Radioactive data: tracing through training

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

Abstract

We want to detect whether a particular image dataset has been used to train a model. We propose a new technique, *radioactive data*, that makes imperceptible changes to this dataset such that any model trained on it will bear an identifiable mark. The mark is robust to strong variations such as different architectures or optimization methods. Given a trained model, our technique detects the use of radioactive data and provides a level of confidence (p-value).

Our experiments on large-scale benchmarks (Imagenet), using standard architectures (Resnet-18, VGG-16, Densenet-121) and training procedures, show that we can detect usage of radioactive data with high confidence ($p < 10^{-4}$) even when only 1% of the data used to trained our model is radioactive. Our method is robust to data augmentation and the stochasticity of deep network optimization. As a result, it offers a much higher signal-to-noise ratio than data poisoning and backdoor methods.

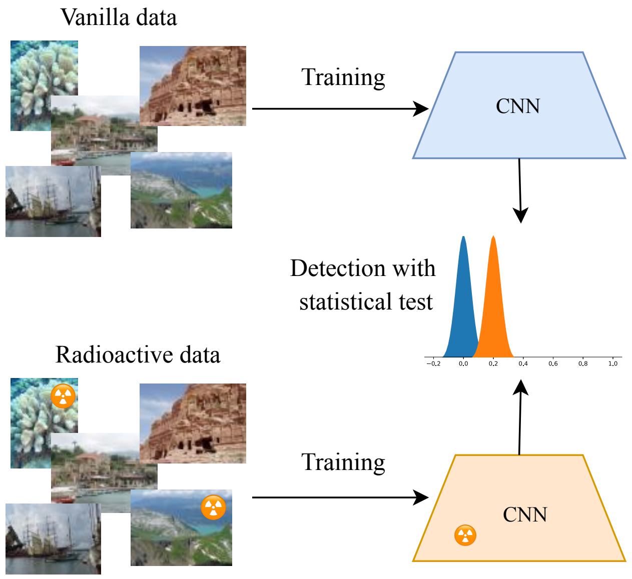

Figure 1. Illustration of our approach: we want to determine through a statistical test (p-value) whether a network has seen a marked dataset or not. The distribution (shown on the histograms) of a statistic on the network weights is clearly separated between the vanilla and radioactive CNNs. Our method 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 accelerated the development of machine learning. The Imagenet collection (Deng et al., 2009) and challenge (Russakovsky et al., 2015) contributed to the success of the deep learning architectures (Krizhevsky et al., 2012). The annotation of precise instance segmentation on the large-scale COCO dataset (Lin et al., 2014) enabled large improvements of object detectors and instance segmentation models (He et al., 2017). Even in weakly-supervised (Joulin et al., 2016; Mahajan et al., 2018) and unsupervised learning (Caron et al., 2019) where annotations are scarcer, state-of-the-art results are obtained on large-scale datasets collected from the Web (Thomee et al., 2015).

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 the 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 for datasets. By introducing a specific mark in a dataset, we want to provide a strong signal that a dataset has been used to train a model. We thus slightly change the dataset, effectively substituting the data for similar-looking marked data (*isotopes*).

Let us assume that this data, as well as other collected data, is used to train a convolutional neural network (convnet). After training, the model is inspected to assess the use of radioactive data. 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 answer the question of whether any radioactive data has been used to train the model, or if only vanilla data was used. We want to provide a statistical guarantee with the answer, in the form of a p-value.

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

Passive techniques such as those employed to measure dataset bias (Torralba et al., 2011) or to do membership inference (Sablayrolles et al., 2019; Shokri et al., 2017) cannot provide sufficient empirical or statistical guarantees. More importantly, their measurement is relatively weak and therefore cannot be considered as an evidence: they are likely to confuse datasets having the same underlying statistics. In contrast, we target a p-value much below 0.1%, meaning there is a very low probability that the results we observe are obtained by chance.

Therefore, 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 reasonably neutral with respect to the end-task, i.e., the accuracy of the model trained with the marked dataset should not be significantly modified; (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 that employ incorrect labels as a mark, which are easy to detect 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), whose goal is to imprint a mark into an image such that it can be reidentified with high probability. We point out that traditional image-based watermarking is ineffective in our context: the learning procedure ignores the watermarks if they are not useful to guide the classification decision (Tishby et al., 2000). Therefore regular watermarking leaves no exploitable trace after training. 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 (data *isotopes*) 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 behaviour is confirmed by an analysis of the latent space before classification. It shows that the network devotes a small part of its capacity to keep track of our "radioactive tracers".

