CS 8803-MDS
Human-in-the-loop Data Analytics

Lecture 18

10/26/22

Logistics

- Project update next Monday (10/31)
  - https://bit.ly/3gBLCPz
  - 5min each group
    - What did you propose to do?
    - What have you tried?
    - What are you going to do next?

Today's class

Domino: Discovering Systematic Errors with Cross-Model Embeddings

Authors: Cuong, Jingfan

Reviewer: Tanya, Abhinav

Archaeologist: Sankalp

Researcher: Shubham

ICLR | 2022

Tenth International Conference on Learning Representations

The Tenth International Conference on Learning Representations (Virtual)
Mon Apr 25th through Fri the 29th

Registration

Pricing »

Register starting Jan 30 02 PM PST »

Registration Cancellation Policy »

Certificate of Attendance

Announcements

- Call for Papers
- Call for Workshops

Virtual Site

The 2022 Virtual Site is now free to the public.

Tweet

Sponsors

The generous support of our sponsors allowed us to reduce our ticket price by about 50%, and support diversity at the meeting with travel awards. In addition, many accepted papers at the conference were contributed by our sponsors.

View ICLR 2022 sponsors »

~~Become a 2023 Sponsor »~~ (not currently taking applications)

2022 ICLR Organizing Committee

General Chair

- Katja Hofmann, Microsoft
- Deputy GC: Alexander (Sasha) Rush, Cornell Tech

Senior Program Chair

- Yan Liu, University of Southern California

Program Chairs

- Chelsea Finn, Stanford University
- Yejin Choi, University of Washington / AI2
- Marc Deisenroth, University College London

Workshop Chairs

- Feryal Behbahani, DeepMind
- Vukosi Marivate, University of Pretoria

Area Chairs

- Area Chairs »

Ethics Review Committee

Diversity Equity & Inclusion Chairs

- Krystal Maughan, University of Vermont
- Rosanne Liu, Google & ML Collective

Virtual Chairs - Virtual & Volunteers

- Jumanah Alshehri, Temple University
- Archana David, Infosys Ltd

Engagements Chairs - Socials & Sponsors

- Ehi Nosakhare, Microsoft
- William Agnew, University of Washington

Blog Track Chairs

- Sebastien Bubeck, Microsoft
- David Dobre, MILA
- Charlie Gauthier, MILA
- Gauthier Gidel, MILA
- Claire Vernade, DeepMind

Workflow Chairs

Domino: Discovering Systematic Errors With Cross-Modal Embeddings

Eyuboglu et al. (2022) - Presented at ICLR 2022

Paper Presentation by: Johnny Nguyen and Jingfan Meng

Georgia Tech

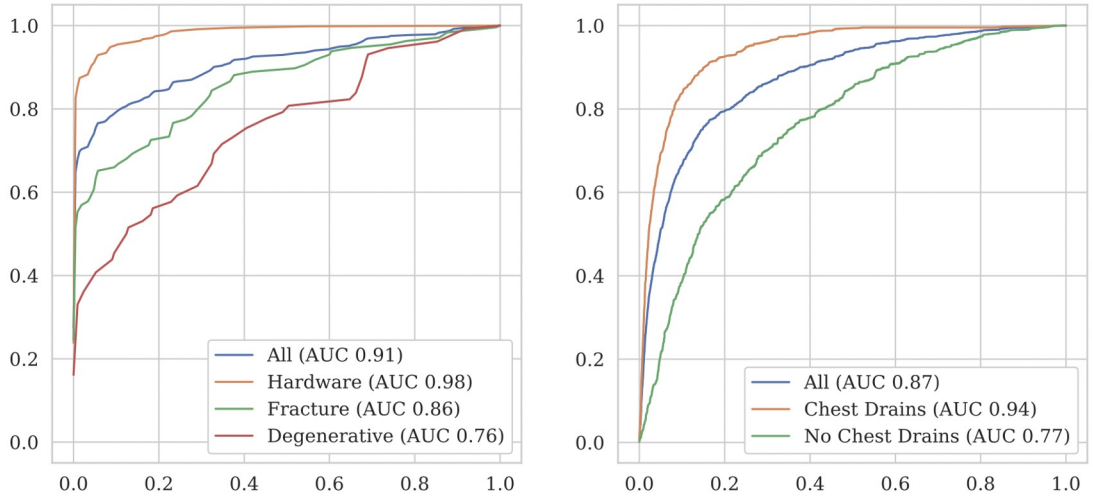
Introduction

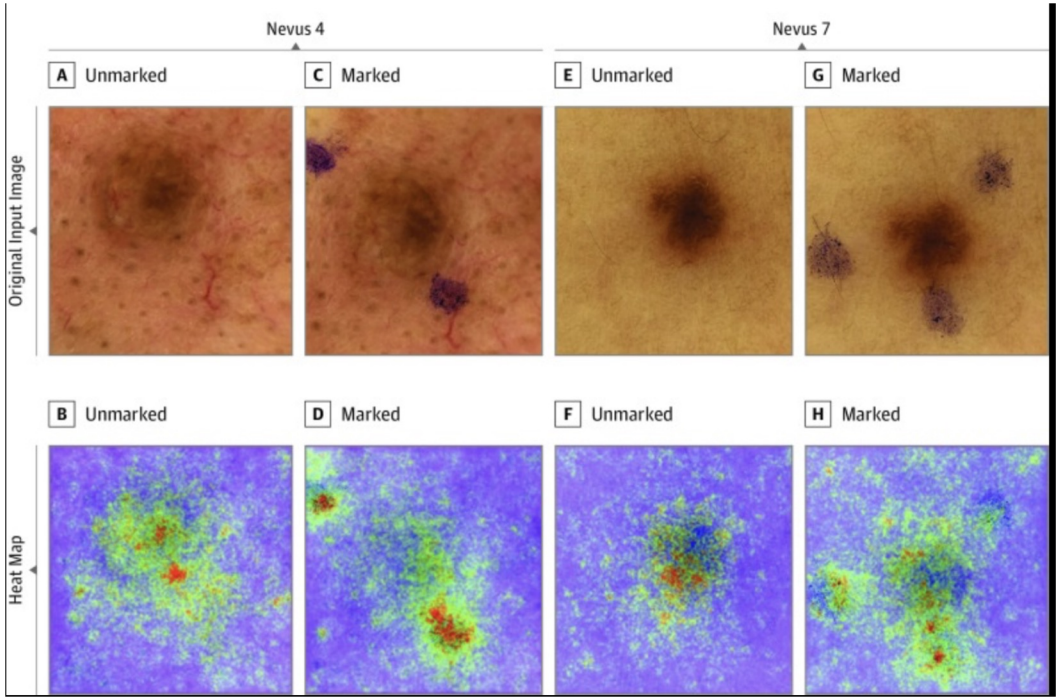
Discovering subsets (or slices) of data where Machine Learning (ML) models significantly underperform compared to entire dataset an important task for ML Fairness, Accountability, Transparency, and Ethics (FATE)

Slice	Log Loss	Size	Effect Size
All	0.35	30k	n/a
Sex = Male	0.41	20k	0.28
Sex = Female	0.22	10k	-0.29
Occupation = Prof-specialty	0.45	4k	0.18
Education = HS-grad	0.33	9.8k	-0.05
Education = Bachelors	0.44	5k	0.17
Education = Masters	0.49	1.6k	0.23
Education = Doctorate	0.56	0.4k	0.33

Georgia Tech

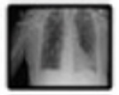
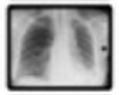
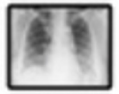
Application: AI Accountability in Medine



