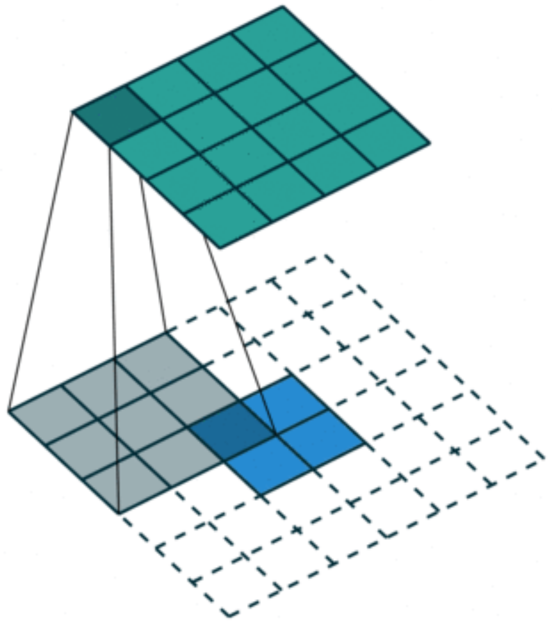


https://arxiv.org/pdf/1511.06434.pdf

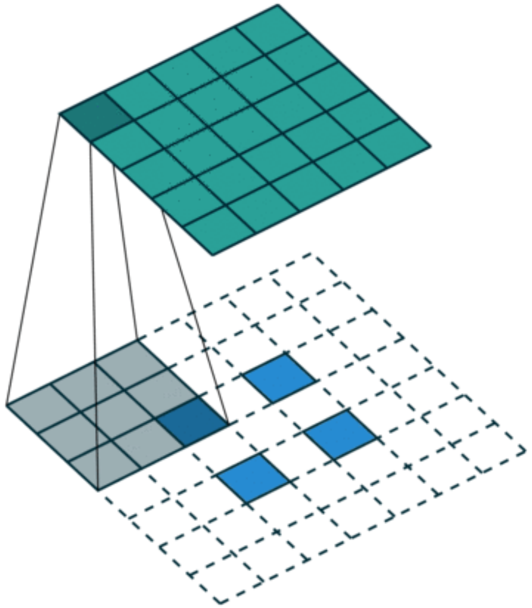
GAN generator uses “transposed” convolutions

Goal is to make the output larger than the input

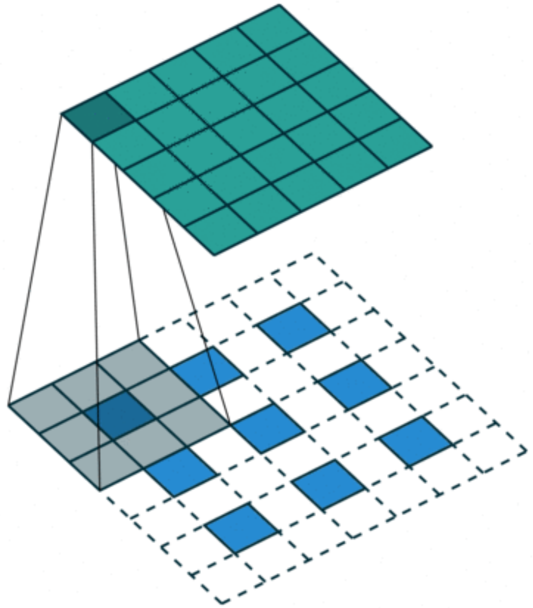
Blue maps are inputs, and cyan maps are outputs.



