ABLATION PATH SALIENCY

Anonymous authors
Paper under double-blind review

ABSTRACT

We consider the saliency problem for black-box classification. In *image* classification, this means highlighting the part of the image that is most relevant for the current decision. We cast the saliency problem as finding an optimal ablation path between two images. An ablation path consists of a sequence of ever smaller masks, joining the current image to a reference image in another decision region. The optimal path will stay as long as possible in the current decision region. This approach extends the *ablation tests* in Sturmfels et al. (2020). The gradient of the corresponding objective function is closely related to the *integrated gradient method* Sundararajan et al. (2017). In the saturated case (when the classifier outputs a binary value) our method would reduce to the *meaningful perturbation approach* Fong & Vedaldi (2017), since crossing the decision boundary as late as possible would then be equivalent to finding the smallest possible mask lying on the decision boundary. Our interpretation provides geometric understanding of existing saliency methods, and suggests a novel approach based on ablation path optimisation.

1 INTRODUCTION

The basic idea of *saliency* or *attribution* is to provide something from which a human can judge how a classifier arrived at its decision of the prediction it gives for a certain input. It is difficult to give a more mathematical definition, but various properties that such a method should fulfill have been proposed.

Sundararajan et al. (2017) give axioms, of which *sensitivity* comes closest to the notion of saliency. Essentially, the features on which the output is most sensitive should be given a higher saliency value. The authors give further axioms to narrow it down – implementation invariance, completeness, linearity and symmetry-preservation – and obtain a corresponding method: the *integrated gradient* method. Note that we have another way to arrive at a similar method, see § 4.1

Fong & Vedaldi (2017) is closer to our work: the authors directly compute the saliency of a given pixel by deleting, or altering that pixel, to see how this affects the output of the classifier.

Our method is to define a proper maximisation problem as follows. First, we define *ablation paths* as time dependent smooth masks $\varphi\colon [0,1] \to \mathcal{C}(\Omega, \mathbb{R})$, going a full mask to an empty mask, such that at each pixel the mask value decreases over time (see Figure 1). We also impose constant area

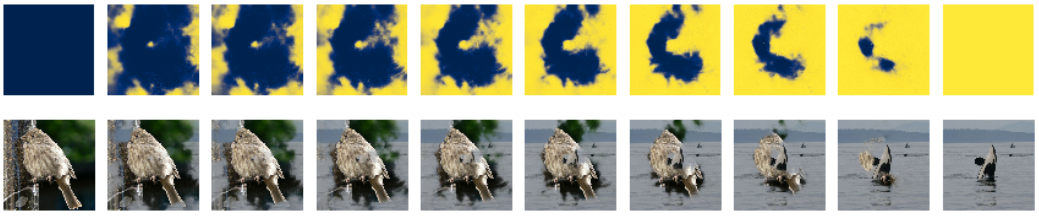


Figure 1: Example of how an ablation path (sequence of masks, top) gives rise to a transition between a current target (image of a house finch) and a baseline (orca whale).

speed: the area covered by the mask should increase linearly over time (see § 3). Let F be the

classifier, that outputs a probability between zero and one. We choose a current image of interest x_0 and a *baseline image* x_1. The objective function is then $P(\varphi) = \int_0^1 F(x_0 + \varphi(t)(x_1 - x_0))\,\mathrm{d}t$ (see §4). Assuming that $F(x_0) \simeq 1$ and $F(x_1) \simeq 0$, maximising the objective function means that we try to find an ablation path that stays as long as possible in the decision region of x_0. Intuitively, we try to replace as many pixels of x_0 by pixels of x_1 while staying in the same class as x_0.

The main contribution of this paper is to formulate the saliency problem as an optimisation problem on paths. Doing so connects previous notions of saliency, notably integrated gradients Sundararajan et al. (2017) (see §4.1), the ablation tests Sturmfels et al. (2020) (see §4.2), and meaningful perturbations Fong & Vedaldi (2017); Fong et al. (2019) (see §4.3). Our formulation is also resolution invariant (it does not assume that images are defined on pixels); this allows to make a clear difference between images and their duals, for instance, which gives guidance as to where regularisation is needed.

2 Related Work

Simonyan et al. (2013) defines a saliency map as the gradient of the network output at the given image. This would appear to be a sensible definition, but the resulting saliency is very noisy because the network output is roughly constant around any particular image. Selvaraju et al. (2016) improves the situation by computing the gradient after each layer instead. This is, however, not a black-box method such as the one we propose. Koh & Liang (2017) computes an *influence function*, that is, a function that measures how the parameters would be changed by a change in the training data. Although it is a black-box method, it is not a saliency method per se. They use the gradient of the network output to find the pixel most likely to have a high saliency. The pixel that have most effect are given a higher saliency. By contrast, Petsiuk et al. (2018) proposes to directly evaluate the saliency by finding out which pixels most affect the output, similarly to Fong & Vedaldi (2017), but without using any gradients.

There are a number of meta-studies of saliency methods. Adebayo et al. (2018) lists essential properties, for instance the requirement that the results should depend on the training data in a sense that perturbing *model parameters* should change the saliency. Kindermans et al. (2017) proposes a number of property that saliency methods should satisfy. Ancona et al. (2017) compares several saliency methods and proposes a method to evaluate them (the sensitiviy-n property).

3 Ablation Paths

3.1 Images and Masks

We consider a compact domain Ω. Note that Ω may be discrete or continuous: in fact, we assume that Ω is endowed with a measure which could be the discrete measure (if Ω is a set of pixels) or the Lebesgue measure (if Ω is a domain in $\mathbb{R}^2$, for instance). In the sequel, $\int_\Omega$ denotes integration with respect to that measure. Without loss of generality, we assume the mass of that measure to be one, i.e., $\int_\Omega 1 = 1$.