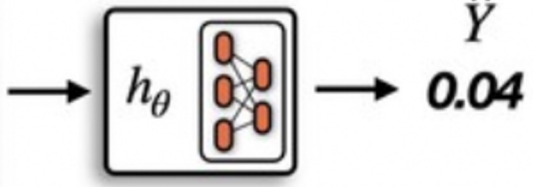


Georgia Tech

Diagnosing collapsed lungs with machine learning

X	Y	$\hat{Y}$
	0	0.04
	1	0.98
	0	0.13
	1	0.86
	0	0.24
	0	0.08
	1	0.18
	1	0.99

X



$\hat{Y}$

0.04

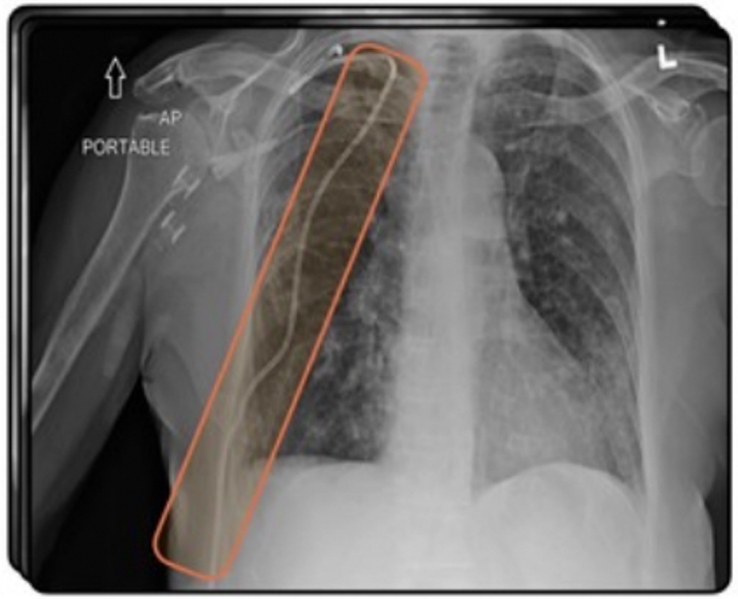
$$\text{AUROC}(Y, \hat{Y}) = \mathbf{87\%}$$

Oakden-Rayner *et al.* CHIL (2020)

What if we slice the data?

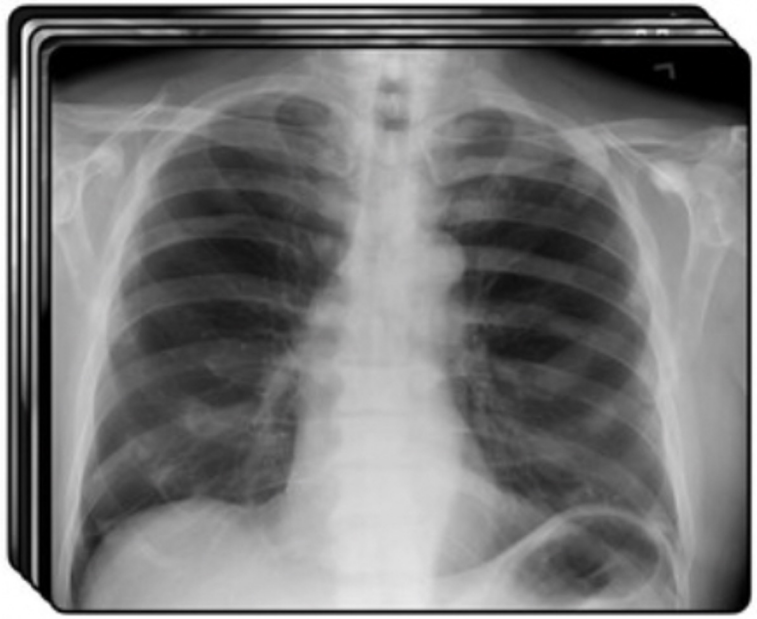
chest tube

$\text{AUROC}(Y, \hat{Y} \mid S = 1) = \mathbf{94\%}$



no chest tube

$\text{AUROC}(Y, \hat{Y} \mid S = 0) = \mathbf{77\%}$



Oakden-Rayner *et al.* CHIL (2020)
Note: negatives (*i.e.* Y=0) from both slices were included when computing AUROC for each slice.

Problem Definition, Slice Discovery Methods

Slice Discovery Problem

- **Inputs**: a trained classifier h_θ and labeled dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ with n samples drawn from $P(X, Y)$.
- **Output**: a set of $\hat{k}$ slicing functions $\Psi = \{\psi^{(j)} : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}\}_{j=1}^{\hat{k}}$ that partition the data into $\hat{k}$ subgroups.

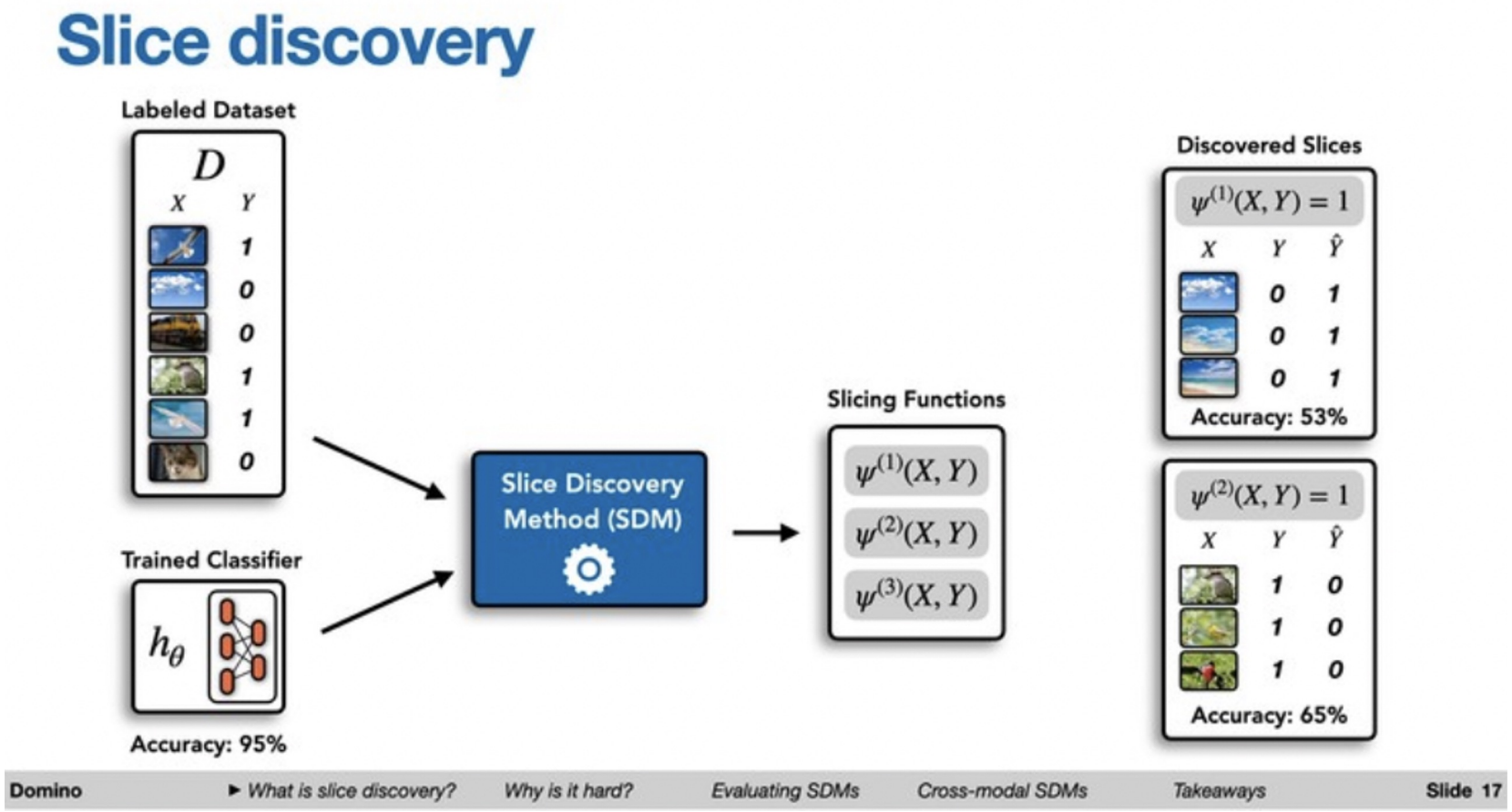
Problem Definition, Slice Function Successfulness