No padding, no strides, transposed

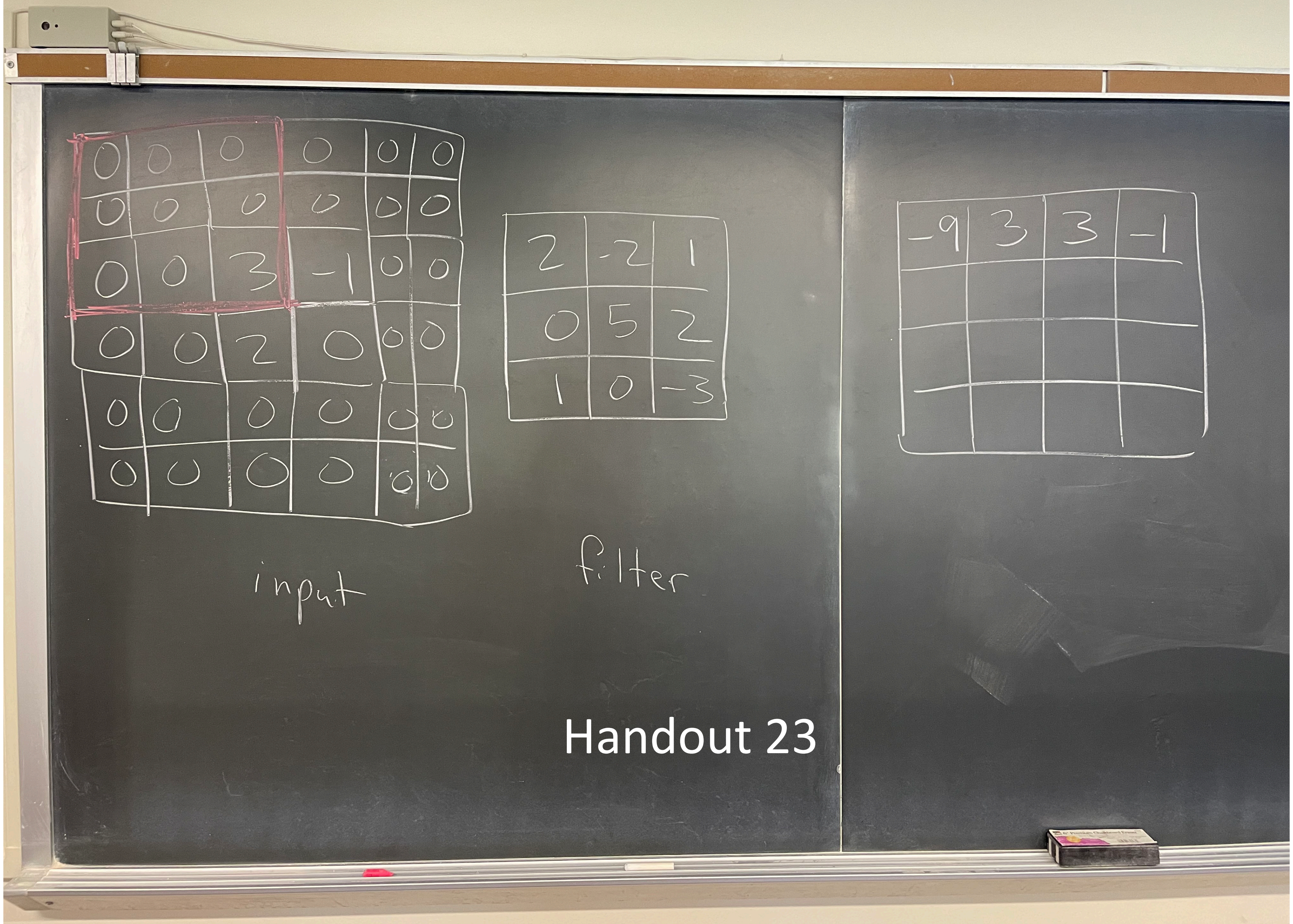


No padding, strides, transposed



Padding, strides, transposed

Vincent Dumoulin, Francesco Visin - [A guide to convolution arithmetic for deep learning]



0	0	0	0	0	0
0	0	0	0	0	0
0	0	3	-1	0	0
0	0	2	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

input

2	-2	1
0	5	2
1	0	-3

filter

-9	3	3	-1

Outline for April 16

- Finish GAN (CNN generators)
- Interpretability (LIME paper)
- Interpretability (saliency maps)

Interpretability and Explainability

- **Local interpretability**
  - Explaining a model's prediction on a specific example
  - What parts/features of the example were most important
    - We already did this in CS260!