We consider a module M of functions on Ω with values in a vector space V (the dimensions of V represent the *channels*, and elements of M represent *images*). This module is equipped with a commutative ring R which represents *masks*. Concretely, in most of the paper we choose

$$M \coloneqq \mathcal{C}(\Omega, V) \qquad R \coloneqq \mathcal{C}(\Omega, \mathbb{R}).$$

The module structure simply means that masks can multiply images, i.e., that the operation θx gives a new image in M when $\theta \in R$ and $x \in M$, and that this operation is bilinear.

3.2 Ablation Paths

Definition 3.1. *We define the set $\mathcal{A}$ of* ablation paths *as the set of functions $\varphi\colon [0,1] \to R$ fulfilling the following properties:*

Boundary conditions $\varphi(0) = 0$ *and* $\varphi(1) = 1$

Monotonicity $t_1 \leq t_2 \implies \varphi(t_1) \leq \varphi(t_2) \qquad t_1, t_2 \in [0,1]$

Constant speed $\int_\Omega \varphi(t) = t \qquad t \in [0,1]$.

We will call monotone paths *the paths that verify the first two conditions but not the third.*

Note that the set $\mathcal{A}$ of ablation paths is a *convex subset* of $\mathcal{L}^\infty([0,1], R)$.

Some comments on each of those requirements are in order. (i) 0 and 1 denote here the constant functions zero and one (which corresponds to the zero and one of the algebra R) (ii) $\varphi(t_1) \leq \varphi(t_2)$ should be interpreted as usual as $\varphi(t_2) - \varphi(t_1)$ being in the cone of nonnegative elements[1]. (iii) If $t \mapsto \int_\Omega \varphi(t)$ is differentiable, this requirement can be rewritten as $\frac{\mathrm{d}}{\mathrm{d}t} \int_\Omega \varphi(t) = 1$, so it can be regarded as a *constant speed* requirement. This requirement is more a normalisation than a requirement, as is further detailed in Remark 3.3.

There is a canonical, simplest, ablation path between x_0 and x_1:

$$\ell(t) \coloneqq t. \tag{1}$$

The mask is thus constant in space at each time t. The reader should check that all the requirements for an ablation path are fulfilled.

Note that an ablation path without the constant-speed property can always be transformed into one that does fulfil it. This is clear if the function $t \mapsto \int_\Omega \varphi(t)$ is strictly increasing, as this is then just a time reparameterisation, but this is in fact always possible, in a canonical sense. The proof is in §A.1.

Lemma 3.2. *To any monotone path there corresponds a canonical ablation path.*

Since R is itself a function space, an ablation path φ is in fact a function of two arguments. In the sequel, we will abuse the notations and write φ as a function of one or two arguments depending on the context: $\varphi(t) \equiv \varphi(t, \cdot)$. For instance, in the definition Definition 3.1 above, $\int_\Omega \varphi(t) \equiv \int_\Omega \varphi(t, \cdot) \equiv \int_\Omega \varphi(t, \mathbf{r}) \, \mathrm{d}\mathbf{r}$.

Remark 3.3. *If the ablation path φ is differentiable in time, the requirements in Definition 3.1 admit a remarkable reformulation. Define $\psi(t) \coloneqq \frac{\mathrm{d}}{\mathrm{d}t}\varphi(t)$. All the requirements in Definition 3.1 are equivalent to the following requirements for a function $\psi \colon [0,1] \times \Omega \to \mathbb{R}$:*

$$\psi(t, \mathbf{r}) \geq 0, \quad \int_\Omega \psi(t, \mathbf{r}) \, \mathrm{d}\mathbf{r} = 1, \quad \int_{[0,1]} \psi(t, \mathbf{r}) \, \mathrm{d}t = 1 \qquad t \in [0,1], \mathbf{r} \in \Omega$$

The corresponding ablation path φ is then recovered by $\varphi(t) \coloneqq \int_0^t \psi(s) \, \mathrm{d}s$. What this means is that differentiable ablation paths can be parameterised as densities of doubly stochastic Markov transition kernels *on* $[0,1] \times \Omega$.

3.3 Regularity of Ablation Paths

Lemma 3.4. *If φ is an ablation path, then*

$$\|\varphi(t_1) - \varphi(t_0)\|_{\mathcal{L}^1} = |t_1 - t_0|.$$

In particular, $t \mapsto \varphi(t, \cdot)$ is continuous as a function $[0,1] \to \mathcal{L}^1(\Omega)$.

Proof. Choose t_0, t_1 in $[0,1]$. Without loss of generality, assume $t_1 \geq t_0$. Then, $\int_\Omega |\varphi(t_1) - \varphi(t_0)| = \int_\Omega (\varphi(t_1) - \varphi(t_0)) = t_1 - t_0$, from which we conclude that $\varphi(t_1) - \varphi(t_0)$ is in $\mathcal{L}^1$ and fulfils the equation above. □

4 Score of an Ablation Path

We now fix two points x_0 (the *current image*) and x_1 (the *baseline image*) in the space of images M. We propose the following measure of the *score* of an ablation path (see Definition 3.1) with respect

[1] Here we can define the cone of nonnegative functions by $\{ f \in \mathcal{C}(\Omega, \mathbb{R}) \mid f \geq 0 \}$. In a general star algebra, this cone would be defined as $\{ x \in R \mid \exists y \in R \quad x = y^* y \}$.