A slicing function is successful if all slices are predicted at precision greater than threshold β

$$\forall u \in [k]. \quad \exists v \in [\hat{k}]. \quad P(S^{(u)} = 1 | \psi^{(v)}(X, Y) = 1) > \beta.$$

Georgia Tech

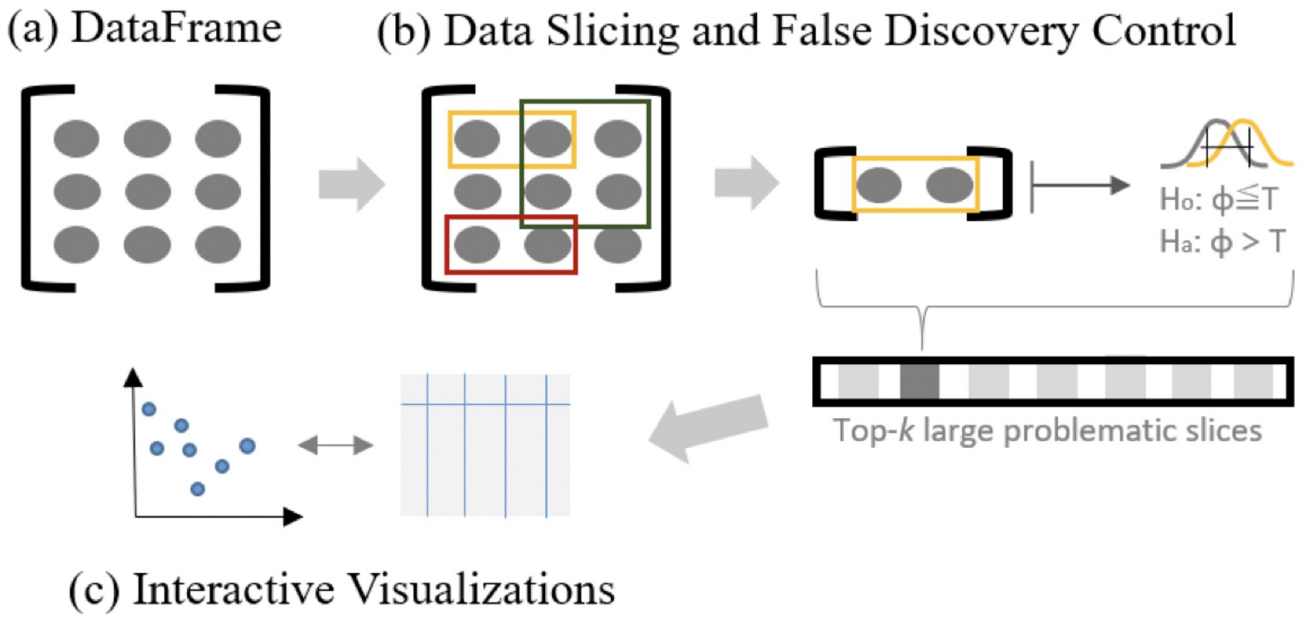
Summary of Slide Discovery Problem



Related Works - SliceFinder

Find problematic “slices” of data via hypothesis testing + visualization

Only works for tabular data

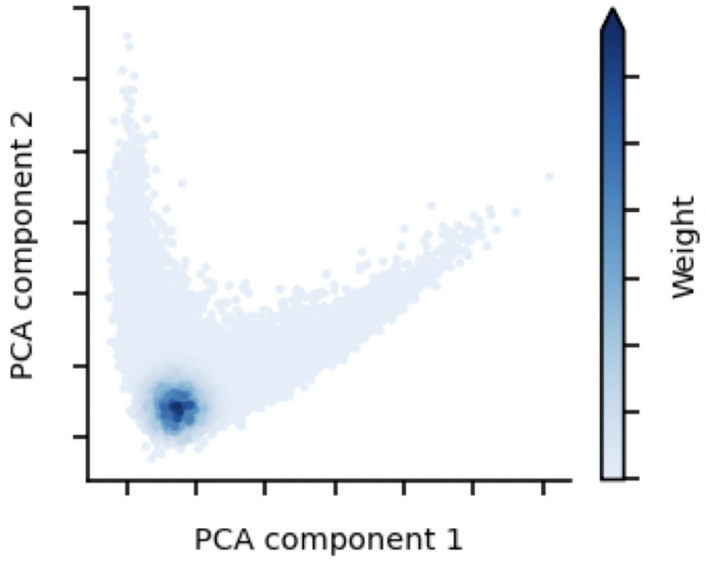


Georgia Tech

Related Works - The SpotLight

Finding continuous regions within the embedding space where model consistently underperforms

Works for deep learning models, but unable to interpret what the “spotlights” represent from an expert perspective



Georgia Tech

Limitations of Previous Work

Previous studies have proposed automated slice discovery methods (SDMs). However…

1. No quantitative evaluation framework has been proposed for rigorously assessing SDMs with respect to both performance **AND** coherence
2. Qualitative evaluations in previous papers have shown that existing SDMs often identify slices that are incoherent from the perspective of domain experts

Georgia Tech

How to Evaluate SDMs

We evaluate SDMs on a large number of *discovery settings*.

Each setting has

1. A labelled **dataset**.
2. A **trained ML model** that underperforms on some slices of dataset.
3. Ground truth **slice annotations** on which the model underperforms.

We evaluate SDMs on how well they discover underperforming slices given the dataset and model, measured by precision@10.

SDM Evaluation Framework

Our evaluation framework consists of 1235 discovery settings generated from the following four **base datasets**:

1. ImageNet and CelebA are natural image datasets with hierarchical structure and 40 labelled attributes, respectively.
2. MIMIC-CXR is a medical image dataset with 14 labelled conditions.
3. A dataset of 12-second EEG (electroencephalography) signals for prediction the onset of seizures.

SDM Evaluation Framework

We observe that the underlying reasons of underperformance can be mainly categorized into (the existence of) three types of slices:

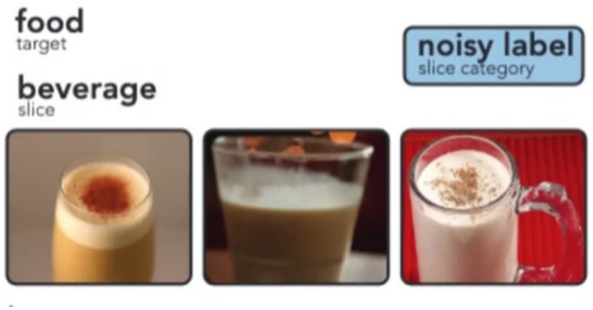
1. **Rare** slices: the dataset does not contain enough information to train the model (e.g., rare diseases).
2. **Correlated** slices: the model tends to choose obvious (but not decisive) features as criterion. For example, birds are often seen with the blue sky as background, but it not always the case.
3. **Noisy label** slices: noisy labels may mislead the model.

