A Background on Optimal Transport

The Wasserstein space $\mathbb{W}_2(\Omega)$ with Ω a convex and compact subset of $\mathbb{R}^d$ is the space $\mathcal{P}(\Omega)$ of probability measures over Ω, equipped with the distance W_2 given by the solution to the optimal transport problem

$$W_2^2(\alpha,\beta) = \min_{\gamma\in\Pi(\alpha,\beta)} \int_{\Omega\times\Omega} \|x-y\|^2 \,\mathrm{d}\gamma(x,y) \tag{11}$$

where $\Pi(\alpha,\beta)$ is the set of probability distribution over $\Omega\times\Omega$ with first marginal α and second marginal β, i.e. $\Pi(\alpha,\beta) = \{\gamma\in\mathcal{P}(\Omega\times\Omega) \mid \pi_{1\sharp}\gamma = \alpha,\ \pi_{2\sharp}\gamma = \beta\}$ where $\pi_1(x,y) = x$ and $\pi_2(x,y) = y$. The optimal transport problem can be seen as looking for a transportation plan minimizing the cost of displacing some distribution of mass from one configuration to another. This problem indeed has a solution in our setting (see for example [48, 53]). If α is absolutely continuous and $\partial\Omega$ is α-negligible then the problem in (11) (called the Kantorovich problem) has a unique solution and is equivalent to the following problem, called the Monge problem,

$$W_2^2(\alpha,\beta) = \min_{T \text{ s.t. } T_\sharp\alpha=\beta} \int_{\Omega} \|T(x)-x\|^2 \,\mathrm{d}\alpha(x) \tag{12}$$

and this problem has a unique solution $T^\star$ linked to the solution $\gamma^\star$ of (11) through $\gamma^\star = (\mathrm{id}, T^\star)_\sharp\alpha$. Another equivalent formulation of the optimal transport problem in this setting is the dynamical formulation ([5]). Here, instead of directly pushing samples of α to β using T, we can equivalently displace mass, according to a continuous flow with velocity $v_t : \mathbb{R}^d \to \mathbb{R}^d$. This implies that the density α_t at time t satisfies the *continuity equation* $\partial_t\alpha_t + \nabla\cdot(\alpha_t v_t) = 0$, assuming that initial and final conditions are given by $\alpha_0 = \alpha$ and $\alpha_1 = \beta$ respectively. In this case, the optimal displacement is the one that minimizes the total action caused by v :

$$\begin{aligned} W_2^2(\alpha,\beta) = \min_v \int_0^1 \|v_t\|^2_{L^2(\alpha_t)} \,\mathrm{d}t \\ \text{s.t. } \partial_t\alpha_t + \nabla\cdot(\alpha_t v_t) = 0,\ \alpha_0 = \alpha, \alpha_1 = \beta \end{aligned} \tag{13}$$

Instead of describing the density's evolution through the continuity equation, we can describe the paths ϕ_t^x taken by particles at position x from α when displaced along the flow v. Here ϕ_t^x is the position at time t of the particle that was at $x\sim\alpha$ at time 0. The continuity equation is then equivalent to $\partial_t\phi_t^x = v_t(\phi_t^x)$. See chapters 4 and 5 of [48] for details. Rewriting the conditions as necessary, Problem (13) becomes

$$\begin{aligned} W_2^2(\alpha,\beta) = \min_v \int_0^1 \|v_t\|^2_{L^2((\phi_t^\cdot)_\sharp\alpha)} \,\mathrm{d}t \\ \text{s.t. } \partial_t\phi_t^x = v_t(\phi_t^x),\ \phi_0^\cdot = \mathrm{id}, (\phi_1^\cdot)_\sharp\alpha = \beta \end{aligned} \tag{14}$$

and the optimal transport map $T^\star$ that solves (12) is in fact $T^\star(x) = \phi_1^x$ for ϕ that solves the continuity equation together with the optimal $v^\star$ from (14). The

optimal vector field is related to the optimal map through $v_t^\star = (T^\star - \mathrm{id}) \circ (T_t^\star)^{-1}$, where $T_t^\star = (1-t)\mathrm{id} + tT^\star$ and is invertible. This simply means that the points move in straight lines and with constant speed from x to $T^\star(x)$. The $\mathbb{W}_2(\Omega)$ space is a metric geodesic space, and the geodesic between α and β is the curve α_t found while solving (13). It is also given by $\alpha_t = (\pi_t)_\sharp \gamma^\star = (T_t^\star)_\sharp \alpha$, where $\pi_t(x,y) = (1-t)x + ty$. We refer to Section 5.4 of [48] for these results on optimal transport.

Optimal transport maps have some regularity properties under some boundedness assumptions. We mention the following result from [17]:

Theorem 4. *Suppose there are X, Y, bounded open sets, such that the densities of α and β are null in their respective complements and bounded away from zero and infinity over them respectively.*
Then, if Y is convex, there exists $\eta > 0$ such that the optimal transport map T between α and β is $C^{0,\eta}$ over X.
If Y isn't convex, there exists two relatively closed sets A, B in X, Y respectively such that $T \in C^{0,\eta}(X \setminus A, Y \setminus B)$, where A and B are of null Lebesgue measure. Moreover, if the densities are in $C^{k,\eta}$, then $C^{0,\eta}$ can be replaced by $C^{k+1,\eta}$ in the conclusions above. In particular, if the densities are smooth, then the transport map is a diffeomorphism.

A final result that we mention is the following, which says that the inverse of the optimal transport map between α and β is the optimal transport map from β to α,

Theorem 5. *If α and β are absolutely continuous measures supported respectively on compact subsets X and Y of $\mathbb{R}^d$ with negligible boundaries, then there exists a unique couple (T, S) of functions such that the five following points hold*

- *$T : X \to Y$ and $S : Y \to X$*
- *$T_\# \alpha = \beta$ and $S_\# \beta = \alpha$*
- *T is optimal for the Monge problem from α to β*
- *S is optimal for the Monge problem from β to α*
- *$T \circ S \overset{\beta - a.s.}{=} \mathtt{id}$ and $S \circ T \overset{\alpha - a.s.}{=} \mathtt{id}$*

B Background on Numerical Methods for ODEs

We refer to [45] for this quick background on numerical methods for ODEs. Consider the Cauchy problem $x' = f(t,x)$ with initial condition $x(t_0) = x_0$ and a subdivision $t_0 < t_1 < .. < t_N = t_0 + T$ of $[t_0, t_0 + T]$. Denote the time-steps $h_n := t_{n+1} - t_n$ for $0 \leq n < N$ and define $h_{\max} := \max h_n$. In a one-step method, an approximation of $x(t_n)$ is x_n given by

$$\frac{x_{n+1} - x_n}{h_n} = \phi(t_n, x_n, h_n)$$

For $\phi(t,x,h) = f(t,x)$, we get Euler's method: $x_{n+1} = x_n + h_n f(t_n, x_n)$.