to these two images. Given a *mask* $\theta \in R$, we define the *interpolated image* $[x_0, x_1]_\theta \in M$ as

$$[x_0, x_1]_\theta \coloneqq (1-\theta)x_0 + \theta x_1.$$

We now define the *score function* $P\colon \mathcal{A} \to \mathbb{R}$ from ablation paths to $\mathbb{R}$ by the integral

$$P(\varphi) \coloneqq \int_0^1 F([x_0, x_1]_{\varphi(t)})\mathrm{d}t. \tag{2}$$

Note that, as F is bounded between zero and one, so is $P(\varphi)$ for any ablation path φ. The main idea is that if $F(x_0) \simeq 1$ and $F(x_1) \simeq 0$, the higher this value of P is, the better the path is to describe the salient parts of the image.

Note that the function P is defined regardless of the constraints placed on ablation paths, i.e., the score function P is defined on the vector space of functions $\varphi\colon [0,1] \to R$. It is straightforward to compute its differential $\mathrm{d}P$ on that space:

$$\langle \mathrm{d}P, \delta\varphi\rangle = \int_0^1 \langle \underbrace{\mathrm{d}F_{[x_0,x_1]_{\varphi(t)}}}_{\in M^*}, \underbrace{(x_1 - x_0)}_{\in M} \underbrace{\delta\varphi(t)}_{\in R}\rangle \,\mathrm{d}t.$$

So if we define the product of $D \in M^*$ and $x \in M$ producing an element in R^* by $\langle xD, \varphi\rangle \coloneqq \langle D, x\varphi\rangle$ as is customary, we can rewrite this differential as

$$\langle \mathrm{d}P, \delta\varphi\rangle = \int_0^1 \big\langle (x_1 - x_0)\mathrm{d}F_{[x_0,x_1]_{\varphi(t)}}, \delta\varphi(t)\big\rangle \,\mathrm{d}t.$$

Note that we know that any ablation path is bounded, so $\varphi \in \mathcal{L}^\infty([0,1], R)$, so the differential of P at φ can be identified with the function $\mathrm{d}P_\varphi = \Big[t \mapsto (x_1 - x_0)\mathrm{d}F_{[x_0,x_1]_{\varphi(t)}}\Big]$ in $\mathcal{L}^1([0,1], R^*)$.

4.1 Relation with the Integrated Gradient Method

When this differential is computed on the interpolation path ℓ (1) and then *averaged*, then this is exactly the integrated average gradient Sundararajan et al. (2017). More precisely, the integrated gradient is exactly $\int_0^1 \mathrm{d}P_{\ell(t)}\mathrm{d}t$. Note that this is in fact an integrated *differential*, since we obtain an element in the dual space M^*, and this differential should be appropriately smoothed along the lines of §5.1.

4.2 Relation to Pixel Ablation

Given a saliency function $\sigma \in R$ we can define a path by $\tilde{\varphi}(t) \coloneqq \mathbf{1}_{\sigma \le \log(t/(1-t))}$ when $t \in (0,1)$ and define $\tilde{\varphi}(0) \coloneqq 0$, $\tilde{\varphi}(1) \coloneqq 1$. This path is a monotone path, except in the module of images $M = \mathcal{L}^2(\Omega, V)$, equipped with the ring of masks $R = \mathcal{L}^\infty(\Omega)$. To be an ablation path, it still needs to be transformed into a constant speed path, which is always possible as explained in §A.1.

Note that this is a generalisation of the ablation method in Sturmfels et al. (2020). In that case, the set Ω would be a discrete set of pixels. Note that in the ranking, pixels with the same saliency would be ranked in an arbitrary way and added to the mask in that arbitrary order. In the method above, we add them all at once, but the time reparameterisation keeps that function constant longer for however many pixels were ranked the same. As long as the ranking is strict (no two pixels have the same saliency), the two methods are the same.

4.3 Relation to Meaningful Perturbations

In the saturated case, that is, if F only takes values zero and one (or in the limit where it does), our method basically reduces to finding the interpolation with the largest mask on the boundary, in essence the approach of Fong & Vedaldi (2017). Indeed, suppose that the ablation path φ crosses the boundary at time t^*. It means that $F\big([x_0, x_1]_{\varphi(t)}\big)$ has value one until t^* and zero afterwards, so the score P defined in (2) is $P(\varphi) = t^*$. By the constant speed property, $t^* = \int_\Omega \varphi(t^*)$, so we end up maximising the mask area on the boundary.

5 OPTIMISATION PROBLEM AND ALGORITHM

We proceed to define the optimisation problem at hand and how to solve it numerically.

Conceptually we try to find the ablation path (see Definition 3.1) that maximises the score $P(\varphi)$:

$$\max_{\varphi \in \mathcal{A}} P(\varphi).$$

Recall that the set $\mathcal{A}$ of ablation paths is convex; however, since the objective function P is not convex, this is not a convex optimisation problem.

The method we suggest is to follow a gradient direction. Such an approach is in general not guaranteed to approximate a global maximum, but empirically it does manage that quite well here.

5.1 GRADIENT AND METRIC

Note that the differential is an element of $\mathcal{L}^1([0,1], R^*)$, so we need a map from that space to $\mathcal{L}^\infty([0,1], R)$. For now we assume that $\varphi \in \mathcal{L}^2([0,1], R)$ and $\mathrm{d}P \in \mathcal{L}^2([0,1], R^*)$.