For each base dataset, we generate multiple discovery datasets and slices that simulate one of these patterns above.

Georgia Tech

SDM Evaluation Framework

Examples of common underperforming slices in evaluation settings.



Georgia Tech

SDM Evaluation Framework

Given datasets and slices, we use two types of ML models in our framework:

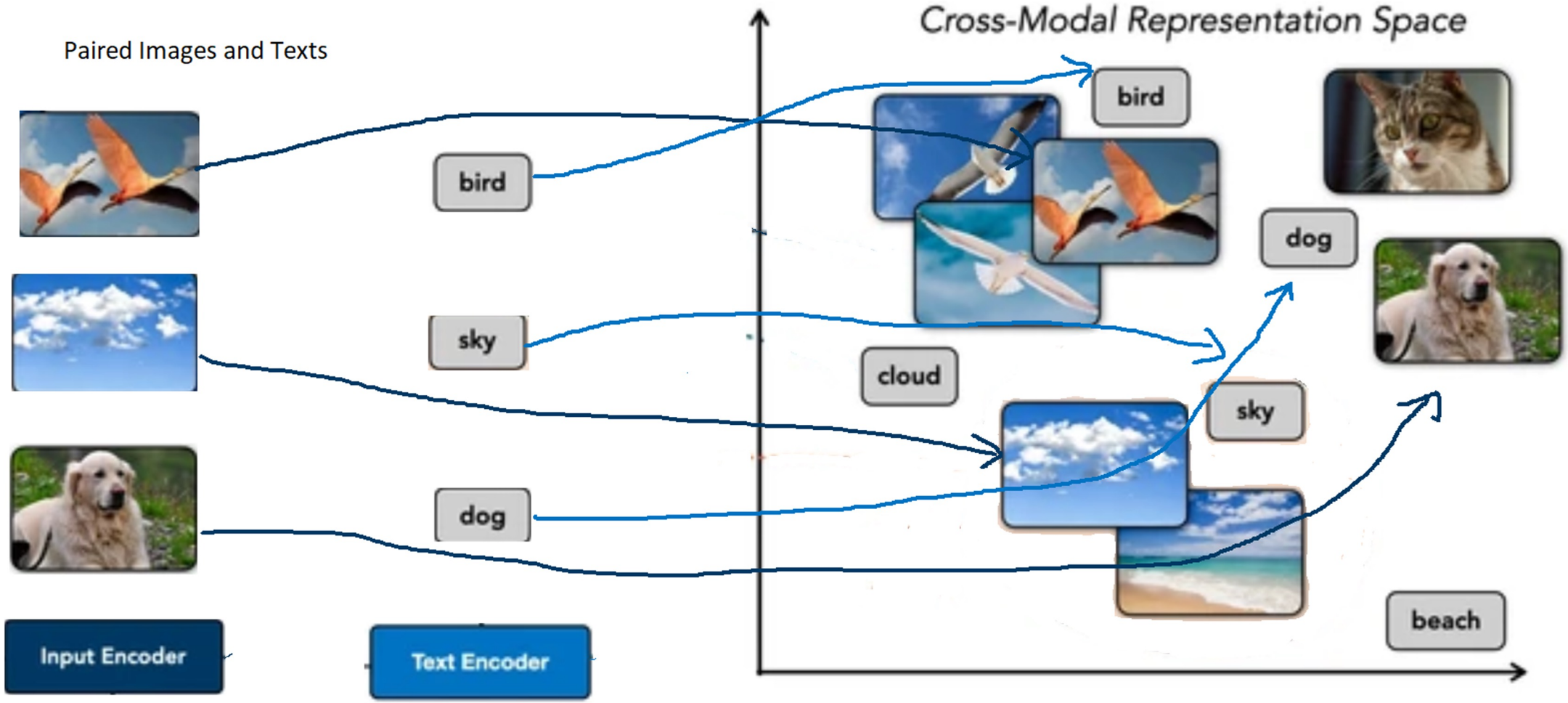
1. **Trained** models that actually have degraded performance on the slices. These models are realistic, but may also have degraded performance on other slices.
2. **Synthesized** models that output random predictions conditional on the classes and slices. They are easier for SDM, because the given slices are the only explanation for underperformance.

Domino Pipeline

1. **Embed** with cross-model embeddings.
2. **Slice** with error-aware mixture model.
3. **Describe** with natural language (keyword).

Georgia Tech

Domino: Embedding



Modified from slides of original authors.

Georgia Tech

Domino: Embedding

- Cross-model representation learning:

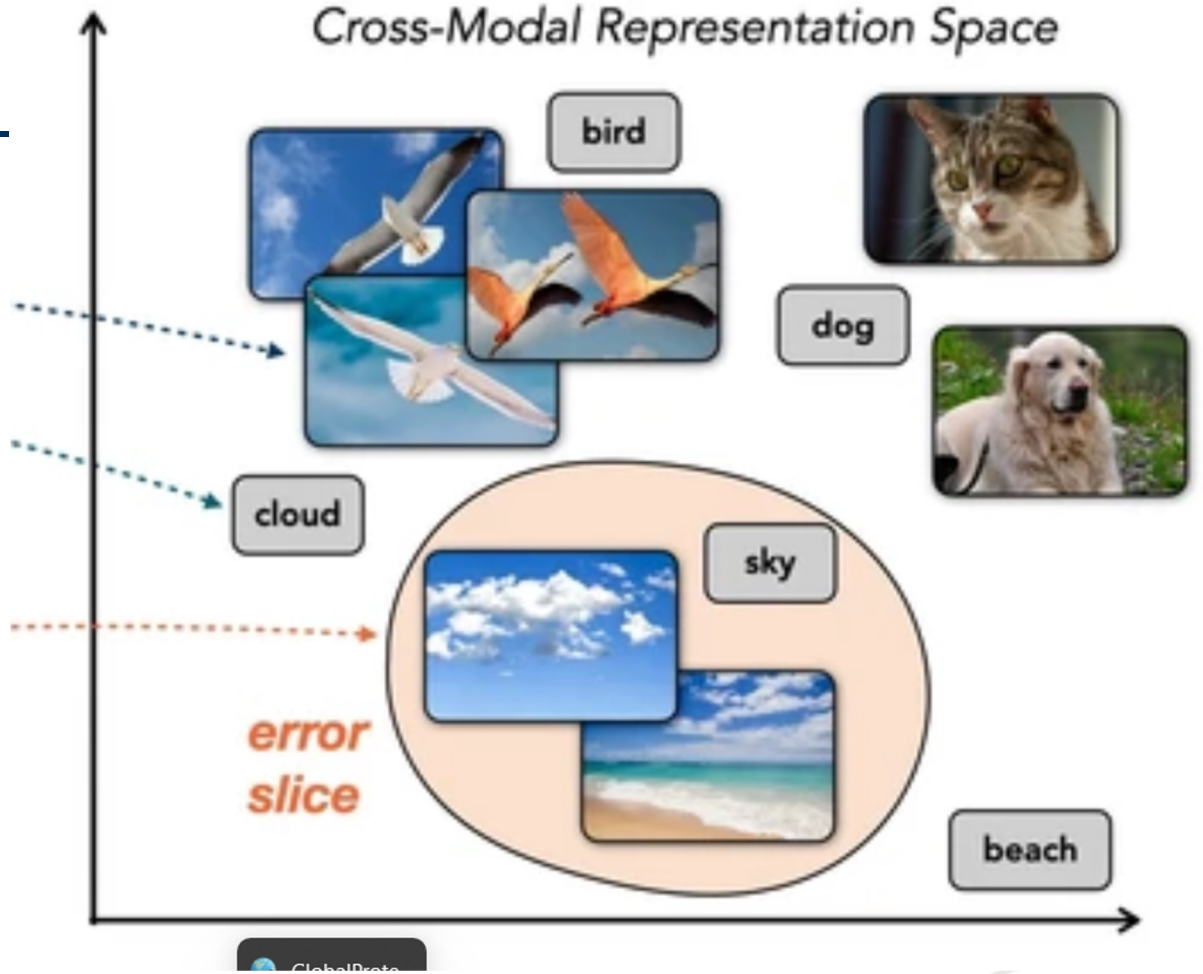
1. Input: images paired with descriptive text.
2. Output: embeddings of both images and text in the same cross-model representation space such that images are mapped close to their semantic texts.