Our experiments on Imagenet confirm that our radioactive marking technique is effective: with almost 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 strong confidence. Note that our radioactive marks, while visually imperceptible, might be detected by a statistical analysis of the latent space of the network. Our aim in this paper is to provide a proof of concept that marking data is possible with statistical guarantees, and the analysis of defense mechanisms lies outside the scope of this paper. 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.

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 random data augmentations. As an example, for a ResNet-18 trained from scratch, we achieve a p-value of 10^{-4} when only 1% of the training data is radioactive. The accuracy of the network is not noticeably changed ($\pm 0.1\%$).

The paper is organized as follows. Section 2 reviews the related literature. We discuss related works in watermarking, and explain how the problem that we tackle is related to and differs from 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 the cone generated by the carrier. Zero-bit watermarking detects whether an image is marked or not, but 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 so-called adversarial examples (Carlini & Wagner, 2017; Goodfellow et al., 2015; Szegedy et al., 2014): 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 and the privacy budget ϵ trades off between learning statistical facts and hiding the presence of individual records in the training set. 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. These classification approaches do not provide any guarantee: if a membership inference model predicts that an image belongs to the training set, it does not give a level of statistical significance. Furthermore, 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 consists of three stages: the *marking stage* where the radioactive mark is added to the vanilla training images, without changing their labels. The *training stage* 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.

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

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

3.1. Statistical preliminaries

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 $a = \frac{1}{2}$ and $b = \frac{d-1}{2}$:

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

$$= \frac{B_{\tau^2}\left(\frac{1}{2}, \frac{d-1}{2}\right)}{B\left(\frac{1}{2}, \frac{d-1}{2}\right)} \tag{3}$$

$$= \frac{1}{B\left(\frac{1}{2}, \frac{d-1}{2}\right)} \int_0^{\tau^2} \frac{\left(\sqrt{1-t}\right)^{d-3}}{\sqrt{t}} dt \tag{4}$$

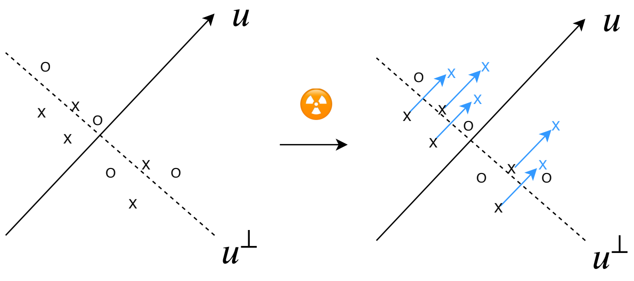

Figure 2. Illustration of our method in high dimension. In high dimension, the linear classifier that separates the class is almost orthogonal to u with high probability. Our method shifts points belonging to a class in the direction u, therefore aligning the linear classifier with the direction u.

with

$$B_x\left(\frac{1}{2}, \frac{d-1}{2}\right) = \int_0^x \frac{\left(\sqrt{1-t}\right)^{d-3}}{\sqrt{t}} dt \tag{5}$$

and

$$B\left(\frac{1}{2}, \frac{d-1}{2}\right) = B_1\left(\frac{1}{2}, \frac{d-1}{2}\right). \tag{6}$$

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 $T_1, \dots, T_k$, independent under the null hypothesis $\mathcal{H}_0$. Under $\mathcal{H}_0$, the corresponding p-values $p_1, \dots, p_k$ 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 $T_1, \dots, T_k$ 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 tracing problem. In the marking stage, we add a random isotropic unit vector $\alpha u \in \mathbb{R}^d$ with $\|u\|_2 = 1$ to the features of all training images of one class. This direction u is our carrier.

If radioactive data is used at training time, the linear classifier of the corresponding class w 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 w 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 $a = \frac{1}{2}$ and $b = \frac{d-1}{2}$. 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 (the probability of it happening under the null hypothesis $\mathcal{H}_0$) 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 them 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 backpropagating gradients in 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:

$$\min_{\tilde{x},\ \|\tilde{x}-x\|_\infty \leq R} \quad \mathcal{L}(\tilde{x}) \tag{7}$$

where the radius R is a hard upper bound on the change of color levels of the image that we can accept. The loss is a combination of three terms:

$$\mathcal{L}(\tilde{x}) = -\left(\phi(\tilde{x}) - \phi(x)\right)^\top u + \lambda_1\|\tilde{x} - x\|_2 + \lambda_2\|\phi(\tilde{x}) - \phi(x)\|_2. \tag{8}$$

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.

