

Yale

Introduction to Trustworthy AI

CPSC 471/571: Trustworthy Deep Learning

Rex Ying

CPSC 471 / 571 Course Logistics

- **Welcome to CPSC 471 / 571**
- **The class meets Monday and Wednesday 2:30 PM-3:45 PM**
 - This is still a **seminar-like, graduate-level** course
 - Students are expected to master the background of deep learning, and are expected to have the ability to explore cutting-edge research topics on their own
 - Most of the lectures will be in-person
 - There will be a few guest lectures, some of which might be held remotely.
- **This is a relatively new course, and feedbacks are especially welcomed!**
 - It's also a course that can benefit a lot from discussions

Course Outline

- **Introduction** (3 lectures)
 - Trustworthy AI intro; preliminaries of deep neural networks
- **Explainable AI** (3-4 lectures)
 - Model interpretability and explainability; evaluation of explainable models
- **Adversarial Robustness** (4 lectures)
 - Evasion attacks; poisoning attacks; defense against adversarial attacks; verification
- **Fairness** (2 lectures)
 - Biases in deep learning models; fairness algorithms in deep learning models
- **Privacy** (3 lectures)
 - Privacy attacks; differentially private deep models; federated learning; unlearning
- **Efficient Deep Learning** (2-3 lectures)
 - Model pruning; sparse transformers; efficient LLM; Quantization
- **Trustworthy LLM** (1-2 lectures)
 - Hallucination; retrieval-augmented generation (RAG); various aspects of trustworthy AI in LLMs

Logistics: Canvas

- **Check Canvas often for course materials and communications**
 - Slides will be posted before or shortly after the class
- **Readings:**
 - Pre-reading for next lecture will be announced at the end of the previous class
- **Optional readings:**
 - Papers and pointers to additional literature (suggested on slides)
 - **These will be very useful for course projects**
- **Student presentations:**
 - Presenter(s) will post the slides on canvas after the presentation

Logistics: Communication

- **Canvas has a discussion panel:**
 - Please participate and help each other!
 - It's also a great way to show engagement and understanding in case you missed classes or didn't join discussions in class
 - Search for answers before you ask
- **Mailing list:**
 - Teaching staff: cpsc471_staff@googlegroups.com
 - Send email to the mailing list if you have requests or questions (not individual instructor / TA)
 - Only email the instructor if the message is highly private / not related to teaching in general
- **Office Hours:**
 - Instructor: Monday 1:30 – 2:30 pm
 - TAs: see canvas announcements

Work for the Course & Grading (1)

- **The final grade will be composed of the course project, discussions, exams**
- **Course project: 50% (Code and Report to be submitted on Canvas)**
 - In-class discussion session: 5%
 - Proposal: 5%
 - In-class “Hackathon”: 5%
 - Milestone: 15%
 - Final report: 20%
 - Consistent work is valued the most

Work for the Course & Grading (2)

- **Final grade will be composed of course project, discussions, exams**
- **4 Assignments**
 - Written assignment 1 – Explainability
 - Coding assignment 1 – Explainability
 - Written assignment 2 – Adversarial Robustness
 - Coding assignment 2 – Adversarial Robustness

Work for the Course & Grading (3)

- **In-class Exam (last lecture)**
- **Will potentially cover all content taught in this class**
 - Close-book but we will provide all necessary knowledge in the exam
 - No memorization of complex equations needed
 - I will highlight the parts that are important to prepare
 - We do assume all pre-requisites (multi-variable calculus, linear algebra) in the exam.
They should be the second instinct for an ML student imo

Honor Code

- **We strictly enforce the Yale Honor Code**
 - Violations of the Honor Code include:
 - Copying or allowing another to copy from one's own paper
 - Unpermitted collaboration
 - Plagiarism
 - Giving or receiving unpermitted aid on an examination
 - Representing as one's own work the work of another
 - Giving or receiving aid on an assignment under circumstances in which a reasonable person should have known that such aid was not permitted
 - The sanction for even a first offense is severe

Course Projects

- **Course project:**
 - Development of methods related to trustworthy AI and perform benchmarking
 - Propose new trustworthy DL models and validate on non-trivial datasets
- **Performed in groups of up to 3 students:**
 - Fine to have groups of 1 or 2. The team size will be taken under consideration when evaluating the scope of the project in breadth and depth.
 - Project is the **important work** for the class
 - Can be very beneficial if you aim to work on research in this topic
 - Graduate and undergraduate students will have the same criteria
 - **Even for group projects each student needs to submit their own separate method and experiment sections**
- **More information will be posted on Canvas in 3 weeks**

Student Participation

- Trustworthy AI is both about people and machine learning
- **Participation** (5% of overall grade) is highly encouraged
- I will likely also pick different students to ask questions or give answers each lecture
- The two in-class discussion and work sessions are mandatory

Course Schedule

Week	Milestones	Due on (11:59pm ET)
3	Finalize project groups	Fri, Feb 2nd
5	In-class project discussions	Wed, Feb 14th
7	Project proposal	Fri, March 1st
8	In-class work session	Wed, March 6th
12	Project milestone	Fri, April 5th
15	Exam	Wed, April 24th
	Project Report	Fri, May 5th 11:59pm

Prerequisites

- **The course has relevance to a wide range of topics and background in DL is needed!**
- Trustworthy DL for different architectures will all be in scope of the class
 - However, although beneficial, students are not required to understand all model architectures
 - **The course will give a brief overview of common model architectures**
- **Minimum Pre-requisites**
CPSC 201, 223 and one of S&DS 265a, 381/581 or 452
 - **Familiarity with Linear algebra**
 - **Familiarity with Multi-variable calculus**

Machine Learning Tools

- PyTorch
- There are many libraries for specific architectures
 - [Hugging Face](#) is a good place to check out the tools
 - Most papers have associated codebase on [GitHub](#) so check out what tools / libraries that the researchers use
(in fact, I do not recommend presenting a paper that does not have open-source implementation)
- Computational resource from [YCRC](#) can be utilized
 - I will follow up with more information about computation resources after the proposals are submitted

Yale

Introduction to Trustworthy AI

CPSC 471/571: Trustworthy Deep Learning

Rex Ying

Readings

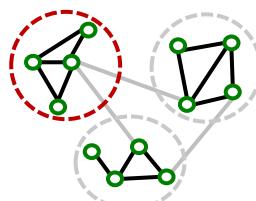
- Readings are updated on the website (syllabus page)
- **Lecture 1 readings:** [AI Sustainability](#)

What Deep Learning Looks Like

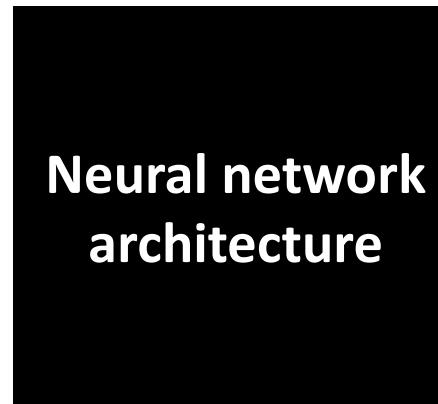
Why do we need Trustworthy AI?

Input

Sentence: [SOS, “graph”,
“neural”, “networks”, “are”,
“powerful”, EOS]



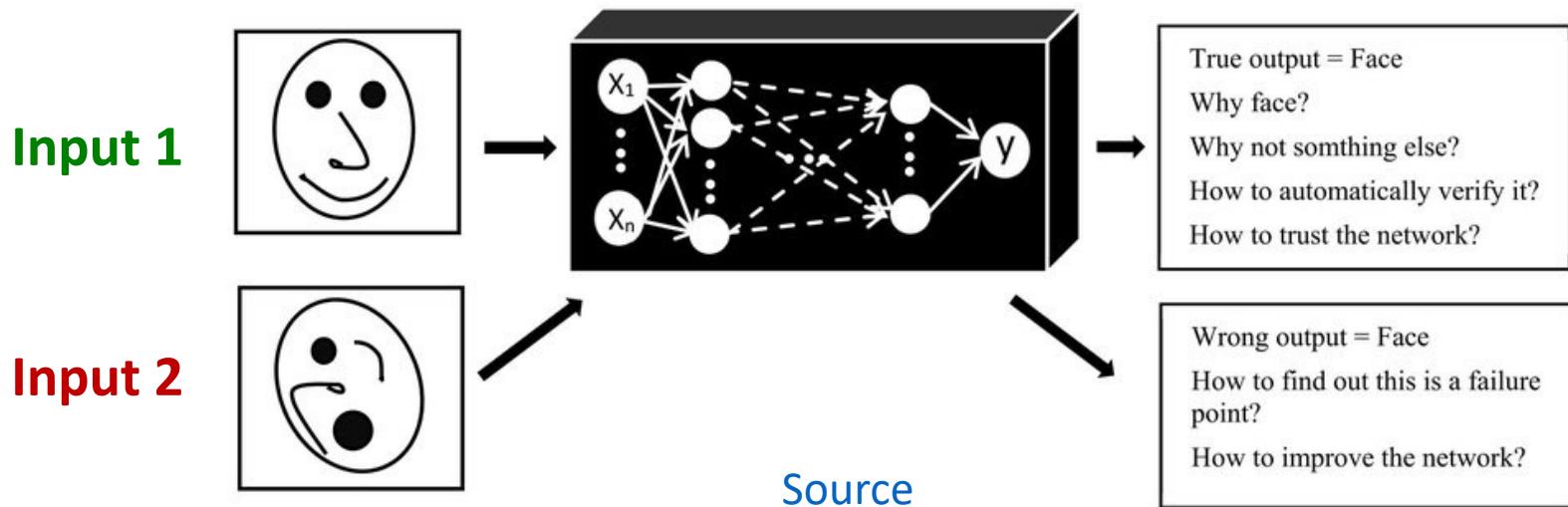
None / Noise



**Labels,
generated content,
agent decision
etc.**

What People Want

- Model debugging
- Phenomenon understanding
- Robustness and consistency
- Efficiency and human-like behavior