Definition 2. (Consistency and order) *For a one-step method, the consistency errors e_n, for $0 \leq n < N$, are*

$$e_n = \frac{x(t_{n+1}) - \Phi(t_n, x(t_n), h_n)}{h_n} = \frac{x(t_{n+1}) - x(t_n)}{h_n} - \phi(t_n, x(t_n), h_n)$$

where x is solution. The local (truncation) errors are $h_n e_n$. The method is consistent if $\max|e_n|$ goes to zero as h_{max} goes to zero. For $p \in \mathbb{N}^*$, the method has order p if $\max|e_n| \leq Ch_{max}^p$ for a constant C that depends on f, t_0 and T.

Theorem 6. (Consistency criterion) *If f and ϕ are continuous then the one-step method is consistent if and only if $\phi(t, x, 0) = f(t, x)$ for all (t, x).*

Theorem 7. (Order criterion) *If f is $\mathcal{C}^p$ and ϕ is $\mathcal{C}^p$ in h then the one-step method is of order p if and only if $\partial_h^k \phi(t, x, 0) = \frac{1}{k+1} f^{[k]}(t, x)$ for all (t, x) and $0 \leq k < p$ where $f^{[0]} = f$ and $f^{[k]} = \partial_t f^{[k-1]} + f\partial_x f^{[k-1]}$.*

Corollary 1. (Consistency and order of Euler's method) *If f is continuous then Euler's method is consistent. If f is $\mathcal{C}^1$ then Euler's method has order 1.*

Definition 3. (Zero-stability) *A one-step method is zero-stable (or stable) if $\exists\ S > 0$ such that for all $(x_n)_{0 \leq n \leq N}$, $(\tilde{x}_n)_{0 \leq n \leq N}$ and $(\varepsilon_n)_{0 \leq n < N}$ satisfying*

$$\frac{x_{n+1} - x_n}{h_n} = \phi(t_n, x_n, h_n)$$

and

$$\frac{\tilde{x}_{n+1} - \tilde{x}_n}{h_n} = \phi(t_n, \tilde{x}_n, h_n) + \epsilon_n$$

for $0 \leq n < N$, we have

$$\max_n \|\tilde{x}_n - x_n\| \leq S(\|\tilde{x}_0 - x_0\| + T \max_n |\epsilon_n|)$$

, where $\epsilon_n = \varepsilon_n / h_n$. The constant S is the stability constant of the method.

Theorem 8. (Zero-stability criterion) *If ϕ is uniformly L-Lipschitz in its second variable, then the one-step method is stable with constant e^{LT}.*

Corollary 2. (Zero-stability of Euler's method) *If f is Lipschitz in its second variable, then Euler's method is stable.*

Definition 4. (Convergence) *A numerical method converges if its global error $\max_n \|x(t_n) - x_n\|$ goes to zero as h_{max} goes to zero.*

Theorem 9. (Convergence criterion) *If a method is consistent and stable with stability constant S, then it converges and $\max_n \|x(t_n) - x_n\| \leq ST \max|e_n|$. If the method is of order p with constant C, then $\max_n \|x(t_n) - x_n\| \leq STCh_{max}^p$*

Corollary 3. (Convergence of Euler's method) *Euler's method converges if f is $\mathcal{C}^0$ and Lipschitz in x. If f is also $\mathcal{C}^1$ then it converges with speed $O(h_{max})$.*

C Proofs

C.1 Proof of Theorem 1

Proof. A solution v to (5) exists and is linked to an optimal transport map T that is a solution to (4) through $v_t = (T - \mathtt{id}) \circ T_t^{-1}$ where $T_t := (1-t)\mathtt{id} + tT$ which is invertible (see Appendix A).

By Theorem 4 in Appendix A, being an optimal transport map, T is η-Hölder on X. So for all $a, b \in \text{support}(\alpha_t)$ and $t \in [0, 1[$, where $\alpha_t = (\phi_t^{\cdot})_\sharp \alpha = (T_t)_\sharp \alpha$ with ϕ solving (5) with v, we have

$$\|v_t(a) - v_t(b)\| \leq \|T_t^{-1}(a) - T_t^{-1}(b)\| + C\|T_t^{-1}(a) - T_t^{-1}(b)\|^\eta$$

Since $(\alpha_t)_{t=0}^1$ is a geodesic between α and $\beta = \alpha_1 = T_\sharp \alpha$, then $(\alpha_s)_{s=0}^t$ is a geodesic between α and α_t (modulo reparameterization to $[0, 1]$). And since $\alpha_s = (T_s)_\sharp \alpha$, the map T_t is an optimal transport map between α and α_t. Therefore its inverse T_t^{-1} is an optimal transport map (see Theorem 5 in Appendix A) and is η_t-Hölder with $0 < \eta_t \leq 1$ (being a push-forward by T_t, the support of α_t satisfies the conditions of Theorem 4 in Appendix A). Therefore, for all $a, b \in \text{support}(\alpha_t)$

$$\|v_t(a) - v_t(b)\| \leq C_t \|a - b\|^{\eta_t} + CC_t^\eta \|a - b\|^{\eta\eta_t} \tag{15}$$

and for all $a, b \in \text{support}(\alpha_{t_m})$

$$\|r_m(a) - r_m(b)\| \leq \varepsilon + C_{t_m} \|a - b\|^{\eta_{t_m}} + CC_{t_m}^\eta \|a - b\|^{\eta\eta_{t_m}} \tag{16}$$

by the hypothesis on r and the triangle inequality. Let $K := \max_m C_{t_m} + CC_{t_m}^\eta$, $\zeta_1 := \eta \min_m \eta_{t_m}$ and $\zeta_2 := \max_m \eta_{t_m}$. Then, we have the desired result immediately from (16).

Remark 1. If the convexity hypothesis on the support Y of the target distribution β is too strong, we still get the same results almost everywhere. More precisely, if the set Y such that β is bounded away from zero and infinity on Y and is zero on $Y^\complement$ is open and bounded but not convex, then the solution map T is η-Hölder almost everywhere on X (see Appendix A).

Remark 2. If the distributions α and β in Theorem 1 are $\mathcal{C}^{k,\eta}$ (i.e all derivatives up to the k-th derivative are η-Hölder), then the optimal transport map T is $\mathcal{C}^{k+1,\eta}$. This means that the more regular the data, the more regular the network we find.

C.2 Proof of Theorems 2 and 3

Proof. Since $T_t^{-1}(\phi_t^x) = x$, we have for any $a_0, b_0 \in X$ by the triangle inequality

$$\begin{aligned}\|r_m(a_m) - r_m(b_m)\| \leq\ & \|r_m(a_m) - r_m(\phi_{t_m}^{a_0})\| + \|r_m(\phi_{t_m}^{a_0}) - v_{t_m}(\phi_{t_m}^{a_0})\| + \\ & + \|v_{t_m}(\phi_{t_m}^{a_0}) - v_{t_m}(\phi_{t_m}^{b_0})\| + \|r_m(\phi_{t_m}^{b_0}) - v_{t_m}(\phi_{t_m}^{b_0})\| + \\ & + \|r_m(b_m) - r_m(\phi_{t_m}^{b_0})\|\end{aligned}$$

So

$$\|r_m(a_m) - r_m(b_m)\| \leq \varepsilon + \|a_0 - b_0\| + C\|a_0 - b_0\|^\eta + \\ + L(\|a_m - \phi_{t_m}^{a_0}\| + \|b_m - \phi_{t_m}^{b_0}\|)$$

where $L = \max_m L_m$ and L_m is the Lipschitz constant of r_m (which is Lipschitz being a composition of matrix multiplications and activations such as ReLU). This the bound in Theorem 2.

