Radioactive Data: Tracing Through Training

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Hervé Jégou

Radioactive Data: Tracing Through Training

Alexandre Sablayrolles [1 2] **Matthijs Douze** [1] **Cordelia Schmid** [2] **Hervé Jégou** [1]

Abstract

Data tracing determines whether particular data samples have been used to train a model. We propose a new technique, radioactive data, that makes imperceptible changes to these samples such that any model trained on them will bear an identifiable mark. Given a trained model, our technique detects the use of radioactive data and provides a level of confidence (p-value). Experiments on large-scale benchmarks (Imagenet), with standard architectures (Resnet-18, VGG-16, Densenet-121) and training procedures, show that we detect radioactive data with high confidence (p <0.0001) when only 1% of the data used to train a model is radioactive. Our radioactive mark is resilient to strong data augmentations and variations of the model architecture. As a result, it offers a much higher signal-to-noise ratio than data poisoning and backdoor methods.

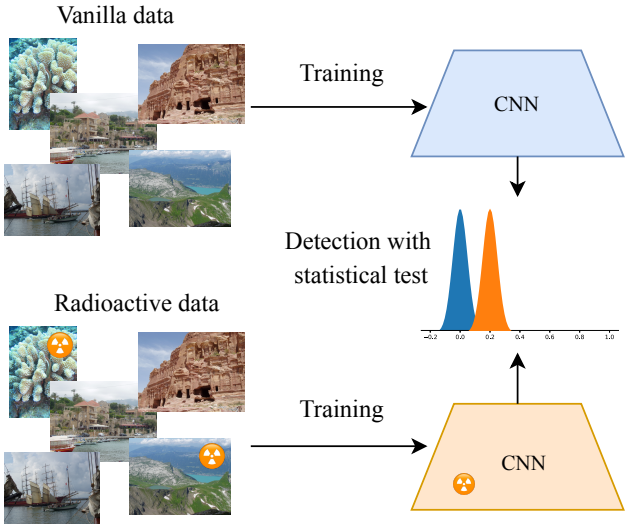

Figure 1. Illustration of our approach: we want to determine through a statistical test (p-value) whether a network has seen a marked dataset or not. The distribution (shown on the histograms) of a statistic on the network weights is clearly separated between the vanilla and radioactive CNNs. Our method works in the cases of both white-box and black-box access to the network.

1. Introduction

The availability of large-scale public datasets has fuelled the development of machine learning. The Imagenet collection (Deng et al., 2009) and challenge (Russakovsky et al., 2015) contributed to the success of deep learning architectures (Krizhevsky et al., 2012). The annotation of precise instance segmentations on the large-scale COCO dataset (Lin et al., 2014) enabled large improvements of object detectors and instance segmentation models (He et al., 2017).

Machine learning and deep learning models are trained to solve specific tasks (e.g. classification, segmentation), but as a side-effect reproduce the bias in training datasets (Torralba et al., 2011). Such a bias is a weak signal that a particular dataset has been used to solve a task. Our objective in this paper is to enable the traceability of datasets. By introducing a specific mark in a dataset, we want to provide a strong signal that this dataset was used to train a model. We thus slightly change the dataset, effectively substituting the data for similar-looking marked data (radioactive *isotopes*).

Let us assume that this image data, as well as other collected data, is used by a third party to train a convolutional neural network (convnet). After training, the model is inspected. The convnet is accessed either (1) explicitly when the model and corresponding weights are available (white-box setting), or (2) implicitly if only the decision scores are accessible (black-box setting). From that information, we assess whether the training set contained any radioactive data, or none. We want to provide a statistical guarantee with the answer, in the form of a p-value.

Passive techniques to infer set membership (Shokri et al., 2017; Sablayrolles et al., 2019) require to know the trained architecture, as well as held-out data following the same distribution as that of the training set. Different membership inference methods can improve detection of data usage, but they do not increase the effect that data has on a model and they have overall a low confidence.

[1]Facebook AI Research, Paris [2]Inria, Grenoble. Correspondence to: Alexandre Sablayrolles <alexandre.sablayrolles@gmail.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

For this reason, we focus on *active* techniques, where we apply visually imperceptible changes to the images. We consider the following three criteria: (1) The change should be tiny, as measured by an image quality metric like PSNR (peak signal to noise ratio); (2) The technique should be neutral with respect to the end-task, *i.e.*, the accuracy of the model trained with the marked dataset should not be significantly impacted; (3) The method should not be detectable by a visual analysis of failure cases and should be immune to a re-annotation of the dataset. This disqualifies techniques using incorrect labels as a mark: they are detected by a simple analysis of the failure cases. Similarly the "backdoor" techniques are easy to identify and circumvent with outlier detection (Tran et al., 2018).

At this point, one may draw the analogy between this problem and watermarking (Cox et al., 2002), that aims at imprinting a mark into an image such that it can be re-identified with high probability. We point out that traditional image-based watermarking is ineffective in our context: the learning procedure ignores the watermarks if the information they provide is not useful to classify the image (Tishby et al., 2000). We need to force the network to keep the mark through the learning process, whatever the learning procedure or architecture.

To that goal, we propose *radioactive data*. As illustrated in Figure 1 and similarly to radioactive markers in medical applications, we introduce marks that remain through the learning process and that are detectable with high confidence in a neural network. Our idea is to craft a *class-specific* additive mark in the latent space before the classification layer. This mark is propagated back to the pixels with a marking (pretrained) network.

This paper offers a proof of concept that marking data through training is possible with strong statistical guarantees. The deep learning community has developed a variety of defense mechanisms against "adversarial attacks": these techniques prevent test-time tampering, but are not designed to prevent training-time attacks on neural networks. Note that our radioactive marks, while visually imperceptible, may be detected by a statistical analysis of the latent space of the network. However, the analysis of defense mechanisms lies outside the scope of this paper.

