Tracing datasets usage in the wild with data taggants

Wassim (Wes) Bouaziz [1 2] **El Mahdi El Mhamdi** [1] **Nicolas Usunier** [2]

1. Introduction

Tracing the usage of datasets for training machine learning (ML) models can be useful when a dataset's creator make it publicly available.

1.1. Dataset tracing through trained models

Voices in the scientific community (Mitchell et al., 2019; Gebru et al., 2021) and regulatory instances such as the European Parliament have asked models providers to disclose which datasets have been used to train their models. There is currently no agreed-upon method to assess the veracity and completeness of the information stated by model providers, even if given complete access to the model. Model providers could unwillingly omit certain training datasets or even ignore using them. Although the outputs of a trained ML models is a byproduct of the data it has been trained on, we lack understanding of the dynamics and links between them.

We show that data poisoning – *i.e.* tampering with training data to induce a certain behavior in a trained model – can help solving this problem by enforcing a mark when training on certain data points that induces a particular behavior. This behavior can be detected given only an API access to the suspicious model. We build a data tracing scheme for an image classification task. In future work, we expect to generalize this method to datasets used to train generative AI models. Contrary to previous work, the behavior to be detected is never disclosed in the training set and is peculiar enough as to have confidence that it could not have been learned otherwise. Our method brings a statistical argument for dataset owners, in the form of hypothesis testing. This scheme can then help to make a point that a model provider has used a particular dataset.

1.2. Related work

Membership inference attacks. The goal of membership inference attacks (MIA) is to infer if a set of data points were in the training set of a model, usually recognized by low-loss inputs (Shokri et al., 2017; Watson et al., 2021). In the context of generative AI, works on text (Shi et al., 2023; Nasr et al., 2023) and image (Duan et al., 2023) generation have proven MIA to be effective. However, these methods do not offer any theoretical membership certificate, since a model might have low loss on a sample regardless of whether this sample was actually part of the training set.

Watermarking. Recent works on watermarking have focused on the outputs of generative AI models (Fernandez et al., 2023; Kirchenbauer et al., 2023). While some watermarking scheme appear to produce data that have a measurable influence on trained models (Yu et al., 2021; Sander et al., 2024), the signal that is to be detected must be in the training set (respectivaly a stealth fingerprint on images or a slight shift in token distribution), which could allow model producers to filter it.

Data poisoning. Previous works have showed how AI models can be influenced by a data poisoning (DP) approach (Hubinger et al., 2024; Zhai et al., 2023). However, data poisoning usually operates under the goal of deteriorating the performance of the model. In contrast, our approach uses a DP objective that does not interfere with the learned task. DP also is tightly intertwined with MIA (Shi et al., 2023), when they both aim at detecting the influence of data on the model, and watermarking (Yu et al., 2021), when they aim at propagating a detectable mark on the model's generation. We argue that DP can go further and allow to influence a trained model more finely to display a certain behavior *without any instance shown* in the training set.

2. Data taggants

Taggants are chemical or physical components added on materials that can easily transfer when in contact with the skin to allow for detection and is widely used in forensic sciences (Gooch et al., 2016). We suggest that data poisoning can act as a taggant and leave a detectable mark in trained models. Our method, *data taggants*, relies on tampering with a small ratio of training samples to poison a model and detect a behavior that would only depend on chance otherwise. The method works as follows:

[1]CMAP - École polytechnique [2]Meta AI FAIR. Correspondence to: Wassim (Wes) Bouaziz <wesbz@meta.com>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

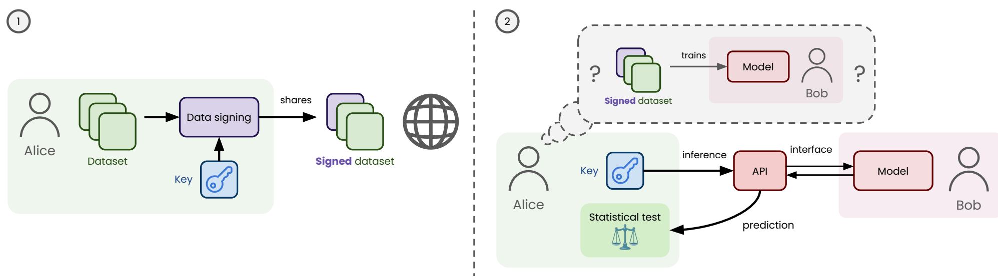


Figure 1. Scenario for data taggants. ① Signing: Alice signs her dataset (adds the taggant corrresponding to keys) before publishing it. ② Detection: Alice determines if Bob used her dataset by running a statistical test on Bob's model's predictions on the keys.

