

GENERATION OF UNIVERSAL ADVERSARIAL PERTURBATIONS AGAINST DEEPFAKE DETECTORS

Aymane SNOUSSI

M2 TDSI – SMART IOT

SUPERVISORS :

Christophe CHARRIER

Emmanuel GIGUET

ACADEMIC SUPERVISOR :

David HELBERT

GREYC Laboratory, Caen — University of Poitiers
2025 - 2026

Defence – 10 September 2026

Introduction & Plan

- 2024, Arup. An employee joins a video call with his CFO. Every face on the call is fake. €24 million transferred.
- Deepfakes grew fourfold between 2023 and 2024.
- The only defence at scale: automatic detection, built on neural networks.
- **Those networks can be fooled. That is my subject.**

Context → Problem → Solution → Project management → Assessment → Perspectives

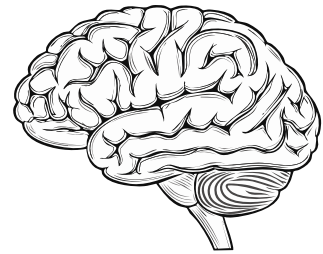
Where the internship took place

GREYC – Caen

- CNRS joint research unit UMR 6072, shared with the University of Caen and ENSICAEN.
- Around 230 members, six research teams.
- **SAFE team** – Security, Architecture, Forensics, Biometrics. Digital investigation and biometrics.

My position – Research intern, Five months, within the COMPROMIS project (national cybersecurity programme, 2024–2027).

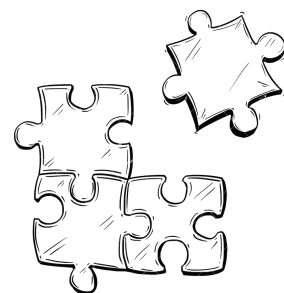




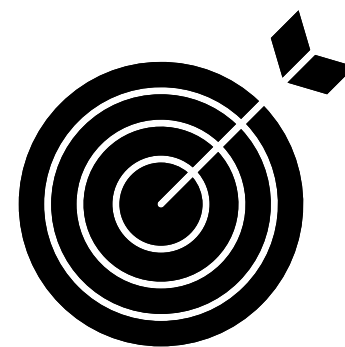
Detectors – CNN architectures such as XceptionNet or EfficientNet, trained to spot what generation leaves behind: lighting inconsistencies, compression artefacts, local deformations around the face.



Training data – large annotated video corpora, FaceForensics++ being the standard.



The flaw – a detector is itself a neural network. Add a computed pattern to an image and it changes its answer, while a human sees no difference. It does not fail loudly: it keeps answering, with confidence, and it is wrong.



My subject – not to attack for the sake of attacking, but to measure how fragile a detection pipeline becomes once it rests on a convolutional network.

Requirements for a realistic attack

1. Universality

- one single matrix δ , computed once, added to any face: $x + \delta$.
- **measured by the fooling rate (FR)**: how often the model changes its answer.

2. Imperceptibility

- The attacked face must look untouched.
- **measured by SSIM**, from **0 to 1**: how close the attacked image stays to the original.

3. Transferability

- **White box**: full access to the model, so access to its gradients.
- **Black box**: only the output verdict is visible. A deployed detector is always a black box.
- **measured by the same FR**, on models never seen



δ is built on models I own, and it must still work on models it has never seen.

Per-image attacks

Method	Reference	Principle	Why not retained
FGSM	Goodfellow et al., 2015	Modify the image once, in one shot	Fast but weak
DeepFool	Moosavi-Dezfooli et al., 2016	Find the smallest change that flips the answer	Works only on that exact image
JSMA	Papernot et al., 2016	Modifies only the most influential pixels	Very slow
PGD	Madry et al., 2018	Repeat small modifications until the answer flips	The strongest, but one image at a time

→ One perturbation per image. Not applicable in real time. Fails universality.

Universal attacks

(built from data)

Method	Reference	Principle	Why not retained
UAP	Moosavi-Dezfooli et al., 2017	One single δ for every image	Needs the dataset
Fast-UAP	Dai & Shu, 2020	Same result, half the images	Needs the dataset
Deepfake UAP	Neekhara et al., 2020	Applied to real deepfake detectors	Needs the dataset

→ Universality solved. But the attacker must own the detector's database. Data-free is the only realistic setting.