- **Global interpretability**
  - Explaining what the model has learned overall
  - Example: looking at the filters of a CNN

Goal: explain a model's predictions

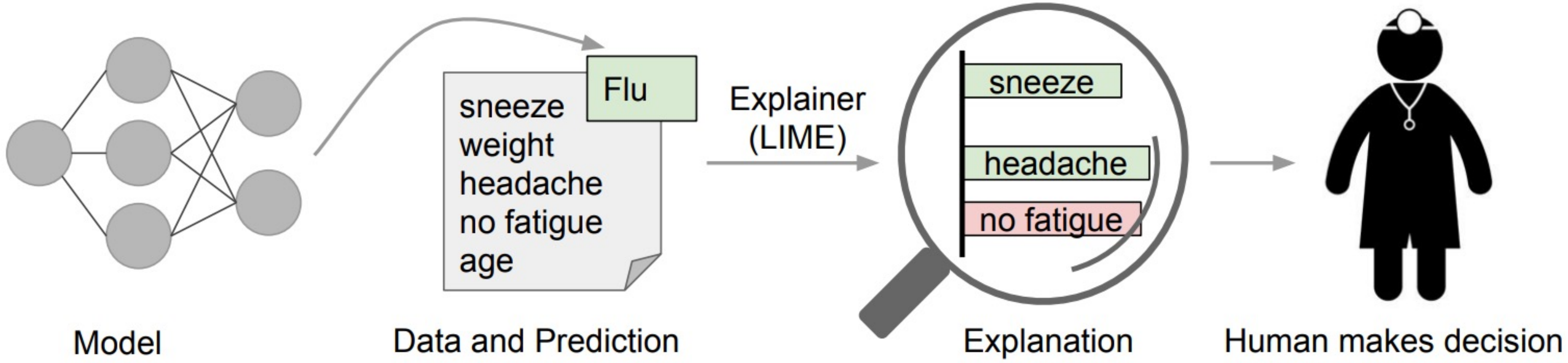
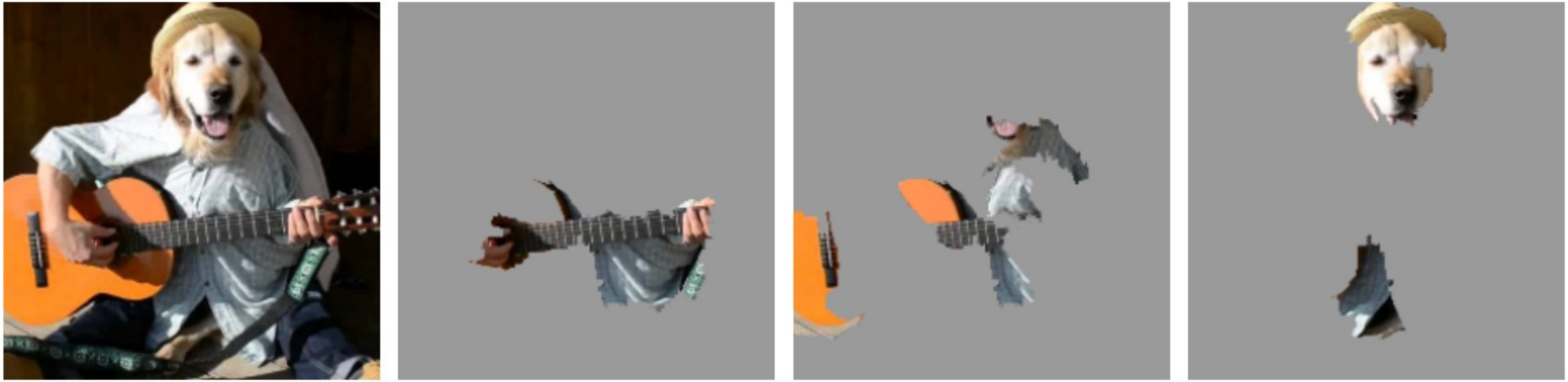


Figure 1: Explaining individual predictions. A model predicts that a patient has the flu, and LIME highlights the symptoms in the patient's history that led to the prediction. Sneeze and headache are portrayed as contributing to the "flu" prediction, while "no fatigue" is evidence against it. With these, a doctor can make an informed decision about whether to trust the model's prediction.