However, we still need a covariance operator $K\colon R^* \to R$. In practice, we use a covariance operator associated to a smoothing operator. For a measure $\mu \in R^*$, $\langle K\mu, \theta\rangle \coloneqq \langle \mu, \int_\Omega k(\cdot - \mathbf{r})\theta(\mathbf{r})\,\mathrm{d}\mathbf{r}\rangle$, where k is a suitable smoothing function, for instance $k(\mathbf{r}) = \exp(-\|\mathbf{r}\|^2/\sigma)$. This allows us to consider the *gradient* of P. Note that different choices of metric will influence on the algorithm.

Since the optimisation problem is *constrained* (since φ is constrained by the requirements in Definition 3.1), following the gradient direction will lead us to violate the constraints. Since the constraints are convex, it is straightforward enough to project each gradient-updated version back to something that does fulfill them, and indeed that is the idea behind our algorithm, however in practice it does by itself not yield convergence without impractically many iterations. See § A.3 for the details of how we actually proceed.

6 EXAMPLES

To test out our path-scoring approach and the saliency method based on its optimisation, we use a standard image classifier deep-CNN (Inception v4 Szegedy et al. (2016), pre-trained on ImageNet), with a selection of real-world images for both the *current target* and *baseline* inputs. For each example pair, we compare multiple saliency methods.

Figure 2 is a particularly clear-cut example, with the vibrantly coloured rubber eraser sufficient to retain the classification even in an almost completely ablated image. All of the compared methods manage to find a mask that focuses in on that, though with differences in the details. The unconstrained optimisation narrows it down to a few individual pixels, which gives an extremely good score (somewhat unsurprisingly, since that is what is being optimised), but the result is hardly useful for interpretability purposes: this should be considered an adversarial example. One interpretation of this is that identifying the gradient with the differential implies that the space of masks R is essentially bounded functions wihout further regularity (see § 5.1), similar to the mask space in § 4.2. The region boundary in that space seems to be extremely close to the baseline, and the optimisation method finds those adversarial examples.

The integrated gradient method in principle also has this issue, and indeed the corresponding mask (pixel ranking, cf. § 4.2) is quite grainy/noisy, without however behaving adversarially (the entire pencil is highlighted).

The authors in Fong & Vedaldi (2017) were confronted with similar problems. We use smoothing filters to compute the gradient, in order to both avoid adversarial examples and to have less noisy saliency masks. Using that, the path-optimisation still manages to achieve a high score, but now highlights the eraser as a single, clearly interpretable feature.

Filtering can also be applied to integrated gradient before pixel-ranking. That does avoid the noisyness, but it also leads to a *blobby* shape of the masks.

Comparison with a random-order ablation confirms that the good scores of the saliency methods really are a result of non-trivial feature highlighting, rather than something that comes out of the

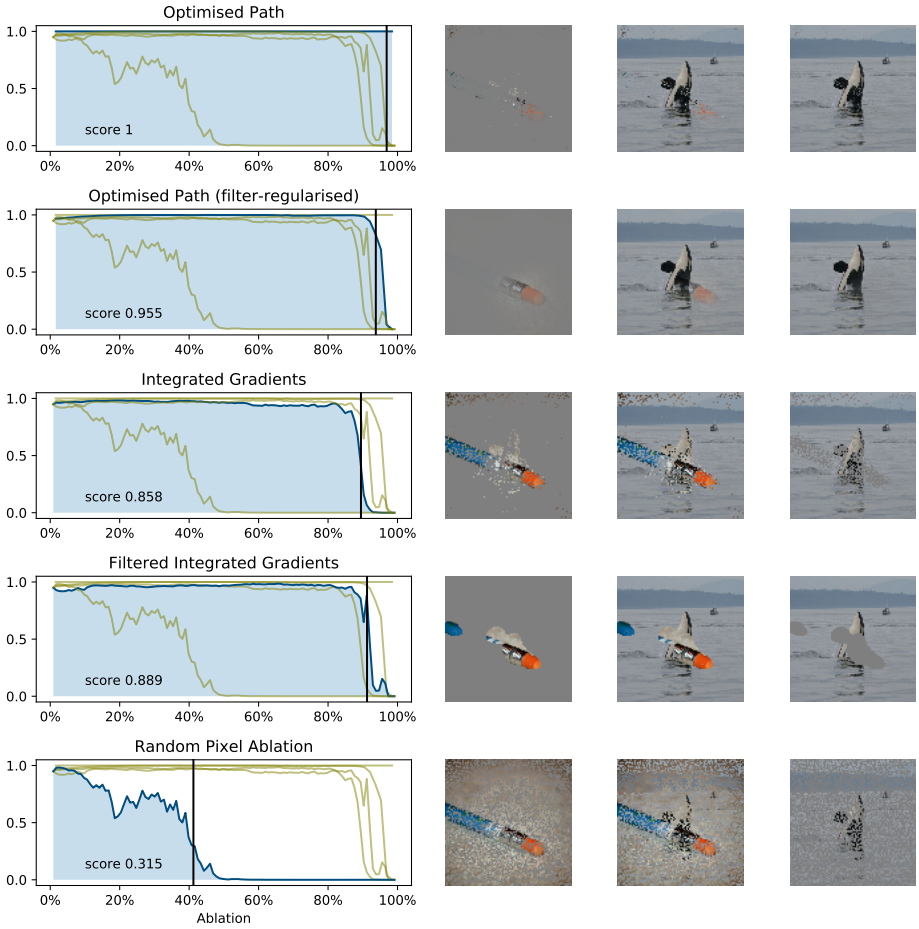


Figure 2: Comparison of saliency methods for an image of a pencil, against one of an orca whale as the baseline.

transition between the images automatically.[2] In Figure 3 it is evident that the saliency methods do *not* in general agree so clearly. Here, the filtered optimal path again highlights a small, single region

[2]We performed a small, non-rigorous statistical study suggesting that scores > 0.9 are $p < 0.01$ significant against a null hypothesis of smoothly random masks, and extremely unlikely with pixelwise-random masks.