DATA-Free Method

PSP-UAP



Ski: 96.7%

Whole Image



Dogsled: 42.8%



Grey whale: 26.4%



Ski: 99.9%

Cropped and Resized Images

A natural image



Safe: 75.8%

Whole Image



Brain coral: 53.0%



Puck: 70.6%



Slot: 20.4%

Cropped and Resized Images

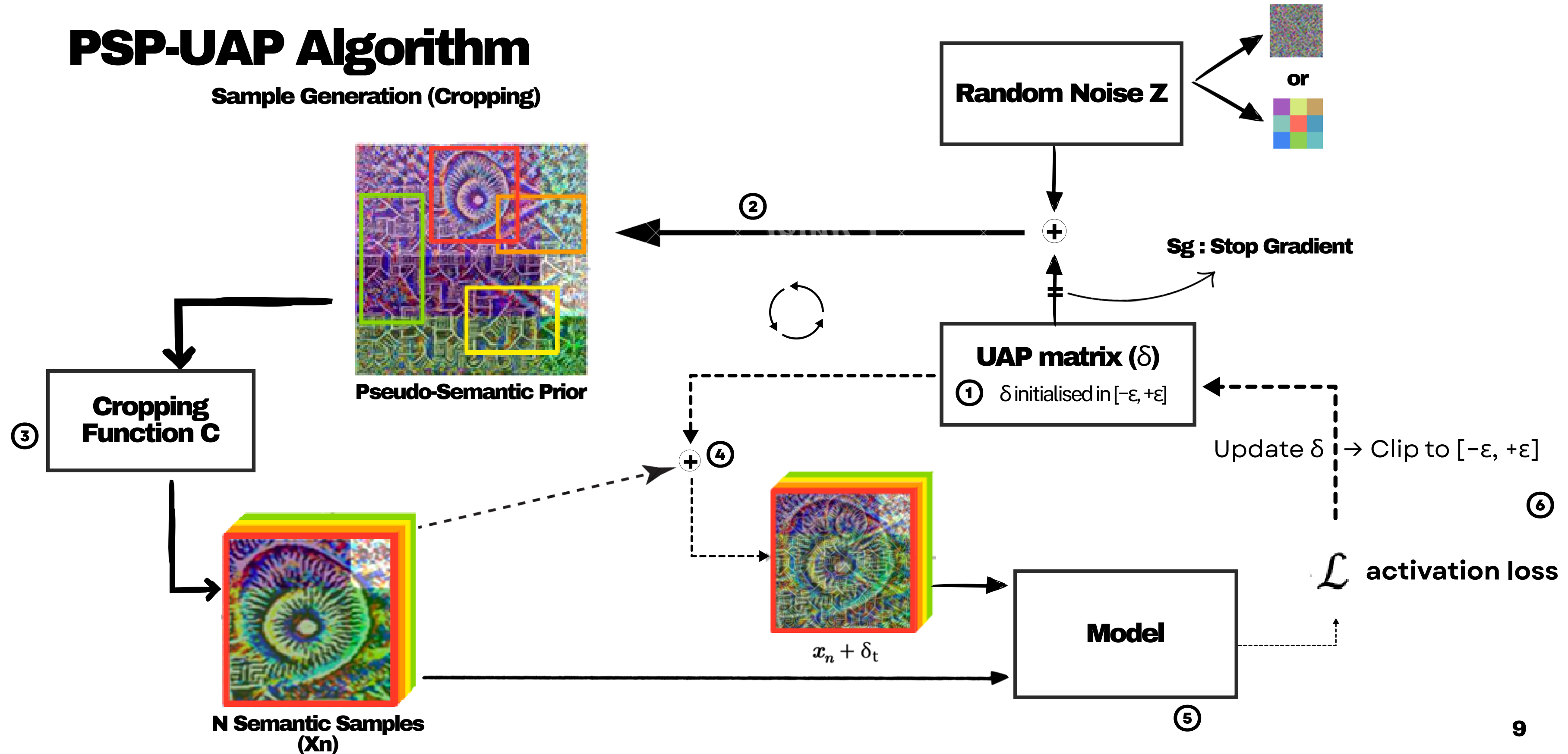
A trained perturbation

- Crop a natural photo: each piece still gets a label, with high confidence.
- Do the same on a trained perturbation: each piece also gets its own label – brain coral, puck, slot.
- The perturbation already contains textures and shapes. It carries semantic content.

→ The idea: use pieces of the perturbation itself as training images. No real photo needed.

PSP-UAP Algorithm

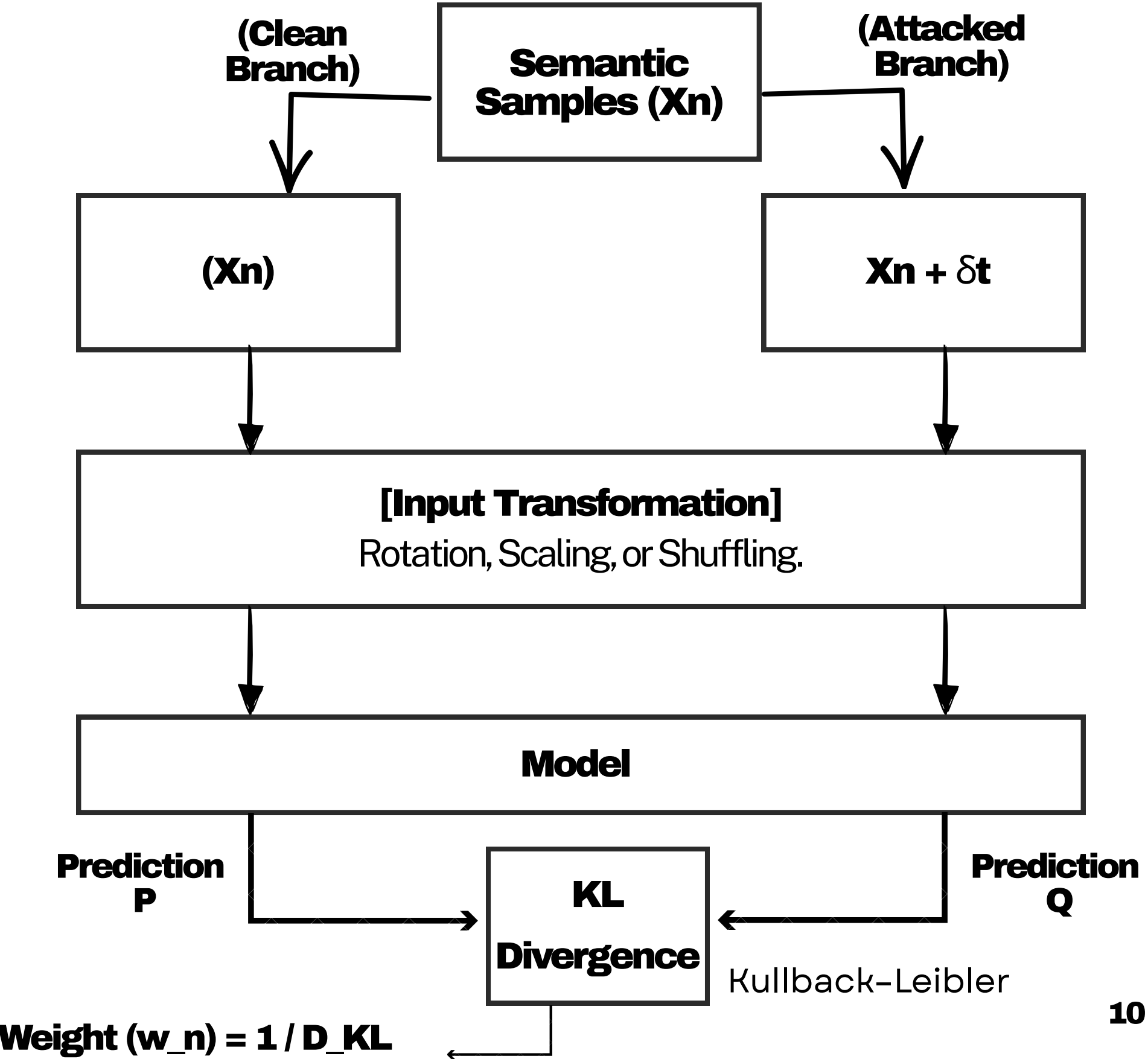
Sample Generation (Cropping)