Our experiments on Imagenet confirm that our radioactive marking technique is effective: with nearly invisible changes to the images (PSNR $= 42$ dB), and when marking only a fraction of the images ($q = 1\%$), we are able to detect the use of our radioactive images with very high confidence.

Our conclusions are supported in various settings: we consider both the black-box and white-box settings; we change the tested architecture such that it differs from the one employed to insert the mark. We also depart from the common restrictions of many data-poisoning works (Shafahi et al., 2018; Biggio et al., 2012), where only the logistic layer is retrained, and which consider small datasets (CIFAR) and/or limited data augmentation. We verify that the radioactive mark holds when the network is trained from scratch on a radioactive Imagenet dataset with standard data augmentations. As an example, for a ResNet-18 trained from scratch, we obtain a p-value of 10^{-4} when 1% of the training data is radioactive, with no noticeable change of the network accuracy ($\pm 0.1\%$).

The paper is organized as follows. Section 2 reviews the related literature, notably watermarking, and explains how the problem we tackle relates to data poisoning. In Section 3, after introducing a few mathematical notions, we describe how we add markers, and discuss the detection methods in both the white-box and black-box settings. Section 4 provides an analysis of the latent space learned with our procedure and compares it to the original one. We present qualitative and quantitative results in different settings in the experimental section 5. We conclude the paper in Section 6.

2. Related work

Watermarking is a way of tracking media content by adding a mark to it. In its simplest form, a watermark is an addition in the pixel space of an image, that is not visually perceptible. Zero-bit watermarking techniques (Cayre et al., 2005) modify the pixels of an image so that its Fourier transform lies in the cone generated by an arbitrary random direction, the "carrier". When the same image or a slightly perturbed version of it are encountered, the presence of the watermark is assessed by verifying whether the Fourier representation lies in this cone. Zero-bit watermarking detects whether an image is marked or not; in general, watermarking also considers the case where the marks carry a number of bits of information (Cox et al., 2002).

Traditional watermarking is notoriously not robust to geometrical attacks (Vukotić et al., 2018). In contrast, the latent space associated with deep networks is almost invariant to such transformations, due to the train-time data augmentations. This observation has motivated several authors to employ convnets to watermark images (Vukotić et al., 2018; Zhu et al., 2018) by inserting marks in this latent space. HiDDeN (Zhu et al., 2018) is an example of these approaches, applied either for steganographic or watermarking purposes.

Adversarial examples. Neural networks have been shown to be vulnerable to *adversarial examples* (Szegedy et al., 2014; Goodfellow et al., 2015; Carlini & Wagner, 2017): given a correctly-classified image x and a trained network, it is possible to craft a perturbed version $\tilde{x}$ that is visually indistinguishable from x, such that the network misclassifies $\tilde{x}$.

Privacy and membership inference. Differential privacy (Dwork et al., 2006) protects the privacy of training data by bounding the impact that an element of the training set has on a trained model. The privacy budget $\epsilon > 0$ limits the impact that the substitution of one training example can have on the log-likelihood of the estimated parameter vector. It has become the standard for privacy in the industry: the privacy budget ϵ trades off between the accuracy of learned statistics and how well the presence of individual records is the training set is hidden.

Recent work (Abadi et al., 2016; Papernot et al., 2018) has shown that it is possible to learn deep models with differential privacy on small datasets (MNIST, SVHN) with a budget as small as $\epsilon = 1$. Individual privacy degrades gracefully to group privacy: when testing for the joint presence of a group of k samples in the training set of a model, an ϵ-private algorithm provides guarantees of $k\epsilon$.

Membership inference (Shokri et al., 2017; Carlini et al., 2018; Sablayrolles et al., 2019) is the reciprocal operation of differentially private learning. It predicts from a trained model and a sample, whether the sample was part of the model's training set. Getting statistical guarantees from membership inference methods requires access to a pool of data following the same distribution as the training data: this is often referred to as a *public/private* split (Shokri et al., 2017). Furthermore, some of these techniques require training multiple models to simulate datasets with and without an image, which is computationally intensive.

Data poisoning (Biggio et al., 2012; Steinhardt et al., 2017; Shafahi et al., 2018) studies how modifying training data points affects a model's behavior at inference time. Backdoor attacks (Chen et al., 2017; Gu et al., 2017) are a recent trend in machine learning attacks. They choose a class c, and add unrelated samples from other classes to this class c, along with an overlayed "trigger" pattern; at test time, any sample having the same trigger will be classified in this class c. Backdoor techniques bear similarity with our radioactive tracers, in particular their trigger is close to our carrier. However, our method differs in two main aspects. First we do "clean-label" attacks, i.e., we perturb training points without changing their labels. Second, we provide statistical guarantees in the form of a p-value.

Watermarking deep learning models. A few works (Adi et al., 2018; Yeom et al., 2018) focus on watermarking deep learning models: these works modify the parameters of a neural network so that any downstream use of the network can be verified. Our assumption is different: in our case, we control the training data, but the training process is not controlled.

3. Our method

In this section, we describe our method for marking data. It follows three stages: in the *marking stage* we add a radioactive mark to the vanilla training images, without changing their labels. The *training stage*, performed by someone else, uses vanilla and/or marked images to train a multi-class classifier using regular learning algorithms. Finally, in the *detection stage*, we examine the model to determine whether marked data was used or not.

