ECML/PKDD 2020

Attacking Optical Character Recognition (OCR) Systems with Adversarial Watermarks

Lu Chen[1], Jiao Sun[2], Wei Xu[1]

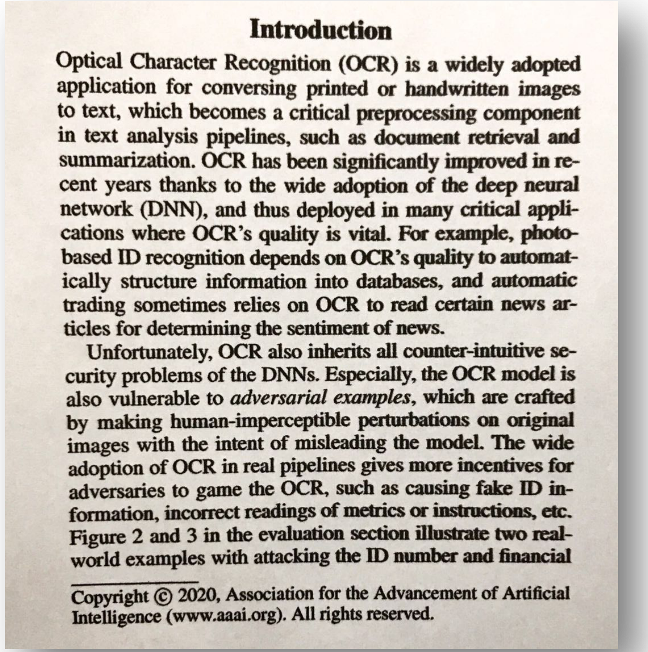
[1]IIIS, Tsinghua University

[2]CS, University of Southern California

清華大學
Tsinghua University

交叉信息研究院
Institute for Interdisciplinary Information Sciences

Optical Character Recognition



Introduction

Optical Character Recognition (OCR) is a widely adopted application for conversing printed or handwritten images to text, which becomes a critical preprocessing component in text analysis pipelines, such as document retrieval and summarization. OCR has been significantly improved in recent years thanks to the wide adoption of the deep neural network (DNN), and thus deployed in many critical applications where OCR's quality is vital. For example, photobased ID recognition depends on OCR's quality to automatically structure information into databases, and automatic trading sometimes relies on OCR to read certain news articles for determining the sentiment of news.

Unfortunately, OCR also inherits all counter-intuitive security problems of the DNNs. Especially, the OCR model is also vulnerable to *adversarial examples*, which are crafted by making human-imperceptible perturbations on original images with the intent of misleading the model. The wide adoption of OCR in real pipelines gives more incentives for adversaries to game the OCR, such as causing fake ID information, incorrect readings of metrics or instructions, etc. Figure 2 and 3 in the evaluation section illustrate two realworld examples with attacking the ID number and financial

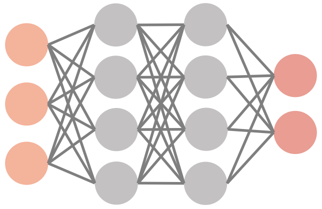
PDF FILE

IMAGE FILE

SCANNED DOCUMENT

OCR

Deep Neural Network



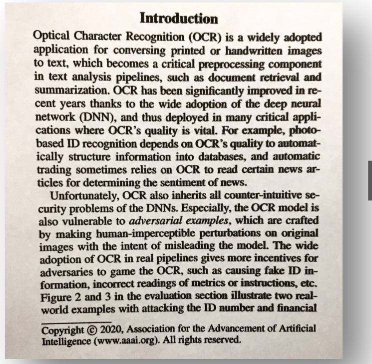
Introduction

Optical Character Recognition (OCR) is a widely adopted application for conversing printed or handwritten images to text, which becomes a critical preprocessing component in text analysis pipelines, such as document retrieval and summarization. OCR has been significantly improved in recent years thanks to the wide adoption of the deep neural network (DNN), and thus deployed in many critical applications where OCR's quality is vital. For example, photobased ID recognition depends on OCR's quality to automatically structure information into databases, and automatic trading sometimes relies on OCR to read certain news articles for determining the sentiment of news.