Trustworthy Graph Learning

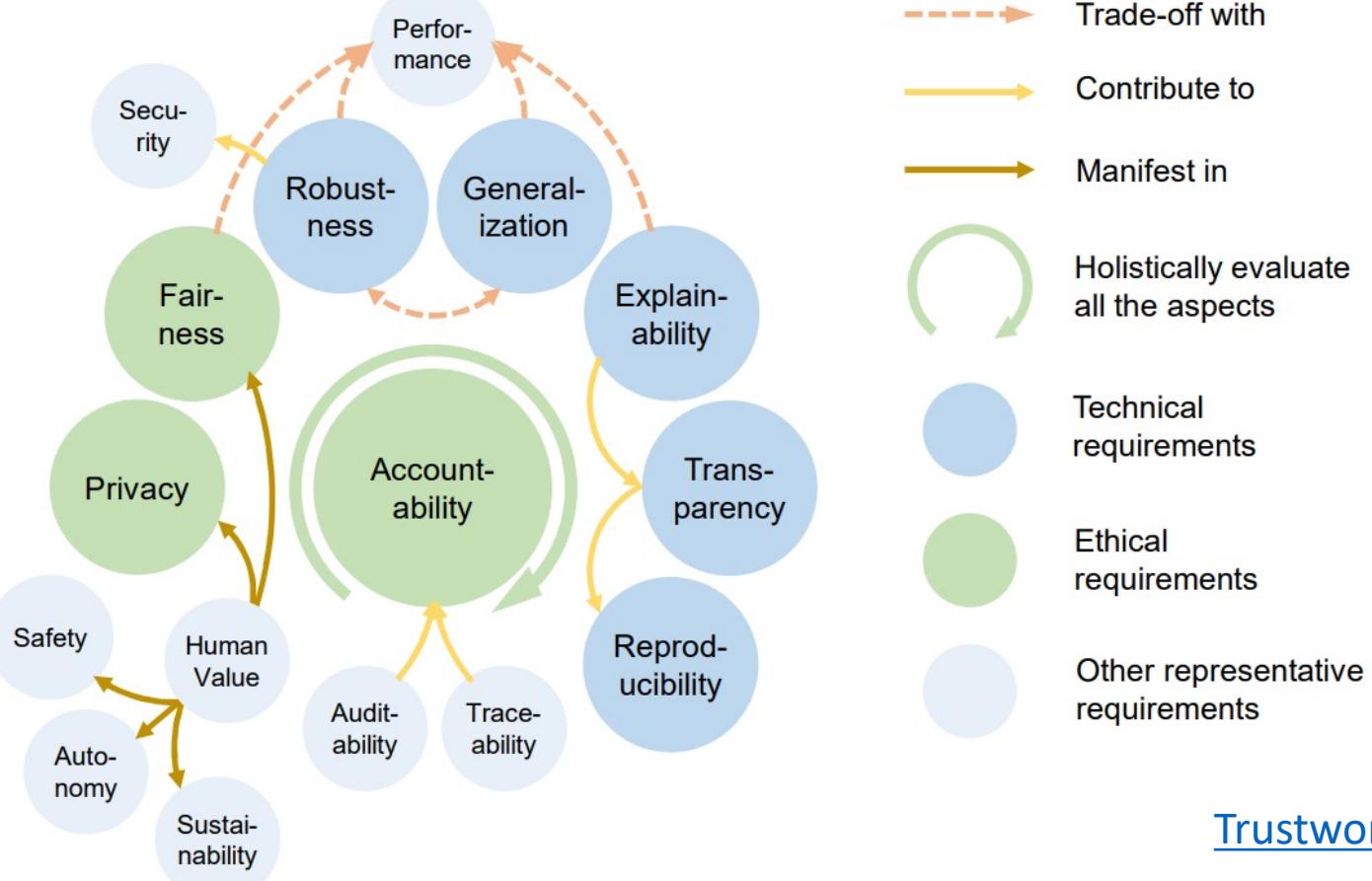
Trustworthy AI includes many components

- Explainability, fairness, robustness, privacy, efficiency ...
- The goal is to develop algorithms to tackle one or a combination of these aspects

Challenges

- Deep learning is typically regarded as blackbox
 - High dimensionality, multimodal data, larger and larger parameter space
- Trustworthiness is defined by human, and in particular, domain experts in many applications
- Sometimes performance and runtime tradeoffs seem inevitable

Trustworthy DL Topics



[Trustworthy AI: From Principles to Practices](#)

Aspects of Trustworthy Deep Learning

- **Robustness** (often against adversarial attacks)
- **Explainability**
- Privacy
- Fairness
- Efficiency / Environmental well-being
- Others

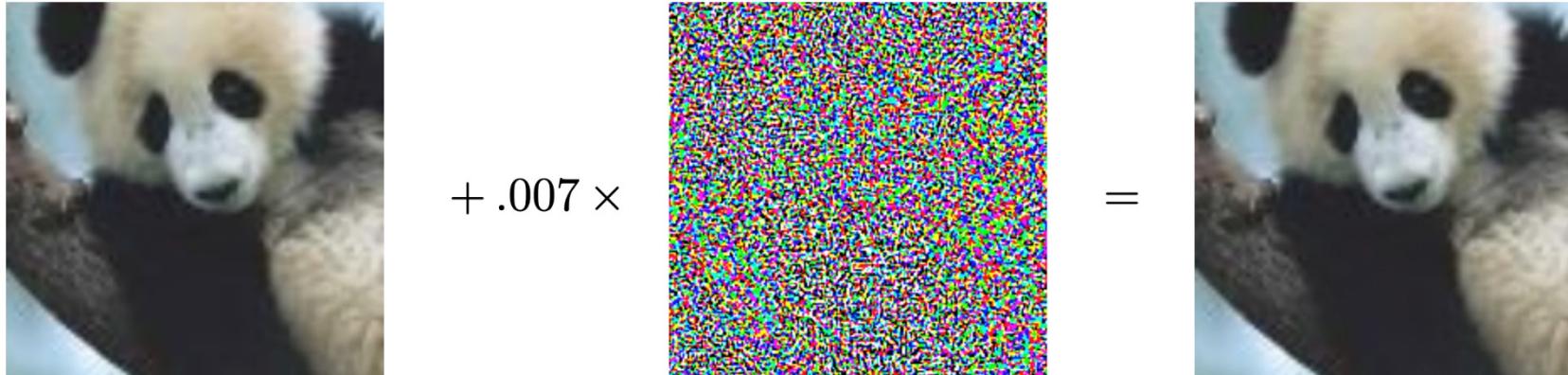
How does each aspect play a role
in gaining trust from users of
machine learning models?

Deep Learning Performance

- Recent years have seen **impressive performance of deep learning models in a variety of applications.**
 - Deep generative models (e.g. diffusion models)
 - AlphaFold
 - Language models
 - Multi-modal models
- **Are these models ready to be deployed in real world?**

Adversarial Examples

- Deep convolutional neural networks are vulnerable to **adversarial attacks**:
 - Imperceptible noise changes the prediction.

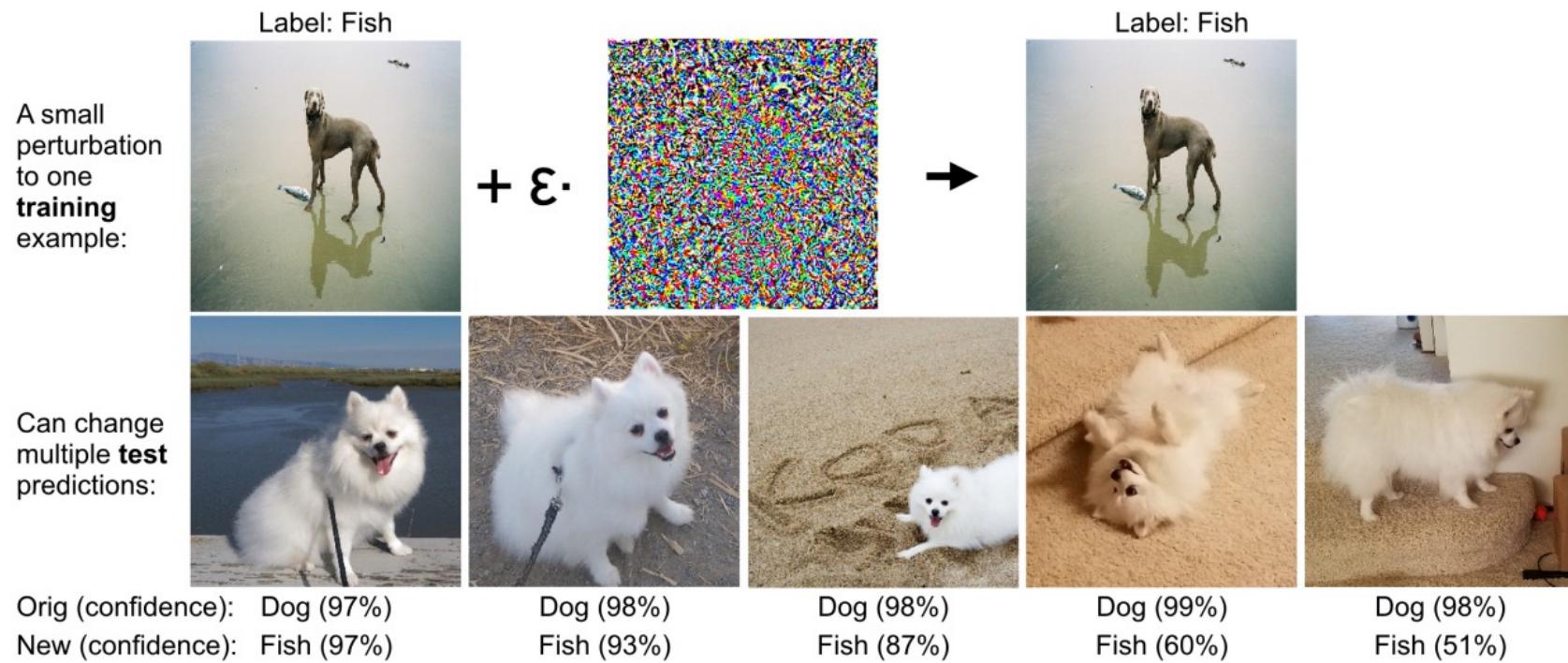
$$\text{“Panda”} \quad 57.7\% \text{ confidence} + .007 \times \text{Carefully calculated noise} = \text{“Gibbon”} \quad 93.3\% \text{ confidence}$$


Goodfellow, I., Shlens, J., & Szegedy, C.. (2014). Explaining and Harnessing Adversarial Examples.