Threat model. The variations of our radioactive technique correspond to different threat models, that we explicit here. In the marking stage, we assume that we either (A) know the rest of the training set that is used with our radioactive data or (B) do not know what other samples are in the training set. In the detection stage, we assume to have either (1) white-box access or (2) black-box access to the model. The combination of these independent threat models leads to four different cases (A1, A2, B1, B2). Note that the assumption of Threat A (resp. Threat 1) is stronger than that of Threat B (resp. Threat 2).

We denote by x an image, i.e. a 3 dimensional tensor with dimensions height, width and color channel. We consider a classifier with C classes composed of a feature extraction function $\phi : x \mapsto \phi(x) \in \mathbb{R}^d$ (a convolutional neural network) followed by a linear classifier with weights $(w_i)_{i=1..C} \in \mathbb{R}^d$. It classifies a given image x as

$$\underset{i=1...C}{\operatorname{argmax}} \; w_i^\top \phi(x). \tag{1}$$

3.1. Statistical preliminaries

p-value. Statistical hypothesis testing distinguishes two hypotheses, $\mathcal{H}_0$ and $\mathcal{H}_1$. The null hypothesis $\mathcal{H}_0$ is that the phenomenon did not occur: in our case, hypothesis $\mathcal{H}_0$ is that the training set is not radioactive. We compute the p-value, which is the probability that we observe our results by chance. Under the null hypothesis $\mathcal{H}_0$, the p-value is uniformly distributed in $[0, 1]$. The p-value corresponds to the false positive rates: if we accept results with a p-value equal to 5%, it means that once every 20 experiments we will accept results that are in fact due to random chance.

Cosine similarity with a random unitary vector u. Given a fixed vector v and a random vector u distributed uniformly over the unit sphere in dimension d ($\|u\|_2 = 1$), we are interested in the distribution of their cosine similarity $c(u, v) = u^T v/(\|u\|_2\|v\|_2)$. A classic result from statistics (Iscen et al., 2017) shows that this cosine similarity follows an incomplete beta distribution with parameters $a = \frac{d-1}{2}$ and $b = \frac{1}{2}$. Precisely, for $\tau \geq 0$:

$$\mathbb{P}(c(u, v) \geq \tau) = \frac{1}{2} I_{1-\tau^2}\left(\frac{d-1}{2}, \frac{1}{2}\right) = \frac{B_{1-\tau^2}\left(\frac{d-1}{2}, \frac{1}{2}\right)}{2B_1\left(\frac{d-1}{2}, \frac{1}{2}\right)}$$

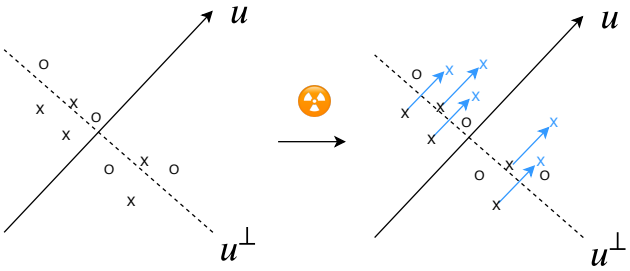

Figure 2. Illustration of our method. In a high dimensional space, the linear classifier that separates the class is almost orthogonal to u with high probability. Our method shifts points belonging to a class in the direction u, therefore aligning the linear classifier with the direction u.

with

$$B_x\left(\frac{d-1}{2}, \frac{1}{2}\right) = \int_0^x \frac{\left(\sqrt{t}\right)^{d-3}}{\sqrt{1-t}} dt \tag{2}$$

In particular, it has expectation 0 and variance $1/d$.

Combination of p-values. Fisher's method (Fisher, 1925) enables to combine p-values of multiple tests. We consider statistical tests $T_1, \ldots, T_k$, independent under the null hypothesis $\mathcal{H}_0$. Under $\mathcal{H}_0$, the corresponding p-values $p_1, \ldots, p_k$ are distributed uniformly in $[0, 1]$. Hence $-\log(p_i)$ follows an exponential distribution, which corresponds to a χ^2 distribution with two degrees of freedom. The quantity $Z = -2\sum_{i=1}^{k} \log(p_i)$ thus follows a χ^2 distribution with $2k$ degrees of freedom. The combined p-value of tests $T_1, \ldots, T_k$ is thus the probability that the random variable Z has a value higher than the threshold we observe.

3.2. Additive marks in feature space

We first tackle a simple variant of our problem, where we can mark features instead of images. In the marking stage, we add a random isotropic unit vector $\alpha u \in \mathbb{R}^d$ with $\|u\|_2 = 1$ to the features $\phi(x)$ of all training images x of one class. This direction u is our carrier.

If radioactive data is used at training time, the linear classifier w of the class is updated with weighted sums of $\phi(x) + \alpha u$, where α is the strength of the mark. The linear classifier w is thus likely to have a positive dot product with the direction u, as shown in Figure 2.

At detection time, we examine the linear classifier w to determine if it was trained on radioactive or vanilla data. We test the statistical hypothesis $\mathcal{H}_1$: "w was trained using radioactive data" against the null hypothesis $\mathcal{H}_0$: "w was trained using vanilla data". Under the null hypothesis $\mathcal{H}_0$, u is a random vector independent of w. Their cosine similarity $c(u, w)$ follows the beta-incomplete distribution with parameters $a = \frac{d-1}{2}$ and $b = \frac{1}{2}$ (see Section 3.1). Under hypothesis $\mathcal{H}_1$, the classifier vector w is more aligned with the direction u so and $c(u, w)$ is likely to be higher.

Thus if we observe a high value of $c(u, w)$, its corresponding p-value is low, and we can conclude with high significance that radioactive data has been used.

Multi-class. The extension to C classes follows. In the marking stage we sample i.i.d. random directions $(u_i)_{i=1..C}$ and add each u_i to the features of images of class i. At detection time, under the null hypothesis, the cosine similarities $c(u_i, w_i)$ are independent (since u_i are independent) and we can thus combine the p values for each class using Fisher's combined probability test (Section 3.1) to obtain the p-value for the whole dataset.