PSP-UAP Algorithm

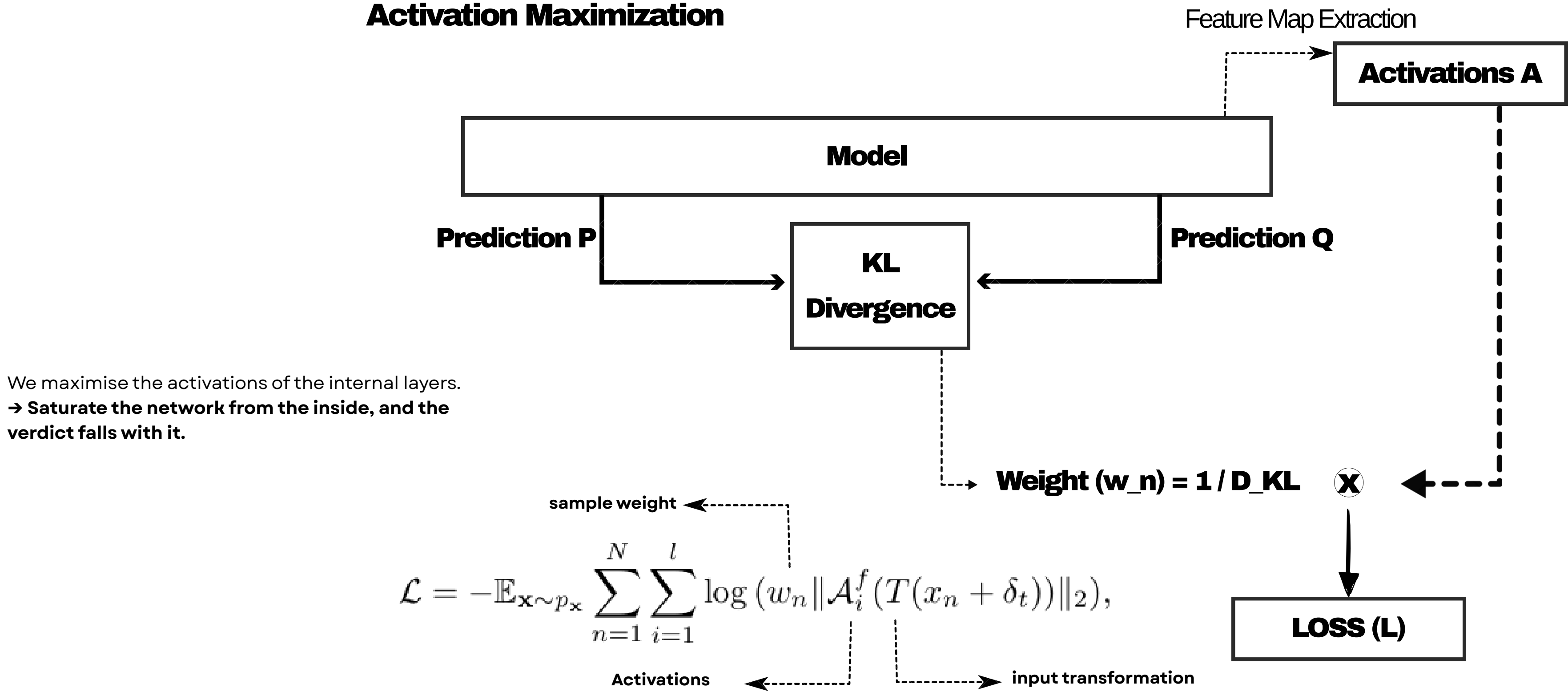
Attack Evaluation
Focusing on the hard samples





PSP-UAP Algorithm

Activation Maximization





Reproducing the reference method

Results measured with the fooling rate (%)

Trained on ↓ / Evaluated on →	VGG16	ResNet50	DenseNet121
VGG16	95	73	63
ResNet50	77	75	66
DenseNet121	82	52	79

→ Both white-box results and transferability have to be improved, so the attack no longer depends on the architecture it was trained on.



