The ‘Rogue’ Data Polluting Generative AI Performance

Originally published at https://www.unite.ai/the-rogue-data-polluting-generative-ai-performance/

A new study finds that many popular image datasets used to train AI models are contaminated with test images or near-duplicates, allowing models to cheat by memorizing answers instead of learning. The leakage is widespread but generally undetected, quietly inflating scores and giving unfair advantages to models trained on web-scale data.

When you take a driving test, you’re not usually told in advance exactly which roads will be used for the test. If you did (and you were a touch lacking in integrity), you might ‘optimize’ for the test by practicing repeatedly on that route, instead of developing broader driving skills that can handle *any* route reasonably well.

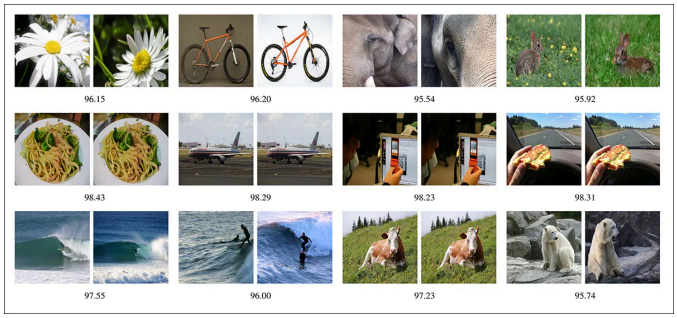
In the training of machine learning models, this is a reasonable analogy for a test split – a dividing-up of training set data between (usually) a 70% split for the data that will be used to train the model, with the remaining 30% used as ‘in the wild’ data.

Since in-the-wild data has never been seen by the model, if the model performs well on that data, it can be assumed effective and performant; if not, the model may have overfitted on a well-balanced set – or else the data needed extra curation and definition.

Either way, *not* evaluating models on their training data is the cornerstone of current method in AI research and development.

Same Again, Please

According to a new research paper from Japan, the computer vision and generative AI research sector has not remotely matched the efforts of LLM researchers to ensure that test data doesn’t pollute training data; in tests, the researchers found that every hyperscale vision dataset they studied, including those powering some of the biggest current generative AI systems, has to some extent allowed its test data to cross over into its training data – meaning that benchmarks and performance reports for models trained on these splits will not be any more accurate than an exam result from someone who sneaked a crib into the exam hall, and will not reflect real-world performance on genuinely novel data.



Examples of cross-contamination of data found by the researchers, where duplicate or near-duplicate data points exist in both the training and testing data. Source: https://arxiv.org/pdf/2508.17416

In the image above, from the new paper, we see examples of either duplicate or near-duplicate data points found in both the core training data and test data of a variety of models – enough to invalidate the model's performance on that data, and lightly inflate its general scores across the board, facilitating the appearance of a level of generalization that the model may not actually have attained.

To make matters more complicated, the contamination appears to occur across a diversity of possible scenarios, including 'pre-training', where the weights of older ancestral models are used to 'kick-start' a new model. If the upstream, older model has some of the same data as the newer dataset that is being pre-trained, then cross-contamination can occur even if the 70/30 or 80/20 split is clean.

Cumulative Effect

This is almost certain to occur even in the very latest datasets: the scope of vision/language datasets has grown enormously over the past five years, taking in not only the newest image data on the web, but re-harvesting much of the same data that populated those older, historical datasets.

Further, automated routines designed to trawl and filter billions of images for duplicates and near-duplicates are now faced with such an onerous task that curation itself – its cost in terms of time and money – must now be considered within the context of budgetary limitations

Meanwhile, image duplication is an inevitable consequence of the kind of *ad hoc* web-trawling behind massive collections such as Common Crawl, due to the common practice of reposting and recompressing images, and applying edits such as crops, and even flipping (to evade detection, when the image may have been used without permission, for instance).