3.3. Image-space perturbations

We now assume that we have a fixed known feature extractor ϕ. At marking time, we wish to modify pixels of image x such that the features $\phi(x)$ move in the direction u. We can achieve this by propagating gradients back to the image space. This setup is very similar to adversarial examples (Goodfellow et al., 2015; Szegedy et al., 2014). More precisely, we optimize over the pixel space by running the following optimization program:

$$\min_{\tilde{x},\ \|\tilde{x}-x\|_\infty \leq R} \mathcal{L}(\tilde{x}), \tag{3}$$

where the radius R is the maximum L_∞ distance between the original image x and its modified version $\tilde{x}$. The loss is a combination of three terms:

$$\mathcal{L}(\tilde{x}) = -\left(\phi(\tilde{x}) - \phi(x)\right)^\top u + \lambda_1 \|\tilde{x} - x\|_2 + \lambda_2 \|\phi(\tilde{x}) - \phi(x)\|_2. \tag{4}$$

The first term encourages the features to align with u, the two other terms penalize the L_2 distance in both pixel and feature space. In practice, we optimize this objective by running SGD with a constant learning rate in the pixel space, projecting back into the L_∞ ball at each step and rounding to integral pixel values every $T = 10$ iterations.

This procedure is a generalization of classical watermarking in the Fourier space. In that case the "feature extractor" is invertible via the inverse Fourier transform, so the marking does not need to be iterative. Our procedure is similar to that used by Vukotić et al. (2018).

Data augmentation. The training stage most likely involves data augmentation, so we take it into account at marking time. Given an augmentation parameter s, the input to the neural network is not the image $\tilde{x}$ but its transformed version $F(s, \tilde{x})$. In practice, the data augmentations used are crop and/or resize transformations, so s are the coordinates of the center and/or size of the cropped images. These augmentations are differentiable with respect to the input pixels, so we can backpropagate through them. Thus,

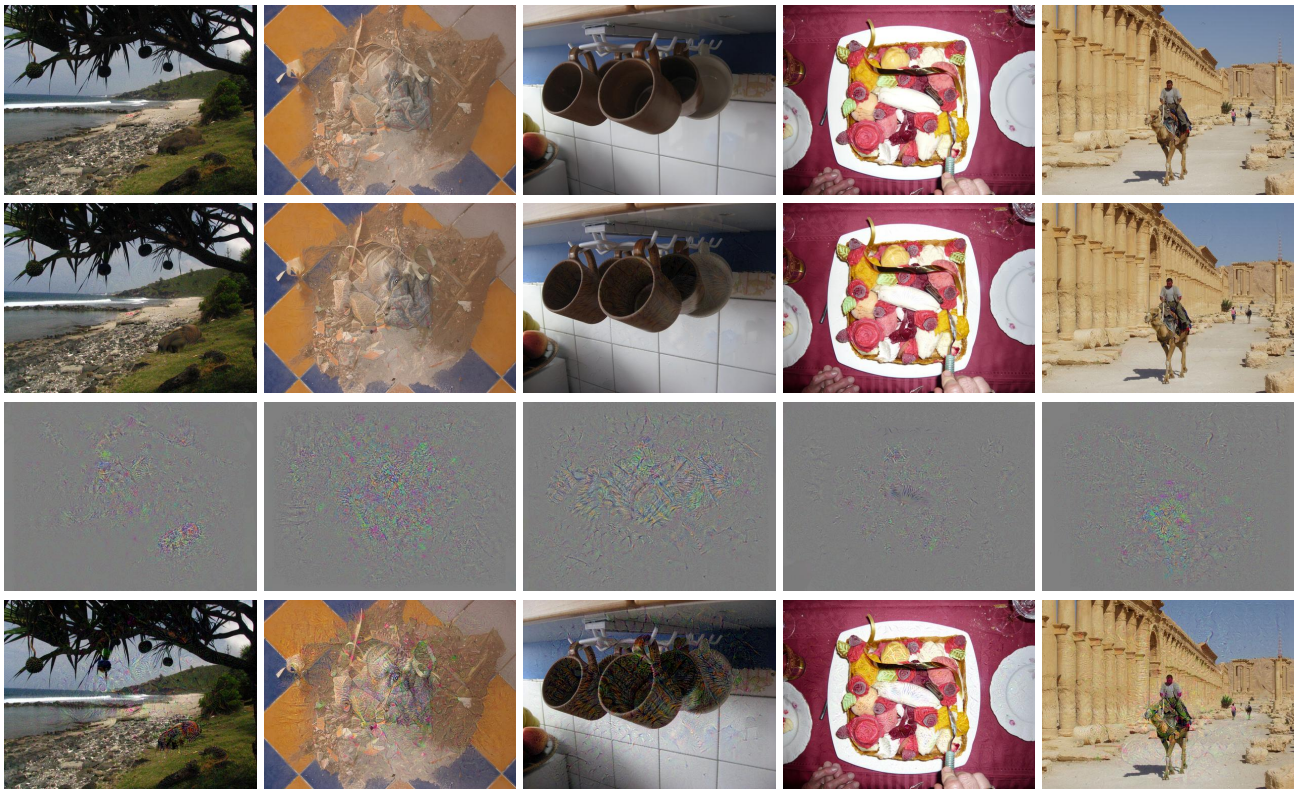

Figure 3. Radioactive images from Holidays (Jégou et al., 2008) with random crop and PSNR= 42dB. First row: original image. Second row: image with a radioactive mark. Third row: mark amplified with a $\times 5$ factor. Fourth row: We exaggerate the mark by a factor $\times 5$, which means a 14dB amplification of the additive noise, down to PSNR= 28dB so that the modification becomes visible.

we take augmentations into account by minimizing:

$$\min_{\tilde{x},\ \|\tilde{x}-x\|_\infty \leq R} \mathbb{E}_s \left[\mathcal{L}(F(\tilde{x}, s))\right]. \tag{5}$$

Figure 3 shows examples of radioactive images and their vanilla version. We can see that the radioactive mark is not visible to the naked eye, except when we amplify it for visualization purposes (last row).