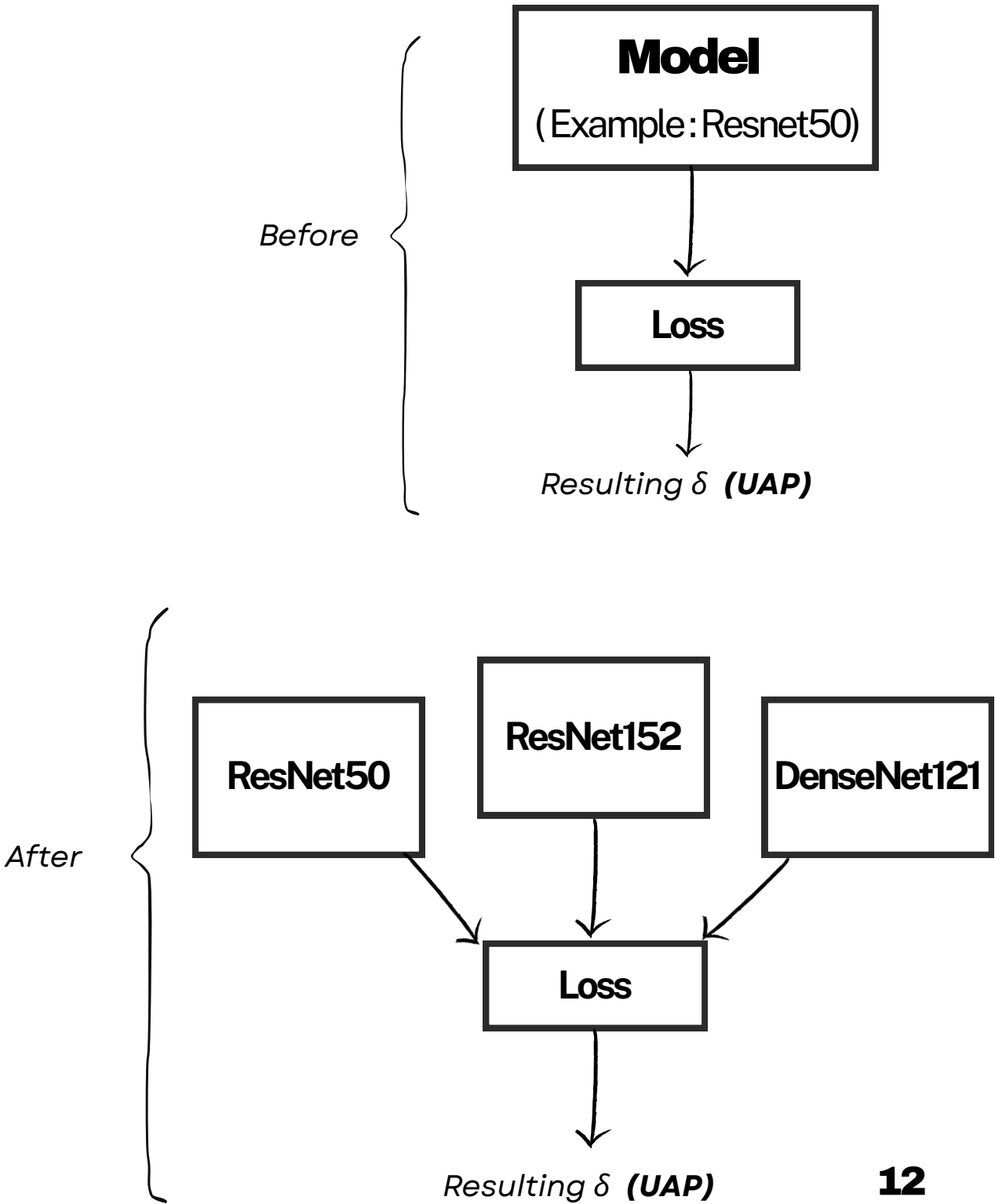
Contribution 1 — surrogate ensemble

1. Surrogate ensemble

- Training on one model gives a δ shaped by that model.
- I train on three at once. Two depths of ResNet(**ResNet50**, **ResNet152**), and **DenseNet121** for a different family.
- Each model contributes at every step. **20,000 iterations**.

→ δ no longer belongs to one architecture.

Stage 1 – ensemble (20k)



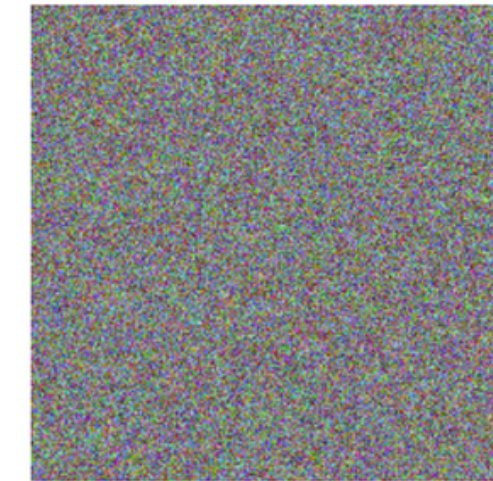
Contribution 2 — diffusion prior

2. Diffusion prior

- The reference prior is the perturbation plus random noise.
- I generate the prior with a diffusion model. Not real images: synthetic patterns that carry structure.

2,000 iterations of refinement, starting from the ensemble result.