"Why Should I Trust You?" Explaining the Predictions of Any Classifier

LIME interpretability method



(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"** ($p = 0.21$)

"Why Should I Trust You?" Explaining the Predictions of Any Classifier

LIME interpretability method

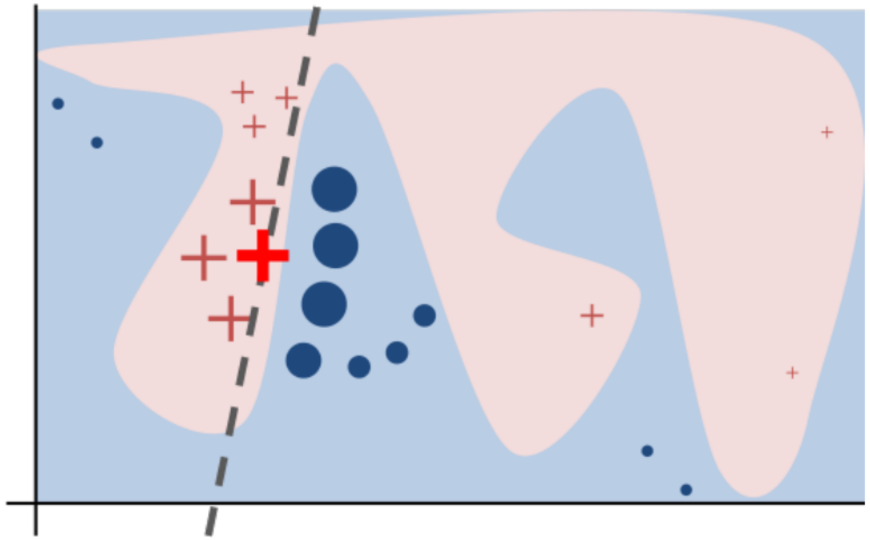


Figure 3: Toy example to present intuition for LIME. The black-box model's complex decision function f (unknown to LIME) is represented by the blue/pink background, which cannot be approximated well by a linear model. The bold red cross is the instance being explained. LIME samples instances, gets predictions using f, and weighs them by the proximity to the instance being explained (represented here by size). The dashed line is the learned explanation that is locally (but not globally) faithful.

Algorithm 1 Sparse Linear Explanations using LIME

Require: Classifier f, Number of samples N
Require: Instance x, and its interpretable version x'
Require: Similarity kernel π_x, Length of explanation K
  $\mathcal{Z} \leftarrow \{\}$
  for $i \in \{1, 2, 3, ..., N\}$ **do**
    $z_i' \leftarrow sample\_around(x')$
    $\mathcal{Z} \leftarrow \mathcal{Z} \cup \langle z_i', f(z_i), \pi_x(z_i) \rangle$
  end for
  $w \leftarrow$ K-Lasso$(\mathcal{Z}, K)$ ▷ with z_i' as features, $f(z)$ as target
  return w

"Why Should I Trust You?" Explaining the Predictions of Any Classifier

LIME

$x \in \mathbb{R}^d$ one example $\Big\}$ what I want to explain

$x' \in \{0,1\}^{d'}$ interpretable version

goal

find w_g

$\Rightarrow$ text $\Rightarrow$ "bag of words"

presence/absense of vocab words (text)

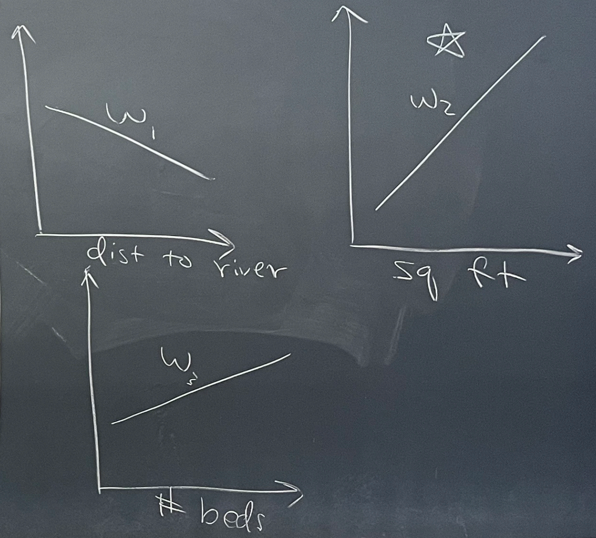
" " of pixel groups (image)

" " of a feature (other)

g is explanation model

☆ $\boxed{g(z') = w_g \cdot z' \quad \text{(linear)}}$

predict house price



$f : \mathbb{R}^d \to \mathbb{R}$ original model

$\pi_x(z)$: kernel (similarity between z & x)

K : length of explanation (# of important features)