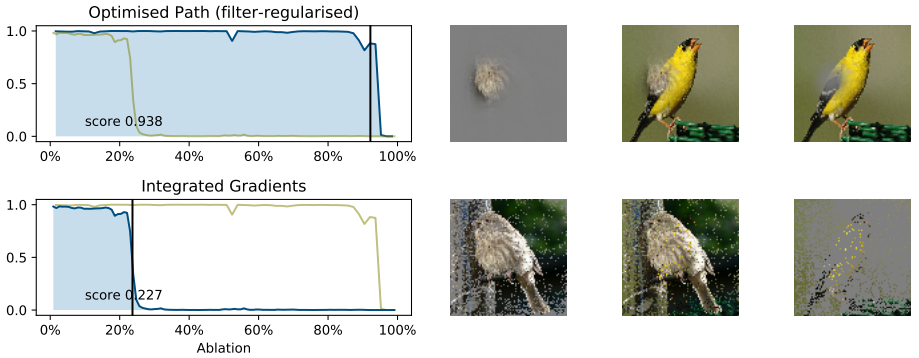


Figure 3: Method comparison for a house finch image against a gold finch baseline.

that is sufficient for a classification of the image as a house- rather than goldfinch. Arguably, this is again adversarial (a human would certainly label the composite image gold- rather than house finch). However it does give rise to a useful interpretation: note that the highlighted region includes the house finch's plumage, whilst covering specifically the wing of the gold finch (which features a distinctive pattern not seen on house finches). So in this case, the saliency tells more about the baseline than about the current target.

The integrated gradient meanwhile hardly manages to mask out anything at all, before the classification switches to gold finch.

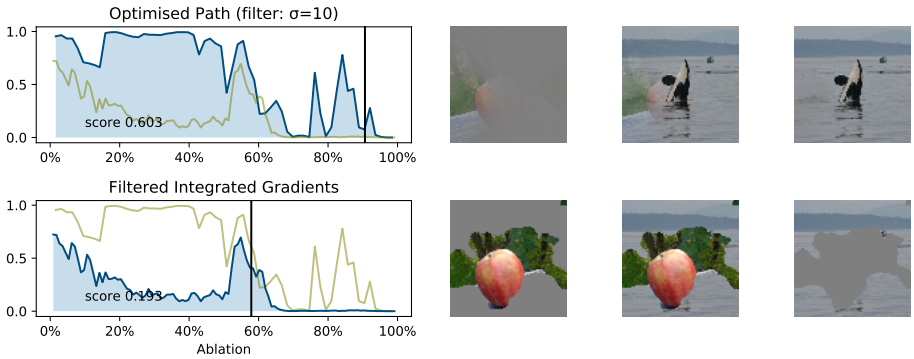


Figure 4: Method comparison for a misclassified image: Inception classifies the apple as a fig instead.

Practically speaking, saliency is particularly interesting in questions of whether a classification was actually correct. Figure 4 shows an example where the model misclassified an apple as a fig. The unstable scores in even the optimised ablation path are perhaps symptomatic of this (we are not sure how representative this is); nevertheless both our method and integrated gradients find a mask of less than half of the image that still retains the fig-classification. Whilst with integrated gradients, this mask includes the apple (which might in a real-world unclear situation lead one to think the classification payed attention to the intended target, increasing credibility), out method highlights mainly a patch of background at the border of the image.

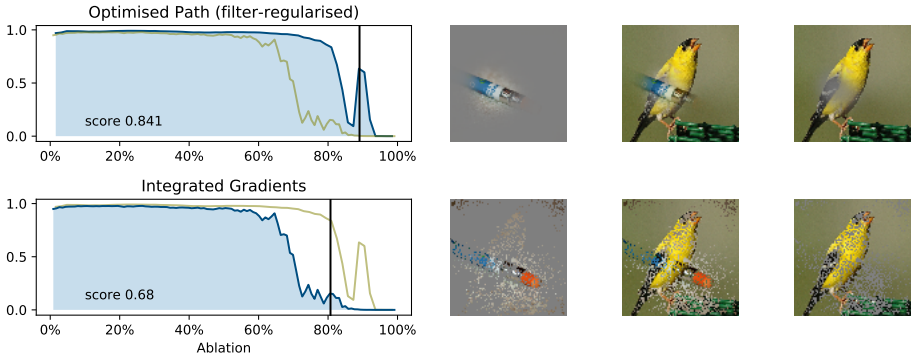


Figure 5: Comparison of saliency methods for an image of a pencil, against one of a gold finch as the baseline.

The optimised paths depend not only on the model and current target, but also on the baseline; notice for example that the pencil, of which in Figure 2 mostly the eraser was highlighted, is in Figure 5 mostly characterised by the eraser's metallic joint as well as part of the blue shaft, which happens to coincide with the gold finch's wings that were also masked out in Figure 3. In Figure 6 it is something in between. Still the results (with filtering) tend to be relatively coherent across baselines, more so than with the Integrated Gradients or the adversarial unfiltered ones.