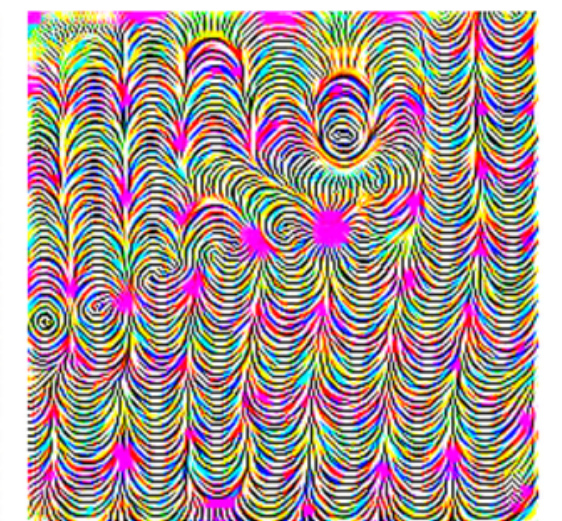
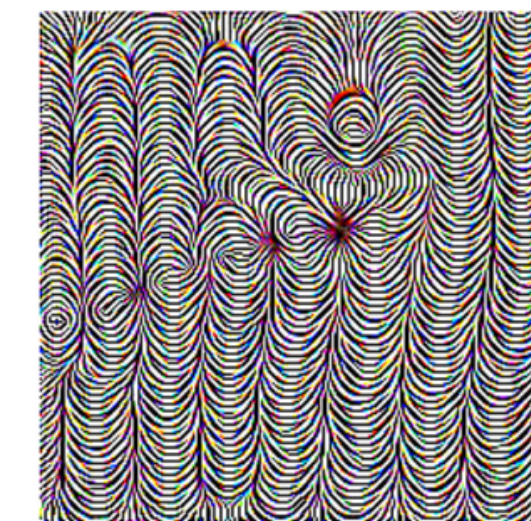
Stage 2 – diffusion prior (2k)



Random noise prior



Diffusion-generated prior



Resulting δ

Contribution 3 — an invisible perturbation

Before: great white shark

After: hammerhead shark



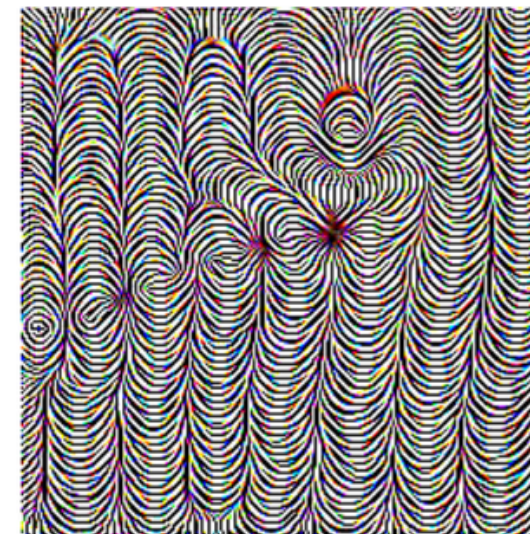
Before: boa constrictor
64.7%

After: hognose snake
16.8%



UAP

FR 77.14 % / SSIM 0.73

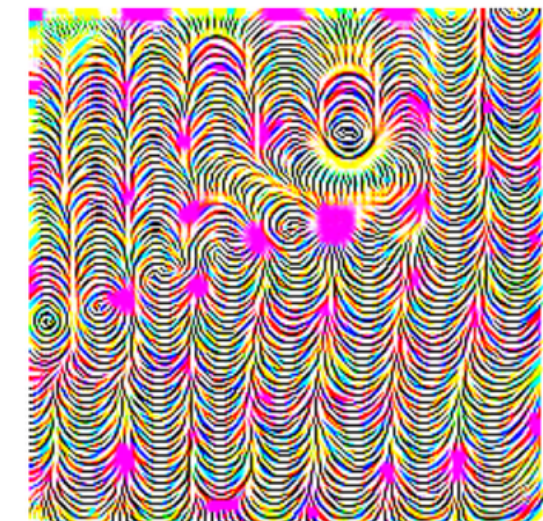


$$\delta \in [-10/255, +10/255]$$

Added everywhere at the same strength –
smooth skin and textured hair get the
same noise

UAP

FR 84.38 % / SSIM 0.80



$$\delta \in [-40/255, +40/255]$$

$$\delta \in [-40/255, +40/255] \times \text{JND map} [0.13 \rightarrow 1]$$

Smooth areas: $\pm 5/255$.
Textured areas: $\pm 40/255$.

----->
JND mask

→ The eye barely sees a change in texture. So that is where I put the energy.

Stage 3 – Perceptual budget(2K)



Each contribution measured separately

Evaluated on 8 architectures – 3 white box (the surrogates), 5 black box

Configuration	SSIM	White-box FR (3 models)	Black-box FR (5 models)	Overall FR
+ Ensemble baseline ($\epsilon = 10$)	0.7303	79.10	75.96	77.14
+ diffusion prior	0.7443	83.17	75.58	78.42
+ perceptual budget (40, 0.13)	0.8025	91.00	80.40	84.38