$\boxed{\text{Algorithm}}$

```
Z = [ ]
for i = 1, 2 ... N   } # samples
    z_i ← sample_around(x)  ← gaussian
    Z.append(z_i', f(z_i), π_x(z_i))
             ↑       ↑        ↑
         features  target   weights
```

outside loop

$w \leftarrow \underbrace{\text{train\_linear\_reg}}_{\text{LASSO}}(Z, K)$

K: $\underbrace{\text{\# of non-zero weights}}_{\text{\# important features}}$

$z \longrightarrow z'$

sentence $\longrightarrow$ bag of words

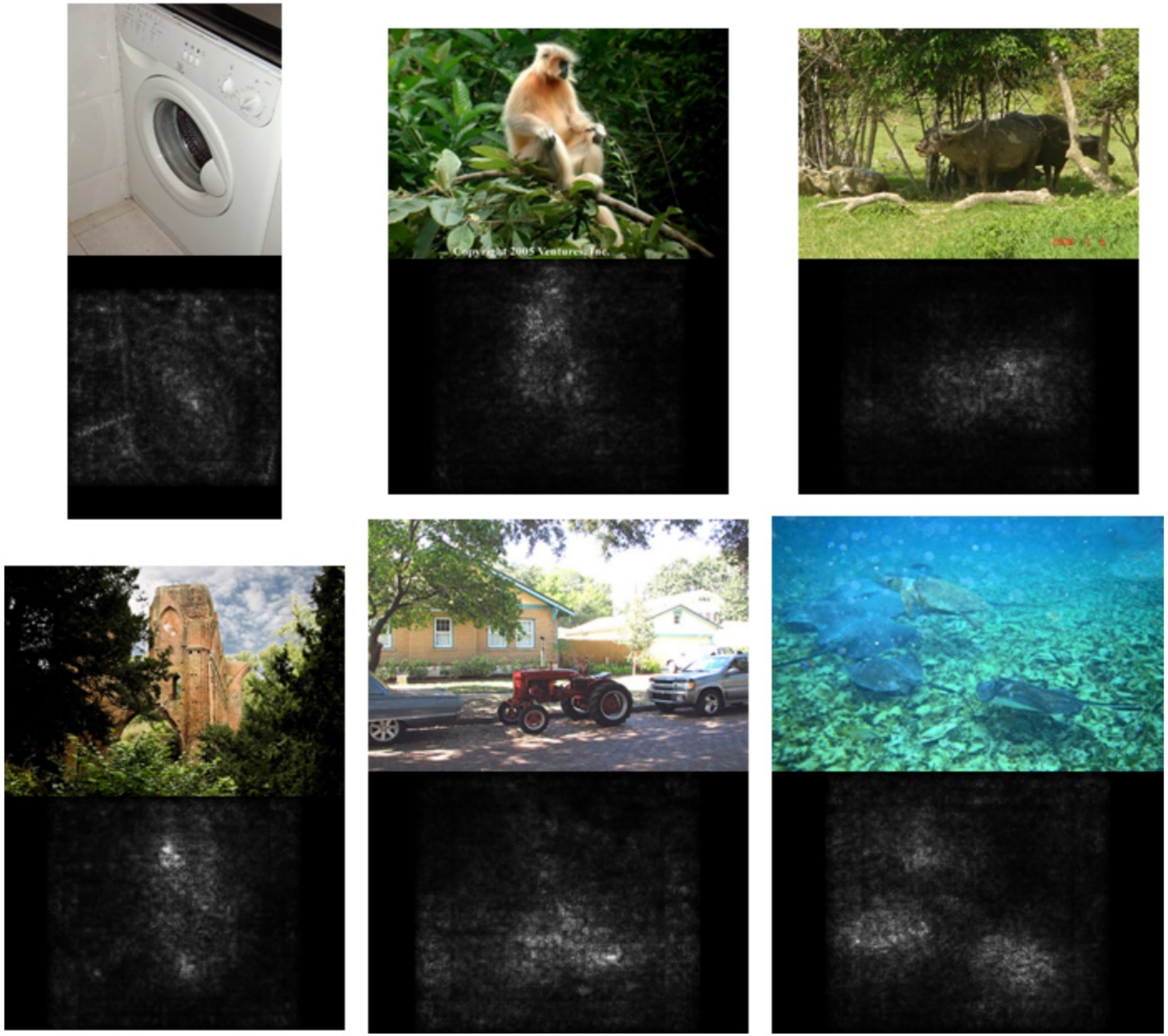
$\cancel{z \leftarrow z'}$

Outline for April 16

- Finish GAN (CNN generators)
- Interpretability (LIME paper)
- Interpretability (saliency maps)