The authors observe*:

> *'Data leakage is a widespread issue, prevalent in most visual datasets. Leakage can obscure the generalization ability of models, which is particularly problematic when comparing models trained on different datasets, leading to unfair comparisons.*
>
> *'We urge dataset designers to carefully consider the implications of these evaluations. For a fairer model evaluation, we recommend the use of duplicate detectors that considers both hard and soft leakage.*
>
> *'Ideally, leaked images should be removed from the training set, and if not possible, they should at least be removed from the test set.'*

The paper elaborates on a number of tests that the researchers conducted on massive and popular datasets – every single one of which demonstrated some level of contamination.

The new paper is titled *Data Leakage in Visual Datasets*, and comes from three researchers at The University of Osaka.

Method

The paper's authors define leakage in terms of three dimensions: *modality*, *coverage*, and *degree*.

Modality distinguishes whether only images are leaked or whether both images and labels are exposed; *coverage* identifies whether the overlap occurs within the same dataset or across different datasets; and *degree* defines whether the duplicated content is exactly the same or merely adjacent.

Regarding leakage, the two scenarios considered in the work are *intra-dataset leakage* (where evaluation images reappear in the training split of the same dataset), and *inter-dataset leakage* (where evaluation images from one dataset are present in a *different* dataset used for training).

Regarding degree, the two levels defined are *soft leakage* (where images are not identical but exhibit minor variations), and *hard leakage* (where images are exactly the same across training and evaluation).

The researchers address the detection of leakage in terms of *image retrieval*, using image encoders to represent each image as a feature vector. The *query set* is the evaluation data, while the *collection* is the training set.

For smaller datasets, every query vector was directly compared to all training vectors using cosine similarity. For larger datasets, a Faiss index was built to enable faster, K-Nearest Neighbors (KNN) search.

Since the encoder needs to capture enough visual information to detect subtle similarities, but still remain efficient in the face of very high volumes of data, the authors relied on precomputed CLIP features made available by dataset creators, in the case of the LAION collection that underpins Stable Diffusion, and later projects.

The authors note that allowing CLIP to use its distilled understanding of the dataset (instead of polling the actual files at scale) sped up the process considerably, and offered improved consistency across comparisons.

Data and Tests

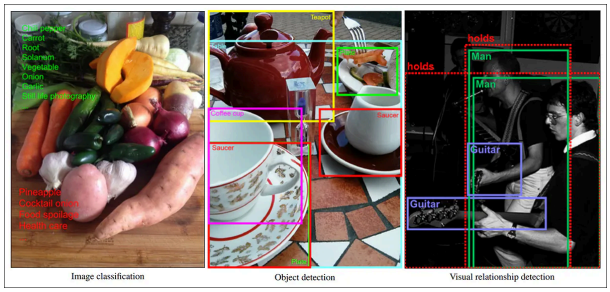
The CLIP image encoder used in the tests for the new work was the default CLIP ViT-B/32 originally used to sift LAION. To establish whether diverse images were related to each other, KNN was used under AutoFaiss.

The datasets were grouped into three types: *pretraining* datasets – large, web-scraped collections used to train generalist models; *training* datasets – smaller, often annotated collections, intended for direct model tuning; and *benchmark* datasets – manually annotated, and used exclusively for evaluation.

The analysis covered twenty splits across seven datasets: Microsoft COCO was used as both a training and evaluation set, incorporating the train, validation, test, and unlabeled splits; Flickr30k served exclusively as a benchmark; and the Google Conceptual Captions (GCC) collection was treated as a pretraining source, with its validation portion also used for evaluation.

Additionally, ImageNet was used for both training and benchmarking, while the LAION-400M dataset was used solely for pretraining.

OpenImages v4 contributed training and benchmark data, and TextCaps provided both training and test splits for evaluation.



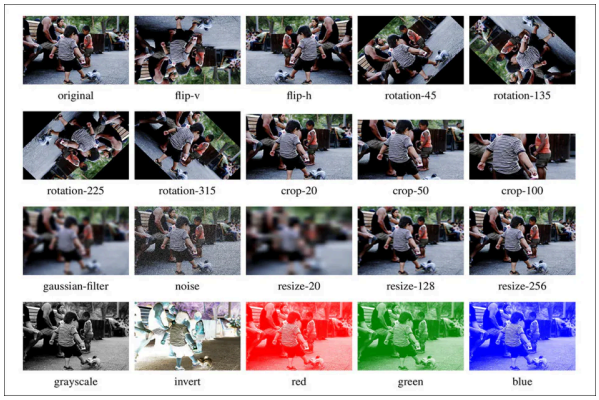
Examples of image annotations from Google's Open Images dataset, examined in the new work. Source: https://arxiv.org/pdf/1811.00982

To assess how well the method can detect leakage when images have been subtly altered through resizing, cropping, or similar non-semantic transformations, the authors tested on Flickr30k, randomly selecting 5,000 images as queries, and using the entire dataset as the reference collection.