Original – class #777



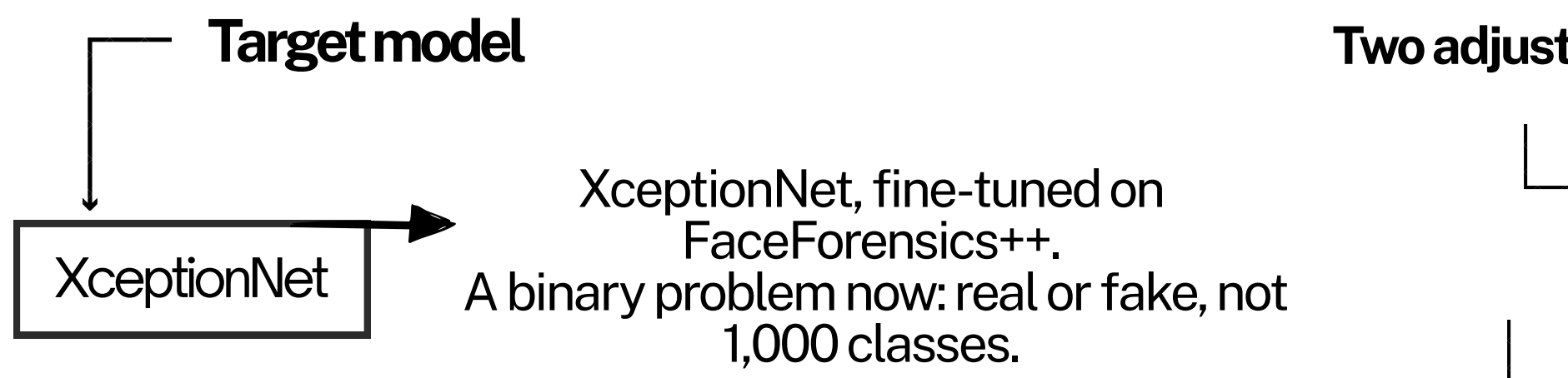
$\epsilon=10$ – class #885,
SSIM 0.769



Perceptual budget–
class #885, SSIM 0.832



Moving the attack to a real deepfake detector



Two adjustments

- 1. Optimise directly at 299 × 299:
Resizing δ afterwards costs about 30 % of its amplitude
- 2. Add a fooling term:

$$\mathcal{L}_{\text{deepfake}} = \mathcal{L}_{\text{act}} + \lambda \mathcal{L}_{\text{fool}}$$

With two classes, saturation alone pushes both ways. This term forces one: fake → real.

Result on 1,998 faces

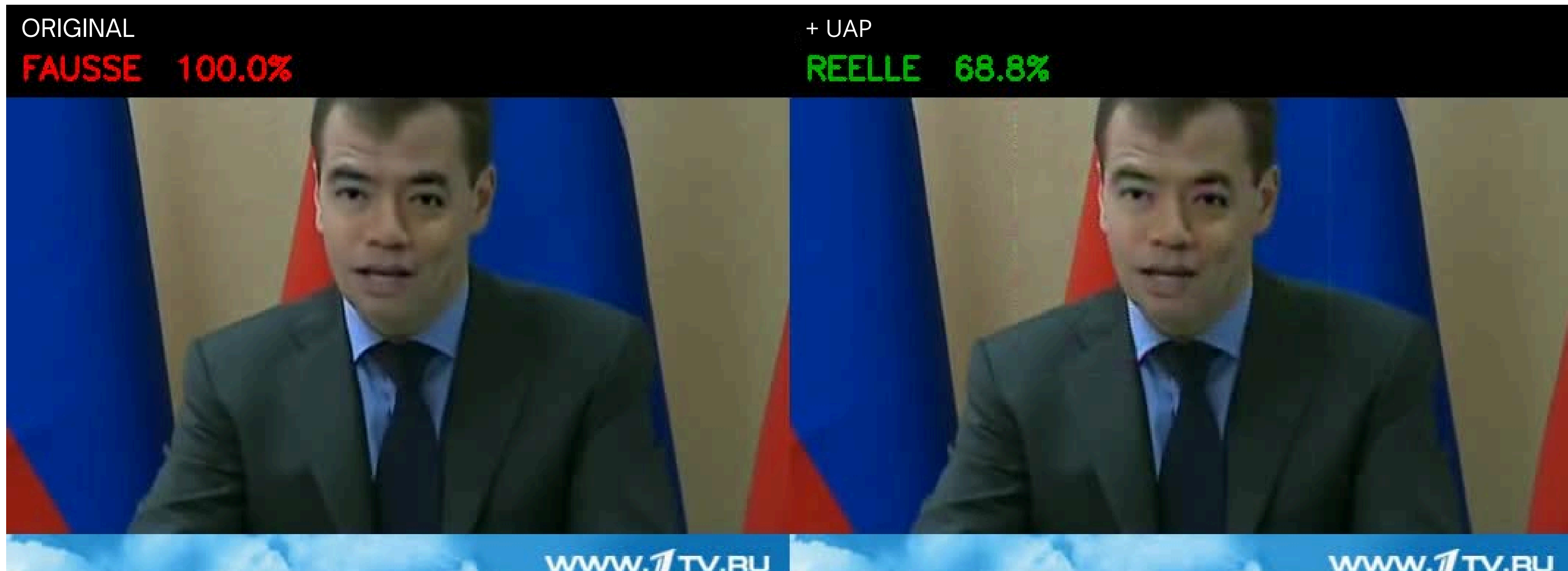
	Before	After
Fake faces detected as fake	92.79 %	0.70 %
Real faces still recognised	97.0 %	98.8 %
Overall accuracy	94.89 %	49.75 %
SSIM	--	0.864

→ 99.3 % of the fakes pass through. The detector keeps its accuracy on real faces, so nothing looks broken.

Applied to a video stream

ORIGINAL – FAKE 100 %

ATTACKED – REAL 64.6 %



+ Same δ on every frame. Nothing is recomputed.

+ **7 videos, all switched from fake to real**

+ Mean SSIM 0.86 – a viewer sees no difference.



Countermeasure 1 — input filtering

Clean the image before detection, to destroy the pattern.

The first column should go up, the other two should stay high.

Filter	Deepfake + UAP <i>% caught</i>	Deepfake, no UAP <i>% caught</i>	Real faces kept <i>% kept</i>
none	0.70 %	92.79 %	97.00%
JPEG quality 50	0.90 %	71.67 %	97.50%
median 5 × 5	45.15 %	96.60 %	71.07%
Gaussian blur $\sigma = 1.5$	81.68 %	93.79 %	55.76 %

→ The filters that block the attack also destroy the detector. No usable defence here.