Unfortunately, OCR also inherits all counter-intuitive security problems of the DNNs. Especially, the OCR model is also vulnerable to adversarial examples, which are crafted by making human-imperceptible perturbations on original images with the intent of misleading the model. The wide adoption of OCR in real pipelines gives more incentives for adversaries to game the OCR, such as causing fake ID information, incorrect readings of metrics or instructions, etc. Figure 2 and 3 in the evaluation section illustrate two realworld examples with attacking the ID number and financial

SEARCHABLE TEXT DOCUMENT

Optical Character Recognition



Introduction

Optical Character Recognition (OCR) is a widely adopted application for conversing printed or handwritten images to text, which becomes a critical preprocessing component in text analysis pipelines, such as document retrieval and summarization. OCR has been significantly improved in recent years thanks to the wide adoption of the deep neural network (DNN), and thus deployed in many critical applications where OCR's quality is vital. For example, photo-based ID recognition depends on OCR's quality to automatically structure information into databases, and automatic trading sometimes relies on OCR to read certain news articles for determining the sentiment of news.

Unfortunately, OCR also inherits all counter-intuitive security problems of the DNNs. Especially, the OCR model is also vulnerable to *adversarial examples*, which are crafted by making human-imperceptible perturbations on original images with the intent of misleading the model. The wide adoption of OCR in real pipelines gives more incentives for adversaries to game the OCR, such as causing fake ID information, incorrect readings of metrics or instructions, etc. Figure 2 and 3 in the evaluation section illustrate two real-world examples with attacking the ID number and financial

Introduction

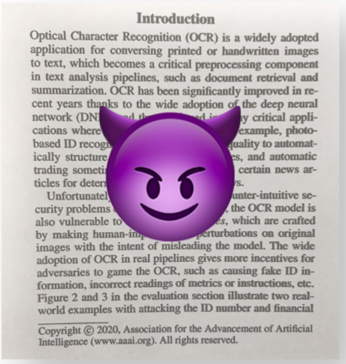
Optical Character Recognition (OCR) is a widely adopted application for conversing printed or handwritten images to text, which becomes a critical preprocessing component in text analysis pipelines, such as document retrieval and summarization. OCR has been significantly improved in recent years thanks to the wide adoption of the deep neural network (DNN), and thus deployed in many critical applications where OCR's quality is vital. For example, photobased ID recognition depends on OCR's quality to automatically structure information into databases, and automatic trading sometimes relies on OCR to read certain news articles for determining the sentiment of news.

Unfortunately, OCR also inherits all counter-intuitive security problems of the DNNs. Especially, the OCR model is also vulnerable to adversarial examples, which are crafted by making human-imperceptible perturbations on original images with the intent of misleading the model. The wide adoption of OCR in real pipelines gives more incentives for adversaries to game the OCR, such as causing fake ID information, incorrect readings of metrics or instructions, etc. Figure 2 and 3 in the evaluation section illustrate two realworld examples with attacking the ID number and financial

Copyright c 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

APPLICATION

Optical Character Recognition



Introduction

Optical Character Recognition (OCR) is a widely adopted application for conversing printed or handwritten images to text, which becomes a critical preprocessing component in text analysis pipelines, such as document retrieval and summarization. OCR has been significantly improved in recent years thanks to the wide adoption of the deep neural network (DN[illegible] y critical applications where [illegible] example, photo-based ID recog[illegible] quality to automatically structure [illegible] es, and automatic trading someti[illegible] certain news articles for deter[illegible] s.

Unfortunatel[illegible] unter-intuitive security problems [illegible] the OCR model is also vulnerable to [illegible] s, which are crafted by making human-im[illegible] perturbations on original images with the intent of misleading the model. The wide adoption of OCR in real pipelines gives more incentives for adversaries to game the OCR, such as causing fake ID information, incorrect readings of metrics or instructions, etc. Figure 2 and 3 in the evaluation section illustrate two real-world examples with attacking the ID number and financial

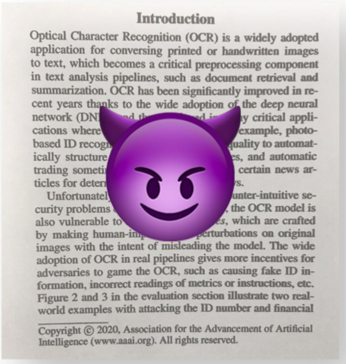
Introduction

Optical Character Recognition (OCR) is a widely adopted application for conversing printed or handwritten images to text, which becomes a critical preprocessing component in text analysis pipelines, such as document retrieval and summarization. OCR has been significantly improved in recent years thanks to the wide adoption of the deep neural network (DNN), and thus deployed in many critical applications where OCR's quality is vital. For example, photobased ID recognition depends on OCR's quality to automatically structure information into databases, and automatic trading sometimes relies on OCR to read certain news articles for determining the sentiment of news.