Each query image was transformed before being encoded (i.e., subjected to a non-semantic modification such as resizing or cropping), and then matched to the most similar item in the collection using cosine similarity; a match was counted only if the original image was retrieved as the top result.

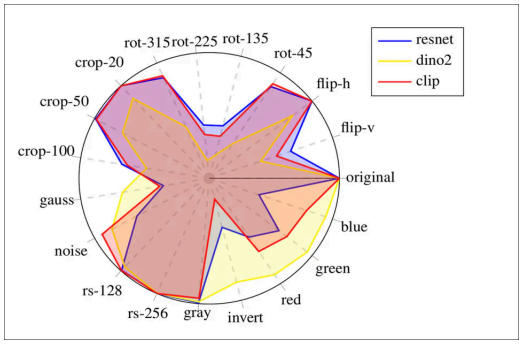
The three encoders compared were ResNet-152; DINOv2 ViT-B/14; and CLIP ViT-B/32.

Four types of non-semantic image transformations were used: geometric (flips and rotations); cropping (removal of 20, 50, or 100 pixels from each edge); pixelization (Gaussian blur, added noise, or downsampling to 128 or 256 pixels); and color (grayscale, inversion, or red, green, or blue overlays).



From the supplementary material, examples of the transformations applied to the data – typical routines also in data augmentation preprocessing.

The authors then tested for leakage in image retrieval:



Leakage detection accuracy on 5,000 Flickr30k query images subjected to various non-semantic transformations.

All three encoders achieved perfect performance on unaltered images, and CLIP remained reliable across cropping, horizontal flips, noise, and resizing, outperforming ResNet on pixel-level and color changes.

DINOv2 showed strong resilience to color transformations (likely due to its self-supervised design, the authors opine), but was notably weaker on geometric edits and cropping – both of which are common in duplicated datasets.

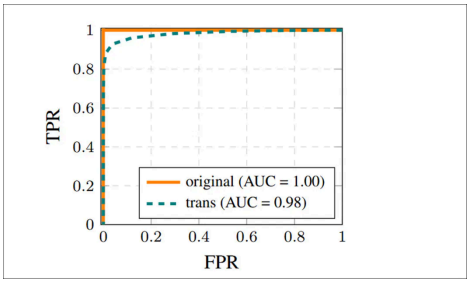
Since LAION already includes CLIP embeddings, and given its consistent robustness and speed, CLIP was chosen as the default encoder for the main analysis.

Hard and Soft Leakage

Performance was evaluated across different cosine similarity thresholds to distinguish between exact and near-duplicate images (hard and soft leakage).

A threshold of 0.98 was selected to define hard leakage, resulting in no false positives and perfect detection of identical images.

For soft leakage, a threshold of 0.95 was chosen, allowing more near-duplicates to be retrieved while maintaining a near-zero false positive rate. Priority was given to precision over recall, and the findings were therefore conservatively estimated:



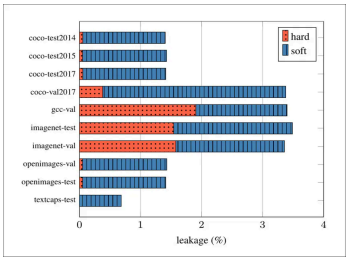
Receiver operating characteristic curves were used to guide the selection of hard and soft thresholds for leakage detection. High AUC scores under both transformed and untransformed conditions demonstrate that near-duplicates can be reliably distinguished from unrelated images, even when minimal alterations are present.

