AI-generated Text Detection

Neil Gong

Why detecting AI-generated text?

- Disinformation propaganda?

- AI-assisted writing
  - Homework
  - Exams
  - Papers and reviews

- Preventing harmful content generation is insufficient

AI-generated text detectors

- Passive
- Retrieval-based
- Watermark-based

Passive detectors

- Outlier detection
- Classification

Outlier detection

- Heuristics for the “artifacts” in AI-generated text
- No access to human-written text

Outlier detection methods

- **log p(x)**: uses the source model's average token-wise log probability
- **Rank**: uses the average observed rank of the tokens in the text
- **Log rank**: uses the average observed log rank
- **Entropy**: model-generated texts will be more 'in-distribution' for the model, leading to more over-confident (thus lower entropy) predictive distributions.
- **DetectGPT**: inspect the local region of a text

Log probability threshold-based detection

- Generated text has a higher log probability

"The cat sat on a mat" →

Word	Probability given context	Log probability
The	0.1	-2.3
cat	0.15	-1.9
sat	0.05	-3.0
on	0.2	-1.6
a	0.3	-1.2
mat	0.1	-2.3

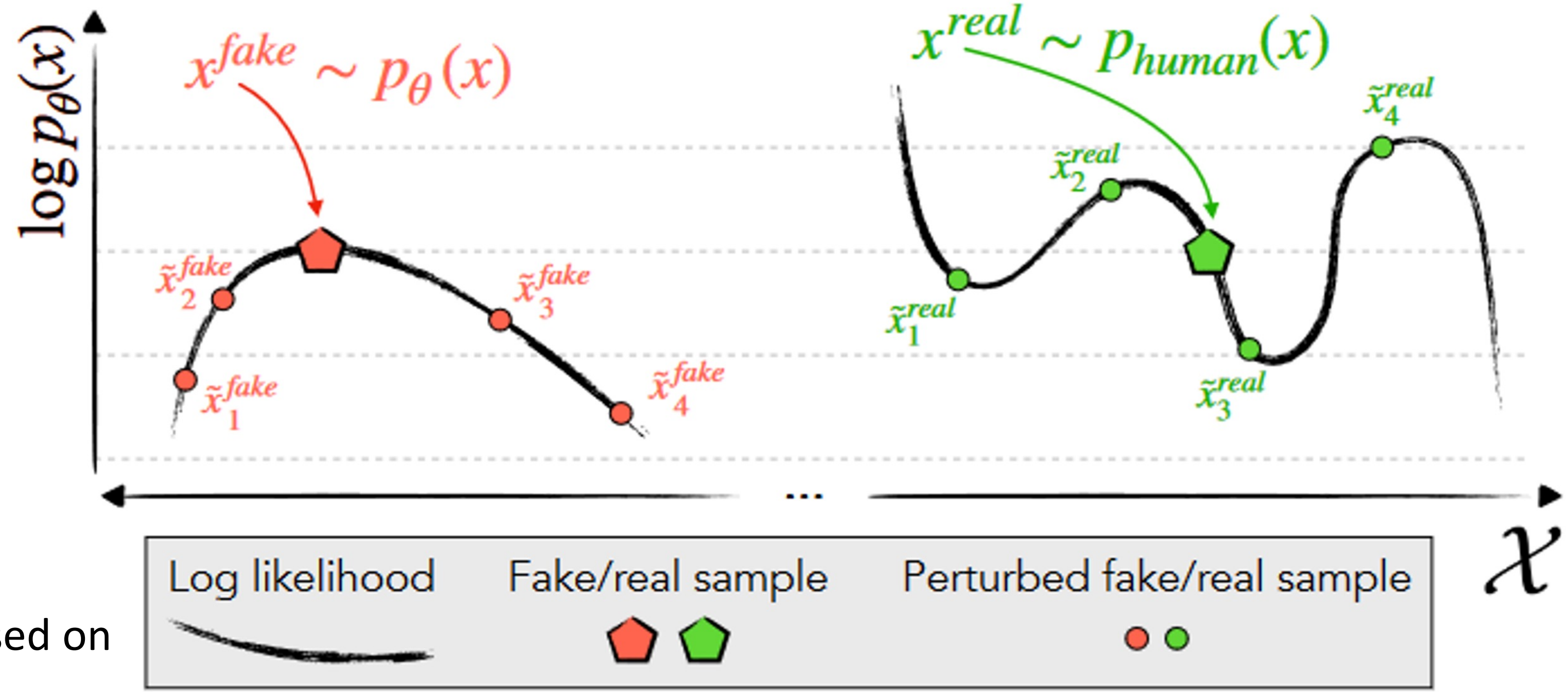
→ **Average Log probability = -2.05**

Threshold = -1 > -2.05

Not generated

DetectGPT - Basic Hypothesis

- Models tend to output the tokens with high probability
- Slight modification to the generated output will decrease the log probability



Slides are modified based on Yanming Xiu

DetectGPT - Basic Hypothesis

- The modified output is defined as the **perturbation**
- Perturbation discrepancy (PD):

$$\mathbf{d}\left(x, p_\theta, q\right) \triangleq \log p_\theta(x) - \mathbb{E}_{\tilde{x} \sim q(\cdot|x)} \log p_\theta(\tilde{x}) \qquad (1)$$

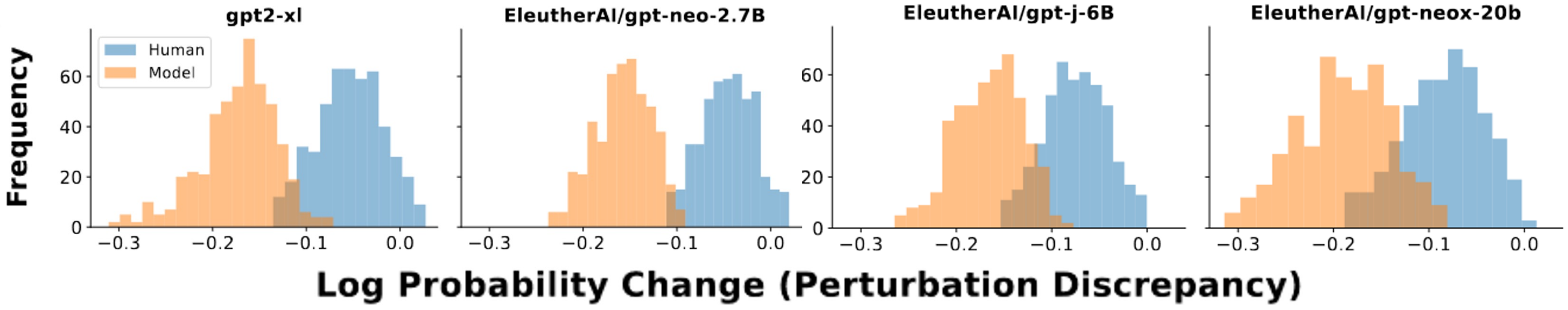
  - x: original text
  - x~: perturbation text
  - q: perturbation function
  - pθ: log probability function of generative model