Unfortunately, OCR also inherits all counter-intuitive security problems of the DNNs. Especially, the OCR model is also vulnerable to adversarial examples, which are crafted by making human-imperceptible perturbations on original images with the intent of misleading the model. The wide adoption of OCR in real pipelines gives more incentives for adversaries to game the OCR, such as causing fake ID information, incorrect readings of metrics or instructions, etc. Figure 2 and 3 in the evaluation section illustrate two realworld examples with attacking the ID number and financial

APPLICATION

Optical Character Recognition



Introduction

Optical Character Recognition (OCR) is a widely adopted application for conversing printed or handwritten images to text, which becomes a critical preprocessing component in text analysis pipelines, such as document retrieval and summarization. OCR has been significantly improved in recent years thanks to the wide adoption of the deep neural network (DN … y critical applications where … example, photobased ID recog… quality to automatically structure … es, and automatic trading someti… certain news articles for deter… s.

Unfortunatel… unter-intuitive security problems … the OCR model is also vulnerable to … es, which are crafted by making human-im… perturbations on original images with the intent of misleading the model. The wide adoption of OCR in real pipelines gives more incentives for adversaries to game the OCR, such as causing fake ID information, incorrect readings of metrics or instructions, etc. Figure 2 and 3 in the evaluation section illustrate two realworld examples with attacking the ID number and financial

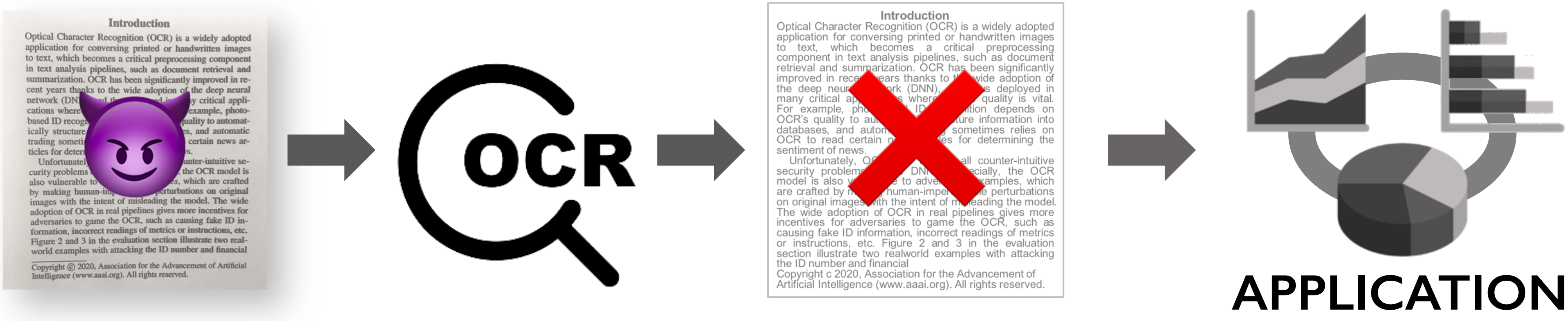
Introduction

Optical Character Recognition (OCR) is a widely adopted application for conversing printed or handwritten images to text, which becomes a critical preprocessing component in text analysis pipelines, such as document retrieval and summarization. OCR has been significantly improved in recent years thanks to the wide adoption of the deep neural network (DNN), and thus deployed in many critical applications where OCR's quality is vital. For example, photobased ID recognition depends on OCR's quality to automatically structure information into databases, and automatic trading sometimes relies on OCR to read certain news articles for determining the sentiment of news.