Data augmentation. The training stage most likely involves data augmentation, so we take it into account at marking time. Given an augmentation parameter θ, the input to the neural network is not the image $\tilde{x}$ but its trans-

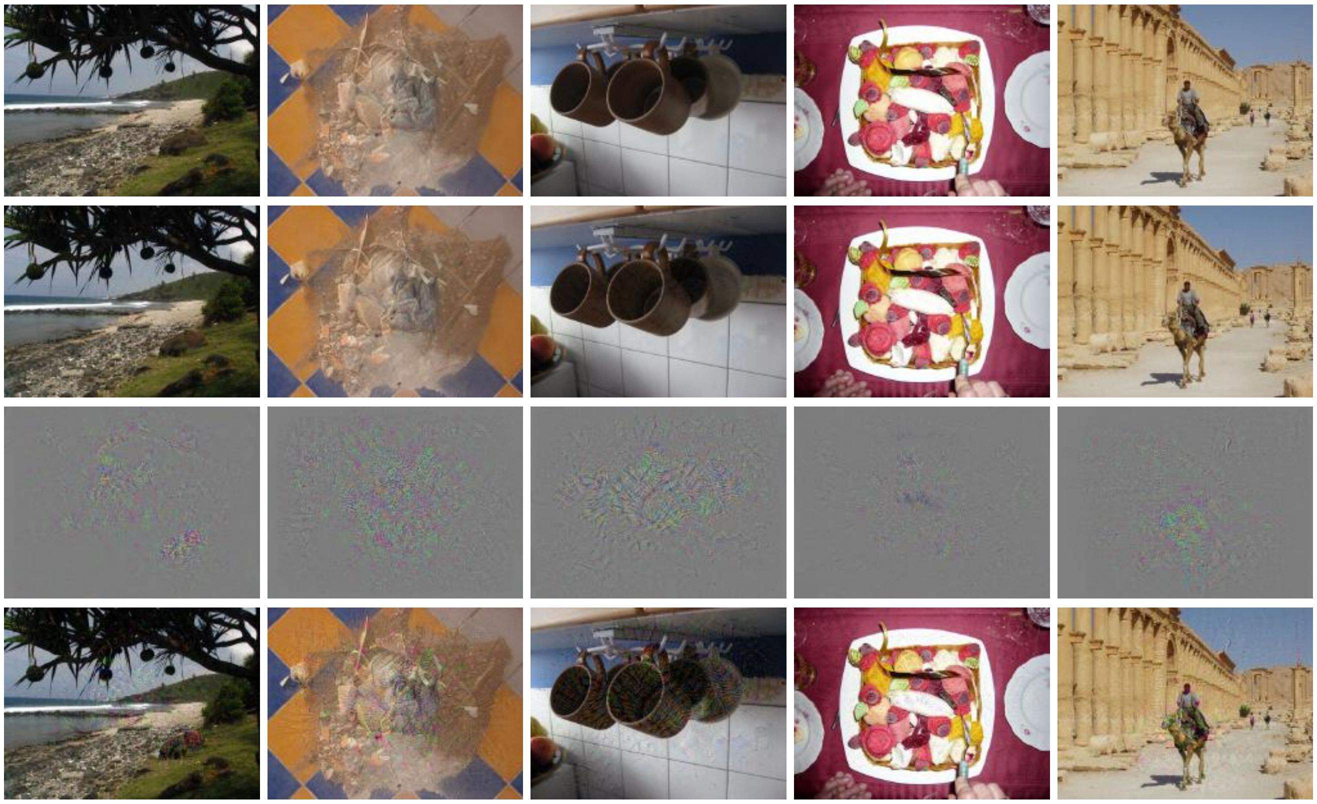

Figure 3. Radioactive images from Holidays (Jégou et al., 2008) with random crop and PSNR= 42dB. First row: original image. Second row: image with a radioactive mark. Third row: visualisation of the mark amplified with a $\times 5$ factor. Fourth row: We exaggerate the mark by a factor $\times 5$, which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

formed version $F(\theta, \tilde{x})$. In practice, the data augmentations used are crop and/or resize transformations, so θ are the coordinates of the center and/or size of the cropped images. The augmentations are differentiable with respect to the pixel space, so we can backpropagate through them. Thus, we emulate augmentations by minimizing:

$$\min_{\tilde{x},\ \|\tilde{x}-x\|_\infty \leq R} \quad \mathbb{E}_\theta \left[\mathcal{L}(F(\tilde{x}, \theta))\right]. \tag{9}$$

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 column).