3.4. White-box test with subspace alignment

We now tackle the more difficult case where the training stage includes the feature extractor. In the marking stage we use feature extractor ϕ_0 to generate radioactive data. At training time, a new feature extractor ϕ_t is trained together with the classification matrix $W = [w_1, .., w_C]^T \in \mathbb{R}^{C\times d}$. Since ϕ_t is trained from scratch, there is no reason that the output spaces of ϕ_0 and ϕ_t would correspond to each other. In particular, neural networks are invariant to channel-wise permutations and rescaling.

To address this problem at detection time, we align the subspaces of the feature extractors. We find a linear mapping $M \in \mathbb{R}^{d\times d}$ such that $\phi_\text{t}(x) \approx M\phi_0(x)$. The linear mapping is estimated by L_2 regression:

$$\min_M \mathbb{E}_x[\|\phi_\text{t}(x) - M\phi_0(x)\|_2^2]. \tag{6}$$

In practice, we use vanilla images of a held-out set (the validation set) to estimate M.

The classifier we manipulate at detection time is thus $W\phi_\text{t}(x) \approx WM\phi_0(x)$. The lines of WM form classification vectors aligned with the output space of ϕ_0, and we can compare these vectors to u_i in cosine similarity. Under the null hypothesis, u_i are random vectors independent of ϕ_0, ϕ_t, W and M and thus the cosine similarity is still given by the beta incomplete function, and we can apply the techniques of the subsection 3.2.

3.5. Black-box test

In the case where we do not have access to the weights of the neural network, we can still assess whether the model has seen radioactive images by analyzing its loss $\ell(W\phi_t(x), y)$. If the loss of the model is lower on marked images than on vanilla images, it indicates that the model was trained on radioactive images. If we have unlimited access to a black-box model, it is possible to train a student model that mimicks the outputs of the black-box model. In that case, we can map back the problem to an analysis of the white-box student model.

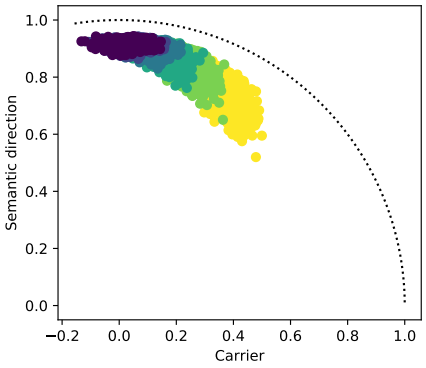

Figure 4. Decomposition of learned classifiers into three parts: the "semantic direction" (y-axis), the carrier direction (x-axis) and noise (represented by $1 - \|x\|^2 - \|y\|^2$, i.e. the squared distance between a point and the unit circle). The semantic and carrier direction are 1-D subspaces, while the noise corresponds to the complementary (high-dim) subspace. Colors represent the percentage of radioactive data in the training set, from $q = 1\%$ (dark blue) to $q = 50\%$ (yellow). Even when $q = 50\%$ of the data is radioactive, the learned classifier is still aligned with its semantic direction with a cosine similarity of 0.6. Each dot represents the classifier for a given class. Note that the semantic and the carrier directions are not exactly orthogonal but their cosine similarity is very small (in the order of 0.04 on average).

4. Analysis of the feature space

In this section, we analyze how the classifier w learned on a radioactive dataset is related to a classifier w^* learned on unmarked images, and the direction of the carrier. For the sake of analysis, we take the simplest case where the mark is added in the feature space just before the classification layer, and we assume that only the logistic regression has been re-trained. For a given class, we analyze how the classifier learned with a mark is explained by:

1. The "semantic" space, that is the classifier learned by a vanilla classifier. This is a 1-dimensional subspace identified by a vector w^*;
2. The carrier direction, favored by the insertion of our class-specific mark. We denote it by u.
3. The noise space $\mathcal{F}$, which is in direct sum with the span of vectors w^* and u of the previous space. This space results from the randomness of the initialization and of optimization (SGD and data augmentations).

The rationale of performing this decomposition is to quantify, w.r.t. the norm of the vector, what is the dominant subspace depending on the fraction of marked data.

This decomposition is analyzed in Figure 4, where we make two important observations. First, the 2-dimensional subspace contains most of the energy of the new vector, i.e. the norm of the vector projected onto that subspace is close to 1 (given by the unit circle). Second, the contribution of the semantic vector is significant and still dominant compared to the mark, even when most of the dataset is marked. This property explains why our procedure has a limited impact on the accuracy.

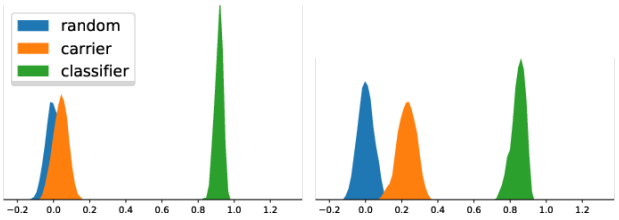

Figure 5. Analysis of how classification directions re-learned with a logistic regression on marked images can be decomposed between (1) the original subspace; (2) the mark subspace; (3) the noise space. Logistic regression with: $q = 2\%$ (*Left*) or $q = 20\%$ (*Right*) of the images marked.