In this bound, the term $\|a_m - \phi_{t_m}^{a_0}\|$ (and likewise $\|b_m - \phi_{t_m}^{b_0}\|$) represents the distance between the point a_m we get after m residual blocks (i.e. after m Euler steps using the approximation r of v) and the point $\phi_{t_m}^{a_0}$ we get by following the solution vector field v up to time t_m. By the properties of the Euler method (consistency and zero-stability, see Corollaries 1, 2 and 3 in Appendix B), under more regularity conditions on v, it is possible to bound this term. Indeed, if v is $\mathcal{C}^1$ and M-Lipschitz in x (this is not stronger than the regularity we get on v through our regularization, because we still need to use (15)), we have for constants $R, S > 0$,

$$\|\phi_{t_m}^{a_0} - a_m\| \leq \|\phi_{t_m}^{a_0} - \tilde{a}_m\| + \|\tilde{a}_m - a_m\| \leq S\varepsilon + SRh$$

where $\tilde{a}_m$ comes from the Euler scheme with access to v (i.e. $\tilde{a}_{m+1}{:=}\tilde{a}_m + hv_{t_m}(\tilde{a}_m)$ and $\tilde{a}_0{:=}a_0$), R is the consistency constant of the Euler method and S is its zero-stability constant. Likewise, we get the same bound for $\|b_m - \phi_{t_m}^{b_0}\|$.

If $a_0, b_0 \notin X$, we need to introduce $\hat{a}_0 := \mathrm{Proj}_X(a_o)$ and $\hat{b}_0 := \mathrm{Proj}_X(b_o)$ to apply (15). We now get

$$\|r_m(a_m) - r_m(b_m)\| \leq \varepsilon + \|a_0 - b_0\| + C\|a_0 - b_0\|^\eta + \\ + L(\|a_m - \phi_{t_m}^{\hat{a}_0}\| + \|b_m - \phi_{t_m}^{\hat{b}_0}\|)$$

Bounding the terms $\|a_m - \phi_{t_m}^{\hat{a}_0}\|$ and $\|b_m - \phi_{t_m}^{\hat{b}_0}\|$ now gives

$$\|\phi_{t_m}^{\hat{a}_0} - a_m\| \leq \|a_m - \tilde{a}_m\| + \|\tilde{a}_m - \phi_{t_m}^{\hat{a}_0}\| \leq S(\|a_0 - \hat{a}_0\| + \varepsilon) + SRh$$

where $\tilde{a}_m$ now comes from the Euler scheme with access to v that starts at $\hat{a}_0$ (meaning $\tilde{a}_{m+1}{:=}\tilde{a}_m + hv_{t_m}(\tilde{a}_m)$ and $\tilde{a}_0{:=}\hat{a}_0$). Likewise, we get the same bound for $\|b_m - \phi_{t_m}^{\hat{b}_0}\|$.

Since $\|a_0 - \hat{a}_0\| = \mathrm{dist}(a_0, X)$ and $\|b_0 - \hat{b}_0\| = \mathrm{dist}(b_0, X)$, we get the bound in Theorem 3. Note that if we use the stability of the ODE instead of the Euler method to bound $\|a_m - \phi_{t_m}^{\hat{a}_0}\|$ we get the same result. Indeed, if $\tilde{a}_m$ again comes from the Euler scheme with access to v that starts at a_0 (meaning $\tilde{a}_{m+1}{:=}\tilde{a}_m + hv_{t_m}(\tilde{a}_m)$ and $\tilde{a}_0{:=}a_0$), we can write, for some constant $F > 0$

$$\|\phi_{t_m}^{\hat{a}_0} - a_m\| \leq \|a_m - \tilde{a}_m\| + \|\tilde{a}_m - \phi_{t_m}^{a_0}\| + \|\phi_{t_m}^{a_0} - \phi_{t_m}^{\hat{a}_0}\| \\ \leq S\varepsilon + SRh + F\|a_0 - \hat{a}_0\|$$

since

$$\begin{aligned}\|\phi_{t_m}^{a_0} - \phi_{t_m}^{\hat{a}_0}\| &\leq \|a_0 - \hat{a}_0\| + \int_0^{t_m} \|v_s(\phi_s^{a_0}) - v_s(\phi_s^{\hat{a}_0})\| \, \mathrm{d}s \\ &\leq \|a_0 - \hat{a}_0\| + M \int_0^{t_m} \|\phi_s^{a_0} - \phi_s^{\hat{a}_0}\| \, \mathrm{d}s \leq F\|a_0 - \hat{a}_0\|\end{aligned}$$

where we get the last line by Gronwall's lemma.

D Additional Experiments

D.1 Adversarial Attacks

White-box attacks have access to the network's weights and architecture. The Fast Gradient Method (FGM) [19] takes a perturbation step in the direction of the gradient that maximizes the loss. Projected Gradient Descent (PGD) [37] and the Basic Iterative Method (BIM) [31] are iterative versions of FGM. We use the Auto-PGD-CE [13] variant of PGD which has an adaptive step size. Two slower but more powerful attacks are DeepFool (DF) [40], which iteratively perturbs an input in the direction of the closest decision boundary, and Carlini-Wagner (CW) [8], which solves an optimization problem to find the perturbation. AutoAttack (AA) [13] is a combination of three white-box attacks (two variants of Auto-PGD [13] and the FAB attack of [12]), and of the black-box Square Attack (SA) [4]. Black-box attacks don't have any knowledge about the network and can only query it. We use two such attacks: Hop-Skip-Jump (HSJ) [10], which estimates the gradient direction at the decision boundary, and the Boundary Attack (BA) [7], which starts from a large adversarial input and moves towards the boundary decision to minimize the perturbation. We use a maximal perturbation of ϵ=0.03 for FGM, APGD, BIM and AA. We use the L_2 norm for CW and HSJ and L_∞ for the other attacks. We use ART [43] and its default hyper-parameter values (except those mentioned) to generate the adversarial samples, except for AA for which we use the authors' original code. The number of iterations is 50 for HSJ, 5000 for BA, 10 for CW and 100 for APGD and DF.

D.2 Implementation Details

For ResNeXt50 [59] on CIFAR100 [30], we train for 300 epochs using SGD with a learning rate of 0.1 (divided by ten at epochs 150, 225 and 250), Kaiming initialization, a batch size of 128 and weight decay of 0.0001. For RCE training, the only changes are that the learning rate is 0.05 and the initialization is orthogonal with a gain of 0.05.

For ResNet110 [21] on CIFAR10 [30], we train for 300 epochs using SGD with a learning rate of 0.1 (divided by ten at epochs 150, 225 and 250), orthogonal initialization with a gain of 0.05, a batch size of 256, weight decay of 0.0001 and gradient clipping at 5. For RCE training, the only change is that we don't use gradient clipping.