3.4. White-box test with subspace alignment

We now tackle the more difficult case where the training stage includes the training of 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 permutation 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 $\phi_0(x) \approx M\phi_t(x)$. The linear mapping is estimated by L_2 regression:

$$\min_M \mathbb{E}_x[\|\phi_0(x) - M\phi_t(x)\|_2^2]. \tag{10}$$

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

The classifier we manipulate at detection time is thus $W\phi_t(x) \approx WM\phi_0(x)$. 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 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 contaminated 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

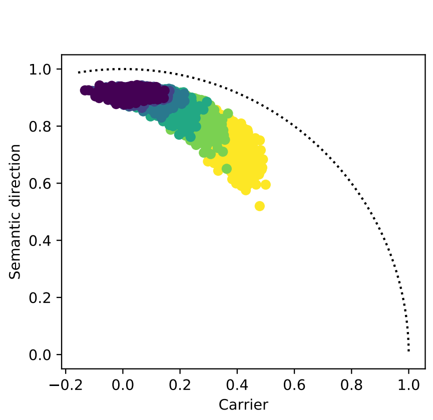

Figure 4. Decomposition of learned classifiers into three parts: the "semantic direction" (y-axis), the carrier direction (x-axis) and noise (represented by $1 - \|x\|^2 - \|y\|^2$, i.e. the distance between a point and the unit circle). The semantic and carrier direction are 1-D subspace, while the 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).

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.

4. Analysis of the latent feature space

In this section, we analyze how the classifier learned on a radioactive dataset is related to (1) a classifier learned on unmarked images ; and (2) the direction of the carrier. For the sake of analysis, we take the simplest case where the mark is added in the latent 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 direction of the carrier, 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 noise space is due to the randomness of the initialization and the optimization procedure (SGD and random data augmentations).

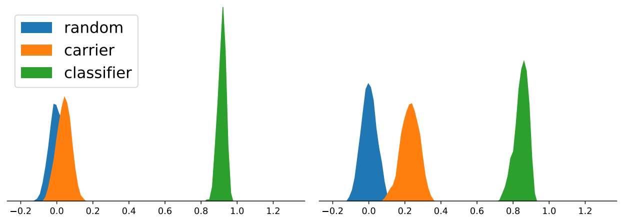

Figure 5. Analysis of how classification directions re-learned with a logistic regression on marked images can be decomposed between (1) the original subspace; (2) the mark subspace; (3) the noise space. Logistic regression with: $q = 2\%$ (*Left*) or $q = 20\%$ (*Right*) of the images marked.

The rationale of performing this decomposition is to quantify, with respect to 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 projection of the new vector, which can be seen by the fact that the norm of the vector projected onto that subspace is close to 1 (which translates visually as to be close to the unit circle). Second and unsurprisingly, 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 only a little impact on the accuracy.

Figure 5 shows the histograms of cosine similarities between the classifiers and random directions, the mark direction and the semantic direction. We can see that the classifiers are well aligned with the mark when $q = 20\%$ or 2% of the data is marked.

5. Experiments

5.1. Image classification setup

In order to provide a comparison on the widely-used vision benchmarks, we use Imagenet (Deng et al., 2009), a dataset of natural images with 1,281,167 images belonging to $1,000$ classes. We first consider the Resnet-18 and Resnet-50 models (He et al., 2016). We perform training using the standard set of data augmentations from Pytorch (Paszke et al., 2017). 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 learning rate schedule: the learning starts at 0.8, (as recommended in (Goyal et al., 2017)) and is divided by 10 every 30 epochs. On a vanilla Imagenet, we obtain a top 1 accuracy of 69.6% and a top-5 accuracy of

89.1% with our Resnet18. We ran experiments by varying the random initialization and the order of elements seen during SGD, and found that 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 using the learning algorithm described above. We then analyze these "contaminated" models for the presence of our mark. We report several measures of performance. On the images, we report the PSNR, i.e. the magnitude of the perturbation necessary to add the 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 conduct experiments where we only mark a fraction q of the data, with $q \in \{0.01, 0.02, 0.05, 0.1, 0.2\}$.

As a sanity check, we ran our radioactive detector on pretrained 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.

5.3. Preliminary experiment: comparison to the backdoor technique

We experimented with the backdoor technique of Chen et al. (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. 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.

5.4. Results

Same architecture. We first analyze the results in Table 1 of a ResNet-18 model with fixed features trained on Imagenet. We can see that we are able to detect that our 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 accuracy of the model (-0.24 as opposed to -0.48 for $q = 1\%$ marked data).