- Adversarial examples are also reported in natural language processing [Jia & Liang et al. EMNLP 2017] and audio processing [Carlini et al. 2018] domains.

Adversarial Examples

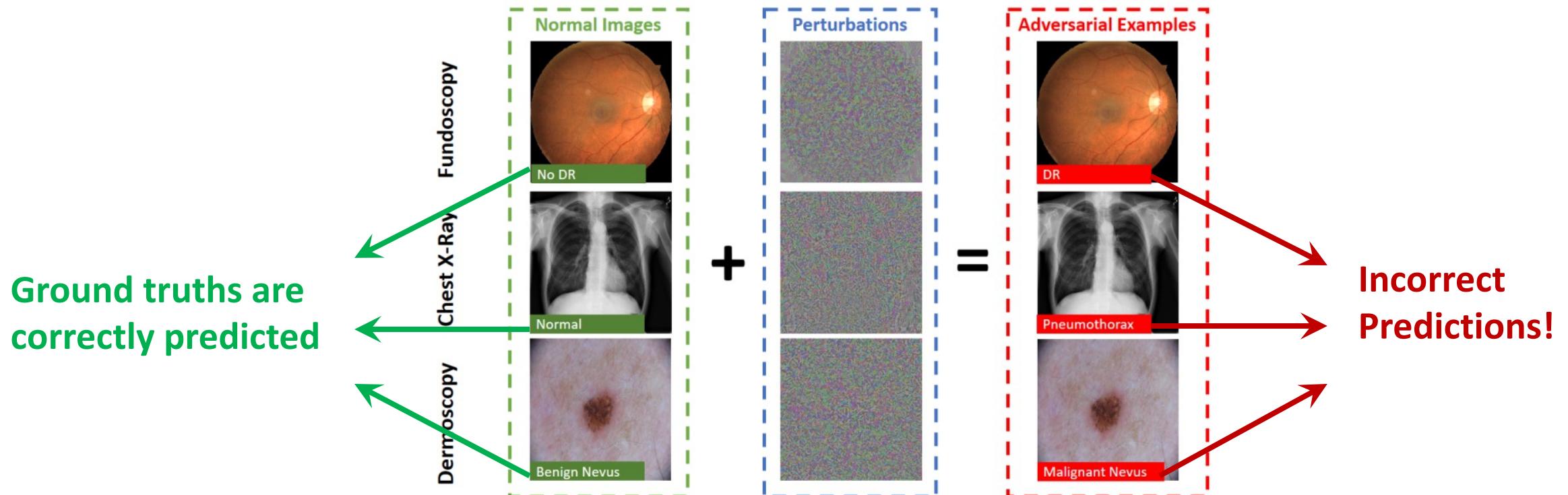
- Adversarial examples are not special cases: they are ubiquitous in deep learning models!



[Source](#)

Adversarial Examples – Medical Applications

- Adversarial attacks crafted by the Projected Gradient Descent (PGD)
- On medical dataset Fundoscopy, Chest X-Ray, Dermoscopy



Implication of Adversarial Examples

- **The existence of adversarial examples prevents the reliable deployment of deep learning models to the real world.**
 - Adversaries may try to actively hack the deep learning models.
 - The model performance can become much worse than we expect.
- **Deep learning models are often not robust.**
 - It is an active area of research to make these models robust against adversarial examples

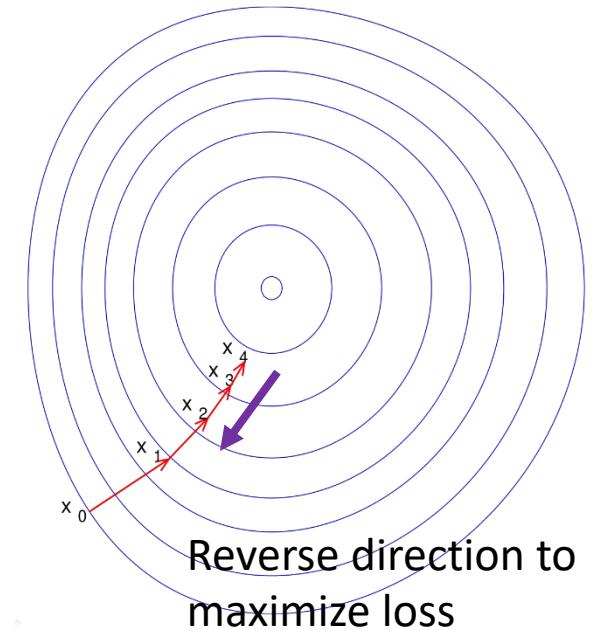
Types of Attacks

- **Whitebox Attack:**

- The attacker has access to model architecture and weights
- Easier to attack
- Gradient-based methods are straightforward and effective

- **Blackbox Attack:**

- The attacker does not have access to the model's parameters
- The type of architecture might be known
- A different model or no model is used to generate adversarial examples in the hopes that these will transfer to the target model



Types of Attacks

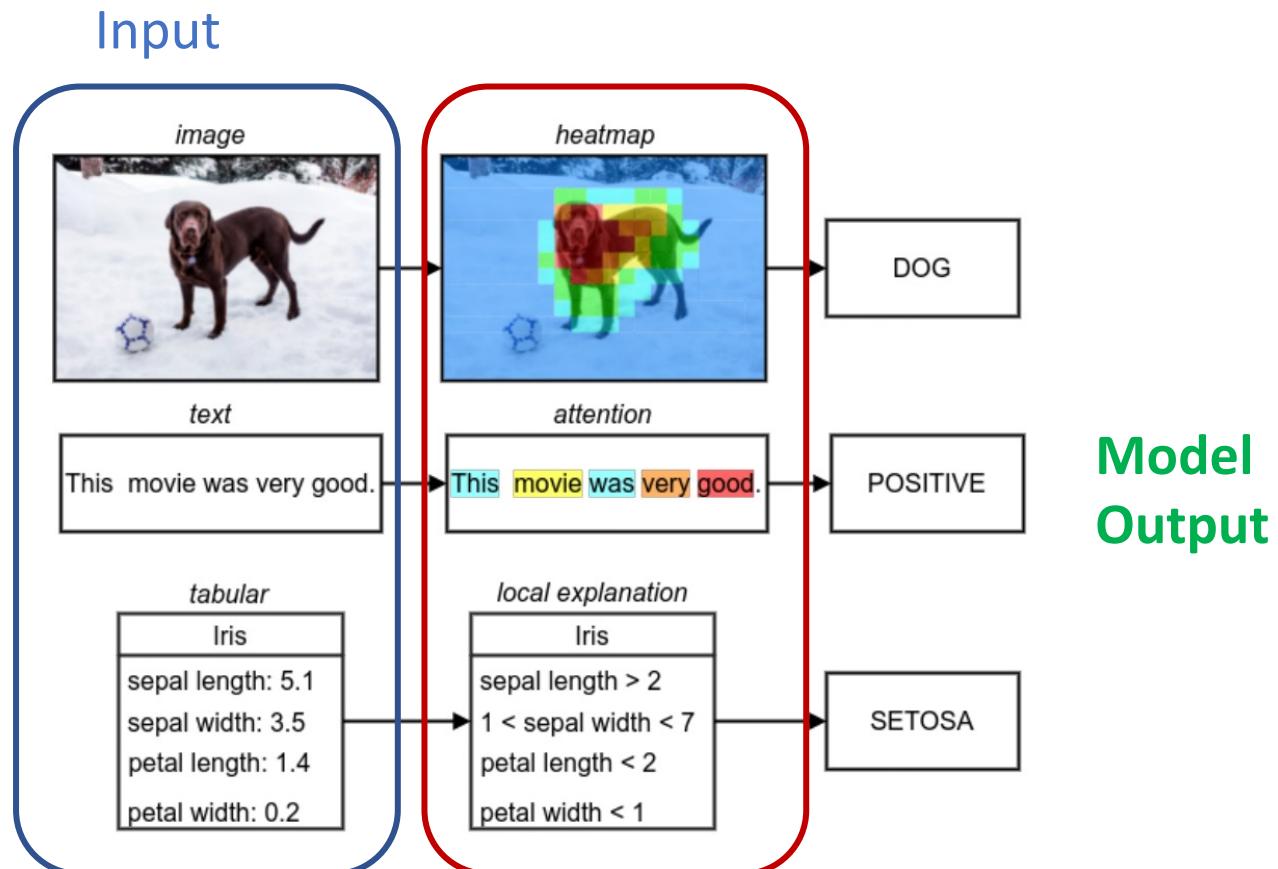
- **Data Poisoning**
 - Contamination of training data
- **Byzantine Attacks**
 - Commonly occur in federated learning setting
 - One or multiple client / edge device attacking against the federated learning system
- **Evasion**
 - Evasion of detection models (face/person detection; anomaly detection; fraud detection, spam detection ...)
- **Model Extraction**
 - Gain understanding of model architecture and model weights