DetectGPT - Basic Hypothesis

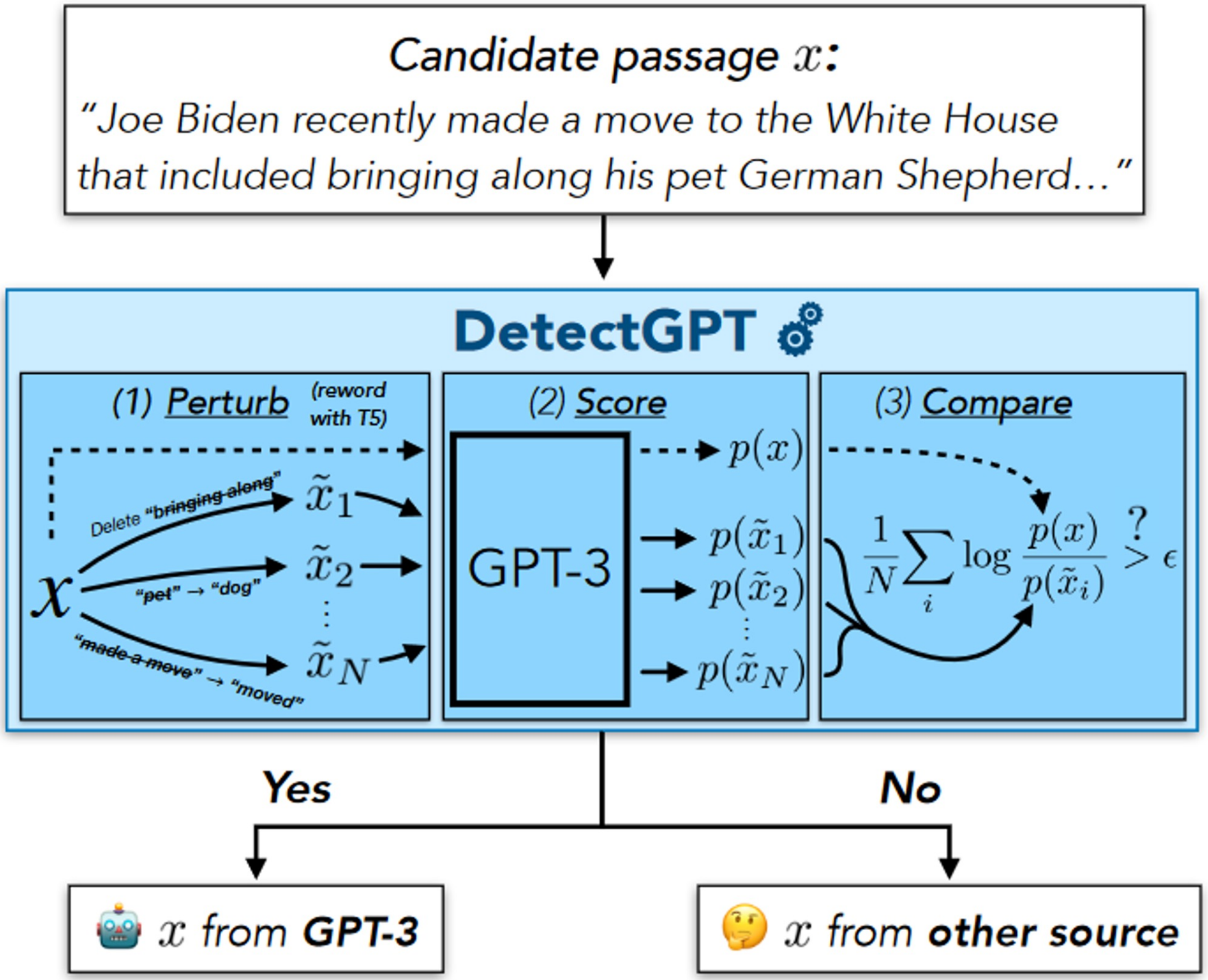
- Formal hypothesis:

Perturbation Discrepancy Gap Hypothesis. *If q produces samples on the data manifold, $\mathbf{d}(x, p_\theta, q)$ is positive with high probability for samples $x \sim p_\theta$. For human-written text, $\mathbf{d}(x, p_\theta, q)$ tends toward zero for all x.*

- Empirical result:



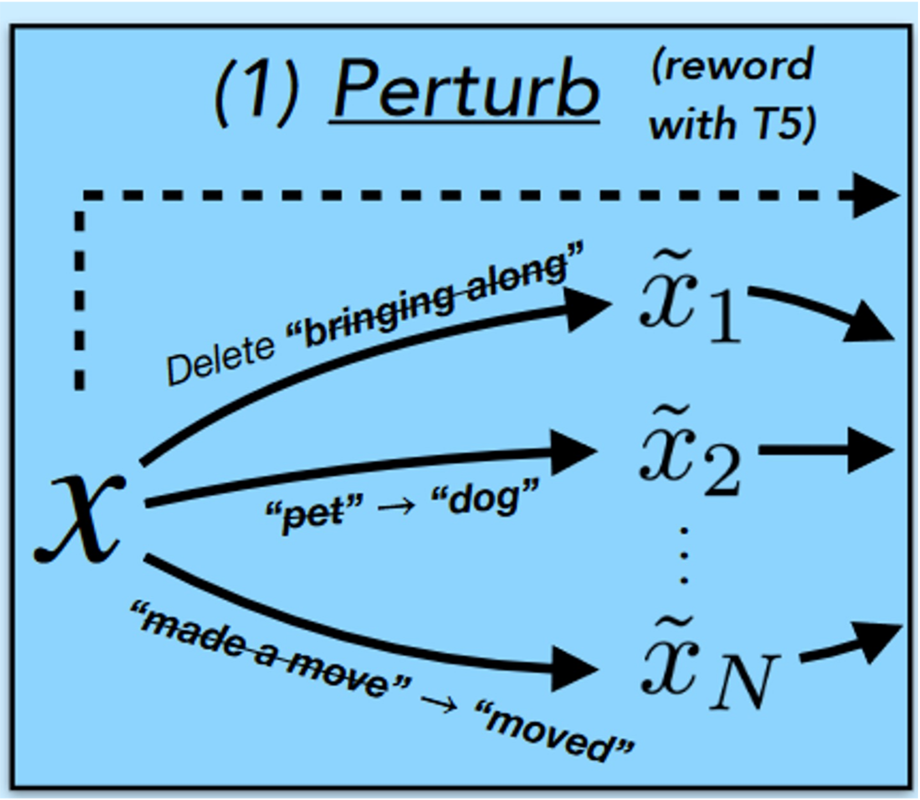
DetectGPT - System Architecture

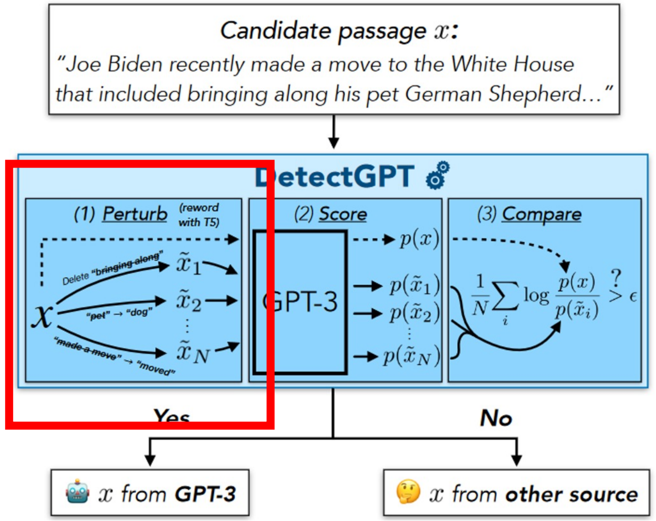


DetectGPT - System Architecture

- Step 1: make the perturbed samples

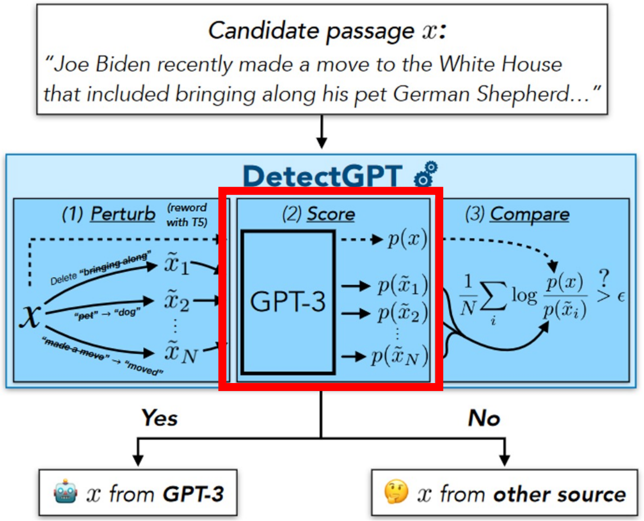
Candidate passage x:
"Joe Biden recently made a move to the White House that included bringing along his pet German Shepherd…"