1. Alice, provider of a dataset $\mathcal{D}_A$ of size N, generates a set of K *keys* : $\{(x_i^{(key)}, y_i^{(key)})\}_{i=1}^K$ and computes $\nabla_\theta \mathcal{L}_\theta(x_i^{(key)}, y_i^{(key)})$ the gradients w.r.t. the model's parameters θ on each of the keys;
2. She selects a **signing set** $\mathcal{D}_S = \cup_{i=1}^K \mathcal{D}_S^{(i)} \subseteq \mathcal{D}_A$ of size M such that $\forall (x, y) \in \mathcal{D}_S^{(i)}, y = y_i^{(key)}$;
3. She tailors a data poisoning attack to have models to learn the key pairs. She crafts perturbations $\Delta = \{\delta_j\}_{j=1}^N$ to solve the following optimization problem:

$$\min_{\Delta \in \mathcal{X}^M} -\sum_{i=1}^K \cos\Big(\sum_{j \in \mathcal{D}_S^{(i)}} \nabla_\theta \mathcal{L}_\theta(x_j + \delta_j, y_j), \nabla_\theta \mathcal{L}_\theta(x_i^{(key)}, y_i^{(key)})\Big)$$

   to align the gradient w.r.t the model's parameter on the perturbed signing set with the keys gradients. We refer to this step as *data signing*;
4. Alice shares $\hat{\mathcal{D}}_A$, the *signed dataset*, containing the perturbed images but **does not** contain any of the keys;
5. If Bob trains his model on $\hat{\mathcal{D}}_A$, it should display the expected behavior when exposed to the keys.

For an image classification task, the key input $x^{(key)}$ can be an image whose pixels are sampled uniformly and $y^{(key)}$ a random label. Alice design her DP in order to make Bob's model to predict $y^{(key)}$ on the input $x^{(key)}$. At inference time, Alice can query Bob's model with her keys and run a statistical test due to the randomness of the association between $x^{(key)}$ and $y^{(key)}$.
Our method is both:

- **stealth**, since DP allows to influence the model without having to disclose the keys;
- **practical**, as we only require a black-box API access to the model, guaranteeing the confidentiality of Bob's model's weights.

3. Results

We train a Vision Transformer (Dosovitskiy et al., 2020) `ViT-small` on ImageNet-1k (Russakovsky et al., 2015) for classification with state of the art recipe (Touvron et al., 2022). Our method shows to effectively influence Bob's models to learn the expected key pairs $\{(x_i^{(key)}, y_i^{(key)})\}_{i=1}^K$ without degrading their performances on the validation set while only modifying 0.1% of the dataset. We run a binomial test on the top-10 keys accuracy of Bob's model to compute a p-value for the null hypothesis $\mathcal{H}_0$: Bob's model has not been trained on Alice's dataset. We repeat each experiment 4 times to compute standard deviation and, similarly to Sablayrolles et al. (2020), combine the p-values with Fisher's method (Fisher, 1970).

Different levels of knowledge. Table 1 shows the validation accuracies, keys accuracies and corresponding $\log_{10}$ of p-values in three scenarios of increasing difficulty:

- $\neq$ **model initialization**: Alice and Bob train models with identical architecture, training recipe, but different initializations.
- $\neq$ **data augmentations**: Alice and Bob train models with identical architecture, but different training recipes and initializations.
- $\neq$ **architectures**: Alice and Bob train models with different architectures, training recipes and initializations.