Explainability

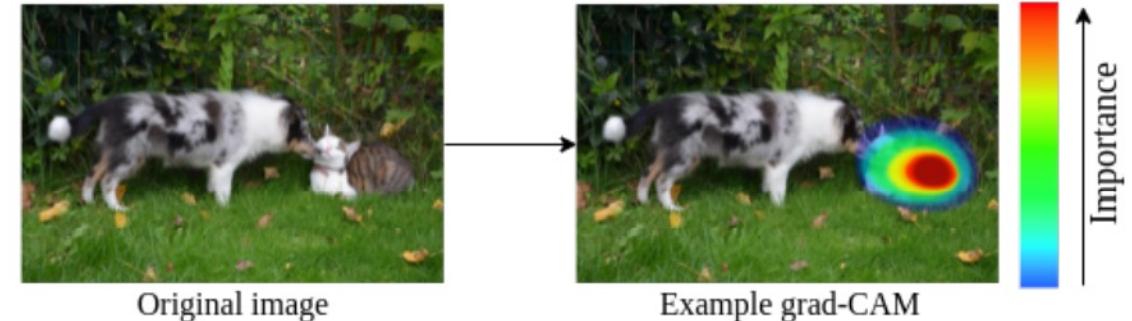
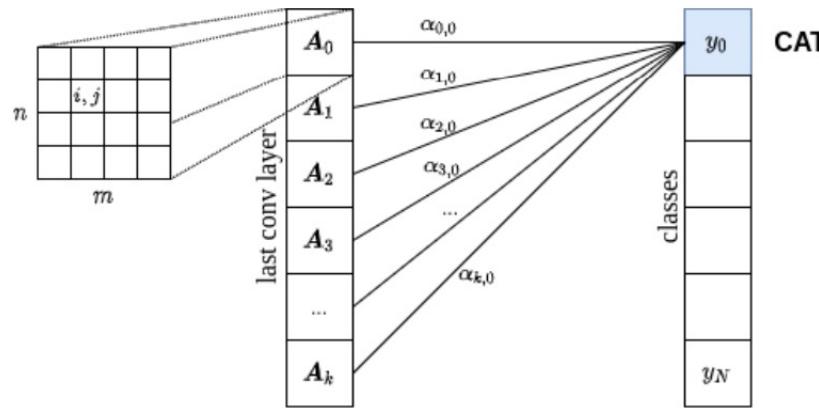
- The goal is to explain what is learned by the model to users / domain experts, in order to gain trust from human users of the deep learning system
- The blackbox nature of deep learning is a major challenge
- Simple-to-read guide: [2004.14545.pdf \(arxiv.org\)](https://arxiv.org/pdf/2004.14545.pdf)

**What was explainable about
previous ML models??**

Forms of Explanation



Example: Computer Vision



**Importance
scores on pixels**

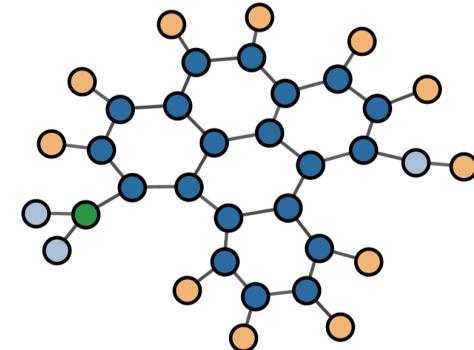
Explanation: A particular region of the image displays a cat

Example: Graphs

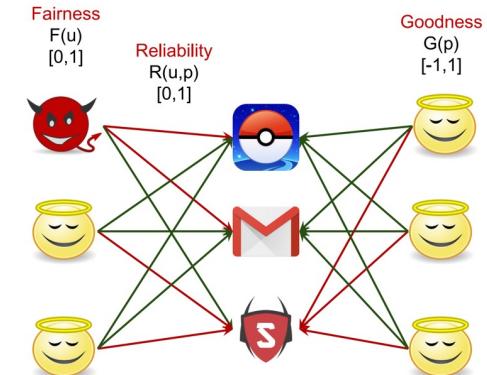
- Example questions after training GNNs:
 - Why is an item recommended to a user?
 - Why is the molecule mutagenic?
 - Why is the user classified as fraudulent?
- How to let the domain experts understand and trust the GNN model?



Recommender System



Mutagenic Molecule



Fraudulent Detection

Explainability: GNNs

- Example questions after training GNNs:
 - Why is an item recommended to a user?
Explain link prediction
 - Why is the molecule mutagenic?
Explain graph classification
 - Why is the user classified as fraudulent?
Explain node classification
- Need to provide explanations to GNN models!

Explainable Models (1)

- **Linear regression**

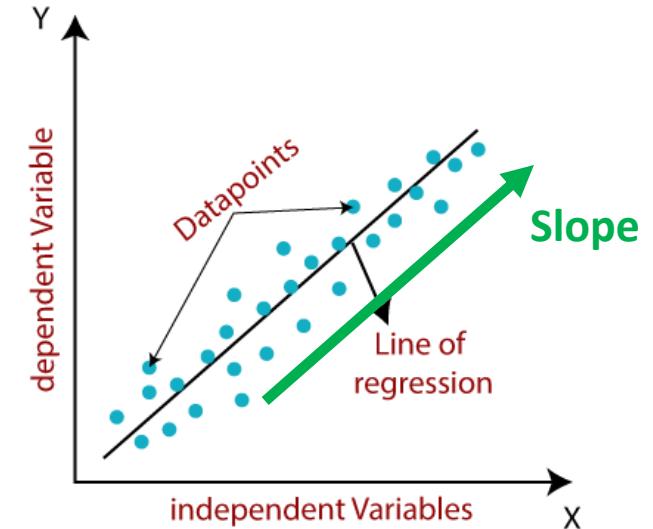
- Slope is explainable (how much does one variable affects a prediction)

$$y = w_1x_1 + w_2x_2 + w_3x_3 + \dots$$

↑
prediction ↑
 weights ↑
 features

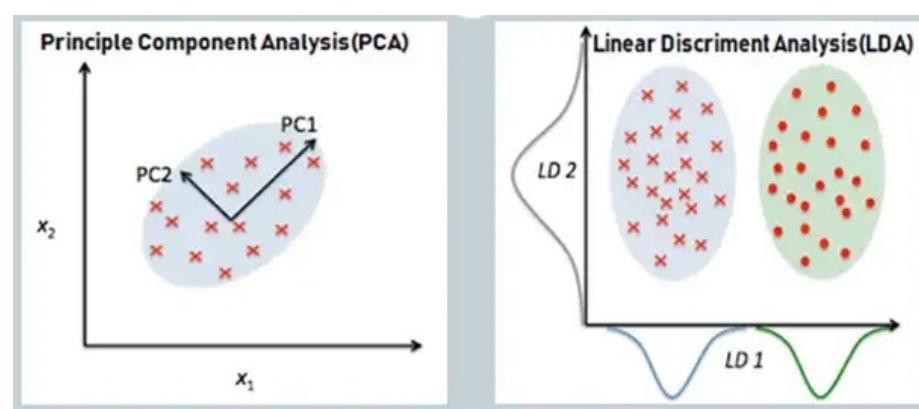
- Each feature has an associated weights, indicating importance

- “A change of Δx amount to feature x_1 will result in increase of prediction by Δy

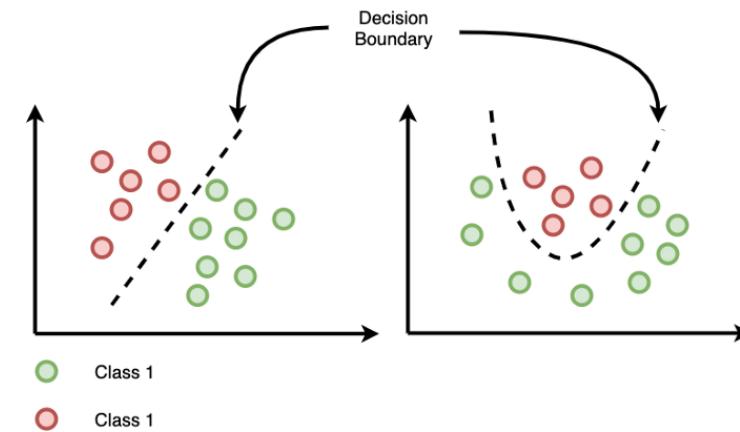


Explainable Models (2)

- Dimension reduction
 - Dimension reduction allows us to visualize the training data distribution



[Source](#)



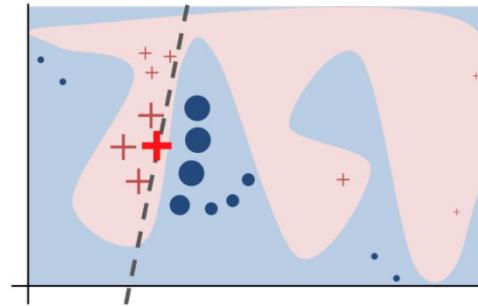
[Source](#)

- Decision boundary can be visualized and understood
 - Instances at the boundary characterizes how different classes are different

Deep Learning Explainability Methods: Examples

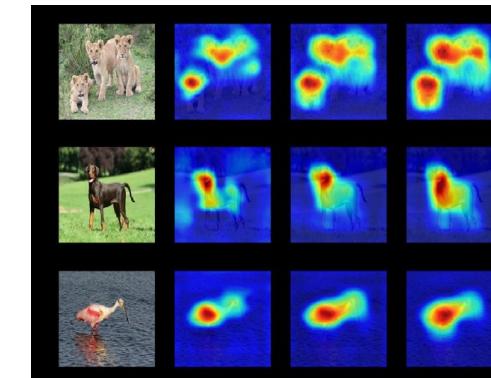
- **Proxy Model**

- Learn a inherently interpretable model locally approximating the original model (e.g. a linear model, interpret by weights).



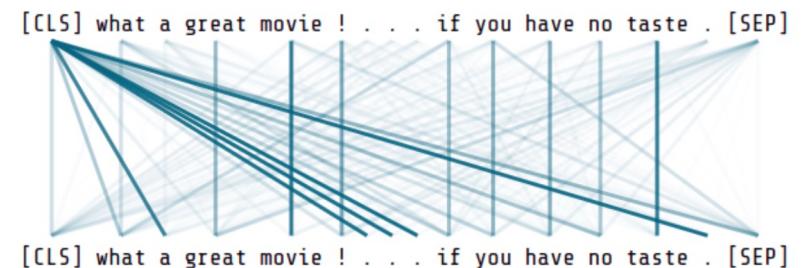
- **Saliency Maps**

- Compute gradients of outputs w.r.t. input pixels.