- Domino assumes either pretrained cross-model embeddings are available or the dataset contains paired images and texts on which the embeddings can be trained.
- In Domino, multiple cross-model embeddings are used including CLIP, conVIRT, etc.

Domino: Slicing

- How do way find error slices (skies without birds) in the cross-model space such that:

1. The slice is semantically coherent.
2. The ML model underperforms on the slice.



Georgia Tech

Domino: Slicing

- Error-aware Mixture Model

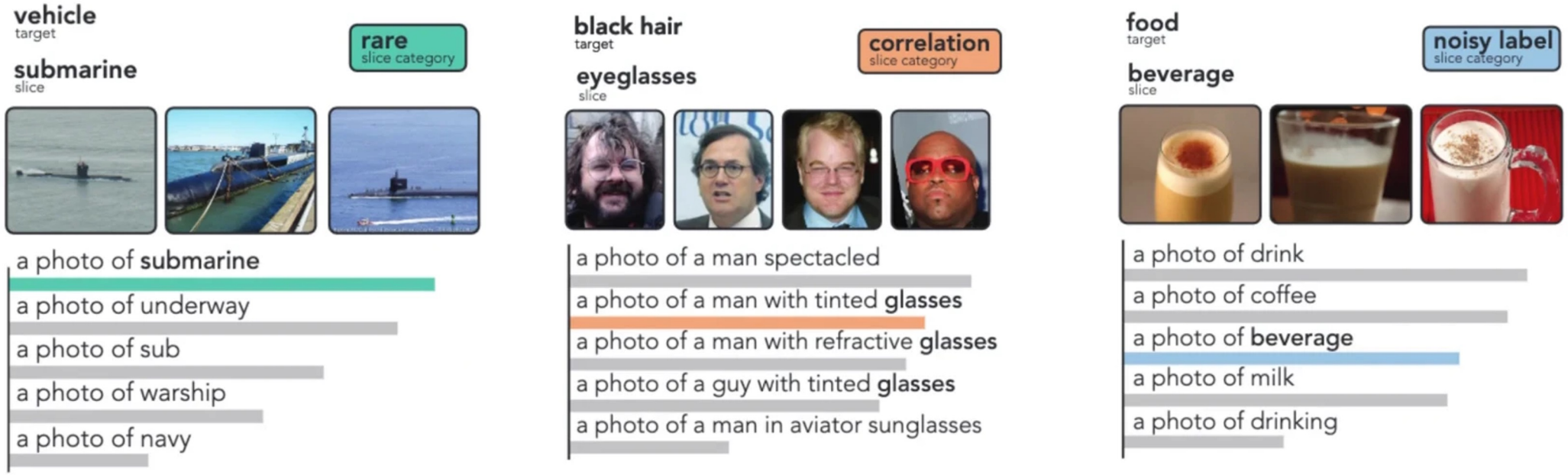
Learns the parameters of the following generative process:

1. Each input example (image) is i.i.d. assigned to a slice.
2. Conditional on the slice:
   a. Its embedding is sampled from Gaussian distribution.
   b. Its ground truth label and ML prediction are independently generated among all classes.

Domino reports slices on which the L1 difference between the learned labelling and prediction probabilities are largest.

Users can tune the weights of 2a and 2b by a hyperparameter γ.

Domino: Describing



The idea is to find texts whose cross-model embedding "best explains" the difference between slice average and class average (food).

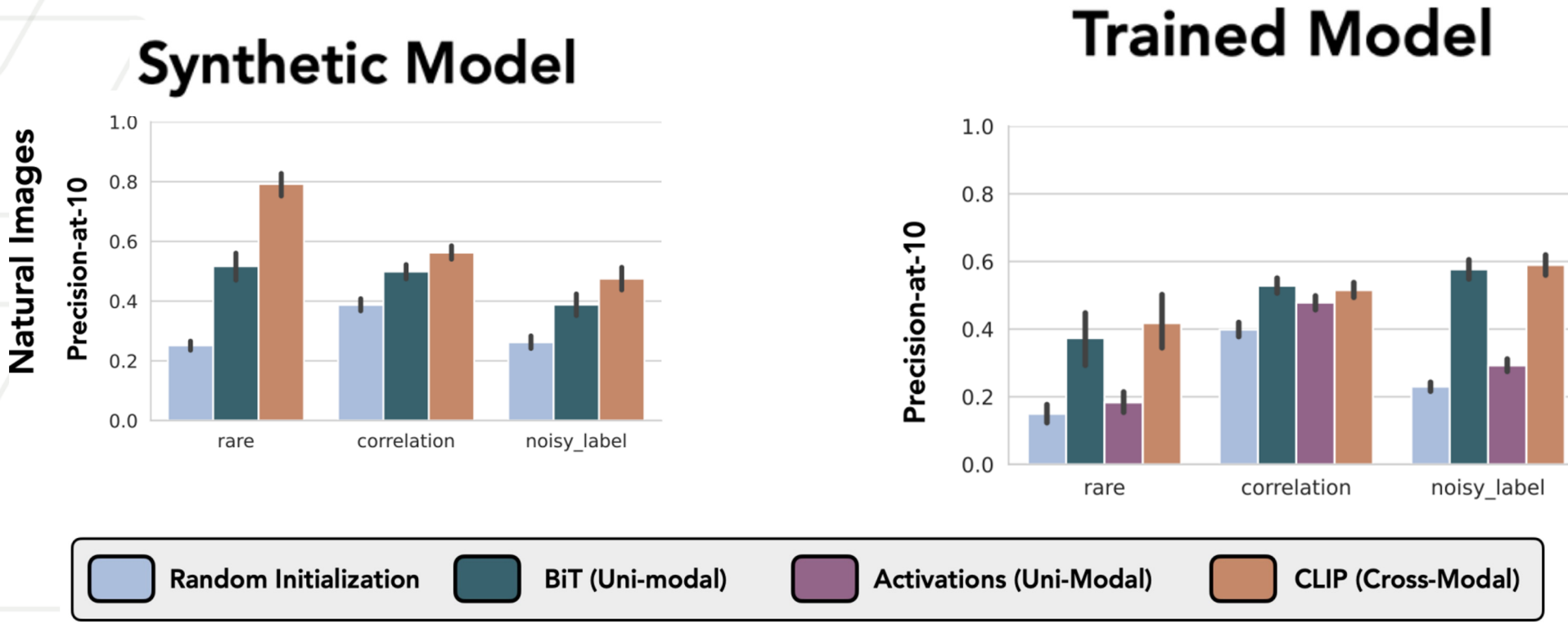
Georgia Tech

Evaluation

- Domino is evaluated on 1235 (trained) SDM settings described before.
- Evaluation Outline:

1. The use of cross-model embeddings.
2. The use of error-aware mixture models.
3. The accuracy of describing found slices.