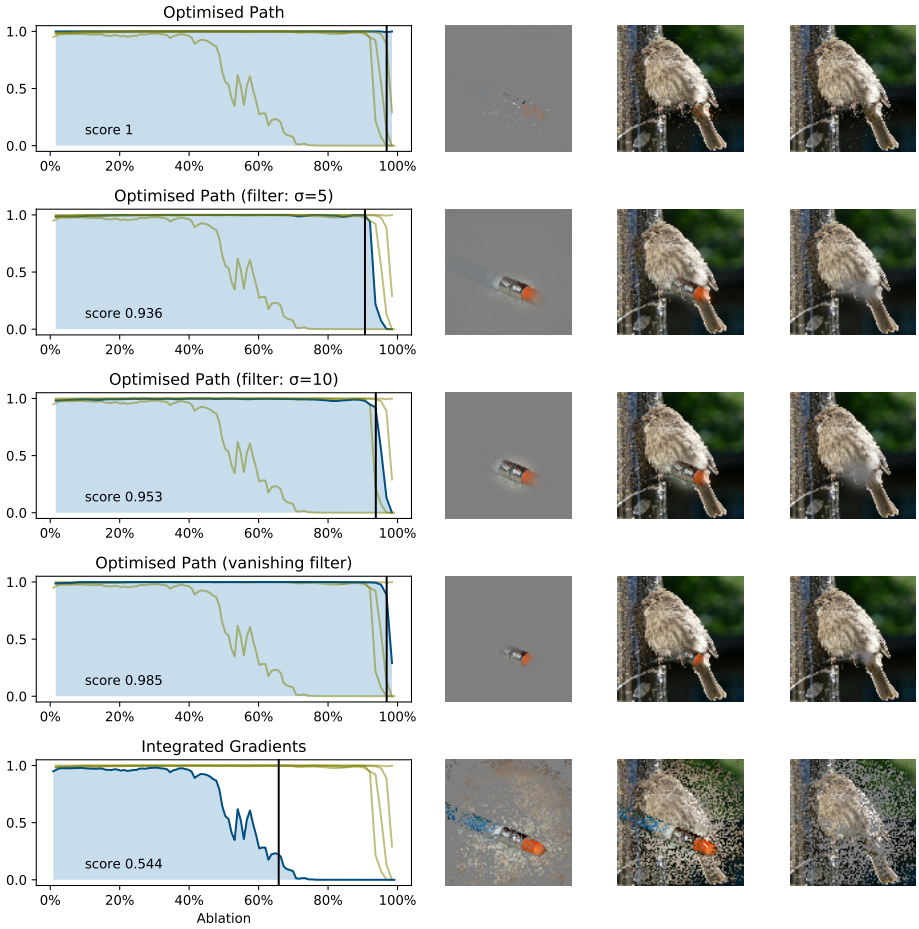


Figure 6: Comparison of saliency methods for an image of a pencil, against one of a house finch as the baseline.

One might ask why to use a true image as a baseline at all (as opposed to a blurred image, a constant colour image, as in Sturmfels et al. (2020) or Fong & Vedaldi (2017)). The problem with artificial baselines is that the network missclassifies them (a blurred goldfinch is classified as a power drill, for example), so the ablation path crosses region where the network is extrapolating wildly. The resulting saliency may be difficult to interpret since the –unknown– parts of the baseline which the networks considers as important (which part of the blurred goldfinch lead to the power drill classification?) have an influence on the saliency of the current image. Ideally, we would like to have some result about saliency that gives good scores across many different baselines. Future research is needed.

Another choice to be made is the regularisation. We used here Gaußian filters; different sizes compared in Figure 6. It is a tradeoff between blurring out the boundaries and inviting noisiness, however even a small filter appears to be enough to avoid the algorithm from converging to adversarial examples (seemingly smaller than what Fong et al. (2019) require). It is even possible to scale down the filter during the optimisation to eventually give very sharp edges but not noise elsewhere, though it is somewhat dubious what sense this makes mathematically, from the point of view that the filtering represents just a metric on R. Again, further research is needed to assess the reliability.

REFERENCES

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 31*, pp. 9505–9515. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/8160-sanity-checks-for-saliency-maps.pdf.

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. *CoRR*, 2017.

Ruth Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. *CoRR*, 2017. URL http://arxiv.org/abs/1704.03296v3.

Ruth Fong, Patrick Mandela, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. *ICCV*, 2019. URL https://arxiv.org/abs/1910.08485.

Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T. Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (un)reliability of saliency methods. *CoRR*, 2017. URL http://arxiv.org/abs/1711.00867v1.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. *CoRR*, 2017. URL http://arxiv.org/abs/1703.04730v2.

Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *CoRR*, 2018.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *CoRR*, 2016. URL http://arxiv.org/abs/1610.02391v4.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, 2013.

Pascal Sturmfels, Scott Lundberg, and Su-In Lee. Visualizing the impact of feature attribution baselines. *Distill*, 2020. doi: 10.23915/distill.00022. https://distill.pub/2020/attribution-baselines.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *CoRR*, abs/1703.01365, 2017. URL http://arxiv.org/abs/1703.01365.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning, 2016.

A APPENDIX

A.1 CANONICAL TIME REPARAMETRISATION

Proof of Lemma 3.2. The function $m\colon [0,1] \to \mathbb{R}$ defined by $m(t) \coloneqq \int_\Omega \varphi(t)$ is increasing and goes from zero to one (since we assume that $\int_\Omega 1 = 1$).

Note first that if $m(t_1) = m(t_2)$, then $\varphi(t_1) = \varphi(t_2)$ from the monotonicity property. Indeed, supposing for instance that $t_1 \leq t_2$, and defining the element $\theta \coloneqq \varphi(t_2) - \varphi(t_1)$ we see that on the one hand $\int_\Omega \theta = 0$, on the other hand, $\theta \geq 0$, so $\theta = 0$ and thus $\varphi(t_1) = \varphi(t_2)$.