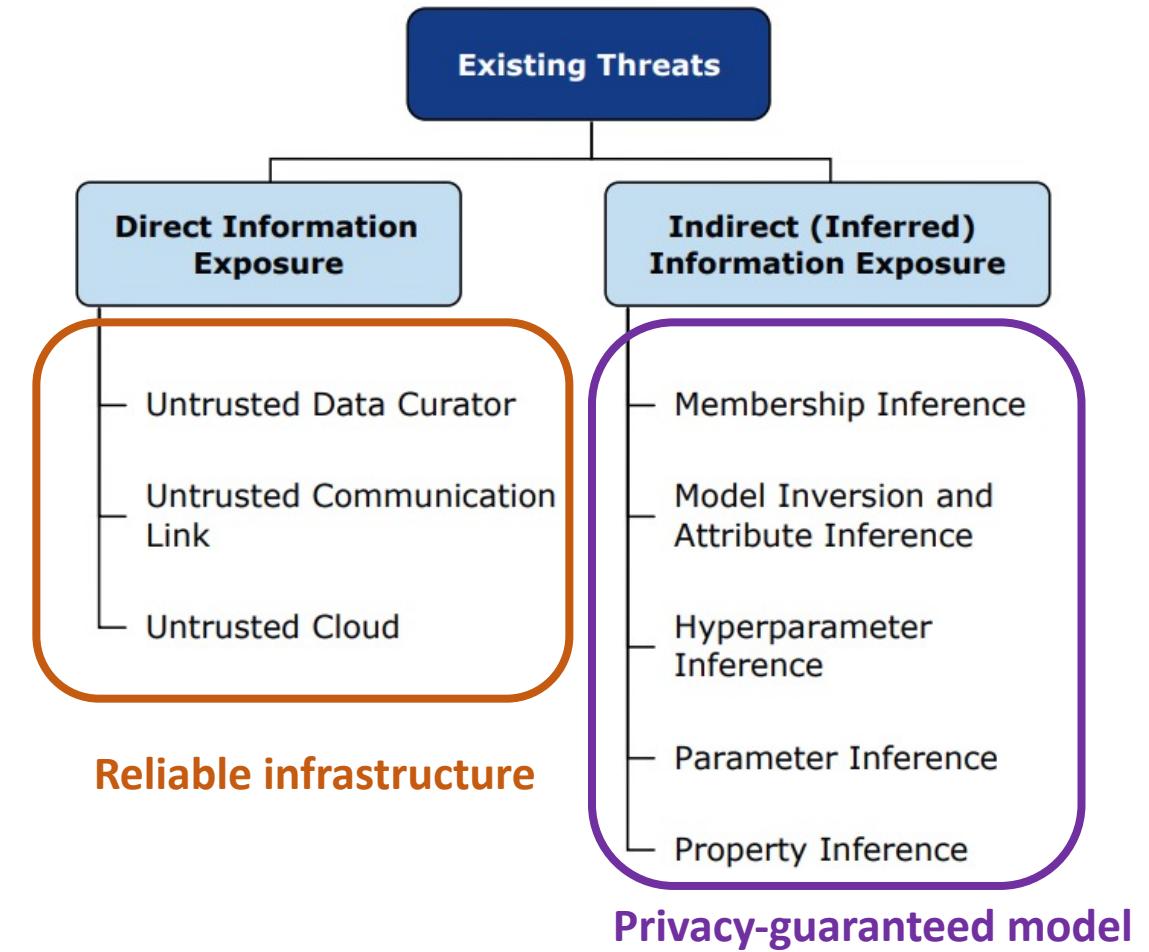
- **Attention Mechanisms**

- Visualize attention weights in a attention models.



Privacy

- Important privacy policies
 - [California Consumer Privacy Act \(CCPA\)](#)
 - [General Data Protection Regulation GDPR](#)
- Prevent private data being leaked
 - Training data
 - Model parameters
- Existing threats:



Privacy-related Attacks

- **Model extraction attacks**
 - Steal architecture and parameters of a deep learning model.
- **Membership inference attacks**
 - Infer whether certain data point belongs to the training set of a model.
- **Model inversion attacks**
 - Infer a model's inputs from their corresponding outputs.
- **Other privacy attacks**

Example Model Inversion



- The image on the left was recovered using the [model inversion attack](#)
- The attacker is given only the person's name and access to a facial recognition system that returns a class confidence score

Privacy of GNNs: Privacy-Preserving Techniques

- **Federated Learning**

- Calculate gradients on **individuals** using their own data
- Aggregate parameters (e.g. gradients/model weights) on the server

- **Differential Privacy**

- Add noise to data, such that
 - **Meaningless** when viewed individually
 - But approximate the analytics results when **aggregated**

- **Insusceptible Training**

$$\min_{\theta} \sum_{v_i \in \mathcal{V}} \mathcal{L}_Y(f_{\theta}(v_i)) + \lambda \mathcal{L}_A(\mathcal{F}_A(v_i))$$

Original task loss Attack function: try to distinguish the private labels

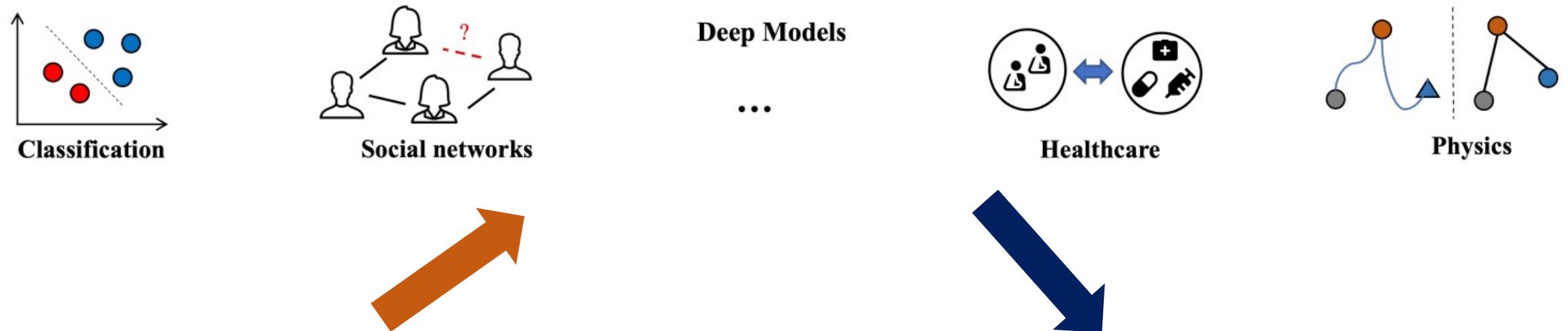
- **Security Computation**

- More related to system/hardware

Privacy-preserving loss: e.g. make the attack function's output probability close to 0.5 for the private labels

Machine Unlearning

- ML regulations give users the **right to eliminate** their data from the trained models as if they never existed in the training dataset



User submits unlearning
request to remove their data

The model is updated (different weights) as if the data to be removed have never been observed

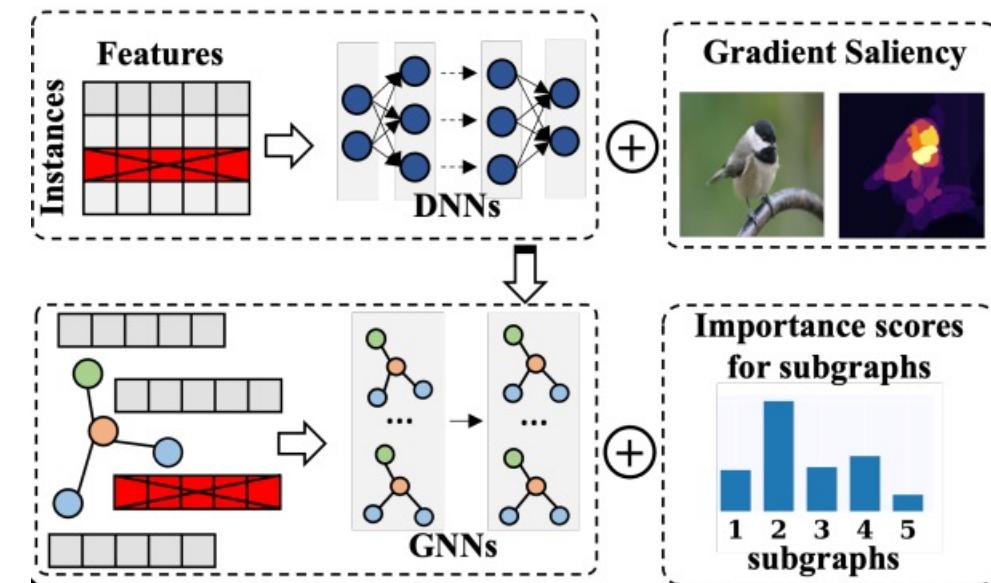
Malicious Unlearning

- **Objective 1: instance-wise removal**

A user / user group may request that its own data (training instance) to be removed from the model

- For example, in image classification, a facial image may be requested to be removed

- In social network predictions, a node in the graph may need to be removed

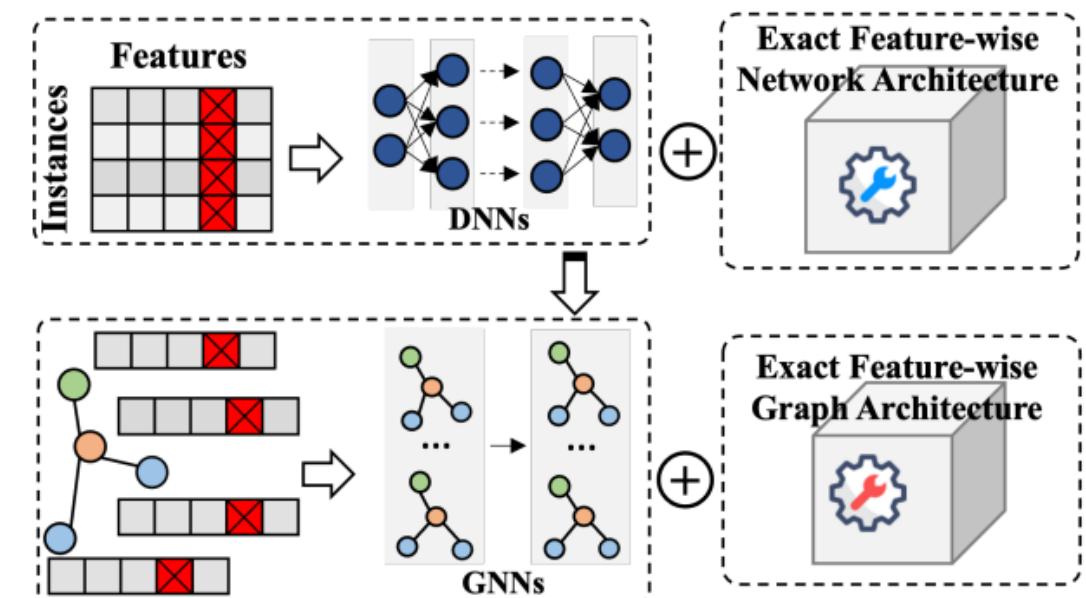


Malicious Unlearning

- **Objective 2: feature-wise removal**

The user of the model may request that a particular feature / group of feature dimensions to be removed from the model