Figure 5 is an alternative visualization: it shows the histograms of cosine similarities between the classifiers and random directions, the carrier and the semantic direction. The classifiers are more aligned with the carrier when $q = 20\%$ of the data is marked, a bit less with $q = 2\%$.

5. Experiments

5.1. Image classification setup

We employ the widely-used benchmarks Imagenet (Deng et al., 2009), a dataset of natural images with 1.2M images belonging to 1,000 classes and Places205 (Zhou et al., 2014), a dataset of 2.4M images from 205 scene categories. We first consider the Resnet-18 and Resnet-50 models (He et al., 2016). We train with SGD with a momentum of 0.9 and a weight decay of 10^{-4} for 90 epochs, using a batch size of 2048 across 8 GPUs.

We use Pytorch (Paszke et al., 2017) and adopt its standard data augmentation settings (random crop resized to 224×224). We use the waterfall schedule for the learning rate: it starts at 0.8 and is divided by 10 every 30 epochs (Goyal et al., 2017). On a vanilla Imagenet, we obtain a top 1 accuracy of 69.6% and a top-5 accuracy of 89.1% with our Resnet-18. Varying the random initialization and the order of elements seen during SGD, the top-1 accuracy varies by 0.1% from one experiment to the other.

5.2. Experimental setup and metrics

We modify Imagenet images by inserting our radioactive mark, and retrain models on this radioactive data. We then analyze these "contaminated" models and report several measures of performance. On the images, we report the PSNR, *i.e.*, the pertubation magnitude necessary to add the

	% radioactive	1	2	5	10
Center Crop	$\log_{10}(p)$	<-150	<-150	<-150	<-150
	Δ_{acc}	-0.48	-0.86	-1.07	-1.33
Random Crop	$\log_{10}(p)$	-38.0	-138.2	<-150	<-150
	Δ_{acc}	-0.24	-0.31	-0.55	-0.99

Table 1. p-value (statistical significance) of radioactivity detection. We train a logistic regression classifier on Imagenet with Resnet-18 features, and only a percentage of the training set is radioactive. Our method can identify with a very high confidence ($\log_{10}(p) < -38$) that the classifier is trained on radioactive data, even when only 1% of the training data is radioactive.

	% radioactive	1	2	5	10
Center Crop	$\log_{10}(p)$	-1.07	-2.09	-8.79	-26.18
	Δ_{acc}	-0.17	-0.18	-0.37	-0.3
Random Crop	passive baseline	-1.5	-6.9	-24.6	-65.4
	$\log_{10}(p)$	-4.85	-12.63	-48.8	<-150
	Δ_{acc}	-0.09	-0.65	-0.24	-0.43

Table 2. p-value (statistical significance) of radioactivity detection. Results for a Resnet-18 trained from scratch on Imagenet, with only a percentage of the data bearing the radioactive mark (Threat A1). We are able to identify models trained from scratch on only $q = 1\%$ of radioactive data. The presence of radioactive data has a low impact on the accuracy of a learned model as long as the fraction of radioactive data is under 10%. Radioactive data outperforms the passive baseline by a significant margin.

radioactive mark. On the model, we report the p-value that measures how confident we are that radioactive data was used to train the model, as well as the accuracy of this model on vanilla (held-out) data. We only mark a fraction q of the data, with $q \in \{0.01, 0.02, 0.05, 0.1, 0.2\}$. Radioactive data are generated by running SGD by optimizing Equation (5) with $R = 10$, $\lambda_1 = 0.0005$ and $\lambda_2 = 0.01$. These parameters where chosen such that the average PSNR of radioactive images is 42dB.

Sanity check. We ran our radioactive detector on pre-trained models of the Pytorch zoo and found p-values of 15% for Resnet-18 and 51% for Resnet-50, which is reasonable: in the absence of radioactive data, these values should be uniformly distributed between 0 and 1. We also ran experiments using the backdoor setup of Chen et al. (2017); these experiments show the limitation of the backdoor technique in two aspects: (1) it requires a much higher perturbation of the images to show an effect (PSNR < 30dB) and (2) it is impossible to provide a level of confidence that a particular marked dataset was used (cf. Appendix C for details).

Baselines. We compare our method to two baselines. The first baseline is individual watermarking of all images of the dataset. When all the training set is marked, we are able to detect this watermark in the trained classifier with a p-value of 10^{-3}: while this is a statistically significant result, it requires marking all elements of the dataset. We also compare to the state-of-the-art membership method of Sablayrolles et al. (2019): we perform membership inference using a threshold attack on a Resnet-18. The attack predicts (correctly) "1" for training elements in about 70% of the cases, and predicts "1" for held-out samples in about 38% of the cases. Note that this baseline assumes that we know the $50,000$ elements of the validation set that follow the same distribution as that of the training set, and are known to not have been used for training. Even in this favorable case, our radioactive technique outperforms the baseline, as can be seen in Table 2.

5.3. Results

Same architecture. Table 1 reports the results of a Resnet-18 model with fixed features trained on Imagenet. We are able to detect that the model was trained on radioactive data with a very high confidence for both center crop and random crop. The model overfits more on the center crop, hence it learns more the radioactive mark, which is why the p-value is lower on center crop images. Conversely on random crops, marking data has less impact on the model accuracy (-0.24 as opposed to -0.48 for $q = 1\%$ marked data).

Training from scratch. Table 2 shows the results of retraining a Resnet-18 from scratch on radioactive data. This setup is more challenging (Section 3.4). However, our watermark is detected reliably when only $q = 1\%$ of the training data is marked. Table 2 shows two interesting results: first, the gap in accuracy is less important than when retraining only the logistic regression layer, in particular using 1% of radioactive data does not impact accuracy (-0.1%); second, data augmentation is helping the radioactive process.