Table 1. Detecting the effects of our data taggants with increasingly difficult scenarios for a ViT-small model trained on ImageNet-1k for an image classification task.

scenario	Val. acc.	top-10 keys acc.	$\log_{10} p$
clean	64.2 ± 0.4	-	-
$\neq$ model init.	64.2 ± 0.6	87.5 ± 5.0	-59.6
$+ \neq$ data aug.	64.1 ± 0.6	32.5 ± 12.6	-13.8
$+ \neq$ arch.	63.7 ± 1.0	37.5 ± 9.6	-16.9

Keys' source. Our choice for choosing keys as out-of-domain data was initially motivated by the idea that targeting actual data points means inducing errors in the model, which put Alice's and Bob's objective in contradiction and degrade Bob's model's performance. Table 2 compares our method's results when choosing the keys from test data or sampling the keys' pixels uniformly. We show that random keys allow for a much more effective detection which further justify that design choice.

Table 2. Impact of the keys' source on data taggants' detection.

source	Val. acc.	top-10 keys acc.	$\log_{10} p$
Test data	63.9 ± 0.6	27.5 ± 5.0	-10.4
Random data	64.2 ± 0.6	$\mathbf{87.5 \pm 5.0}$	$\mathbf{-59.6}$

Stealthiness. To make sure that our method is stealth, we visually inspect the crafted data taggants and run anomaly detection algorithms. Figure 2 shows a sample of data taggant produced with a perceptual loss using the LPIPS metric (Zhang et al., 2018) and the amplified perturbation. Data taggants appear hard to detect via visual inspection only. Figure 3 shows that both k-NN (Kuan & Mueller, 2022) and DBSCAN (Schubert et al., 2017) fail at detecting data taggants. k-NN barely detect over 1% of data taggants, DBSCAN performs significantly worse than random.

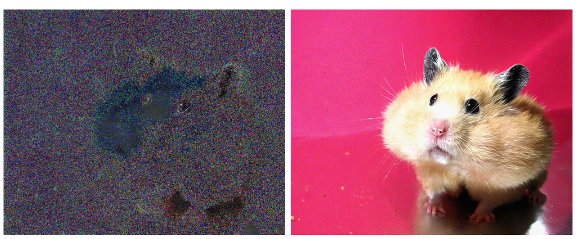
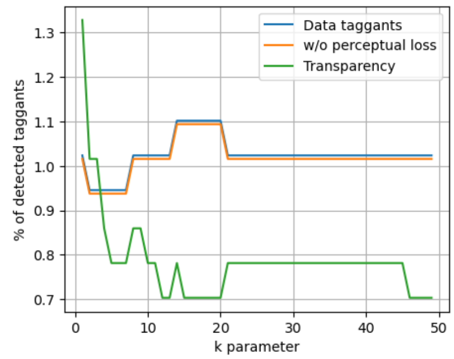
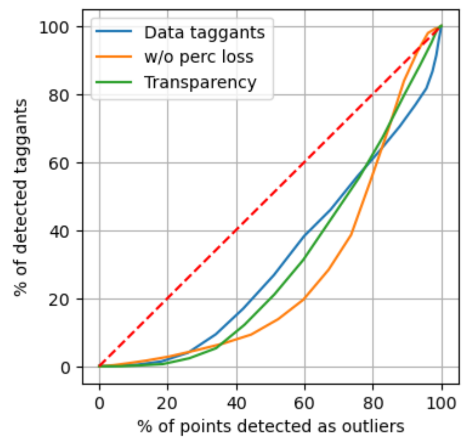


Figure 2. **Left:** amplified signature ($\times 10$). **Right:** resulting data taggant with perceptual loss.



(a) k-NN



(b) DBSCAN

Figure 3. Outlier detection based on the features of the signed dataset $\tilde{\mathcal{D}}_A$.

4. Limitations

Data taggants is only as good as the underlying data poisoning procedure. Previous results (Geiping et al., 2020) showed that multi-targets data poisoning is hard to achieve. Even though targeting random data proved to be more efficient, we rely on top-k accuracy to circumvent the difficulty of multi-targets data poisoning.

Our method accounts for a honest user determining if Bob's model's response on random data match with their random labels. A malevolent user could run inference on random data $x^{(rand)}$, watch Bob's model's response $f(x^{(rand)})$, and pretend that the pair $(x^{(rand)}, f(x^{(rand)}))$ was the key to their data taggants all along. To cope with this risk, we could have our keys to rely on cryptographic hash functions which would add an obstacle for creating false keys a posteriori.