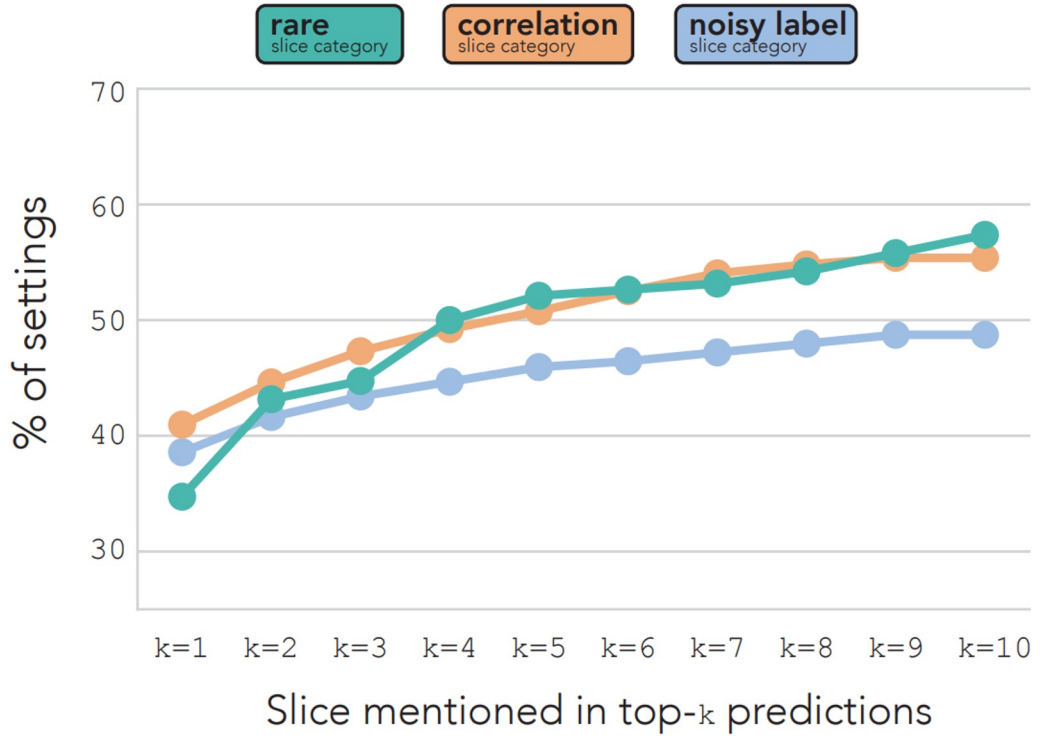
Evaluation: Embedding



Cross-model (images and texts) embeddings outperform uni-model ones on both model types.

Georgia Tech

Evaluation: Describing



In "rare" and "correlation" settings, most settings are explained by first 5 explanations.

Georgia Tech

Conclusion

- We observe the limitations of prior SDM evaluations, and propose a new evaluation framework of two axes: coherence and underperformance.
- We propose Domino, which outperforms existing solutions thanks to cross-model embedding and error-aware mixture models. Furthermore, Domino is the first work that automatically generates slice descriptions.
- Domino only needs black-box access to models.
- One future work is study on how Domino helps users avoid underperforming slices.

Thanks!

Georgia Tech

Domino: Discovering Systematic Errors with Cross-Modal Embeddings

Eyuboglu et. al.

Archaeologist role : Sankalp

Georgia Tech

Summarizing…

- Domino paper contributed two things
  - *Quantitative*, *programmable* evaluation framework for SDMs
  - A new SDM leveraging cross-modal embeddings (Domino)

- Important novelty of SDM approach: describes in words, the “common concept” in a slice

- Three steps: Embed, Slice, Describe

- Clustering using error-aware GMM:
  Objective function incorporates not just feature vector, but also true label and model prediction

- To generate natural language descriptions for a slice, generates a candidate set of phrases from a template like “an image of {object}” using language models like BERT. Transform them in the joined representation space and pick the one with maximum cosine similarity to the representative slice vector.

Georgia Tech

Inspiration for cross-modal embeddings

Motivated by the recent development of large cross-modal representation learning approaches (*e.g.* CLIP) that embed inputs and text in the same latent representation space, in Section 4 we present *Domino*, a novel SDM that uses cross-modal embeddings to identify coherent slices. Cross-modal

Domino uses four types of cross-modal embeddings to enable slice discovery across our input domains: CLIP (Radford et al., 2021), ConVIRT (Zhang et al., 2020), MIMIC-CLIP, and EEG-CLIP.

Large-scale pretrained cross-modal embedding functions can be used to generate accurate representations of input examples. For instance, if our inference dataset consists of natural images, we can use a pre-trained CLIP model as embedding functions g_{input} and g_{text} to obtain image embeddings that lie in the same latent representation space as word embeddings.

Georgia Tech

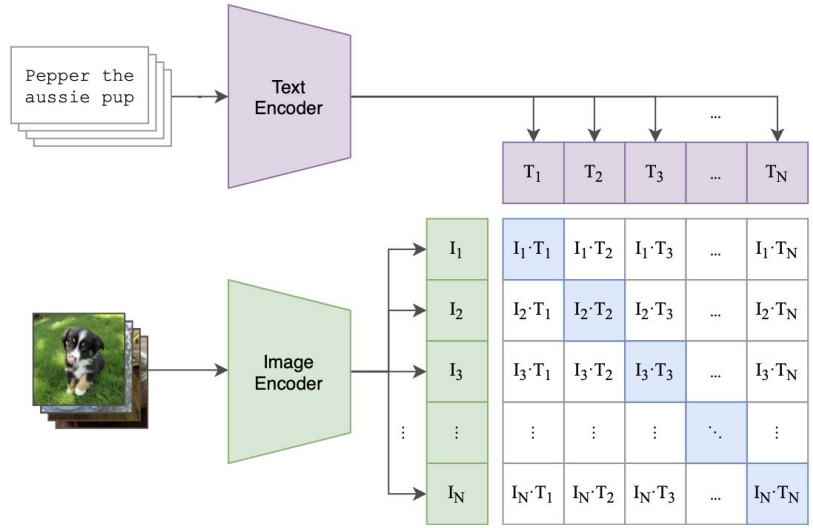
Lets talk about CLIP

- Paper called “Learning Transferable Visual Models From Natural Language Supervision”, by Radford et. al. (ICML 2021)
- CLIP (Contrastive Language-Image Pre-training) is the model built by the paper

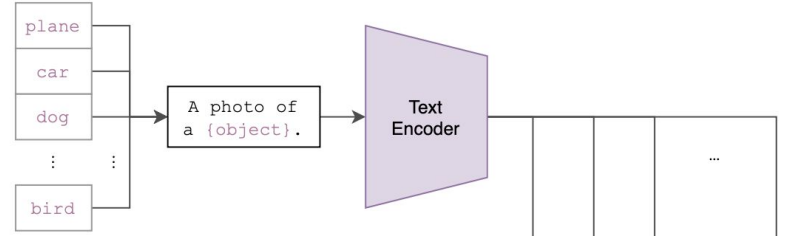
- Motivation? Problem of Image classification
  - SOTA CV systems predict a *fixed* set of *predetermined* object categories
  - They do not generalize well either
  - CLIP instead uses Contrastive Learning instead of traditional supervised learning
  - Approach isn’t new, but none of the previous approaches have done it at CLIPs scale
- CLIP is trained on 400 million (image, caption) pairs collected from the internet
  - Note: NOT (image, label) pairs but (image, caption) pairs