Unfortunately, OCR also inherits all counter-intuitive security problems of the DNNs. Especially, the OCR model is also vulnerable to adversarial examples, which are crafted by making human-imperceptible perturbations on original images with the intent of misleading the model. The wide adoption of OCR in real pipelines gives more incentives for adversaries to game the OCR, such as causing fake ID information, incorrect readings of metrics or instructions, etc. Figure 2 and 3 in the evaluation section illustrate two realworld examples with attacking the ID number and financial

APPLICATION

Optical Character Recognition



Traditional Attacks

[Goodfellow, 2015]

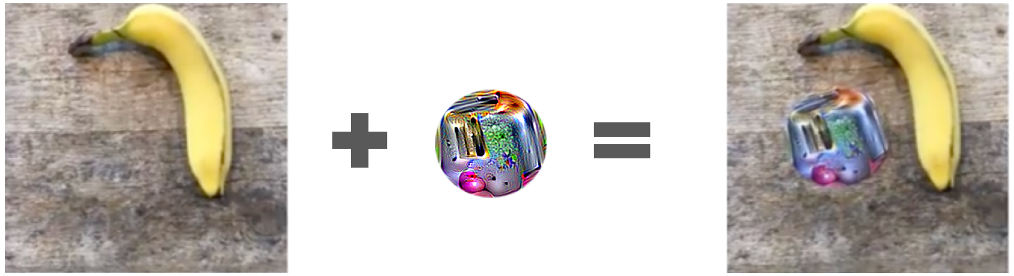
Original image
"Panda" 57.7%

noise
"nematode" 8.2%

Adverarial example
"gibbon" 99.3%

Visually imperceptible

[Brown, 2017]



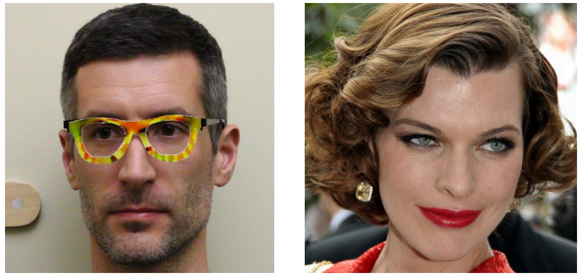
Original image
"Banana" 97%

Patch

Adversarial example
"toaster" 99%

Adversarial patch

[Sharif, 2016]



[Thys, 2019]

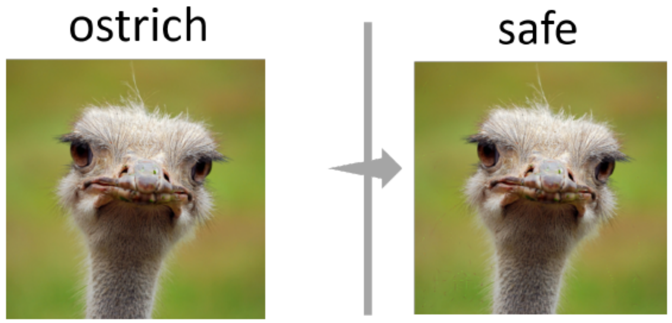
[Eykholt, 2018]

Adversarial attacks in the real world

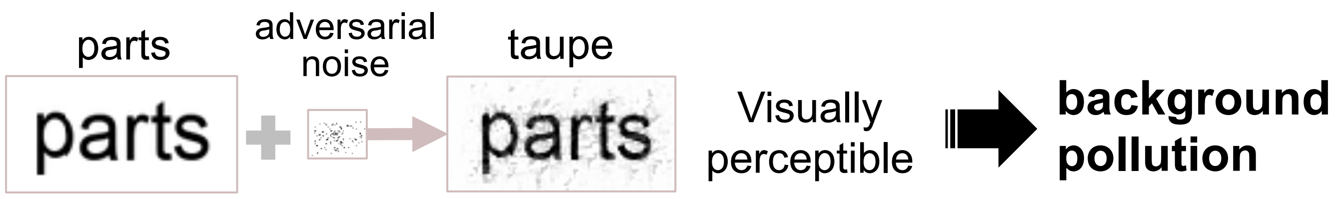
Traditional Attack vs. OCR Attack:
1. Image Backgrounds

- **colorful** background vs. **white** background

[Chen, 2018]



Colorful Background
(Photos)