Countermeasure 2 — adversarial retraining

Retrain XceptionNet with attacked images included.

- **Known attack** → the one used during retraining
- **Unknown attack** → a different δ the model never saw
- **Re-optimised attack** → the attacker recomputes δ against the retrained model

Table 1 – Original detector

Input	Fakes caught	Real faces kept
Clean images	92.86 %	96.92 %
Known attack	0.65 %	97.26 %
Unknown attack	0.97 %	98.63 %

Table 2 – After adversarial retraining

Input	Fakes caught	Real faces kept
Clean images	99.03 %	98.29 %
Known attack	97.73 %	97.95 %
Unknown attack	81.82 %	96.92 %
Re-optimised attack	0.00 %	100 %

→ Retraining works, until the attacker computes a new pattern against the new model.



Countermeasure 3 — detecting the perturbation

Detect whether the image has been attacked

- 1. **Residual** — original image minus a blurred version. What is left is the fine noise.
- 2. **Features** — 9 statistics measuring how irregular that noise is
- 3. **Training** — logistic regression on 2,796 images (1,398 clean / 1,398 attacked).
- 4. **Threshold** — fixed above the highest score reached by clean images.

5. Results:

Detector	Input	UAP detected
Original	Clean images	0.33 %
Original	Known attack	100 %
Retrained	Clean images	0.33 %
Retrained	Re-optimised attack	100 %

→ 0.33 % false alarms,
100 % detection –
including on a pattern
never seen before.

Context

Problem

Solution

Project management

Assessment

Perspectives

How the work was organised

Phase 1

Immersion & literature

Phase 3

Evaluation campaign

Phase 5

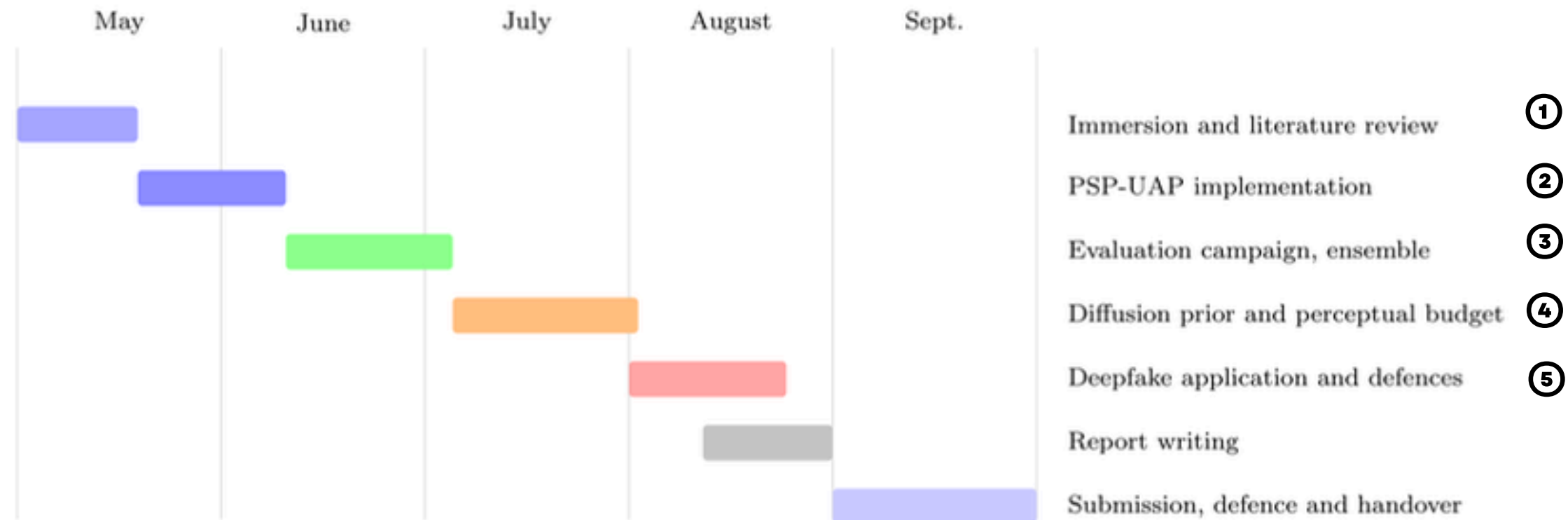
Deepfake application & defences

Phase 2

PSP-UAP implementation

Phase 4

Diffusion prior & perceptual budget



① Objective

fixing the objective for the week

② Implementation

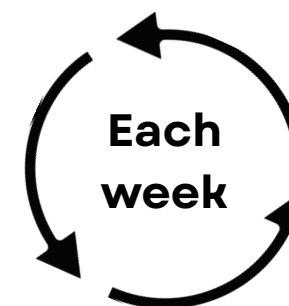
implementing the idea

③ Measurement

testing and measuring

④ Decision

taking the decision



iterative cycles · weekly · aligned with the supervision meetings



Tasks and deviations

→ **The subject set five objectives**

Compare existing UAP methods, apply them to deepfake detectors, evaluate fooling rate, transferability and perceptibility, extend to video, propose defences.

Earlier work in the team (2025) confirmed these detectors are vulnerable – but in white box, on a single detector, using real images.