For WideResNet [60] on TinyImageNet, we train for 300 epochs using SGD with a learning rate of 0.1 (divided by ten at epochs 150, 225 and 250), orthogonal initialization with a gain of 0.1, a batch size of 114 and weight decay of 0.0001.

For the magnitude parameter of the Mahalanobis detector, we try all the values tried in their paper for the magnitude and we report the best results.

D.3 Adversarial detection training data

See Figure 3.

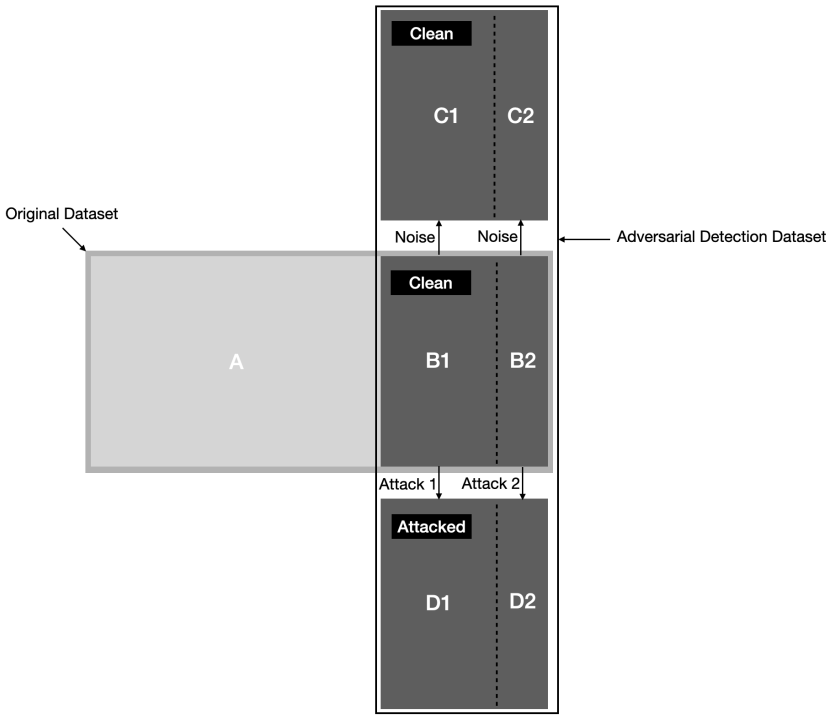

Fig. 3. Adversarial detection dataset creation. A∪B1∪B2 is the original dataset, where A is the training set and B1∪B2 is the test set. We create a noisy version of B1∪B2 by adding random noise to each sample in B1∪B2 to get C1∪C2. Noisy samples are considered clean (i.e. not attacked) in adversarial detection training. We create an attacked version of B1∪B2 by creating an attacked image from each image in B1∪B2 to get D1∪D2. In the case of generalization to unseen attacks, Attack 2 used to create D2 from B2 is different from Attack 1 used to create D1 from B1. Otherwise, Attack 1 and Attack 2 are the same. B1∪C1∪D1 is the adversarial detection training set and B2∪C2∪D2 is the adversarial detection test set.

D.4 Preliminary Experiments

We see in Figure 4 below that training deep ResNets without batch-normalization is near impossible, whereas LAP-ResNets maintain the same performance and stability without ResNets for up to 50 blocks. LAP-ResNets are also compared in this regard to the small step method of [63], which simply adds a small weight h of around 0.1 in front of the residue function to make ResNets more stable. The Least Action Principle has the same improved stability when training without batch-normalization in Figure 4 as this method, while also improving the test accuracy when batch-normalization is used [26] which the small step method does not claim.

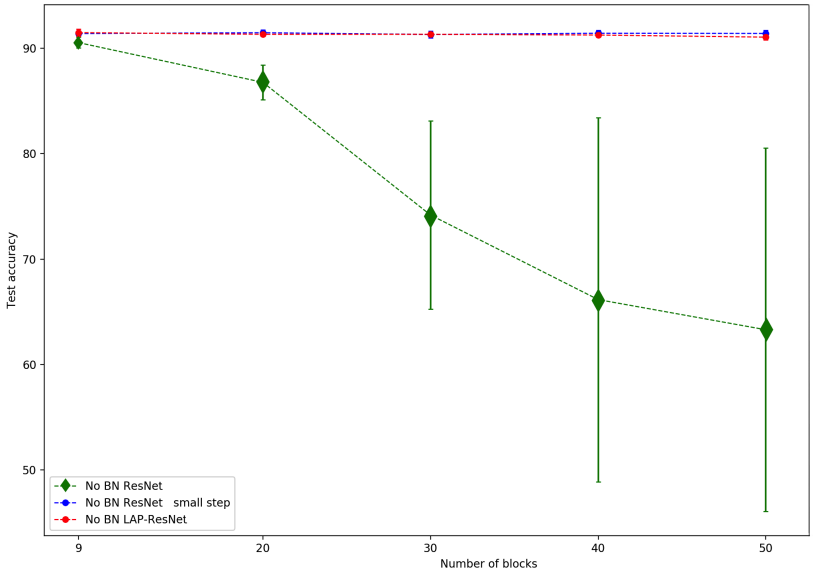

Fig. 4. Test accuracy of ResNets of various depths without batch-normalization on CIFAR10.

D.5 Detection of Seen Attacks

In the tables below, VAN corresponds to detectors trained on a vanilla network, RCE on an RCE-network and LAP on a LAP-network.

Table 3: Average adversarial detection accuracy of seen attacks and standard deviation over 5 runs using ResNet110 on CIFAR10.

	Attack			
Det	FGM	APGD	BIM	DF
VAN T	97.1 ± 0.6	94.1 ± 0.4	97.5 ± 0.5	**100** ± 0.5
RCE T	96.0 ± 0.5	95.3 ± 0.8	96.2 ± 0.6	99.9 ± 0.1
LAP T	**98.7** ± 0.3	**97.5** ± 0.4	**99.3** ± 0.3	99.8 ± 0.1
VAN M	87.8 ± 4.2	82.1 ± 4.0	86.8 ± 4.7	91.5 ± 3.2
RCE M	93.0 ± 0.6	87.9 ± 0.5	92.3 ± 1.0	95.0 ± 0.4
LAP M	95.6 ± 0.6	90.7 ± 0.7	95.4 ± 0.5	96.7 ± 0.7
VAN N	94.6 ± 0.7	94.3 ± 0.5	95.0 ± 0.5	99.8 ± 0.1

Table 4: Average adversarial detection accuracy of seen attacks and standard deviation over 5 runs using ResNet110 on CIFAR10.

	Attack			
Det	CW	AA	HSJ	BA
VAN T	**98.0** ± 0.5	88.9 ± 1.3	**99.9** ± 0.1	96.6 ± 0.6
RCE T	89.4 ± 0.5			
LAP T	98.0 ± 0.4	**94.1** ± 0.8	99.9 ± 0.1	**97.0** ± 0.2
VAN M	85.6 ± 2.6	80.5 ± 2.2	85.5 ± 1.8	80.2 ± 2.1
RCE M	83.4 ± 0.5			
LAP M	93.4 ± 0.6	90.0 ± 0.9	94.6 ± 0.4	89.6 ± 0.4
VAN N	93.9 ± 9.2	88.8 ± 1.4	99.7 ± 0.1	92.1 ± 0.5

Table 5: Average adversarial detection accuracy of seen attacks and standard deviation over 5 runs using ResNeXt50 on CIFAR100.