- For example, gender information may be requested to be removed in medical data prediction models

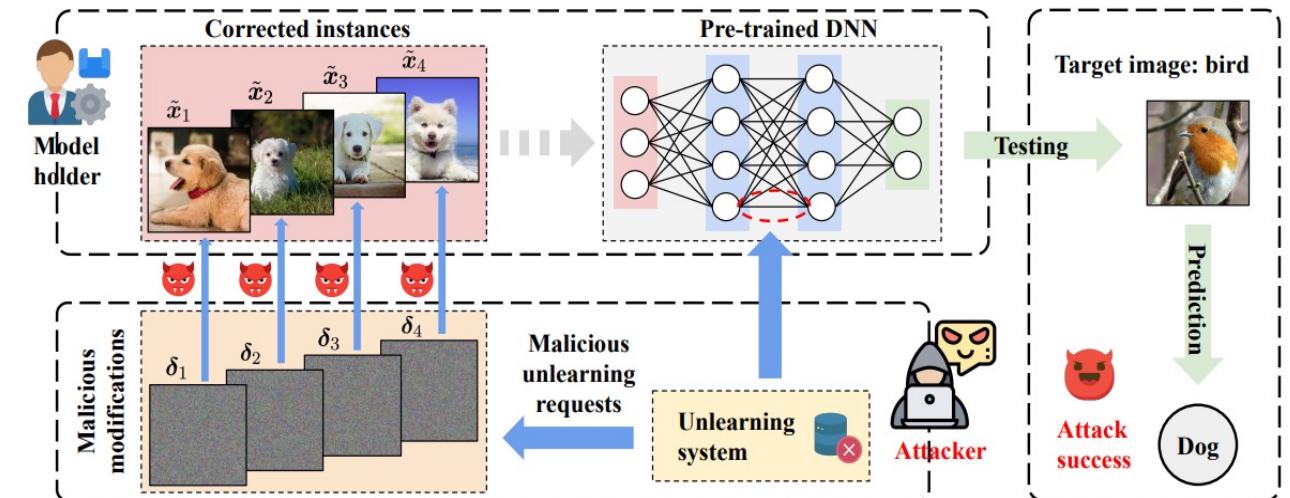


Malicious Unlearning

- **Objective 3: adversarial robustness**

An adversary may send malicious unlearning requests to remove certain features and training instances, to make the model perform drastically worse

- For example, the attacker can ask certain images to be altered, so that the system **unlearns the previous instances** and instead uses corrupted instances
- Resulting in lower test accuracy

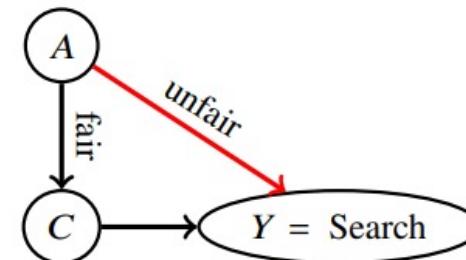
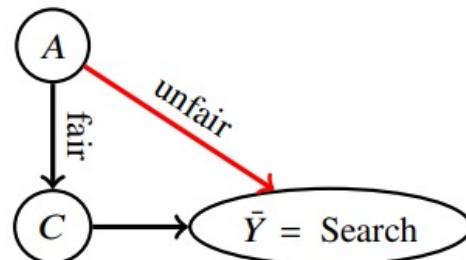
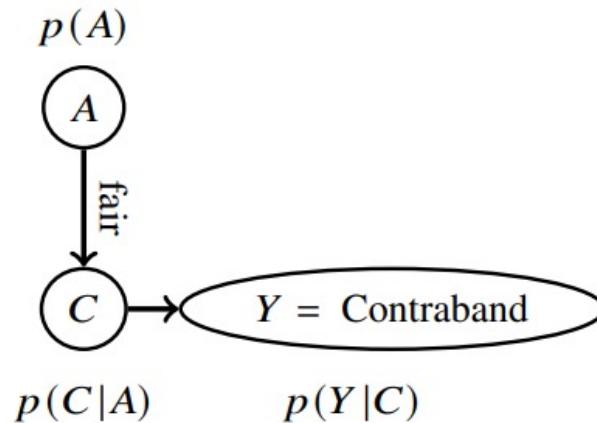


Fairness in Deep Learning

- **Goal:** exclude prejudice or favoritism towards an individual or a group.
 - For example, in a bank's transaction network, the model should not learn to make predictions of loans based on gender, race or other protected characteristics.
- **Prevent Bias & Discrimination**
 - **Bias:** unfair operation in data collection, sampling, measurements, ...
 - **Discrimination:** incorporation of intentional or unintentional human prejudices and stereotyping in deep learning models

Why is Fairness an Issue

- Illogical conclusions may be made due to biased model or biased training data.
- Contraband example:



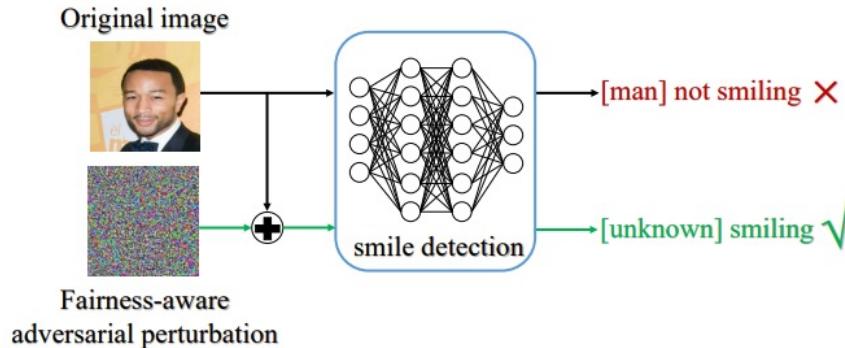
- Examples
 - image classification biasing towards a certain race
 - Recommender system biasing towards popular or generic items

Fairness and Attacks

- A lot of times adversarial attacks exacerbates fairness issues

Microsoft chatbot incident

- On the other side, adversarial training can actually **improve fairness!**



Original image is falsely recognized due to **model unfairness**, i.e., tending to predict males as “not smiling”.
The fairness-aware adversarial perturbation helps the input image to **hide the protected attribute and get fair treatment**.
[CVPR 2022](#)



“We are deeply sorry for the unintended offensive and hurtful tweets from Tay, which do not represent who we are or what we stand for, nor how we designed Tay,” Lee wrote in a [blog post](#), adding that the bot will come back online only after the company is sure that it’s ready to deal with “malicious intent.”

Indeed, Lee said that a small number of people “exploited a vulnerability” in Tay and thus were to blame for the tweets, which spoke positively of Hitler, among other things.

Fairness

- **Fair representation learning methods**
 - Learn representations, from which one cannot infer sensitive attributes.
 - A common technique is **adversarial training**
- **Fair prediction enhancement methods**
 - **Data augmentation**
 - Perturbation of protected features
 - **Fair graph**
 - Modify graph structures (e.g. drop edges that may induce bias)
 - **Regularisation**
 - Ex) any two individuals who are similar should receive similar algorithmic outcome

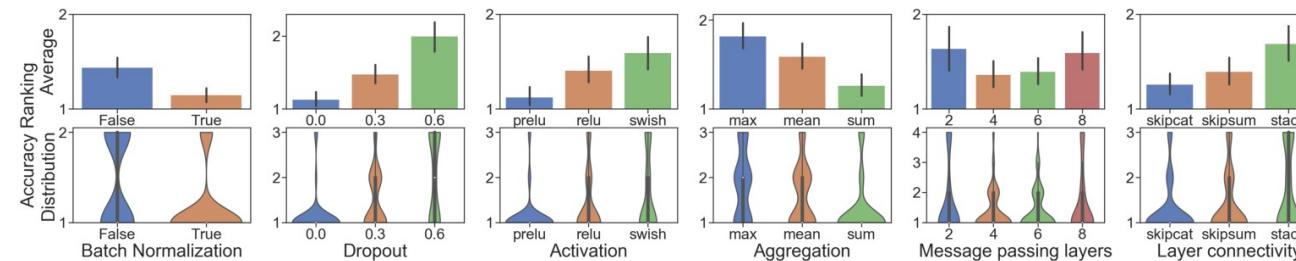
$$\| \mathbf{Y}[i, :] - \mathbf{Y}[j, :] \|_F^2 \mathbf{S}[i, j] \leq \delta$$

Predictions of node i

Similarity between node i and j

Accountability

- **Accountability** refers to the extent to which people can trust models by assessing a complex DL system
- **Benchmarking**: detect violation of utility
 - Architecture design, Model training, Model validation



- **Security evaluation**: detect violation of security
 - Data integrity verification
 - Procedure integrity verification

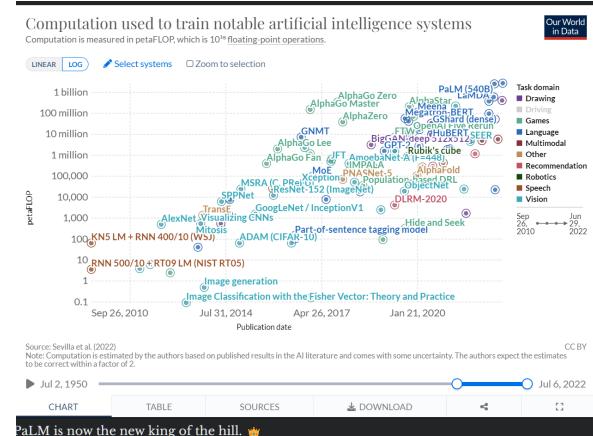
Environmental Well-Being

- When developing DL models, one should consider the **cost of training and inference**
 - Large-scale (unlabeled) datasets → requires execution efficiency
 - Complex pre-trained foundation models → large parameter space (**hundreds of billions**)
 - Deeper or more complex architectures → challenge in deployment on edge devices



Compute Clusters

Credit: Imaginima/E+/gettyimages



Large model training involves 10^{24} flops
<https://blog.heim.xyz/palm-training-cost/>

Rex Ying, CPSC 471: Trustworthy Deep Learning

tl;dr What would it cost you to train PaLM using cloud computing (and you're not Google)?
Something around \$9M to \$28M.