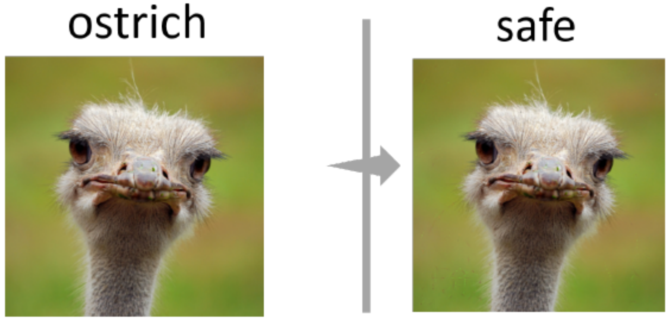


White Background
(Printed-text Images)

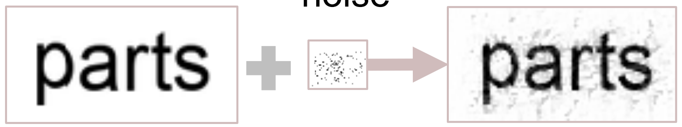
Traditional Attack vs. OCR Attack:
1. Image Backgrounds

- **colorful** background vs. **white** background

[Chen, 2018]



Colorful Background
(Photos)



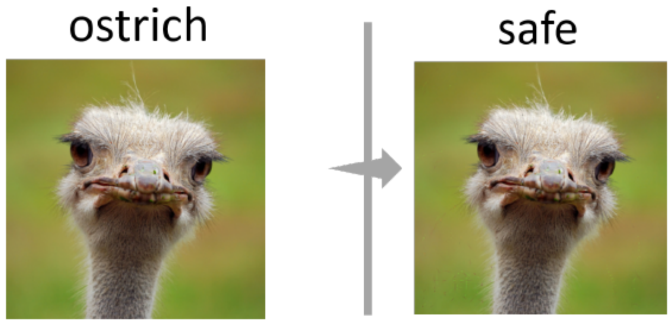
Visually perceptible → **background pollution**

White Background
(Printed-text Images)

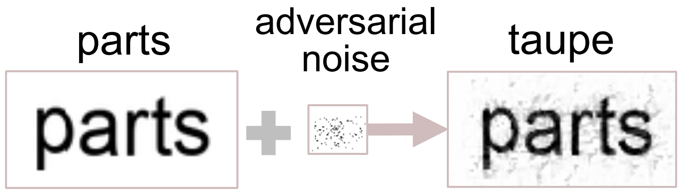
Traditional Attack vs. OCR Attack:
1. Image Backgrounds

- **colorful** background vs. **white** background

[Chen, 2018]



Colorful Background
(Photos)



Visually perceptible ➡ **background pollution**

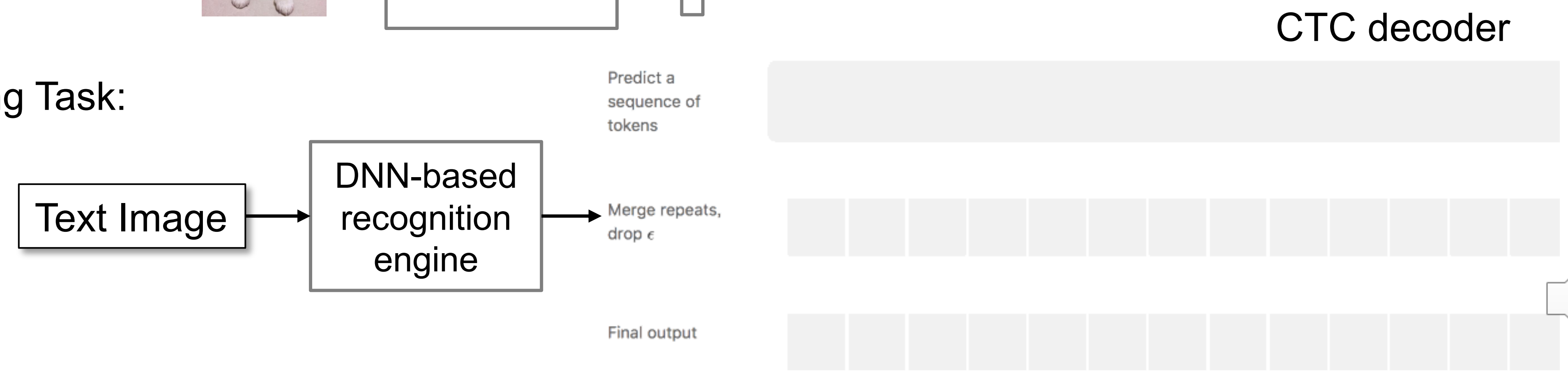
White Background
(Printed-text Images)