	Attack			
Det	FGM	PGD	BIM	AA
VAN T	97.3 ± 0.5	96.0 ± 0.5	98.0 ± 0.3	84.9 ± 0.7
RCE T	97.4 ± 0.4	97.0 ± 0.1	97.8 ± 0.2	50.1 ± 0.1
LAP T	**98.3** ± 0.3	**97.8** ± 0.5	**98.9** ± 0.1	**87.6** ± 0.6
VAN M	95.8 ± 0.5	93.9 ± 0.5	96.1 ± 0.6	83.9 ± 0.7
RCE M	96.5 ± 0.4	94.7 ± 0.4	96.6 ± 0.6	50.1 ± 0.1
LAP M	96.8 ± 0.4	94.6 ± 0.7	97.8 ± 0.5	86.6 ± 0.5
VAN N	94.7 ± 0.7	94.2 ± 1.0	94.7 ± 0.6	84.8 ± 0.8

Table 6: Average adversarial detection accuracy of seen attacks and standard deviation over 5 runs using ResNeXt50 on CIFAR100.

	Attack	
Detector	DF	CW
VAN TR	**99.80** ± 0.15	**97.04** ± 0.88
RCE TR	99.04 ± 0.14	92.52 ± 0.34
LAP TR	99.58 ± 0.18	97.80 ± 0.18
VAN MH	97.30 ± 0.45	95.38 ± 0.56
RCE MH	97.64 ± 0.44	88.36 ± 0.62
LAP MH	97.12 ± 0.28	96.42 ± 0.42
VAN NS	99.56 ± 0.21	90.72 ± 1.39

Table 7: Average adversarial detection accuracy of seen attacks and standard deviation over 5 runs using WideResNet on TinyImageNet.

	Attack			
Det	FGM	APGD	BIM	AA
VAN T	**95.4** ± 0.4	**95.2** ± 0.5	**95.3** ± 0.5	**81.4** ± 0.4
LAP T	95.1 ± 0.5	95.2 ± 0.7	95.1 ± 0.7	81.2 ± 0.5
VAN M	81.1 ± 1.1	79.7 ± 1.0	81.2 ± 1.3	78.4 ± 0.7
LAP M	85.3 ± 1.0	85.1 ± 0.6	82.5 ± 1.6	78.4 ± 1.0
VAN N	94.9 ± 0.7	94.9 ± 0.9	95.0 ± 0.6	81.3 ± 0.2

D.6 Detection of Unseen Attacks

Table 8: Average adversarial detection accuracy of unseen attacks after training on FGM and standard deviation over 5 runs using ResNet110 on CIFAR10.

	Attack			
Detector	APGD	BIM	AA	DF
VAN TR	89.3 ± 1.6	96.0 ± 0.7	**85.1** ± 1.1	**91.0** ± 0.9
RCE TR	91.8 ± 1.1	93.6 ± 1.1	50.0 ± 0.1	63.4 ± 1.1
LAP TR	**92.8** ± 0.5	**98.8** ± 0.4	84.2 ± 0.5	75.5 ± 1.2
VAN MH	77.3 ± 4.7	77.2 ± 4.8	72.1 ± 3.1	80.1 ± 3.4
RCE MH	81.5 ± 0.6	82.6 ± 1.3	50.0 ± 0.1	81.2 ± 0.7
LAP MH	87.9 ± 0.8	84.9 ± 0.4	81.9 ± 1.2	81.6 ± 0.7
VAN NS	92.1 ± 0.5	93.9 ± 0.4	51.8 ± 0.6	51.4 ± 0.58

Table 9: Average adversarial detection accuracy of unseen attacks after training on FGM and standard deviation over 5 runs using ResNet110 on CIFAR10.

	Attack		
Detector	CW	HSJ	BA
VAN TR	**93.2** ± 1.0	**93.0** ± 0.9	**90.9** ± 0.6
RCE TR	60.5 ± 0.9	63.9 ± 1.0	52.5 ± 0.5
LAP TR	75.2 ± 1.0	76.8 ± 0.6	75.0 ± 0.4
VAN MH	79.9 ± 3.7	79.7 ± 3.0	79.3 ± 3.0
RCE MH	76.0 ± 0.9	81.6 ± 0.9	68.5 ± 1.2
LAP MH	81.5 ± 0.6	81.5 ± 0.3	81.4 ± 0.8
VAN NS	50.84 ± 1.1	52.1 ± 0.7	59.9 ± 5.4

Table 10: Average adversarial detection accuracy of unseen attacks after training on FGM and standard deviation over 5 runs using ResNeXt50 on CIFAR100.

	Attack			
Detector	APGD	BIM	AA	DF
VAN T	91.9 ± 0.8	95.0 ± 0.5	73.3 ± 1.0	**85.2** ± 0.6
RCE T	87.7 ± 0.5	95.1 ± 0.9	50.0 ± 0.1	72.3 ± 0.4
LAP T	89.3 ± 0.8	**97.7** ± 0.3	74.0 ± 1.3	76.0 ± 1.0
VAN M	90.9 ± 0.8	93.2 ± 0.3	73.1 ± 0.6	82.7 ± 0.9
RCE M	82.0 ± 0.7	88.6 ± 0.8	50.0 ± 0.1	74.1 ± 0.8
LAP M	86.7 ± 0.9	93.9 ± 0.4	**80.0** ± 0.6	79.4 ± 1.6
VAN NS	**92.2** ± 0.4	93.9 ± 1.0	51.3 ± 0.4	51.6 ± 0.5

Table 11: Average adversarial detection accuracy of unseen attacks after training on FGM and standard deviation over 5 runs using ResNeXt50 on CIFAR100.

	Attack		
Detector	CW	HSJ	BA
VAN T	**78.2** ± 1.0	**85.0** ± 0.4	**92.1** ± 4.8
RCE T	61.9 ± 0.5	72.4 ± 0.4	57.7 ± 0.4
LAP T	74.7 ± 1.1	78.1 ± 3.4	71.9 ± 3.9
VAN M	76.4 ± 0.7	82.8 ± 1.2	84.5 ± 2.0
RCE M	63.0 ± 0.6	74.6 ± 0.3	63.2 ± 0.9
LAP M	80.9 ± 2.0	80.5 ± 3.7	78.2 ± 2.0
VAN NS	51.0 ± 0.4	52.0 ± 0.7	57.9 ± 7.5

Table 12: Average adversarial detection accuracy of unseen attacks after training on FGM and standard deviation over 5 runs using WideResNet on TinyImageNet.