Now, define $\mathsf{M} \coloneqq m([0,1]) = \{\, s \in [0,1] \mid \exists t \in [0,1]\, m(t) = s \,\}$. Pick $s \in [0,1]$.

If $s \in \mathsf{M}$ we define $\psi(s) \coloneqq \varphi(t)$ where $m(t) = s$ (and this does not depend on which t fulfills $m(t) = s$ from what we said above). We remark that $\int_\Omega \psi(s) = \int_\Omega \varphi(t) = m(t) = s$.

Now suppose that $s \notin \mathsf{M}$. Define $s_1 \coloneqq \sup(\mathsf{M} \cap [0,s])$ and $s_2 \coloneqq \inf(\mathsf{M} \cap [s,1])$ (neither set are empty since $0 \in \mathsf{M}$ and $1 \in \mathsf{M}$). Since $s_1 \in \mathsf{M}$ and $s_2 \in \mathsf{M}$, there are $t_1 \in [0,1]$ and $t_2 \in [0,1]$ such that $m(t_1) = s_1$ and $m(t_2) = s_2$. Finally define $\psi(s) \coloneqq \varphi(t_1) + (s - s_1)\frac{\varphi(t_2)-\varphi(t_1)}{s_2 - s_1}$. In this case, $\int_\Omega \psi(s) = m(t_1) + (s - s_1)\frac{m(t_2)-m(t_1)}{s_2-s_1} = s$. The path ψ constructed this way is still monotone, and it has the constant speed property, so it is an ablation path. $\square$

A.2 $\mathcal{L}^\infty$-OPTIMAL MONOTONICITY PROJECTION

The algorithm proposed in § A.3 for optimising monotone paths uses updates that can locally introduce nonmonotonicity in the candidate $\hat{\varphi}_1$, so that it is needed to project back onto a monotone path φ_1. The following routine[3] performs such a projection in a way that is optimal in the sense of minimising the $\mathcal{L}^\infty$-distance[4], i.e.

$$\sup_t \big|\varphi_1(t, \mathbf{r}) - \hat{\varphi}_1(t, \mathbf{r})\big| \leq \sup_t \big|\vartheta(t, \mathbf{r}) - \hat{\varphi}_1(t, \mathbf{r})\big|$$

for all $\mathbf{r} \in \Omega$ and any other monotone path ϑ.

The algorithm works separately for each $\mathbf{r}$, i.e. we express it as operating simply on continuous functions $p : [0, 1] \to \mathbb{R}$. The final step effectively *flattens out*, in a minimal way, any region in

Algorithm 1 Make a function $[0, 1] \to \mathbb{R}$ nondecreasing

$\cup_i [l_i, r_i] \leftarrow \{\, t \in [0, 1] \mid p'(t) \leq 0 \,\}$ ▷ Union of intervals where p decreases
for i **do**
  $m_i \leftarrow \frac{p(l_i)+p(r_i)}{2}$
  $l_i \leftarrow \max\{\, t \in [r_{i-1}, l_i] \mid p(t) \leq m_i \,\}$
  $r_i \leftarrow \min\{\, t \in [r_i, l_{i+1}] \mid p(t) \geq m_i \,\}$
end for
for i, j **do**
  if $[l_i, r_i] \cap [l_j, r_j] \neq \emptyset$ **then**
    if $m_j < m_i$, merge the intervals and recompute m as the new center
  end if
end for
return $t \mapsto \begin{cases} p(t) & \text{if } t \notin \cup_i [l_i, r_i] \\ m_i & \text{if } t \in [l_i, r_i] \end{cases}$

which the function was decreasing.

In practice, this algorithm is executed not on continuous functions but on a PCM-discretised representation; this changes nothing about the algorithm except that instead as real numbers, l, r and t are represented by integral indices.

A.3 PATH OPTIMISATION ALGORITHM

As said in tsection § 5, our optimisation algorithm is essentially gradient descent of a path φ: it repeatedly seeks the direction within the space of all paths that (first ignoring the monotonicity constraint) would affect the largest increase to $P(\varphi)$ as per (2). As discussed before, this choice already requires a metric to obtain a vector-gradient from the covector-differential, which could be either the implicit ℓ^2 metric on the discretised representation (pixels), or a more physical kernel/filter-based metric. We conceptually use the latter, however for technical reasons do not immediately apply the corresponding filter to the differential but rather to the *path*, which is not quite equivalent but does have the effect of avoiding noise from creeping into the state.