Traditional Attack vs. OCR Attack:
2. Model Tasks

- Traditional models ---- **image classification task** ➡️ cross entropy loss
- DNN-based OCR models ---- **sequential labeling task** ➡️ CTC loss

Image Classification Task:

Sequential Labeling Task:



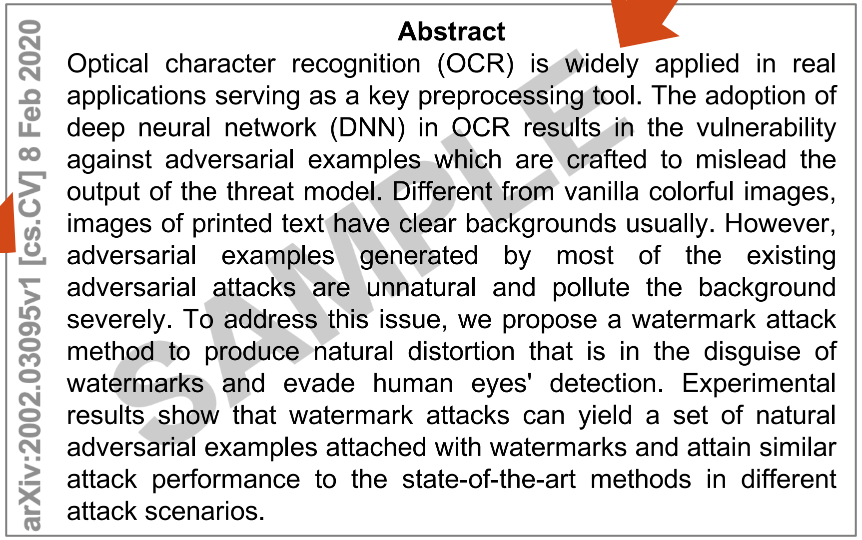
from: https://distill.pub/2017/ctc/

✅ background pollution
✅ sequential labeling task

FAWA: Fast Adversarial Watermark Attack

💡 Basic idea:

Making use of the popularity of watermark (**WM**) in the documents, we hide noise in watermarks.

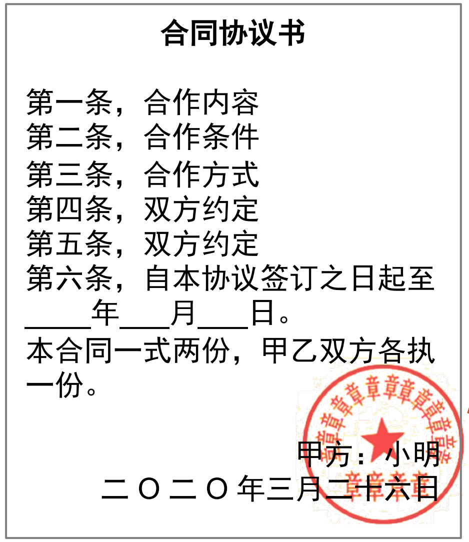


arXiv:2002.03095v1 [cs.CV] 8 Feb 2020

Abstract

Optical character recognition (OCR) is widely applied in real applications serving as a key preprocessing tool. The adoption of deep neural network (DNN) in OCR results in the vulnerability against adversarial examples which are crafted to mislead the output of the threat model. Different from vanilla colorful images, images of printed text have clear backgrounds usually. However, adversarial examples generated by most of the existing adversarial attacks are unnatural and pollute the background severely. To address this issue, we propose a watermark attack method to produce natural distortion that is in the disguise of watermarks and evade human eyes' detection. Experimental results show that watermark attacks can yield a set of natural adversarial examples attached with watermarks and attain similar attack performance to the state-of-the-art methods in different attack scenarios.