	% 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) for the detection of radioactive data usage when only a fraction of the training data is radioactive. Results for a logistic regression classifier trained on Imagenet with Resnet-18 features , with only a percentage of the data bearing the radioactive mark. Our method can identify with a very high confidence ($\log_{10}(p) < -38$) that the classifier was trained on radioactive data, even when only 1% of the training data is radioactive. The radioactive data has an impact on the accuracy of the classifier: around -1% (top-1).

	% radioactive	1	2	5	10
Center Crop	$\log_{10}(p)$	-0.66	-1.64	-4.60	-11.37
Random Crop	$\log_{10}(p)$	-4.85	-12.63	-48.8	<-150
	Δ_{acc}	-0.1	-0.7	-0.3	-0.5

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

Table 2 shows the results of retraining a Resnet-18 from scratch on radioactive data. The results confirm that our watermark can be detected when only $q = 1\%$ of the data is used at train time. This setup is more complicated for our marks because since the network is retrained from scratch, the directions that will be learned in the new feature space have no *a priori* reason to be aligned with the directions of the network we used. 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; second, data augmentation is actually helping the radioactive process. We hypothesize that the multiple crops make the network believe it sees more variety, but in reality all the feature representations of these crops are aligned with our carrier which makes the network learn the carrier direction.

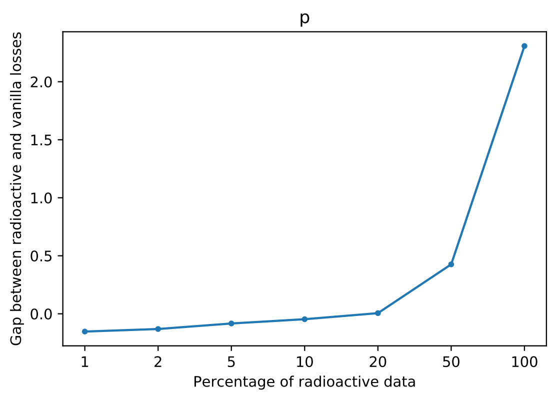

Figure 6. Black-box detection of the usage of radioactive data. The gap between the loss on radioactive and vanilla samples is around 0 when $q = 20\%$ of the data are contaminated.

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.

Distillation. Given only black-box access to a model (assuming access to the full softmax), we experiment distillation of this model, and test the distilled model for 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. We give detailed results in Appendix A.

5.5. 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). This is non-trivial, as there is no reason that the feature space of a VGG-16 would behave in the same way as that of a Resnet-18: yet, after alignment, we are able to detect the presence of our radioactive mark with high statistical significance. Specifically, when $q = 2\%$ of the data is radioactive, we are able to detect it with a p-value of 10^{-15}. This p-value is even stronger than the one we obtain when retraining the same architecture as our marking architecture (Resnet-18). We hypothesize that larger model overfit more in general, and thus in this case will learn the mark more acutely.

% 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. Even though radioactive data was crafted using a ResNet-18, models of other architectures also become radioactive when trained on this data.

% 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. When 10% of the data or more is radioactive, we are able to detect radioactivity with a strong confidence ($p < 10^{-3}$).

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. These experiments show that even if the marking network is fit for a different distribution, the marking still works and we are able to detect it. 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. Given that radioactive data adds a marker in the features that is correlated with the class label, we expect this mark to be learned by the network more when the class accuracy is low. To validate this hypothesis, we compute the Spearman correlation between the class accuracy for each class and the cosine 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 when learning with difficult classes.

5.6. Discussion

The experiments validate that our radioactive marks do indeed imprint on the trained models. We also observe two beneficial effects: data augmentation improves the strength of the mark, and transferring the mask to a larger and more realistic architectures makes its detection more reliable. These two observations suggest that our radioactive method is appropriate for real use cases.

Limitation in an adversarial scenario. We assume that at training time, there is no special procedure to take into account the radioactive data, but rather training is conducted as if it was vanilla data. In particular, a subspace analysis would likely reveal the marking direction. This adversarial scenario becomes 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.

We have shown in this paper 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), 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).

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.

Caron, M., Bojanowski, P., Mairal, J., and Joulin, A. Unsupervised pre-training of image features on non-curated data. In *ICCV*, 2019.

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.

Joulin, A., van der Maaten, L., Jabri, A., and Vasilache, N. Learning visual features from large weakly supervised data. In *ECCV*, 2016.

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.

Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., and van der Maaten, L. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018.

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.

Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., and Li, L.-J. Yfcc100m: The new data in multimedia research. *arXiv preprint arXiv:1503.01817*, 2015.

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.

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

% 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 ($p < 10^{-3}$).

A. Distillation

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 ($p < 10^{-3}$).