Black-box results. We measure the difference between the loss on vanilla samples and the loss on radioactive samples: when this gap is positive, it means that the model fares better on radioactive images, and thus that it has been trained on the radioactive data. We can detect the use of radioactive data when a fraction of $q = 20\%$ or more of the training set is radioactive. When a smaller portion of the data is radioactive, the model performs better on vanilla data than on radioactive data and thus it is difficult to precisely assess the use of radioactive data. Also there is no rigorous hypothesis testing for this use case. Detailed results are provided in Appendix B.

Distillation. Given only black-box access to a model (assuming access to the full softmax), we can perform distillation (Hinton et al., 2015), and test the distilled model for

% radioactive	1	2	5	10	20
Resnet-50	-6.9	-12.3	-50.22	-131.09	<-150
Densenet-121	-5.39	-11.63	-41.24	-138.36	<-150
VGG-16	-2.14	-4.49	-13.01	-33.28	-106.56

Table 3. p-value (statistical significance) for radioactivity detection. Results for different architectures trained from scratch on Imagenet (Threat A1). Even though radioactive data was crafted using a Resnet-18, models of other architectures also become radioactive when trained on this data.

radioactivity. In this setup, it is possible to detect the use of radioactive data on the distilled model, with a slightly lower performance compared to white-box access to the model (this corresponds to Threat A2). We give detailed results in Appendix A.

5.4. Ablation analysis

Architecture transfer. We ran experiments on different architectures with the same training procedure: Resnet-50, VGG-16 and Densenet121. The results are shown in Table 3: the values and trend are similar to what we obtain with Resnet-18 (Table 2). Although there is no reason that the feature space of a VGG-16 would behave in the same way as that of a Resnet-18, after alignment, we detect the presence of our radioactive mark with high statistical significance.

Transfer to other datasets. We conducted experiments on a slightly different setup: we mark images from the dataset Places205, but use a network pretrained on Imagenet for the marking phase. The purpose of these experiments is to show that even if the marking network is fit for a different distribution, the marking still works and we are able to detect it (it corresponds to Threat B1). Results are shown in Table 4. We can see that when a fraction q higher than 10% of the training data is marked, we can detect radioactivity with a strong statistical significance ($p < 10^{-3}$).

Correlation with class difficulty. The radioactive carriers added to features are correlated with the class label, so we expect the learning to rely more on this mark when the class accuracy is low. To validate this hypothesis, we compute the Spearman correlation between the per-class accuracy and the cosine similarity between the classifier and the carrier: this correlation is negative, with a p-value of 4×10^{-5}. This confirms that the network relies more on the mark for difficult classes.

% radioactive	10	20	50	100
$\log_{10}(p)$	-3.30	-8.14	-11.57	<-150

Table 4. p-value of radioactivity detection. A Resnet-18 is trained on Places205 from scratch, and a percentage of the dataset is radioactive (Threat B1). When 10% of the data or more is radioactive, we are able to detect radioactivity with a strong confidence ($p < 10^{-3}$).

5.5. Discussion

We observe that our technique, designed for the feature space, applies in practice with progressively relaxed hypotheses: (1) with pixel-space marks that reproduce the mark imperfectly, (2) transferring to another feature extractor and even between different architectures, (3) model distillation and (4) when transferring to a different dataset. In each of these steps, there is some loss of detection accuracy (p-value) but the detection is still remarkably reliable. These observations suggest that our radioactive method is appropriate for real use cases.

Limitation in an adversarial scenario. We assume that at training time, radioactive and vanilla data are ingested in the same way. However, a subspace analysis could reveal the marking direction. This adversarial scenario is akin to that considered in the watermarking literature, where strategies have been developed to reduce the detectability of the carrier. Our current proposal is therefore restricted to the proof of concept that we can mark a model through training that is only resilient to blind attacks such as architectural or training changes. We hope that follow-up works will address a more challenging scenario under Kerckhoffs assumptions (Kerckhoffs, 1883).

6. Conclusion

The method proposed in this paper, radioactive data, is a way to verify if some data was used to train a model, with statistical guarantees. Our paper shows that such radioactive contamination is effective on large-scale computer vision tasks such as classification on Imagenet with modern architecture (Resnet-18 and Resnet-50, DenseNet, VGG), even when only a very small fraction (1%) of the training data is radioactive. Although it is not the core topic of our paper, our method incidentally offers a way to watermark images in the classical sense (Cayre et al., 2005).

Acknowledgements. The authors would like to thank Laurens van der Maaten, Tom Goldstein and Laurent Meunier for their feedback on earlier versions of this paper. This work was supported by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference

ANR-19-P3IA-0001 (PRAIRIE 3IA Institute). d

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *SIGSAC*. ACM, 2016.

Adi, Y., Baum, C., Cissé, M., Pinkas, B., and Keshet, J. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *USENIX Security Symposium*, 2018.

Biggio, B., Nelson, B., and Laskov, P. Poisoning attacks against support vector machines. In *ICML*, 2012.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *IEEE Symp. Security and Privacy*, 2017.

Carlini, N., Liu, C., Kos, J., Erlingsson, Ú., and Song, D. The secret sharer: Measuring unintended neural network memorization & extracting secrets. *arXiv preprint arXiv:1802.08232*, 2018.

Cayre, F., Fontaine, C., and Furon, T. Watermarking security: theory and practice. IEEE *Transactions on Signal Processing*, 2005.

Chen, X., Liu, C., Li, B., Lu, K., and Song, D. Targeted backdoor attacks on deep learning systems using data poisoning. *CoRR*, abs/1712.05526, 2017.