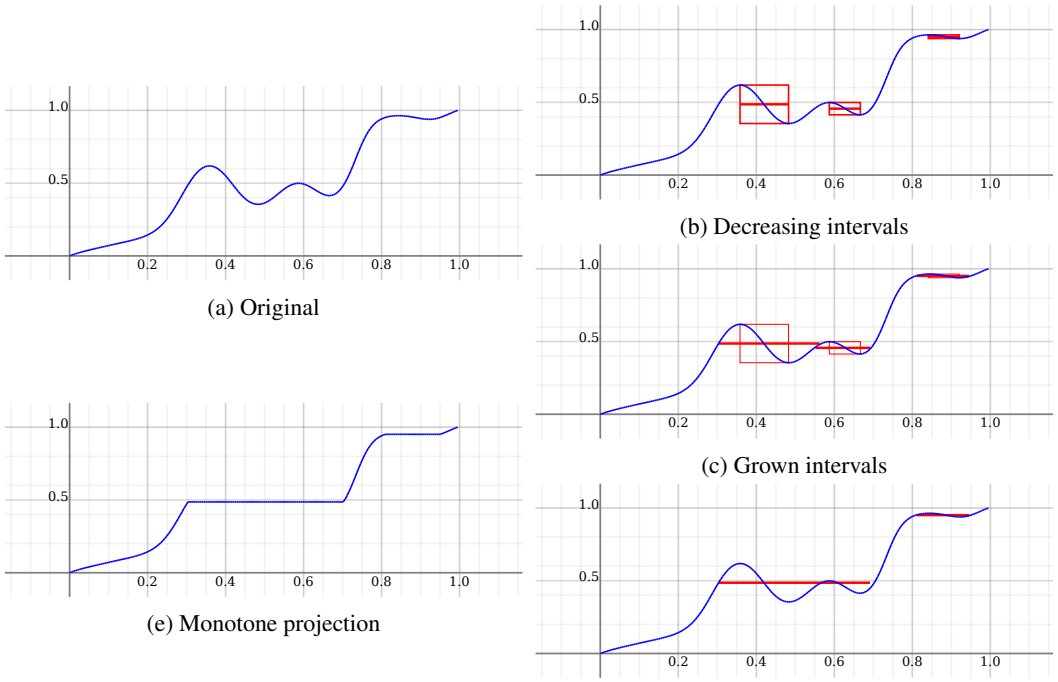
Unlike with the monotonisation condition, the update can easily be made to preserve speed-constness by construction, by projecting for each t the gradient $\mathbf{g}$ on the sub-tangent-space of zero change to $\int_\Omega \varphi(t)$, by subtracting the constant function times $\int_\Omega \mathbf{g}(t)$. Note this requires the measure of Ω to be normalised, or else considered at this point.

Then we apply these gradients time-wise as updates to the path, using a scalar product in the channel-space to obtain the best direction for φ itself (as opposed to the corresponding image composit $x_{\varphi,t}$).

Simply projecting the updates path then again to the set of legal (in the sense of Definition 3.1) ablation paths would presumably enough to converge towards a saturated path with high score,

[3] It is easy to come up with other algorithms for monotonising a (discretised) function. One could simply *sort the array*, but that is not optimal with respect to any of the usual function norms; or clip the derivatives to be nonnegative and then rescale the entire function, but that is not robust against noise pertubations.

[4] Note that the optimum is not necessarily unique.



(a) Original

(b) Decreasing intervals

(c) Grown intervals

(d) Merged intervals

(e) Monotone projection

Figure 7: Example view of the monotonisation algorithm in practice. (a) contains decreasing intervals, which have been localised in (b). For each interval, the centerline is then extended to meet the original path non-decreasingly (c). In some cases, this will cause intervals overlapping; in this case merge them to a single interval and re-grow from the corresponding centerline (d). Finally, replace the path on the intervals with their centerline (e).

Algorithm 2 Projected Gradient Descent

$\varphi \leftarrow ((t, \mathbf{r}) \mapsto t)$ ▷ Start with linear-interpolation path
while φ is not sufficiently saturated **do**
  for t in $[0, 1]$ **do**
    $x_{\varphi,t} := (1 - \varphi(t))\, x_0 + \varphi(t)\, x_1$
    compute $F(x_{\varphi,t})$ with gradient $\mathbf{g} := \nabla F(x_{\varphi,t})$
    let $\hat{\mathbf{g}} := \mathbf{g} - \int_\Omega \mathbf{g}$ ▷ ensure $\hat{\mathbf{g}}$ does not affect mass of $\varphi(t)$
    update $\varphi(t, \mathbf{r}) \leftarrow \varphi(t, \mathbf{r}) - \gamma\ \langle \hat{\mathbf{g}}(\mathbf{r}) \mid |x_1 - x_0\rangle$, for $\mathbf{r}$ in Ω ▷ γ is learning rate
    (optional) apply a filter to $\varphi(t)$
  end for
  (optional) apply nonlinear gain to φ
  for $\mathbf{r}$ in Ω **do**
    re-monotonise $t \mapsto \varphi(t, \mathbf{r})$, using Algorithm 1
  end for
  clamp $\varphi(t, \mathbf{r})$ to $[0, 1]$ everywhere
  re-parametrise φ, such that $\int_\Omega \varphi(t) = t$ for all t (using §A.1)
end while.

however in tests with artificial constant gradient we found out that this requires extremely many iterations, with the number apparently scaling with the image dimension. The problem here is that saturating updates tend to be undone again by the speed-normalising reparametrisation, except for the most affected pixel. (Just increasing the learning rate does not help with this.) If the gradients come from a deep CNN and for every pixel the monotonicity needs to be restored, such many iterations would be prohibitly computation-intensive.

Fortunately the converges can be sped up dramatically, by artificially aiding saturation through the use of a nonlinear gain step, i.e. φ is pointwise tweaked with a sigmoidal function that brings values $< \frac{1}{2}$ closer to 0, values $> \frac{1}{2}$ closer to 1. For this, we found a $\tanh$-based function to work well:

$$\varphi(\mathbf{r}, t) \leftarrow \frac{\tanh(2 \cdot \zeta \cdot (\varphi(\mathbf{r}, t) - \frac{1}{2}))}{2 \cdot \tanh \zeta} + \frac{1}{2}.$$

With gentle parameters like $\zeta = 0.8$, we found convergence to be sped up sufficiently. The examples in §6 required 20-40 iterations.