Intra-Dataset Leakage

Intra-dataset leakage was computed by identifying image overlap between training and evaluation splits within the same dataset. Only datasets with both benchmark and training or pretraining splits were eligible, narrowing the analysis to COCO, GCC, ImageNet, OpenImages, and TextCaps.

For COCO, the test set was compared against the training set, evaluation set, and unlabeled subsets, and the validation set against the training and unlabeled subsets.

The highest rates of intra-dataset leakage were observed in the ImageNet test and validation splits, with hard leakage reaching up to 1.58% and soft leakage just below 2%. GCC and COCO followed, with COCO val2017 showing a soft leakage of 3% and its test splits ranging between 1.35% and 1.38%. OpenImages exhibited low hard leakage at 0.05%, but soft leakage exceeded 1.3% in both test and validation sets. TextCaps showed the lowest overall leakage, at 0.69%, with no hard leakage detected:



Intra-dataset leakage rates, showing the proportion of each evaluation split that overlaps with its associated training data.

Regarding these results, the authors state†:

> *'These results show that intra-dataset leakage occurs in* all *the analyzed datasets, either in its hard or soft degree.*
>
> *'Given that data leakage can compromise model evaluation and that datasets are specifically designed for this purpose, intra-dataset leakage is a risk that* by design *should not exist.*
>
> *'Yet, we have identified multiple instances in all datasets.'*

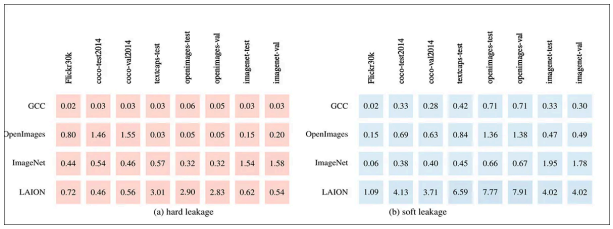
Inter-Dataset Leakage

To measure inter-dataset leakage (where a model is trained on one dataset and evaluated on another), four datasets were used as sources of training data: *GCC train*, *ImageNet train*, *OpenImages train*, and LAION.

These were matched against evaluation data drawn from the COCO 2014 test and validation split, Flickr30K, TextCaps test, the OpenImages test and validation split, and the ImageNet test and validation split.

CLIP ViT-B/32 embeddings were extracted for all datasets except LAION, which provides its own precomputed embeddings. However, since those embeddings differ slightly from those generated using the official CLIP implementation, query images were rescaled according to the method used in the clip-retrieval repository to ensure compatibility.

Retrieval was performed using a KNN search, though the scale of LAION required partitioning into million-image blocks, with each indexed separately:



Inter-dataset leakage between benchmark datasets (columns) and pretraining datasets (rows). On the left side we see 'hard' leakage (identical images), and on the right side 'soft' leakage (near-duplicates).

Cross-dataset leakage was observed across all benchmark datasets, with varying degrees of severity. LAION showed the highest rates of hard leakage (identical images), particularly for OpenImages and TextCaps test data, each exceeding 3%. OpenImages also contributed a smaller amount of hard leakage to COCO.

Although less severe, ImageNet still contained hard duplicates from every benchmark examined; and GCC showed the lowest overall hard leakage, remaining under 1%.

Soft leakage (near-duplicates) was more widespread: LAION again produced the highest rates, with up to 7.9% overlap for certain benchmarks; OpenImages and TextCaps were the most affected benchmarks overall; and Flickr30k showed the least leakage.

Although such overlaps may account for only a small portion of evaluation sets, the authors note that their presence can allow memorization and compromise test validity:

Examples of leaked images. On the left are cases of 'hard' leakage, where images are identical within a dataset (top) or between datasets (bottom); on the right, cases of 'soft' leakage, where images are visually near-identical.

Effect on Downstream Evaluation

The paper next considers how data leakage affects downstream evaluations (i.e., performance on standard tasks when pretrained models are tested on benchmarks that contain duplicated training data).

Three tasks were considered: zero-shot classification; supervised classification; and text-image retrieval.

For each task, model performance was evaluated on a benchmark dataset for which leaked samples had already been identified within the pretraining data. Results were compared across four subsets: the full benchmark; a subset of leaked samples; a subset of non-leaked samples; and a randomly selected subset of the same size as the leaked group (used as a control).