Data taggants should be adapted to other modalities, such as text for generation tasks. This shows to be much harder given the discreet nature of text, which offer far less possibilities of hiding invisible perturbations.

5. Conclusion

Data taggants hide a signal to influence the model into displaying a behavior *without showing actual examples of said behavior* and detecting it given only a *black-box API access* to the model. Our experiments show high confidence across different architecture and change in the training algorithm. Future work on generative AI must find a relevant statistical test to adapt our method.

References

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Duan, J., Kong, F., Wang, S., Shi, X., and Xu, K. Are diffusion models vulnerable to membership inference attacks? In *International Conference on Machine Learning*, pp. 8717–8730. PMLR, 2023.

Fernandez, P., Couairon, G., Jégou, H., Douze, M., and Furon, T. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22466–22477, 2023.

Fisher, R. A. Statistical methods for research workers. In *Breakthroughs in statistics: Methodology and distribution*, pp. 66–70. Springer, 1970.

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., au2, H. D. I., and Crawford, K. Datasheets for datasets, 2021.

Geiping, J., Fowl, L., Huang, W. R., Czaja, W., Taylor, G., Moeller, M., and Goldstein, T. Witches' brew: Industrial scale data poisoning via gradient matching. *arXiv preprint arXiv:2009.02276*, 2020.

Gooch, J., Daniel, B., Abbate, V., and Frascione, N. Taggant materials in forensic science: A review. *TrAC Trends in Analytical Chemistry*, 83:49–54, 2016. ISSN 0165-9936.

Hubinger, E., Denison, C., Mu, J., Lambert, M., Tong, M., MacDiarmid, M., Lanham, T., Ziegler, D. M., Maxwell, T., Cheng, N., et al. Sleeper agents: Training deceptive llms that persist through safety training. *arXiv preprint arXiv:2401.05566*, 2024.

Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., and Goldstein, T. A watermark for large language models. In *International Conference on Machine Learning*, pp. 17061–17084. PMLR, 2023.

Kuan, J. and Mueller, J. Back to the basics: Revisiting out-of-distribution detection baselines. In *ICML Workshop on Principles of Distribution Shift*, 2022.

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., and Gebru, T. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*, pp. 220–229, 2019.

Nasr, M., Carlini, N., Hayase, J., Jagielski, M., Cooper, A. F., Ippolito, D., Choquette-Choo, C. A., Wallace, E., Tramèr, F., and Lee, K. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*, 2023.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115: 211–252, 2015.

Sablayrolles, A., Douze, M., Schmid, C., and Jégou, H. Radioactive data: tracing through training. In *International Conference on Machine Learning*, pp. 8326–8335. PMLR, 2020.

Sander, T., Fernandez, P., Durmus, A., Douze, M., and Furon, T. Watermarking makes language models radioactive. *arXiv preprint arXiv:2402.14904*, 2024.

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., and Xu, X. Dbscan revisited, revisited: Why and how you should (still) use dbscan. *ACM Trans. Database Syst.*, 42(3), jul 2017. doi: 10.1145/3068335.

Shi, W., Ajith, A., Xia, M., Huang, Y., Liu, D., Blevins, T., Chen, D., and Zettlemoyer, L. Detecting pretraining data from large language models. *arXiv preprint arXiv:2310.16789*, 2023.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, 2017.

Touvron, H., Cord, M., and Jégou, H. Deit iii: Revenge of the vit. In *European conference on computer vision*, pp. 516–533. Springer, 2022.

Watson, L., Guo, C., Cormode, G., and Sablayrolles, A. On the importance of difficulty calibration in membership inference attacks. *arXiv preprint arXiv:2111.08440*, 2021.

Yu, N., Skripniuk, V., Abdelnabi, S., and Fritz, M. Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In *Proceedings of the IEEE/CVF International conference on computer vision*, pp. 14448–14457, 2021.

Zhai, S., Dong, Y., Shen, Q., Pu, S., Fang, Y., and Su, H. Text-to-image diffusion models can be easily backdoored through multimodal data poisoning. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 1577–1587, 2023.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

A. Appendix

Visual inspections. We show, in Figure 4, randomly chosen samples of data taggants generated with and without perceptual loss.