Cox, I. J., Miller, M. L., Bloom, J. A., and Honsinger, C. *Digital watermarking*, volume 53. Springer, 2002.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *TCC*, 2006.

Fisher, R. *Statistical methods for research workers*. 1925.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Gu, T., Liu, K., Dolan-Gavitt, B., and Garg, S. Badnets: Evaluating backdooring attacks on deep neural networks. In *Machine Learning and Computer Security Workshop*, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *ICCV*, 2017.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Iscen, A., Furon, T., Gripon, V., Rabbat, M., and Jégou, H. Memory vectors for similarity search in high-dimensional spaces. *IEEE Transactions on Big Data*, 2017.

Jégou, H., Douze, M., and Schmid, C. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, 2008.

Kerckhoffs, A. La cryptographie militaire [military cryptography]. *Journal des sciences militaires [Military Science Journal]*, 1883.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pp. 1097–1105, 2012.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *ECCV*, 2014.

Papernot, N., Song, S., Mironov, I., Raghunathan, A., Talwar, K., and Erlingsson, Ú. Scalable private learning with pate. In *ICLR*, 2018.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. 2017.

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. *IJCV*, 2015.

Sablayrolles, A., Douze, M., Ollivier, Y., Schmid, C., and Jégou, H. White-box vs black-box: Bayes optimal strategies for membership inference. In *ICML*, 2019.

Shafahi, A., Huang, W. R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., and Goldstein, T. Poison frogs! targeted clean-label poisoning attacks on neural networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *NeurIPS*, 2018.

Shokri, R., Stronati, M., and Shmatikov, V. Membership inference attacks against machine learning models. *IEEE Symp. Security and Privacy*, 2017.

Steinhardt, J., Koh, P. W. W., and Liang, P. S. Certified defenses for data poisoning attacks. In *NeurIPS*. 2017.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. Intriguing properties of neural networks. In *ICLR*, 2014.

Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Torralba, A., Efros, A. A., et al. Unbiased look at dataset bias. In *CVPR*, volume 1, pp. 7, 2011.

Tran, B., Li, J., and Madry, A. Spectral signatures in backdoor attacks. In *NeurIPS*. 2018.

Vukotić, V., Chappelier, V., and Furon, T. Are deep neural networks good for blind image watermarking? In *Workshop on Information Forensics and Security (WIFS)*. IEEE, 2018.

Yeom, S., Giacomelli, I., Fredrikson, M., and Jha, S. Privacy risk in machine learning: Analyzing the connection to overfitting. In *CSF*, 2018.

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A. Learning deep features for scene recognition using places database. In *NeurIPS*, 2014.

Zhu, J., Kaplan, R., Johnson, J., and Fei-Fei, L. Hidden: Hiding data with deep networks. In *ECCV*, 2018.

A. Distillation

% radioactive	1	2	5	10	20
$\log_{10}(p)$	-1.58	-3.07	-13.60	-34.22	-137.42

Table 5. p-value for the detection of radioactive data usage. A Resnet-18 is trained on Imagenet from scratch, and a percentage of the training data is radioactive. This marked network is distilled into another network, on which we test radioactivity. When 2% of the data or more is radioactive, we are able to detect the use of this data with a strong confidence ($p < 10^{-3}$).

Given a marked Resnet-18 on which we only have black-box access, we use distillation (Hinton et al., 2015) to train a second network. On this distilled network, we perform the radioactivity test. We show in Table 5 the results of this radioactivity test on distilled networks. We can see that when 2% or more of the original training data is radioactive, the radioactivity propagates through distillation with statistical significance ($p < 10^{-3}$).

B. Black-box results.

We report in Figure 6 the results of our black-box detection test. We measure the difference between the loss on vanilla samples and the loss on radioactive samples: when this gap is positive, it means that the model fares better on radioactive images, and thus that it has been trained on the radioactive data. We can see that the use of radioactive data can be detected when a fraction of $q = 20\%$ or more of the training set is radioactive. When a smaller portion of the data is radioactive, the model fares better on vanilla data than on radioactive data and thus it is difficult to tell.

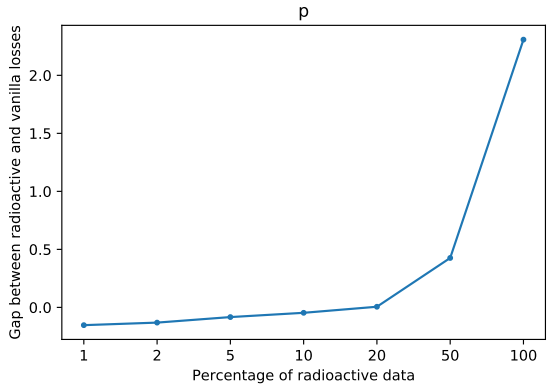

Figure 6. Black-box detection of the usage of radioactive data. The gap between the loss on radioactive and vanilla samples is around 0 when $q = 20\%$ of the data are contaminated.

C. Experiments with backdoors

We experimented with the backdoor technique of Chen et al. (2017) in the context of our marking problem. In general, the backdoor technique adds unrelated images to a class, plus a "trigger" that is consistent across these added images. In their work, Chen et al. (2017) need to poison approximately 10% of the data in a class to activate their trigger. We adapted their technique to the "clean-label" setup on Imagenet: we blend a trigger (a Gaussian pattern) to images of a class. We observed that it is possible to detect this trigger at train time, albeit with a low image quality (PSNR $<$ 30dB) that is visually perceptible. In this case, the model is more confident on images that have the trigger than on vanilla images in about 90% of the cases. However, we also observed that any Gaussian noise activates the trigger: hence we have no guarantee that images with our particular mark were used.