	Attack		
Detector	APGD	BIM	AA
VAN TR	93.26 ± 0.60	94.66 ± 0.49	**77.04** ± 0.74
LAP TR	93.48 ± 0.72	**94.80** ± 0.56	76.58 ± 0.48
VAN MH	76.96 ± 0.94	77.02 ± 1.08	60.36 ± 0.62
LAP MH	77.96 ± 0.49	78.00 ± 0.77	61.96 ± 0.89
VAN NS	**94.06** ± 0.61	94.62 ± 0.64	72.82 ± 1.98

Table 13: Average adversarial detection accuracy of unseen attacks after training on FGM and standard deviation over 5 runs using WideResNet on TinyImageNet.

	Attack	
Detector	DF	CW
VAN TR	**90.62** ± 0.60	91.42 ± 1.06
LAP TR	90.12 ± 0.55	**91.52** ± 0.89
VAN MH	73.18 ± 0.59	75.52 ± 0.82
LAP MH	73.98 ± 1.12	76.22 ± 0.83
VAN NS	71.96 ± 4.03	65.60 ± 2.20

D.7 Detection Rate of Successful Adversarial Samples

As in [33], we might be only concerned with detecting adversarial samples that successfully fool the network and that are created from clean samples that are correctly classified. We find that the detection rate of successful adversarial samples is always very high and close to 100% on our detector. On seen attacks, the results are in Table 14. On unseen attacks, the results are in Table 15.

Table 14: Average detection rate of successful adversarial samples from seen attacks over 5 runs on Network/LAP-Network.

Attack	Detector	ResNet110 CIFAR10	ResNeXt50 CIFAR100	WideResNet TinyImageNet
FGM	TR	97.7/**98.6**	98.2/**98.6**	95.4/**95.9**
	MH	88.3/93.9	96.7/97.1	84.0/85.0
	NS	95.9	95.0	94.4
APGD	TR	99.3/**99.4**	97.1/**97.9**	**96.7**/96.4
	MH	85.9/86.8	95.8/92.7	82.7/84.8
	NS	95.9	94.8	94.5
BIM	TR	98.3/**99.6**	98.6/**99.2**	95.0/**96.1**
	MH	88.1/93.8	96.6/98.0	85.8/86.2
	NS	96.4	94.6	94.3
AA	TR	**100**/**100**	**100**/**100**	**100**/**100**
	MH	88.3/95.4	98.8/98.7	95.6/96.5
	NS	99.9	99.9	99.9
DF	TR	**100**/99.9	**99.9**/99.4	
	MH	93.8/98.2	97.6/97.8	
	NS	99.9	99.3	
CW	TR	98.6/98.7	99.9/99.6	
	MH	83.5/93.9	98.1/98.1	
	NS	**99.9**	**100**	
HSJ	TR	**100**/99.9		
	MH	82.3/95.1		
	NS	99.7		

Table 15: Average detection rate of successful adversarial samples from unseen attacks after training on FGM over 5 runs.

Attack	Detector	ResNet110 CIFAR10	ResNeXt50 CIFAR100	WideResNet TinyImageNet
APGD	TR	**96.94**	**98.76**	**100.0**
	MH	79.80	90.86	79.16
	NS	93.02	91.8	93.20
BIM	TR	**98.68**	**98.56**	**100.0**
	MH	78.66	93.38	86.42
	NS	94.24	93.8	93.98
AA	TR	**98.54**	**74.06**	**91.06**
	MH	81.10	73.74	71.70
	NS	10.22	8.32	54.5
DF	TR	**93.42**	**75.14**	**95.00**
	MH	79.96	73.64	68.34
	NS	8.12	8.06	50.80
CW	TR	**92.22**	**72.90**	**96.00**
	MH	78.96	72.34	76.66
	NS	8.76	7.38	51.96
HSJ	TR	**93.22**	**75.42**	
	MH	78.14	73.66	
	NS	9.60	8.94	
BA	TR	**93.38**	**91.04**	
	MH	79.42	76.44	
	NS	25.50	21.90	

D.8 False Positive Rate

We report here the false positive rate on seen (Table 16) and unseen (Table 17) attacks of both detectors.

Table 16: Average FPR of seen attacks over 5 runs on Network/LAP-Network.

Attack	Detector	ResNet110 CIFAR10	ResNeXt50 CIFAR100	WideResNet TinyImageNet
FGM	TR	3.3/**1.5**	3.4/**1.9**	**3.7**/5.2
	MH	13.9/3.5	5.2/3.3	18.1/16.3
	NS	5.4	5.0	4.5
APGD	TR	6.3/**2**	4.6/**2.3**	5.3/4.9
	MH	18.7/4.7	6.3/3.6	17.5/16.9
	NS	4.9	5.0	**4.5**
BIM	TR	2.7/**0.8**	2.5/**1.9**	**3.6**/4.6
	MH	13.6/3.1	4.6/3.3	18.0/15.9
	NS	4.8	4.8	4.2
AA	TR	1.9/**1.5**	4.0/**4.1**	**6.8**/7.0
	MH	13.3/6.4	13.7/10.8	14.8/15.3
	NS	2.9	5.1	7.4
DF	TR	**0.1**/0.2	**0.2**/0.3	
	MH	10.2/4.6	2.9/2.7	
	NS	0.4	**0.2**	
CW	TR	2.6/2.8	**0.3**/0.4	
	MH	12.4/7.4	2.5/2.2	
	NS	**2.3**	2.8	
HSJ	TR	**0.1**/**0.1**		
	MH	11.9/5.8		
	NS	0.3		

Table 17: Average FPR of unseen attacks after training on FGM over 5 runs.

Attack	Detector	ResNet110 CIFAR10	ResNeXt50 CIFAR100	WideResNet TinyImageNet
APGD	TR	**6.70**	8.96	**9.44**
	MH	21.36	**7.28**	14.48
BIM	TR	**6.02**	7.10	**9.44**
	MH	21.46	**7.08**	14.54
AA	TR	**8.06**	**7.82**	**9.48**
	MH	23.36	7.90	14.48
DF	TR	**6.34**	**3.74**	**9.44**
	MH	18.48	7.78	14.54
CW	TR	**5.64**	**3.74**	**9.44**
	MH	18.86	7.68	14.82
HSJ	TR	**6.70**	**4.86**	
	MH	18.72	7.56	
BA	TR	**6.70**	**5.74**	
	MH	19.10	7.90	

D.9 AUROC

We report in Table 18 the AUROC of seen attacks, and in Table 19 the AUROC of unseen attacks. Note that the AUROC is computed on the class-agnostic random forest detector, not on the ensemble of the class-agnostic and the class-conditional detectors.

Table 18: Average AUROC of seen attacks on Network/LAP-Network.

Attack	Detector	ResNet110 CIFAR10	ResNeXt50 CIFAR100	WideResNet TinyImageNet
AA	TR	94.91/**99.95**	**99.86**/99.70	82.58/**82.95**
	MH	81.94/94.17	87.13/94.32	70.85/71.36
DF	TR	**99.94**/99.92	**99.84**/99.58	
	MH	89.60/94.17	86.88/91.82	
CW	TR	98.77/**99.96**	**99.85**/99.61	
	MH	88.33/94.31	86.33/91.36	
HSJ	TR	**99.95**/99.94		
	MH	86.01/93.45		

Table 19: Average AUROC of unseen attacks after training on FGM over 5 runs.