PaLM a 540B state-of-the-art language model

Google recently published a new paper presenting PaLM (their blogpost) – a 540B parameter large language model.

Input: Jennifer looked out her window and sees a really cool cloud below her. She unbuckles her seatbelt and heads to the bathroom. Is Jennifer probably traveling more than 300 miles per hour relative to the earth?
Model Output: 300 miles per hour is about 480 km/h. This is about the speed of a commercial airplane. Clouds are usually below airplanes, so Jennifer is probably on an airplane. The answer is "yes".

540 B parameter pretrained model

<https://blog.heim.xyz/palm-training-cost/>

Environmental Well-Being: Methods

- **Scalable models**

- For example, model / data distillation, scalable graph neural network training

- **Sparse models**

- Sparse neural networks; sparse transformers

- **Model compression**

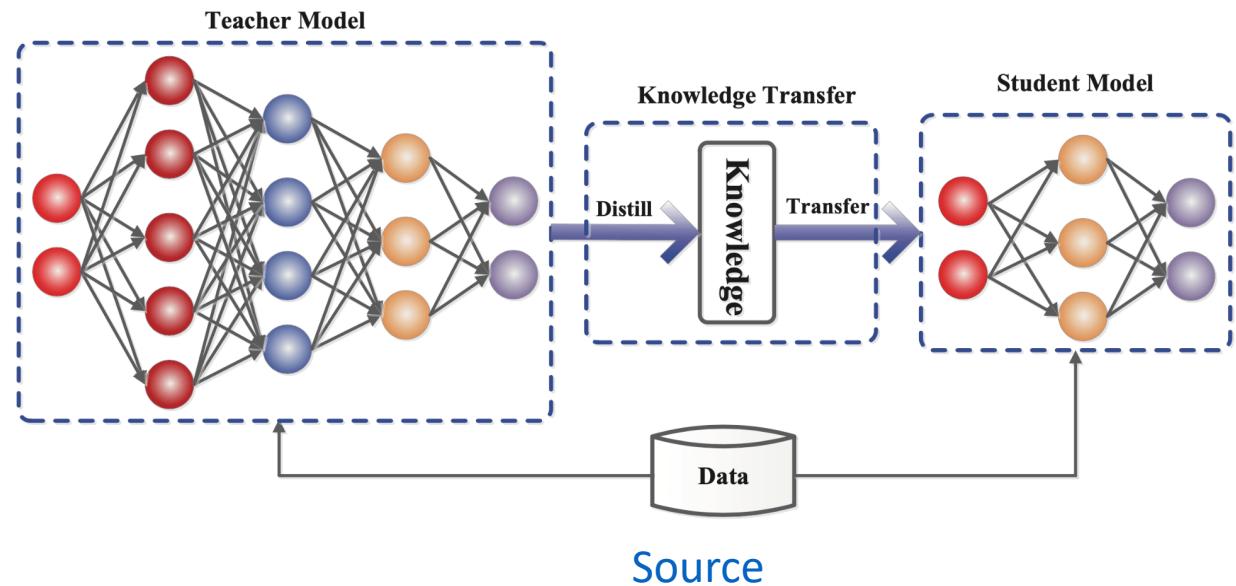
- Knowledge distillation
- Model pruning
- Reducing parameters
- Model quantisation

- **Efficient frameworks and accelerators**

- Sparse computation; efficient distributed training
- Software and hardware-level

Scalable Models: Distillation

- Model distillation
 - Learn **student model(s)** that are light-weight but can be more efficient
 - Simple models also tend to be more **explainable**



Scalable Models: Data Selection

- **Data distillation**

- Find subset of training data that are representative
- Such that model trained on the subset can achieve similarly good performance

- Example: coreset selection

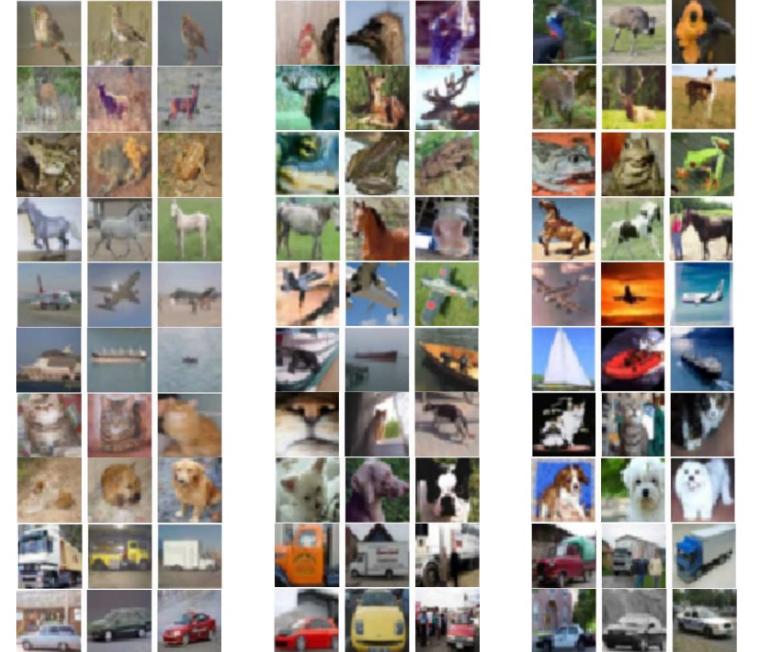
- At every epoch, a subset of **representative** images are selected for training. **Objective:**

Training Subset $S^* = \arg \min_{S \subseteq V, \gamma_j \geq 0} \forall j |S|,$ s.t.

$$\max_{w \in \mathcal{W}} \left\| \sum_{i \in V} \nabla f_i(w) - \sum_{j \in S} \gamma_j \nabla f_j(w) \right\| \leq \epsilon.$$

Model weights

Gradient difference (between subset and the whole training set)



(a) First

(b) Middle

(c) Last

Epoch 1

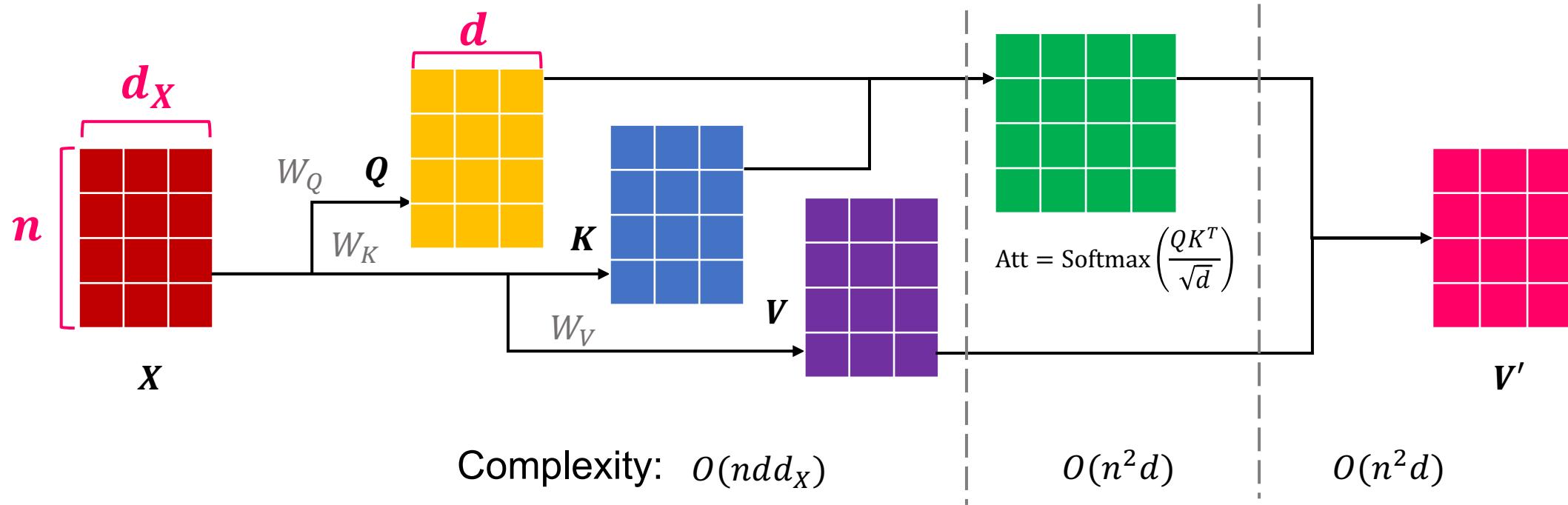
Epoch 100

Epoch 200



Scalable Models: Sparse Transformer (1)

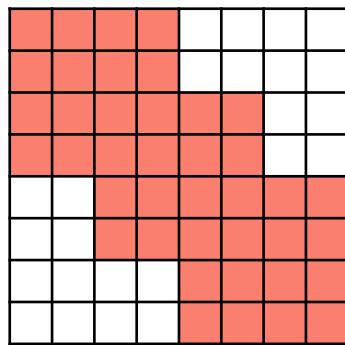
- For an input sequence with n tokens and dimension d_X



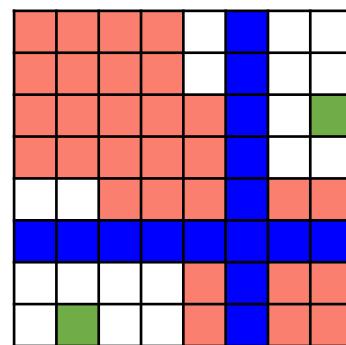
The computation complexity is quadratic to number of tokens n

Scalable Models: Sparse Transformer (2)

- Masking attention (with sparse pattern) can reduce complexity from $O(n^2)$ to $O(n)$.