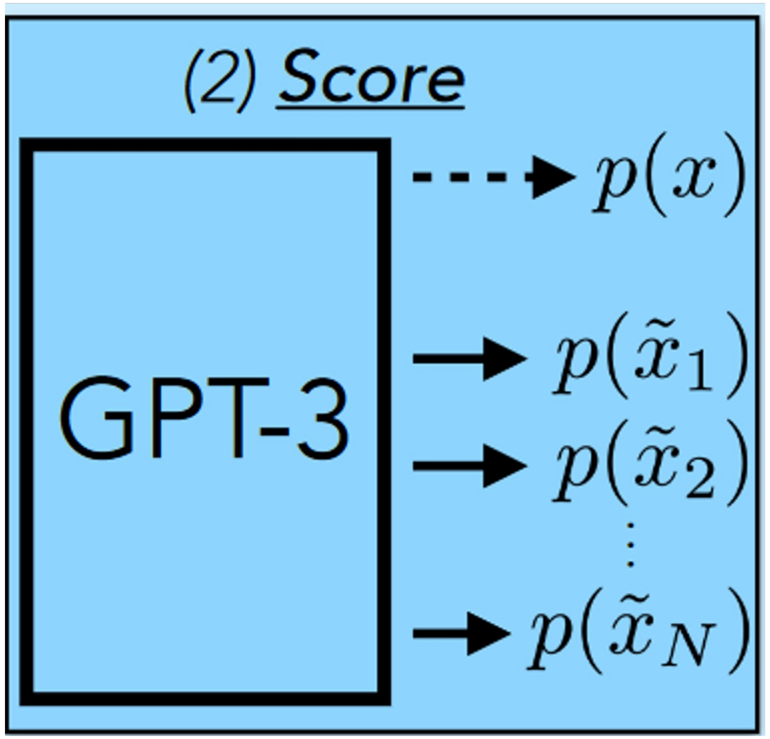




DetectGPT - System Architecture

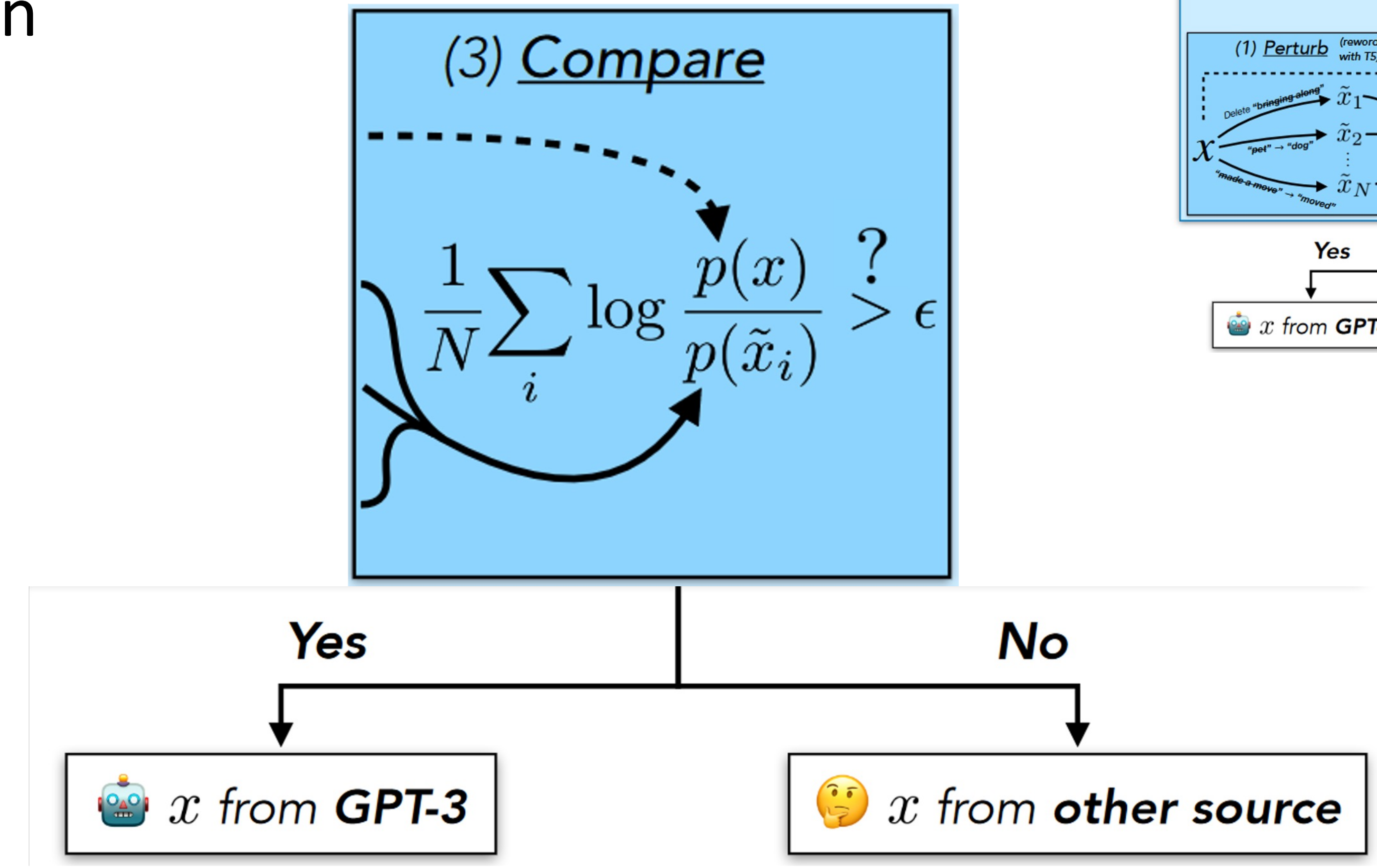
- Step 2: calculate the log probability of perturbed samples