Approach

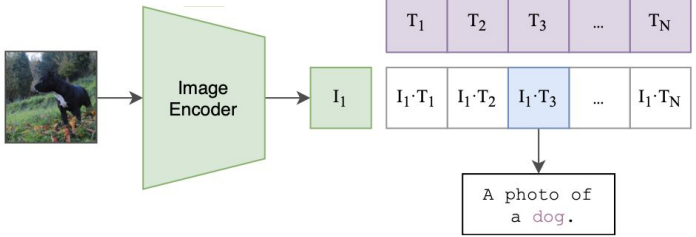
(1) Contrastive pre-training



(2) Create dataset classifier from label text

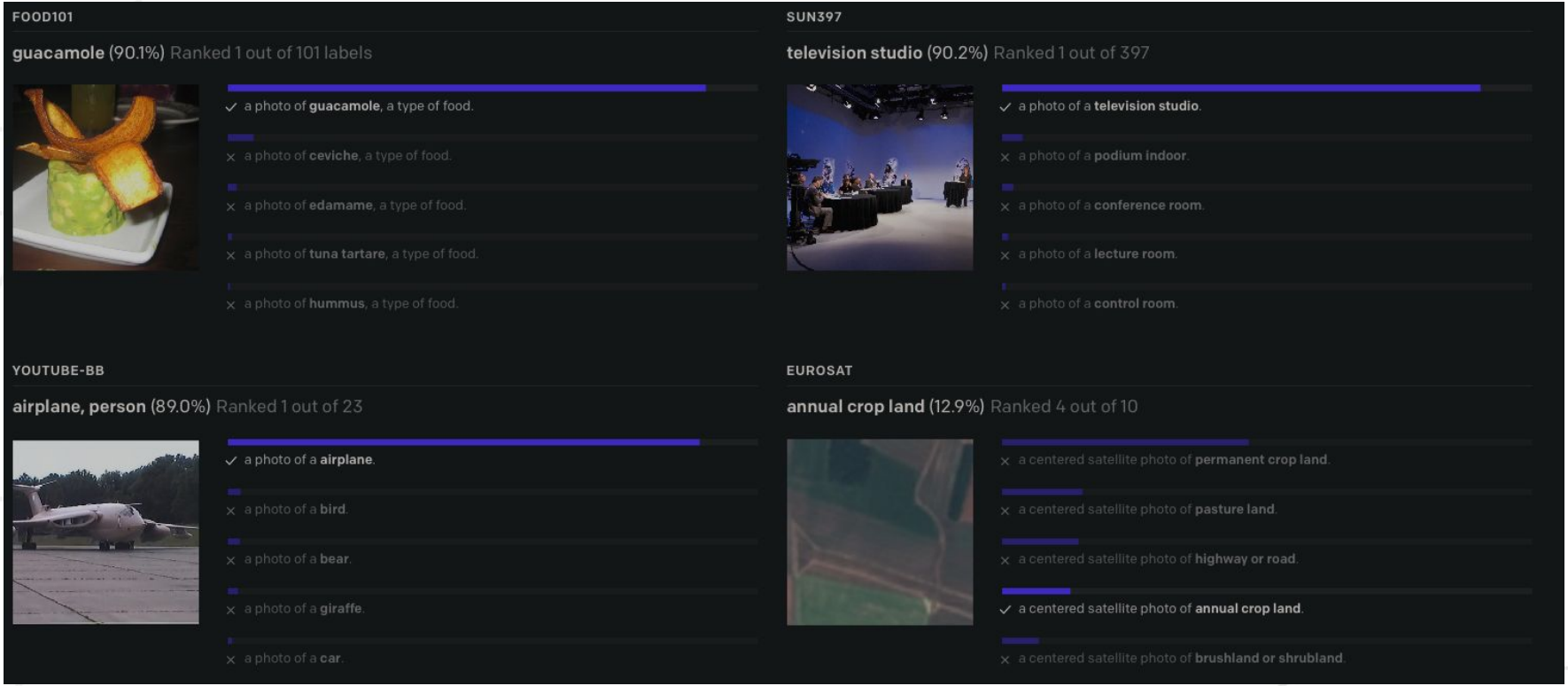


(3) Use for zero-shot prediction



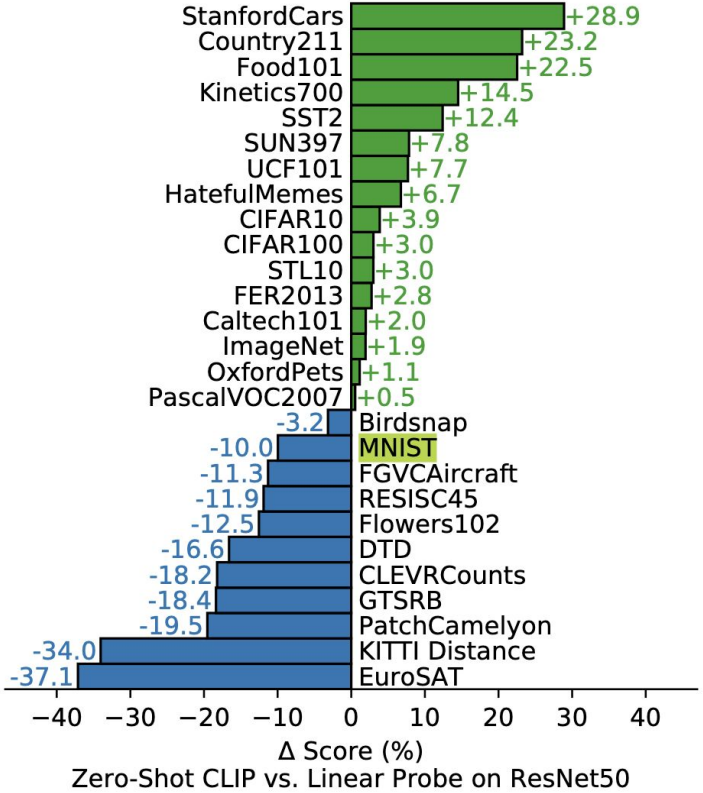
Georgia Tech

Zero-shot learning performance of CLIP



Georgia Tech

Results on zero-shot learning



- Comparison of CLIP versus off-the-shelf baseline: a fully supervised, logistic regression classifier on features of ResNet50.
  Authors suggest that CLIP is performing well on datasets that are more varied, or have limited number of labeled examples.
  It doesn't do well on several specialized datasets / tasks.

Georgia Tech

Some limitations

The performance of zero-shot CLIP is often just competitive with the supervised baseline of a linear classifier on ResNet-50 features. This baseline is now well below the overall SOTA. Significant work is still needed to improve the task learning and transfer capabilities of CLIP. We estimate around a 1000x increase in compute is required for zero-shot CLIP to reach overall SOTA performance across our evaluation suite. This is infeasible to train with current hardware. Further research into improving upon the computational and data efficiency of CLIP will be necessary.

Zero-shot CLIP is competitive with ResNet.
But ResNet itself isn't SOTA, and to get to SOTA performance, authors estimate 1000x more compute

- CLIP is generally flexible enough to generate zero-shot classifiers for a variety of tasks. But to be noted is that it is a caption ranker, and not a caption generator.

- CLIP also seems to have only 88% accuracy on the MNIST dataset (a classic dataset of images of handwritten digits). Probably because not too many images on the internet of handwriting of digits.