SAMPLE



合同协议书

第一条，合作内容
第二条，合作条件
第三条，合作方式
第四条，双方约定
第五条，双方约定
第六条，自本协议签订之日起至____年___月___日。
本合同一式两份，甲乙双方各执一份。

甲方：小明
二〇二〇年三月二十六日

background pollution
sequential labeling task

FAWA: Fast Adversarial Watermark Attack

Basic idea:

Making use of the popularity of watermark (**WM**) in the documents, we hide noise in watermarks.

background pollution
sequential labeling task

FAWA

1. **Natural**

targeted text: ***taupe***

traditional attack

FAWA

2. **Fast** **100% attack succes rate**

78% fewer iterations

3. **Low Perturbation Level**

60% less noise

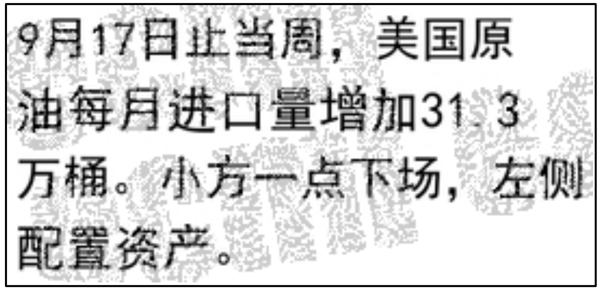
background pollution
sequential labeling task

FAWA

1. Colored Watermark

targeted text: ***<u>randem</u>***

2. OCR Model of Other Language


9月17日止当周，美国原油每月进口量增加31.3万桶。小方一点下场，左侧配置资产。

Target Output :
(Chinese) 9 月 <u>**12**</u> 日<u>上当</u>周，美国原油每<u>且</u>进口量增加 31.3 万桶。<u>孙</u>方<u>二</u>点<u>不</u>下场，<u>右</u>侧配置资产。

3. OCR accuracy-enhancing mechanism

15

FAWA

FAWA:
Attack Settings

- **White-box Model**
  - Attackers have perfect knowledge of the DNN architecture and parameters.

- **Targeted Attack**
  - Attackers aim to generate specific recognition texts.

harmless image

adversarial image

intermediate status

Attack Pipeline

clean image x

$f(x)$: **random**

adversarial image x'