Attack	Detector	ResNet110 CIFAR10	ResNeXt50 CIFAR100	WideResNet TinyImageNet
AA	TR	**82.70**	**71.53**	**9.48**
	MH	76.46	66.60	61.61
DF	TR	**90.09**	**72.61**	**9.44**
	MH	82.25	66.00	71.72
CW	TR	**88.84**	**71.89**	**9.44**
	MH	81.59	66.06	71.57
HSJ	TR	**87.30**	**74.85**	
	MH	75.89	67.07	

D.10 Time Comparison

With a ResNeXt50 on CIFAR100 and a Tesla V100 GPU, it takes our method (including the time to generate FGM attacks) 66 seconds to extract its features from both the clean and the adversarial samples, while it takes the Mahalanobis method 110 seconds. Mahalanobis also extracts some statistics from the training set prior to adversarial detection training, which takes an additional 35 seconds. Our feature vector is of size 32, compared to 5 for the Mahalanobis detector. So our random forest takes only 4 more seconds to train than the Mahalanobis one (7 vs 3 seconds). Computation of the features our detector uses (norms and cosines) is in $O(MD)$, where M is the number of residual blocks and D is the largest embedding dimension inside the network.

D.11 Detection of Out-Of-Distribution Samples

Since our analysis applies to all out-of-distribution (OOD) samples, we test detection of OOD samples in a similar setting to the Mahalanobis paper [33]. We use the same ResNet110 and ResNeXt50 models trained on CIFAR10 and CIFAR100 respectively. Since the detectors need to be trained, we are in the OOD setting where we have a first dataset for training the network (CIFAR10 in Tables 20 and 21 and CIFAR100 in Tables 26 and 27) and a second dataset from another distribution that is not the test OOD distribution to train the detector on. This could be another dataset (CIFAR100 in Table 20), some images found in the wild, or a perturbation of our dataset that we generate using an adversarial attack (CW on CIFAR10 in Table 21, and AA and CW on CIFAR100 in Tables 26 and 27 respectively). Detectors can then be used by training them to distinguish between these first two datasets, and then testing them on distinguishing between the first dataset and a third unseen dataset (SVHN [41] in both tables). The accuracy is in Tables 20 to 27. The AUROC is in Tables 22 to 29. The false positive rate (FPR) at a fixed true positive rate (TPR) of 95% is in Tables 24 to 31. Our detector performs very well and better than the MH detector in three of

the four experiments, and in the fourth case, the MH detector benefits from LAP training by 8 percentage points (Table 26). Without any extra data available, using the CW adversarial attack allows to detect OOD samples from an unseen distribution with more than 90% accuracy and an FPR of less than 10% at a fixed TPR of 95%. The choice of the attack is also important, as CW allows for much better detection of unseen samples from SVHN than AA.

Table 20: Average OOD detection accuracy and standard deviation over 5 runs using ResNet110 trained on CIFAR10.

Detector	CIFAR100 (seen)	SVHN (unseen)
VAN TR	98.30 ± 0.46	97.46 ± 0.49
RCE TR	**98.42** ± 0.40	98.20 ± 0.39
LAP TR	98.30 ± 0.22	**98.50** ± 0.47
VAN MH	86.88 ± 1.52	91.28 ± 0.92
RCE MH	94.82 ± 0.45	92.16 ± 0.57
LAP MH	94.84 ± 0.41	90.46 ± 1.45

Table 21: Average OOD detection accuracy and standard deviation over 5 runs using ResNet110 trained on CIFAR10.

Detector	CW-CIFAR10 (seen)	SVHN (unseen)
VAN TR	**97.42** ± 0.57	**91.38** ± 0.95
RCE TR	91.54 ± 6.06	77.58 ± 6.72
LAP TR	97.28 ± 0.62	85.46 ± 2.64
VAN MH	81.80 ± 1.96	83.76 ± 1.13
RCE MH	76.74 ± 2.75	54.24 ± 3.46
LAP MH	89.68 ± 0.65	76.72 ± 1.73

Table 22: Average OOD detection AUROC and standard deviation over 5 runs using ResNet110 trained on CIFAR10.

Detector	CIFAR100 (seen)	SVHN (unseen)
VAN TR	99.64 ± 0.13	98.74 ± 0.47
RCE TR	99.61 ± 0.09	99.01 ± 0.25
LAP TR	**99.73** ± 0.09	**99.43** ± 0.28
VAN MH	92.74 ± 1.50	97.00 ± 0.80
RCE MH	97.97 ± 0.25	96.26 ± 0.40
LAP MH	98.06 ± 0.38	96.31 ± 0.75

Table 23: Average OOD detection AUROC and standard deviation over 5 runs using ResNet110 trained on CIFAR10.

Detector	CW-CIFAR10 (seen)	SVHN (unseen)
VAN TR	**99.32** ± 0.14	**96.29** ± 0.71
RCE TR	96.53 ± 0.58	86.34 ± 3.98
LAP TR	99.31 ± 0.07	96.12 ± 0.59
VAN MH	88.38 ± 2.94	88.04 ± 4.03
RCE MH	88.28 ± 2.22	79.53 ± 3.78
LAP MH	95.22 ± 0.90	86.54 ± 3.60

Table 24: Average OOD detection FPR at 95% TPR and standard deviation over 5 runs using ResNet110 trained on CIFAR10.

Detector	CIFAR100 (seen)	SVHN (unseen)
VAN TR	1.16 ± 0.66	2.68 ± 1.05
RCE TR	1.16 ± 0.41	1.94 ± 0.86
LAP TR	**1.02** ± 0.44	**1.54** ± 0.56
VAN MH	36.42 ± 4.29	15.66 ± 1.69
RCE MH	7.34 ± 0.90	20.30 ± 4.45
LAP MH	6.98 ± 2.35	17.56 ± 5.76

Table 25: Average OOD detection FPR at 95% TPR and standard deviation over 5 runs using ResNet110 trained on CIFAR10.

Detector	CW-CIFAR10 (seen)	SVHN (unseen)
VAN TR	2.71 ± 1.01	6.68 ± 0.98
RCE TR	19.2 ± 2.35	24.98 ± 5.43
LAP TR	**2.70** ± 0.64	**5.68** ± 1.12
VAN MH	49.46 ± 4.80	36.78 ± 6.97
RCE MH	41.94 ± 6.44	52.72 ± 7.71
LAP MH	23.06 ± 6.29	58.40 ± 14.42

Table 26: Average OOD detection accuracy and standard deviation over 5 runs using ResNeXt50 trained on CIFAR100.

Detector	AA-CIFAR100 (seen)	SVHN (unseen)
VAN TR	84.48 ± 0.59	75.32 ± 0.62
RCE TR	50.04 ± 0.07	55.44 ± 5.76
LAP TR	**87.10** ± 0.12	72.98 ± 2.72
VAN MH	83.44 ± 0.48	78.82 ± 0.48
RCE MH	50.04 ± 0.07	58.74 ± 2.36
LAP MH	86.04 ± 0.31	**86.84** ± 0.68

Table 27: Average OOD detection accuracy and standard deviation over 5 runs using ResNeXt50 trained on CIFAR100.