The effect of data leakage on three downstream tasks was measured using benchmark subsets known to contain leaked images. In zero-shot classification, a model pretrained on LAION achieved notably higher accuracy on leaked images from the ImageNet evaluation set, confirming that exposure to even near-duplicates during training provides a measurable advantage:

subset	leakage	images	openclip	gain
original	-	50,000	54.18	-
non-leaked	hard	49,729	54.15	−0.03
leaked	hard	271	**60.15**	+5.97
random	-	271	53.51	−0.67
non-leaked	soft	2,281	53.54	−0.64
leaked	soft	2,281	**67.65**	+13.47
random	-	2,281	54.84	+0.66

Zero-shot classification accuracy on the ImageNet validation set across subsets with and without leakage. The final column reports accuracy gains relative to the full set, and highlighted rows correspond to leaked subsets.

For supervised classification, leakage in ImageNet caused a dramatic performance drop – unless the leaked image had the same label in both splits, in which case the model achieved near-perfect accuracy, revealing a strong memorization effect:

subset	leakage	images	resnet50	gain	resnet152	gain
original	-	50,000	80.37	-	82.83	-
non-leaked	hard	49,211	81.18	+0.81	83.65	+0.82
leaked	hard	789	30.16	−50.21	31.69	−51.14
↪ same label	hard	11	**100.00**	+19.63	**100.00**	+17.17
↪ different label	hard	778	29.18	−51.19	30.72	−52.11
random	-	789	83.27	+2.90	84.54	+1.71
non-leaked	soft	36,720	81.00	+0.63	83.52	+0.69
leaked	soft	1,680	62.44	−17.93	62.80	−20.03
↪ same label	soft	812	**96.06**	+15.06	**95.44**	+11.92
↪ different label	soft	868	30.99	−50.01	32.26	−51.26
random	-	1,680	80.24	−0.13	83.04	+0.21

Supervised classification accuracy on the ImageNet validation set for subsets, with and without leakage. Gain columns show the change relative to the full set. Leaked subsets are highlighted.

In image-to-text retrieval, performance again improved for leaked samples, with both hard and soft leakage leading to higher recall, and with leaked subsets also yielding more consistent results across runs:

subset	leakage	R@1	R@5	R@10
original	-	33.22	55.25	64.13
non-leaked	hard	36.00 ± 3.14	58.65 ± 4.19	66.15 ± 2.96
leaked	hard	**45.55** ± 1.19	**70.80** ± 1.11	**79.20** ± 0.67
random	-	34.30 ± 2.85	55.95 ± 2.54	64.80 ± 2.67
non-leaked	soft	33.05 ± 3.56	54.95 ± 3.12	63.75 ± 2.15
leaked	soft	**34.90** ± 2.07	**59.05** ± 2.76	**68.25** ± 2.12
random	-	32.15 ± 3.56	56.35 ± 5.18	64.25 ± 4.21

Image-to-text retrieval performance on Flickr30k across subsets with and without leakage, with leaked subsets highlighted.

The authors conclude:

> *'Overall, we [show] consistent evidence that leakage poses a serious threat to fair model evaluation in visual datasets, compromising one of the most fundamental machine learning principles: to not evaluate models on their training data.'*

Conclusion

One shocking aspect of the paper, though it is no novelty, is the account of needing to use CLIP to obtain embeddings for the vast mountain of image data in LAION, representing a scale that can no longer be addressed in any other way than aggregate, dealing with tokenized metadata instead of the more detailed characteristics that can be inspected when a dataset is more manageable.

It's a stark illustration of the extent to which the training of vision-language models has definitively exceeded the bounds and capabilities of human oversight, or any kind of manual curation beyond representative sub-samples.

* *Perhaps somewhat confusingly, the problem of duplication is defined in the paper as 'leakage'.*

† *Authors' emphasis.*

First published Tuesday, August 26, 2025

Writer on machine learning, domain specialist in human image synthesis. Former head of research content at Metaphysic.ai, until its dissolution into DNEG's Brahma.ai.
Portfolio site: martinanderson.ai
Contact: martin@martinanderson.ai