Georgia Tech

Some interesting applications

lab member 001
@l4rz

filtering out noisy photos with CLIP

```
import torch
import clip
from PIL import Image
from pathlib import Path
import sys, os

device = "cuda" if torch.cuda.is_available() else "cpu"
model, preprocess = clip.load("ViT-B/32", device=device)
filelist = sorted(Path(sys.argv[1]).rglob('*.jpg'))
for file in filelist:
    file = str(file)
    image = preprocess(Image.open(file)).unsqueeze(0).to(device)
    text3 = clip.tokenize(["grainy, noisy photo", "high quality photo"]).to(device)

    with torch.no_grad():
        image_features = model.encode_image(image)
        text3_features = model.encode_text(text3)
        logits_per_image, logits_per_text = model(image, text3)
        probs3 = logits_per_image.softmax(dim=-1).cpu().numpy()

        print(file, "probs3:", probs3)

        if probs3[0][0] > 0.8:
            # delete
```

9:51 AM · 22/01/21 · Twitter Web App

3 Retweets **1** Quote Tweet **49** Likes

NightCafe Studio @NightcafeStudio · 18/01/22
New text-to-image algorithm coming soon to NightCafe - "Coherent" (clip-guided diffusion under the hood) gives slightly less artistic, but much better composed images. Here are a few samples.

#aigeneratedart #aiart #aiartists #aiartcommunity #diffusion #clipguideddiffusion

9 10 48

I like butter Vegemite, honey, and cheese on toast @... · 19/01/22
How would you describe clip guided diffusion?

Asking for a dumb friend (me)

1 1

NightCafe Studio
@NightcafeStudio

Replying to @DannyDangerOz

At the risk of over-simplifying, instead of using a GAN to generate images, diffusion starts with a noisy image and gradually refines it. Both methods use CLIP to guide the algorithm towards an image that matches the text prompt.

7:40 PM · 19/01/22 · Twitter Web App

1 Like

Thank you!

Georgia Tech

Biases in CLIP

Class designs have the potential to be a key factor determining both the model performance and the unwanted biases the model may exhibit

Experiment: adding biased probes/classes to the FairFace dataset

- non-human classes: animal, gorilla, chimpanzee, orangutan
- crime-related classes: thief, criminal, suspicious person

Biases in CLIP

Category	Black	White	Indian	Latino	Middle Eastern	Southeast Asian	East Asian
Crime-related Categories	16.4	24.9	24.4	10.8	19.7	4.4	1.3
Non-human Categories	14.4	5.5	7.6	3.7	2.0	1.9	0.0

Table 6. Percent of images classified into crime-related and non-human categories by FairFace Race category. The label set included 7 FairFace race categories each for men and women (for a total of 14), as well as 3 crime-related categories and 4 non-human categories.

Significant disparities in misclassification rates across races (and also gender)

Biases in CLIP

Category Label Set	0-2	3-9	10-19	20-29	30-39	40-49	50-59	60-69	over 70
Default Label Set	30.3	35.0	29.5	16.3	13.9	18.5	19.1	16.2	10.4
Default Label Set + 'child' category	2.3	4.3	14.7	15.0	13.4	18.2	18.6	15.5	9.4

Table 7. Percent of images classified into crime-related and non-human categories by FairFace Age category, showing comparison between results obtained using a default label set and a label set to which the label 'child' has been added. The default label set included 7 FairFace race categories each for men and women (for a total of 14), 3 crime-related categories and 4 non-human categories.

Adding a "child" class drastically reduced the number of images of people under 20 classified in either crime-related categories or non-human animal categories

Domino: Discovering Systematic Errors with Cross-Model Embeddings

Summary

- Focuses on qualitative and quantitative evaluation of Automated Slice Discovery Method (SDMs)
- SDM: Mine input data for slices on which a model performs poorly

- **Problem** - SDMs produce slices of data that aren't grouped coherently
- Contributions:
  - Evaluation framework for SDM for Natural Images, Medical Images and Time-series data
  - Domino: SDM that leverages Cross-modal embeddings to discover and describe coherent slices

Expert Domino

- Domino uses embeddings trained on image-text pairs sourced from web

Research Idea:

- Improve Domino by using embeddings specific from the domain (medical images)
- Seek domain experts for annotations instead of just templates

Align-Domino

- Domino generated a corpus of Natural Language Descriptions
- Align (Jia et al., 2021) - A large scale Image and Noisy text embedding
  - It uses contrastive learning on text and image encoders
  - Pushes matched image-text pairs together and non-matching apart
  - Top-1 accuracy of 78% and 97.4% for top-10

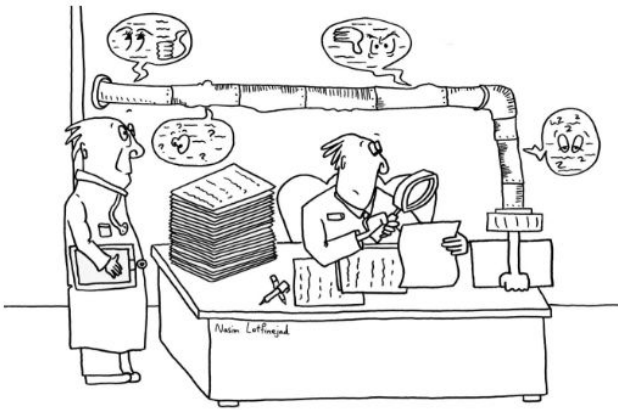
Research Idea:

- Merge Align and Domino to generate descriptions of slices
- Use similarity score on those descriptions to group slices coherently

Jia, Chao & Yang, Yinfei & Xia, Ye & Chen, Yi-Ting & Parekh, Zarana & Pham, Hieu & Le, Quoc & Sung, Yun-Hsuan & Li, Zhen & Duerig, Tom. (2021). Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision.

Domino: Discovering Systematic Errors With Cross-Modal Embeddings

Peer Review



Reviewer: Tanya Garg

Summary

- ➔ Introduces a new slice discovery evaluation framework which quantitatively analyses a slice discovery method which validates the performance of a model built to identify coherent problematic slices.

- ➔ Introduces Domino: A new slice discovery method that uses cross-modal embedding (input-text paired samples) and an error-aware mixture model to discover and describe coherent problematic slices.

- ➔ Domino uses three basic steps : embed, slice and describe to find the top k underperforming slices. The performance was later evaluated using the discussed framework on 1235 settings.

Strong Points

- ➔ First Slice Discovery Method that uses embedded text and input to generate natural language descriptions for the identified under-performing slices.
- ➔ Built in a way that the tool only requires black-box access to to models and can thus be broadly useful in settings where users have API access to models.
- ➔ One of its kind programmable framework to give a quantitative measure of the performance of an SDM based on coherence and underperformance.
- ➔ Method proven to perform better than other SDMs on 1235 SDM settings.
- ➔ Open-source code

Weak Points

- ➔ No evaluation done of the SDM evaluation framework which is used extensively later on.
- ➔ Evaluation of Domino done on an in-house evaluation framework whose credibility is itself not proven.
- ➔ No user study or input from real-life practitioners to understand their needs and making the tool more convenient.

Weak Accept!

Domino: Discovering systematic errors with cross-modal embeddings

Reviewer: Abhi

Summary

1. A large scale method of evaluating SDMs.
2. Domino: A new SDM that uses cross-modal embeddings to identify slices and provide natural language explanations. It outperforms previous methods on the newly proposed metric (1).

Strengths

1. Novelty
    a. This is the first time a quantitative method of evaluation has been proposed for SDMs. Prior methods of evaluation were all hand-wavy.
    b. Using cross modal representations for slice discovery is a new idea. This made the slices more coherent for humans.
2. Well written!
3. Nicely motivated- chest drains in X-rays, melanoma detection
4. Open source! Pip install domino

Weaknesses(?)

1. Domino generates textual descriptions of slices. Are these really useful?
2. Is the classification of slices (rare, correlation and noisy) comprehensive? Are there slices that don’t fall into these categories?

Overall Verdict

ACCEPT

Discussion

- How did Domino and Slice Finder evaluate the accuracy of their slice finding algorithms?
- How did Domino and Slice Finder address the interpretability of the slices?
- How to better involve users in these systems?
- Anything else?

Next class

Project update (10/31)

https://bit.ly/3gBLCPz