Saliency Maps

- Shows which pixels would impact the classification scores the most if changed slightly



https://arxiv.org/pdf/1312.6034.pdf

Caution is required when working with saliency maps, can be largely edge detectors and ignore the model

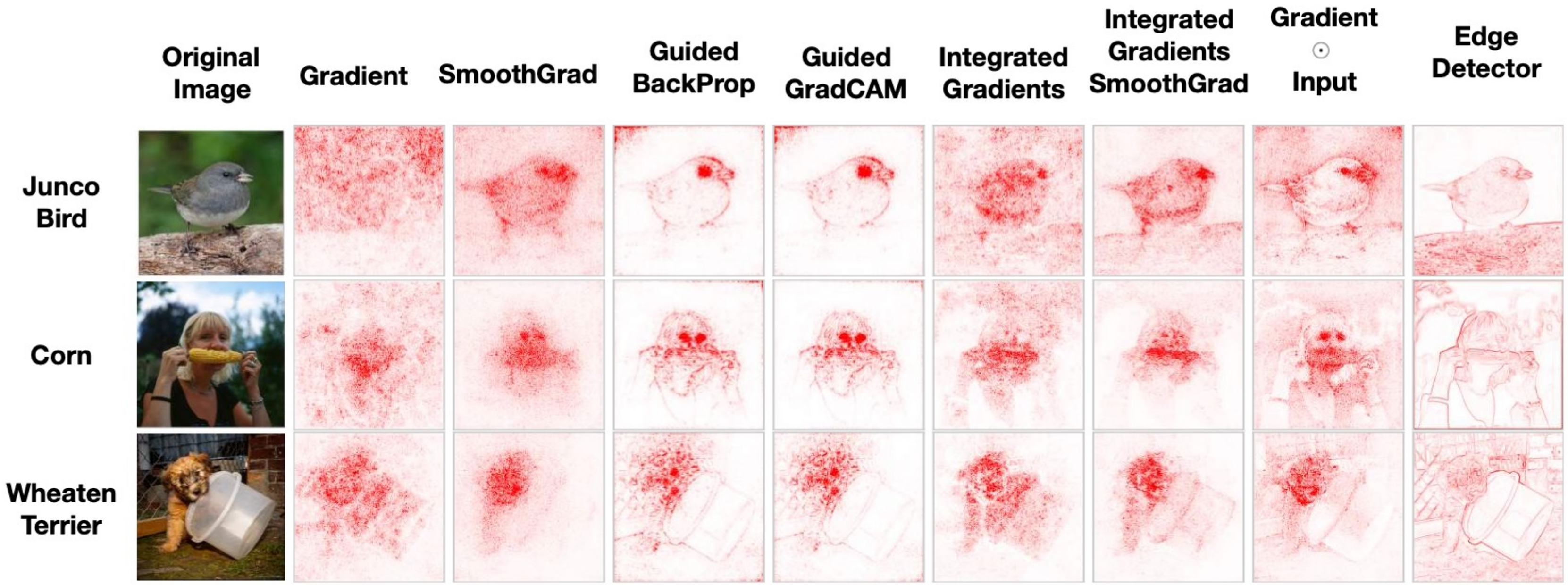
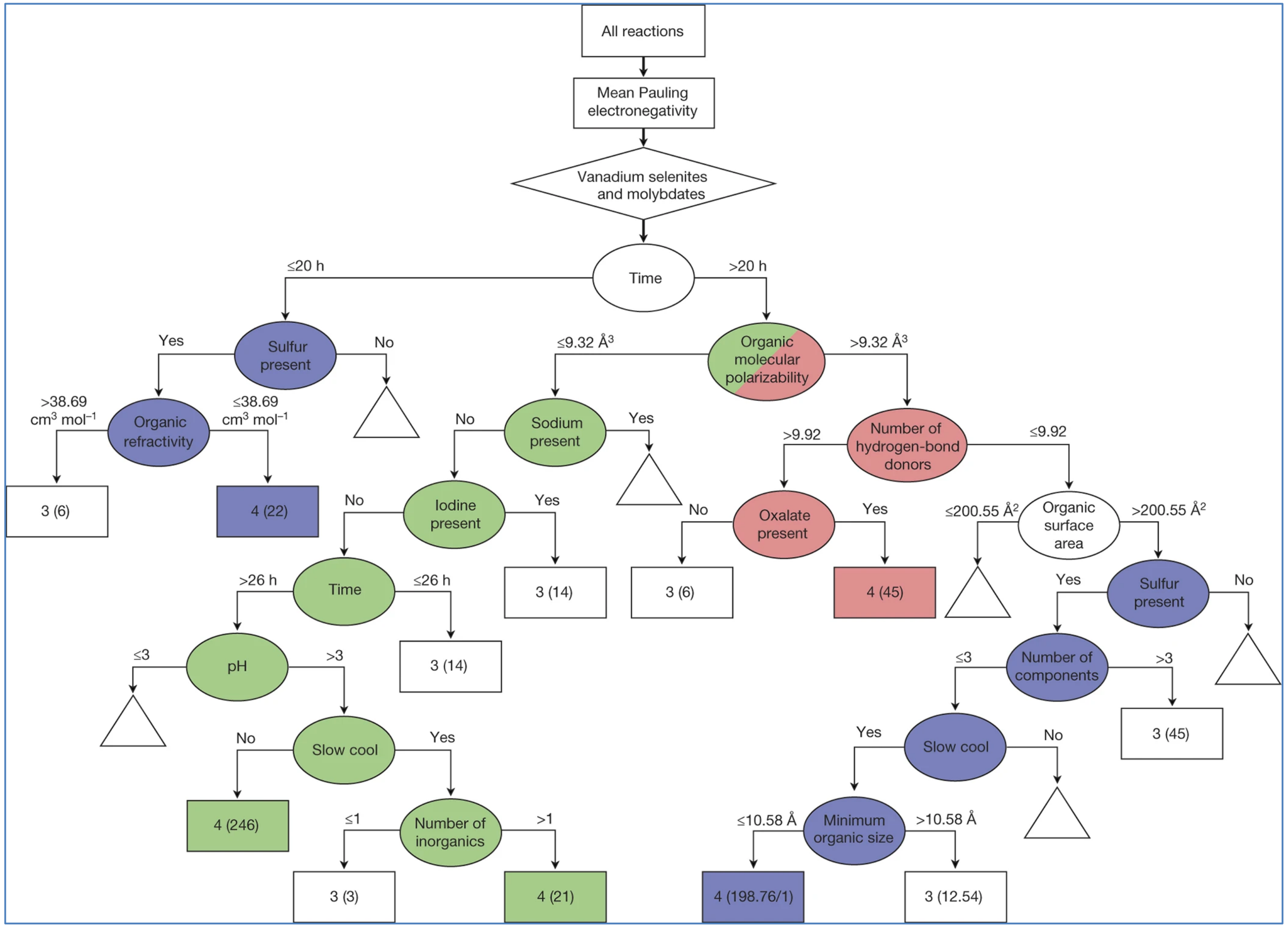