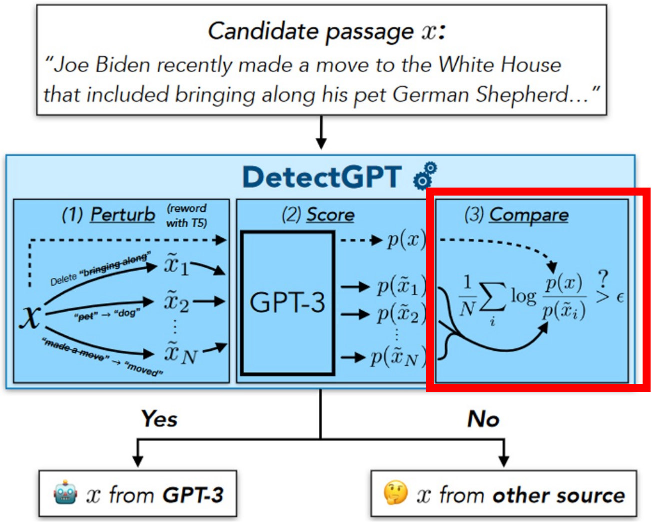




DetectGPT - System Architecture

- Step 2: compare the result and make classification





Experiments

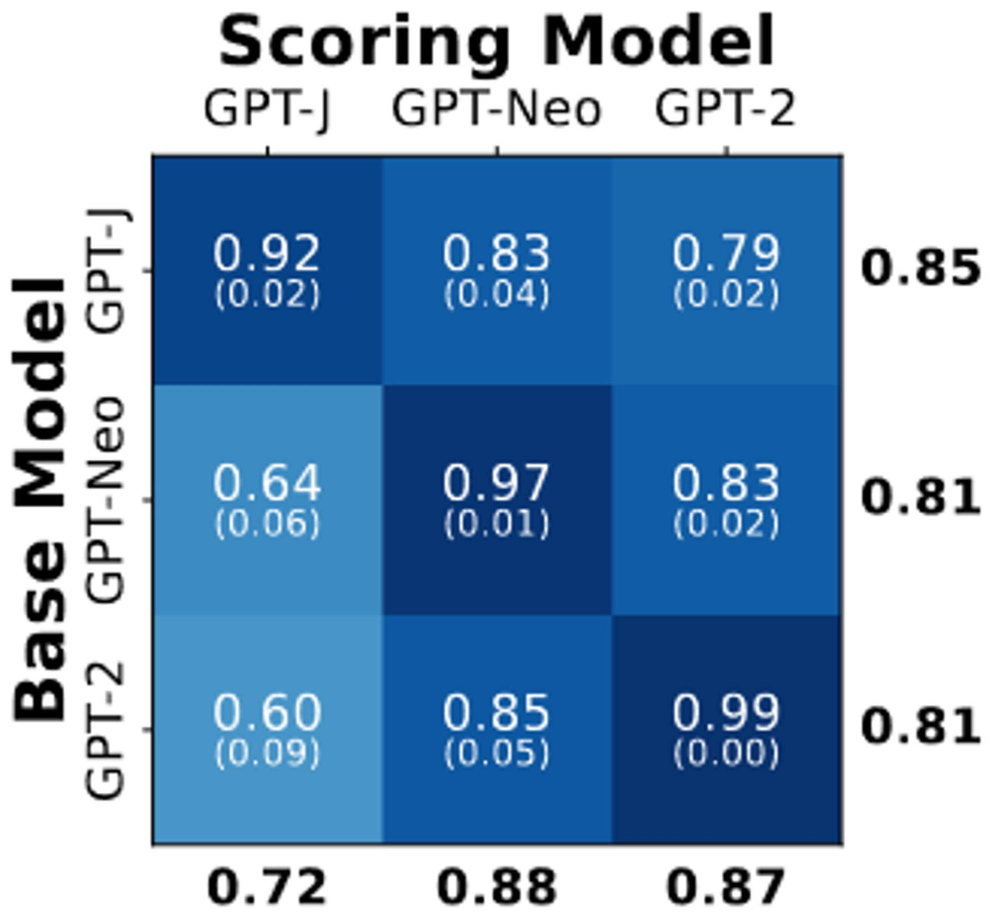
- Generated text: prompting with the first 30 tokens of real text
- Models tested: GPT-2 OPT-2.7 Neo-2.7 GPT-J NeoX
- Perturbation model: T5

AUC Results

	XSum						SQuAD						WritingPrompts					
Method	GPT-2	OPT-2.7	Neo-2.7	GPT-J	NeoX	**Avg.**	GPT-2	OPT-2.7	Neo-2.7	GPT-J	NeoX	**Avg.**	GPT-2	OPT-2.7	Neo-2.7	GPT-J	NeoX	**Avg.**
$\log p(x)$	0.86	0.86	0.86	0.82	0.77	0.83	0.91	0.88	0.84	0.78	0.71	0.82	0.97	0.95	0.95	0.94	0.93*	0.95
Rank	0.79	0.76	0.77	0.75	0.73	0.76	0.83	0.82	0.80	0.79	0.74	0.80	0.87	0.83	0.82	0.83	0.81	0.83
LogRank	0.89*	0.88*	0.90*	0.86*	0.81*	0.87*	0.94*	0.92*	0.90*	0.83*	0.76*	0.87*	0.98*	0.96*	0.97*	0.96*	**0.95**	0.96*
Entropy	0.60	0.50	0.58	0.58	0.61	0.57	0.58	0.53	0.58	0.58	0.59	0.57	0.37	0.42	0.34	0.36	0.39	0.38
DetectGPT	**0.99**	**0.97**	**0.99**	**0.97**	**0.95**	**0.97**	**0.99**	**0.97**	**0.97**	**0.90**	**0.79**	**0.92**	**0.99**	**0.99**	**0.99**	**0.97**	0.93*	**0.97**
Diff	0.10	0.09	0.09	0.11	0.14	0.10	0.05	0.05	0.07	0.07	0.03	0.05	0.01	0.03	0.02	0.01	-0.02	0.01

Detection - Source Model Unknown

- Black-box setting: source model unaccessible, use a different model to score a candidate passage;
- When the surrogate model is different from the source model, detection performance is reduced;



Limitations of DetectGPT

- Access to log probabilities
- A reasonable perturbation function is required
- Computation overhead

Classification methods

- Train binary classifiers
- Key: what features to represent a text?

Retrieval-based detection