Longformer [Beltagy et al., 2020]

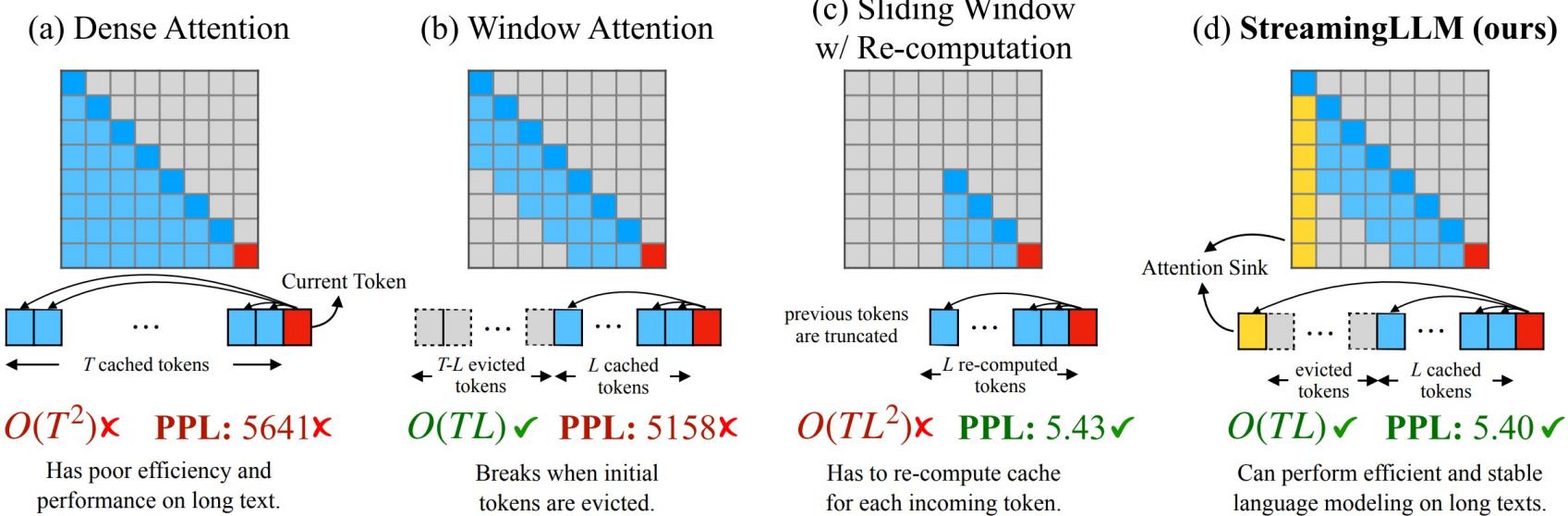


BigBird [Zaheer et al., 2020]

- No attention
- Local attention: tokens attend within a local window (size = 4 in the figure)
- Global attention: one global token attend to all tokens
- Random attention: randomly select attentions

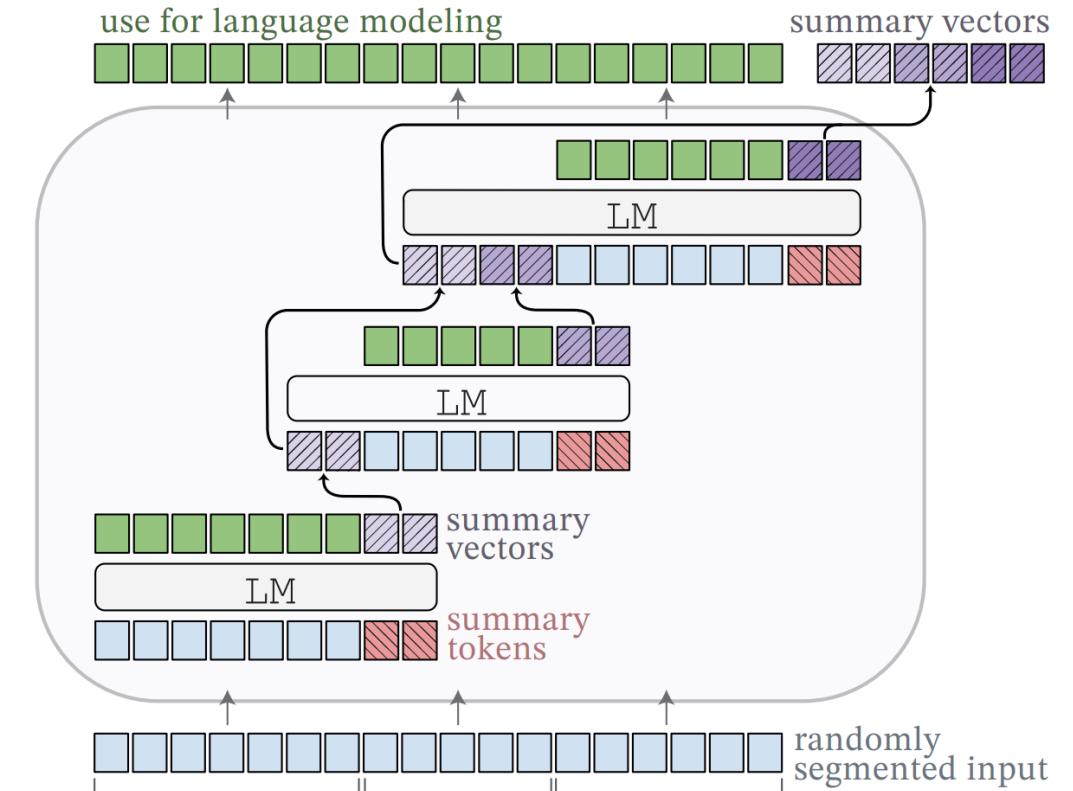
Efficient LLMs (Sliding Window)

- StreamingLLM uses a sliding window and the first token (attention sink)



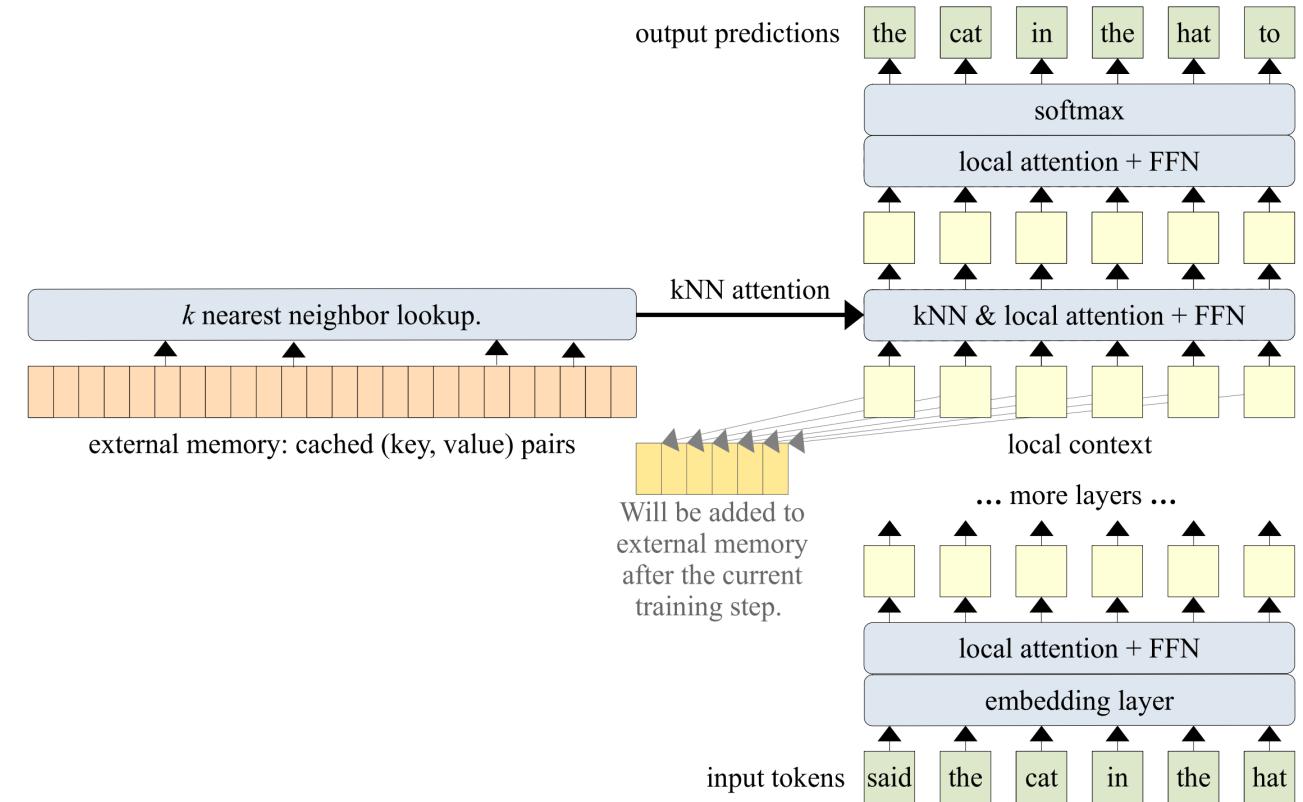
Efficient LLMs (Recurrence)

- AutoCompressors process long documents by recursively generating **summary vectors**
- Pass summary vectors as soft prompts to all subsequent segments.



Efficient LLMs (Retrieval)

- Extend Transformers with access to (key, value) pairs of **previously seen subsequences**
- Retrieve the most relevant tokens in the past for each step of autoregressive generation



Summary

- Trustworthy AI and Deep Learning plays a crucial role when applying models to **real-world applications**
- An intersection of machine learning with many fields: computer security, systems, causal inference, human-computer interaction ...
- Many of the aspects of trustworthy AI are **closely related to each other!**
- It is increasingly **challenging** given the growing complexity of deep learning models in recent years
- Research in this field faces huge challenges due to the **diversity of aspects** in trustworthy AI, as well as the **diversity of deep learning models** (supervised; self-supervised; generative model; RL ...)