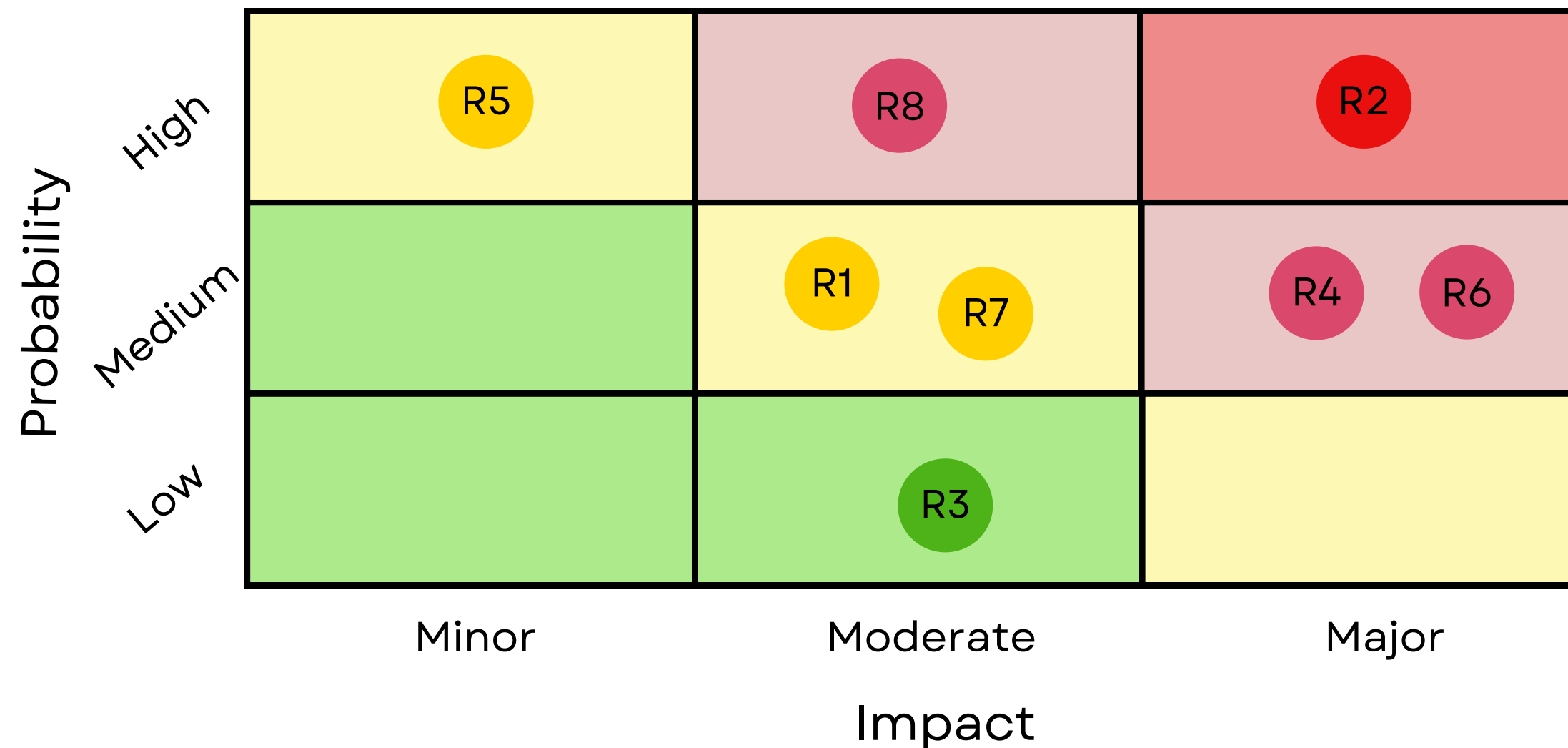
→ **Countermeasures – added to the plan.** Objective 5 of the subject; I scheduled it once the attack worked.

→ **Transformers – not retained.** The perturbation did not transfer to ViT.

→ **MOS – not run.** A human panel rating images from 1 to 5. No time.

Task	Status
Literature review	Done
PSP-UAP reproduction	Done
Evaluation campaign, surrogate ensemble	Done
Diffusion prior	Done
Perceptual budget	Done
Deepfake application	Done
Video extension	Done
Countermeasures	Added
Attack on transformer architectures (ViT)	Not retained
MOS subjective evaluation (ITU-T P.910)	Not done

Risk matrix



R2 – No server access during the August closure
 → Remote access set up before the lab closed.

R4 – The reference method does not reproduce
 → Reproduced and validated first, before any change.

R8 – The attack does not work on a binary detector
 → Fooling term added. Solved.

R1 FaceForensics++ access

R5 Long run interrupted

R7 Report not validated in time

R3 Server outage

R6 Deepfake application not reached in time

Context

Problem

Solution

Project management

Assessment

Perspectives

Final assessment

7/7

Video sequences flipped
to "real"

99.3 %

Evasion on a real
deepfake detector

84.38 %

Overall fooling rate on 8
architectures

91.00 %

White-box fooling rate,
at SSIM 0.80

A data-free universal attack that is both stronger and less visible — and three countermeasures tested against it.

Perspectives



1. Subjective evaluation (MOS)

Run the ITU-T P.910 protocol: a human panel rates the attacked images from 1 to 5. SSIM is only an approximation of what the eye sees. Real viewers would confirm whether the perturbation is invisible.



2. Transformer architectures

The perturbation was built on CNNs and did not transfer to ViT. Attacking transformers is the next step.



3. Better use of the diffusion prior

I generated 64 priors, with no text prompt and few denoising steps. Changing the prompt, the number of steps, or how many priors are generated was never tested.

Conclusion

- Universal attacks no longer need the detector's data. The realistic attack scenario is now the one that works.
- Adding the perceptual budget made the attack both stronger and less visible, where the literature expected a trade-off.
- Detection is the only defence that held. The direction is to detect the attack, not to harden the detector.

THANK YOU

FOR YOUR ATTENTION

Questions welcome