Detector	CW-CIFAR100 (seen)	SVHN (unseen)
VAN TR	95.82 ± 0.67	**92.92** ± 1.36
RCE TR	76.48 ± 0.75	75.66 ± 0.68
LAP TR	**95.94** ± 0.57	85.94 ± 2.88
VAN MH	94.96 ± 0.81	85.10 ± 1.50
RCE MH	76.20 ± 0.72	72.20 ± 1.47
LAP MH	94.82 ± 0.34	88.92 ± 1.26

Table 28: Average OOD detection AUROC and standard deviation over 5 runs using ResNeXt50 trained on CIFAR100.

Detector	AA-CIFAR100 (seen)	SVHN (unseen)
VAN TR	94.96 ± 0.38	78.77 ± 1.15
RCE TR	50.12 ± 0.05	50.30 ± 8.13
LAP TR	**96.45** ± 0.08	76.28 ± 0.73
VAN MH	93.32 ± 0.41	85.02 ± 0.97
RCE MH	50.15 ± 0.08	58.04 ± 3.56
LAP MH	94.87 ± 0.24	**92.76** ± 0.39

Table 29: Average OOD detection AUROC and standard deviation over 5 runs using ResNeXt50 trained on CIFAR100.

Detector	CW-CIFAR100 (seen)	SVHN (unseen)
VAN TR	99.00 ± 0.13	95.17 ± 0.31
RCE TR	87.50 ± 1.65	79.99 ± 3.62
LAP TR	**99.16** ± 0.28	94.84 ± 1.63
VAN MH	98.24 ± 0.32	93.07 ± 0.33
RCE MH	86.73 ± 2.03	77.42 ± 3.38
LAP MH	97.84 ± 0.20	**95.92** ± 0.64

Table 30: Average OOD detection FPR at 95% TPR and standard deviation over 5 runs using ResNeXt50 trained on CIFAR100.

Detector	AA-CIFAR100 (seen)	SVHN (unseen)
VAN TR	27.68 ± 1.84	32.66 ± 1.41
RCE TR	96.12 ± 0.96	94.72 ± 3.10
LAP TR	**21.62** ± 0.30	**29.47** ± 0.59
VAN MH	29.75 ± 1.09	39.06 ± 0.96
RCE MH	95.28 ± 0.27	91.42 ± 1.26
LAP MH	24.74 ± 0.69	34.54 ± 1.16

Table 31: Average OOD detection FPR at 95% TPR and standard deviation over 5 runs using ResNeXt50 trained on CIFAR100.

Detector	CW-CIFAR100 (seen)	SVHN (unseen)
VAN TR	5.42 ± 0.82	**8.90** ± 1.20
RCE TR	44.12 ± 3.31	45.68 ± 2.01
LAP TR	**4.90** ± 0.92	10.52 ± 3.61
VAN MH	8.02 ± 0.81	16.38 ± 0.82
RCE MH	45.12 ± 3.76	48.45 ± 4.55
LAP MH	8.10 ± 0.46	9.80 ± 1.21

D.12 Attacking the Detector

We consider here the case where the attacker also attacks the detector (adaptive attacks). We try two such attacks on the TR and MH detectors on ResNet110 trained on CIFAR10. Both attacks are white-box with respect to the network. The first is black-box with respect to the detector. It only knows if an adversarial sample has been detected or not. The second has some knowledge about the detector. It knows what features it uses and can attack it directly to find adversarial features. We test these attacks by looking at the percentage of detected successful adversarial samples that they turn into undetected successful adversarial samples that fool both the network and the detector.

The first attack proceeds as follows. A strong white-box attack (CW) is used on the network on image x that has label y. If it finds a successful adversarial image $\tilde{x}$ that fools the network into predicting $\tilde{y} \neq y$ but is detected by the detector, the attacker will attempt to modify this image $\tilde{x}$ so that the network and the detector are both fooled. For this, the image $\tilde{x}$ is used as the initialization for an attack (HSJ with a budget of 50 iterations and 10000 evaluations) on a black-box Network-Detector system. The attacker considers that the Network-Detector behaves as follows: it outputs the class prediction of the network if the detector does not detect an attack and outputs an additional 'detected' class if the detector detects an attack. The attacker attacks this Network-Detector on image $\tilde{x}$ targeting the $\tilde{y}$ label. This way the network makes a mistake and the 'detected' class is avoided. On the vanilla ResNet110, this attack turns 16.5% of 1700 detected successful adversarial samples $\tilde{x}$ into undetected successful adversarial samples on our detector, compared to 25.7% on the Mahalanobis detector. These percentages are lower on the LAP-ResNet110 as they drop to 6.8% on our detector and 12.9% on the Mahalanobis detector. This shows that LAP training improves the robustness of both adversarial detectors to being attacked themselves, and that the Transport detector is more robust than the MH detector.

The second attack is very similar to the adaptive attack used in [9] to break the Kernel Density detector of [16]. It proceeds as follows. A strong white-box attack (CW) is used on the network on image x that has label y. If it finds a successful adversarial image $\tilde{x}$ that fools the network but is detected by the

detector, the detection features $\tilde{z}$ that $\tilde{x}$ generates when run through the network are used as the initialization for a black-box attack (HSJ with a budget of 50 iterations and 10000 evaluations) on the detector. If successful adversarial detection features z^* that fool the detector are found, the attacker has to find an adversarial perturbation of x that still fools the network and that generates these features z^* (or close features that also fool the detector) when run through the network. We do this as in [9] by solving the following optimization problem:

$$\min_{x^*} -L(N(x^*), y) + c_1\|D(x^*) - z^*\| + c_2\|x^* - x\| \tag{17}$$

where L is the cross-entropy loss, N is the network, and D is the (differentiable) function that returns the detection features of its input. This optimization problem is differentiable and we try differentiable optimization algorithms such as BFGS and NR to solve it. The initial detected successful adversarial image $\tilde{x}$ is used as initialization as in [9]. This attack turns 14% of detected successful adversarial samples $\tilde{x}$ into undetected successful adversarial samples on our detector on the LAP-ResNet110.

Given that initial detection rates of successful adversarial samples are almost 100% (see Appendix D.7), this shows that adaptive attacks do not (at least not easily) circumvent the detector, as detection rates drop to 85% at worst. Obviously, the second attack is stronger than the first one, but it can probably still be improved by using a white-box attack that is specific to random forests for attacking the detector such as [25] or [61], or a different loss than cross-entropy such as the one used in the CW attack. However, the difficulty of combining the attack on the network with that on the detector remains. It is the non-differentiability of the random forest that forces either this separate treatment of network and detector then the use of a proxy differentiable term for the detector (here $\|D(x^*) - z^*\|$ in (17)) to combine both, or the use of a black-box method as in the first attack. Also, we did not consider here the ensemble of the class-conditional detector and the general detector, which is the best performing version of the detector (see Section 5.2), and should be even more robust to adaptive attacks, as the attacker will have to fool two random forest detectors at once and target a particular label, constraining further the optimization problem he solves.