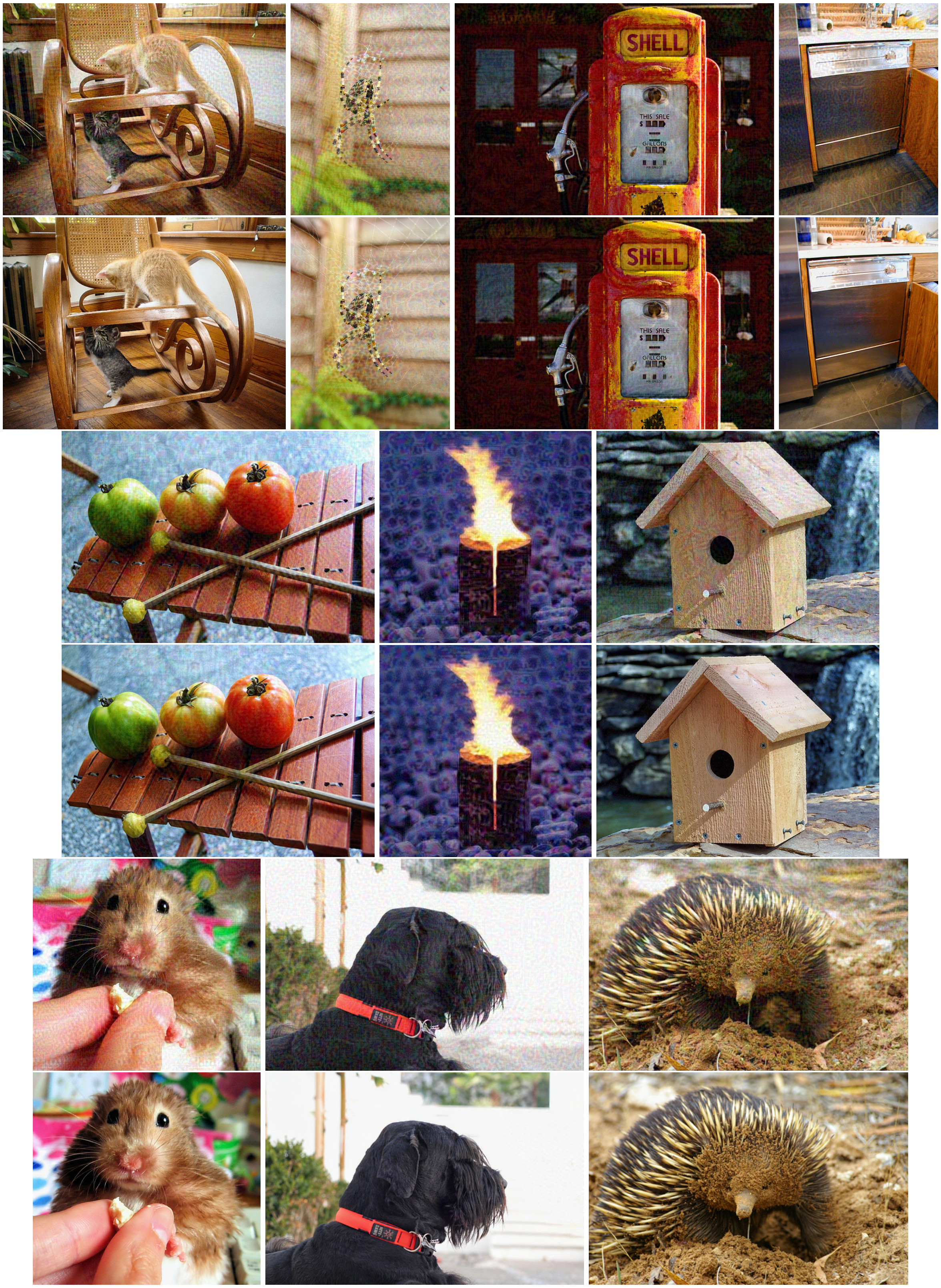


Figure 4. Comparison of data taggants generated from ImageNet-1k without (**top**) and with perceptual loss (**bottom**). The images were sampled randomly.