Figure 1: **Saliency maps for some common methods compared to an edge detector.** Saliency masks for 3 inputs for an Inception v3 model trained on ImageNet. We see that an edge detector produces outputs that are strikingly similar to the outputs of some saliency methods. In fact, edge detectors can also produce masks that highlight features which coincide with what appears to be relevant to a model's class prediction. We find that the methods most similar (see Appendix for SSIM metric) to an edge detector, i.e., Guided Backprop and its variants, show minimal sensitivity to our randomization tests.

Sanity Checks for Saliency Maps

Using an explainable model to predict decisions from an opaque model

- "Model of the model"
- Original model: SVM
- Interpretable model: Decision Tree



Machine-learning-assisted materials discovery using failed experiments

Paul Raccuglia, Katherine C. Elbert, Philip D. F. Adler, Casey Falk, Malia B. Wenny, Aurelio Mollo, Matthias Zeller, Sorelle A. Friedler, Joshua Schrier & Alexander J. Norquist

Nature **533**, 73–76 (2016) | Cite this article

Correlations between CNN filters and interpretable summary statistics

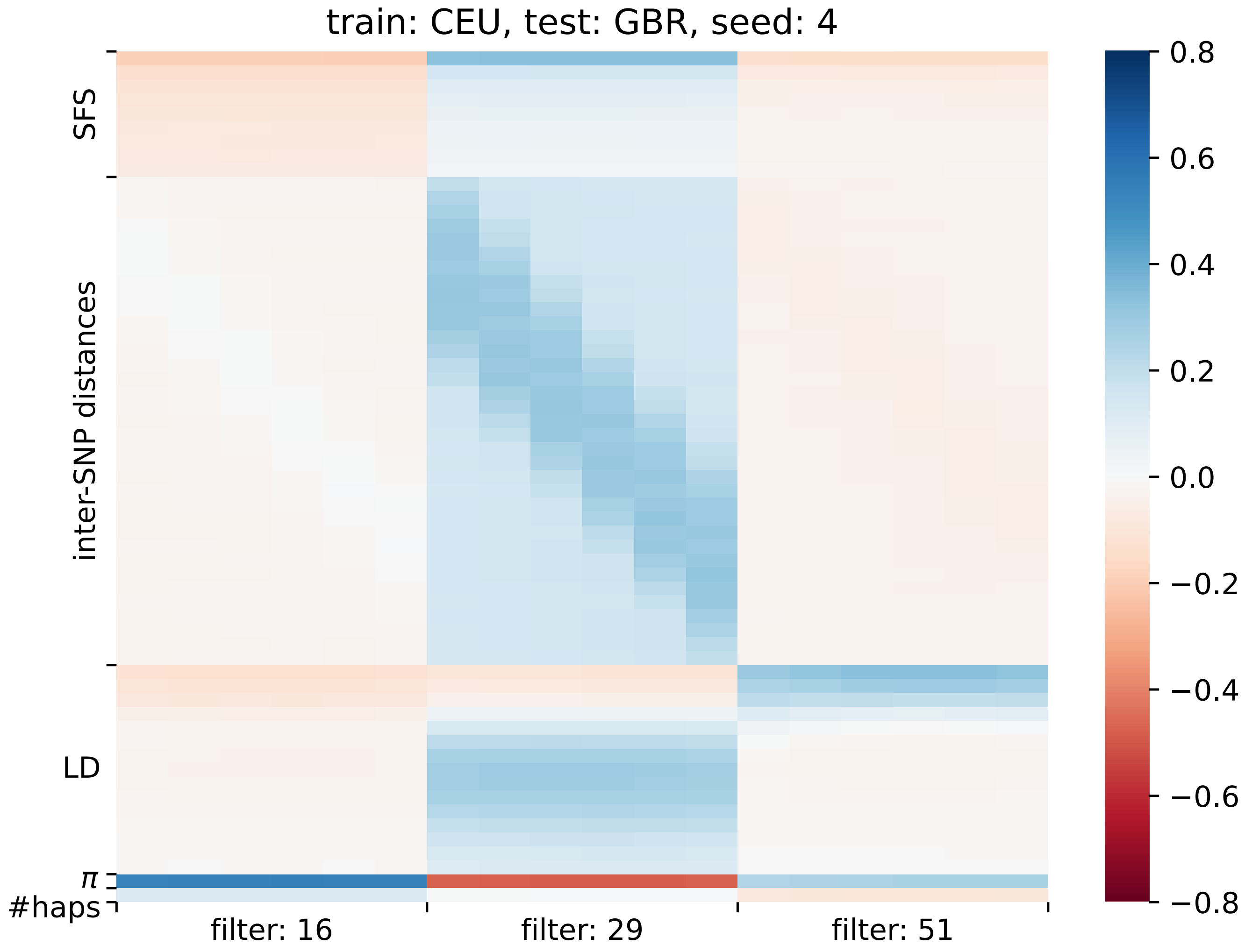


Image specific saliency map

$\underbrace{\text{Image specific}}_{\text{local interpretability}}$

image: I_0

class: c

model prediction: $S_c(I)$

if linear model

$$S_c(I) = w_c \cdot I + b_c$$

I ↑ flat

around I_0, use Taylor expantion to approximate complex model as a linear model

around I_0

$$S_c(I) \approx w_c \cdot I + b_c$$

$$\boxed{\left.\frac{\partial S_c(I)}{\partial I}\right|_{I_0} = w_c}$$