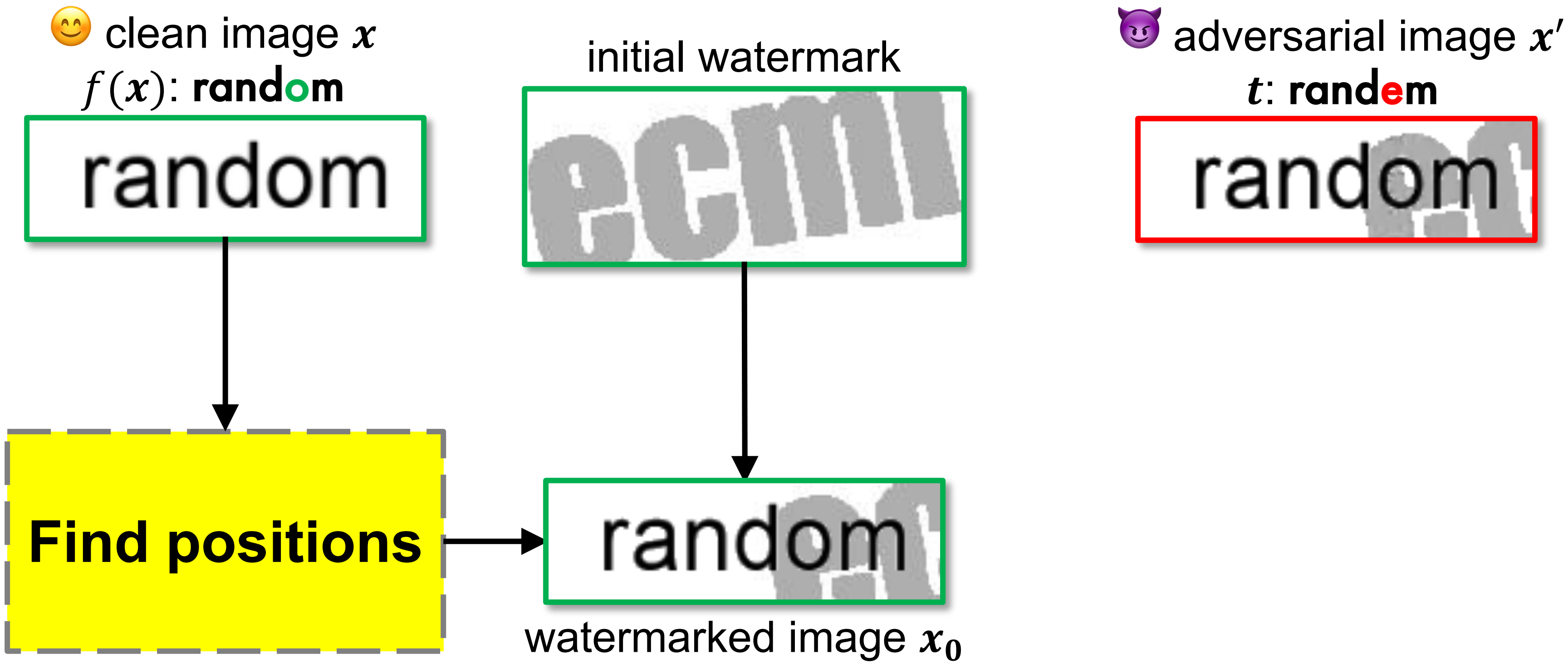
t: **randem**

Attack Pipeline:
1. Find Positions

harmless image

adversarial image

intermediate status

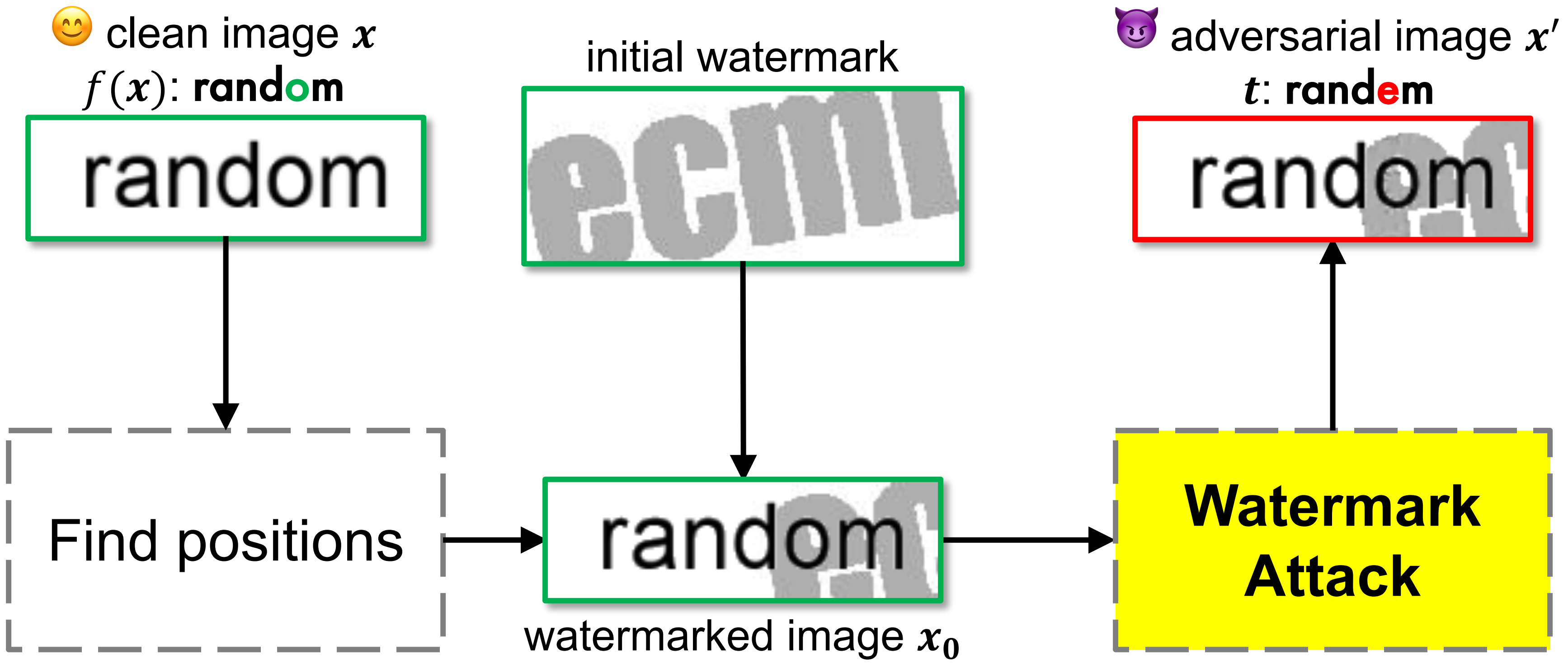


Attack Pipeline:
2. Watermark Attack

harmless image

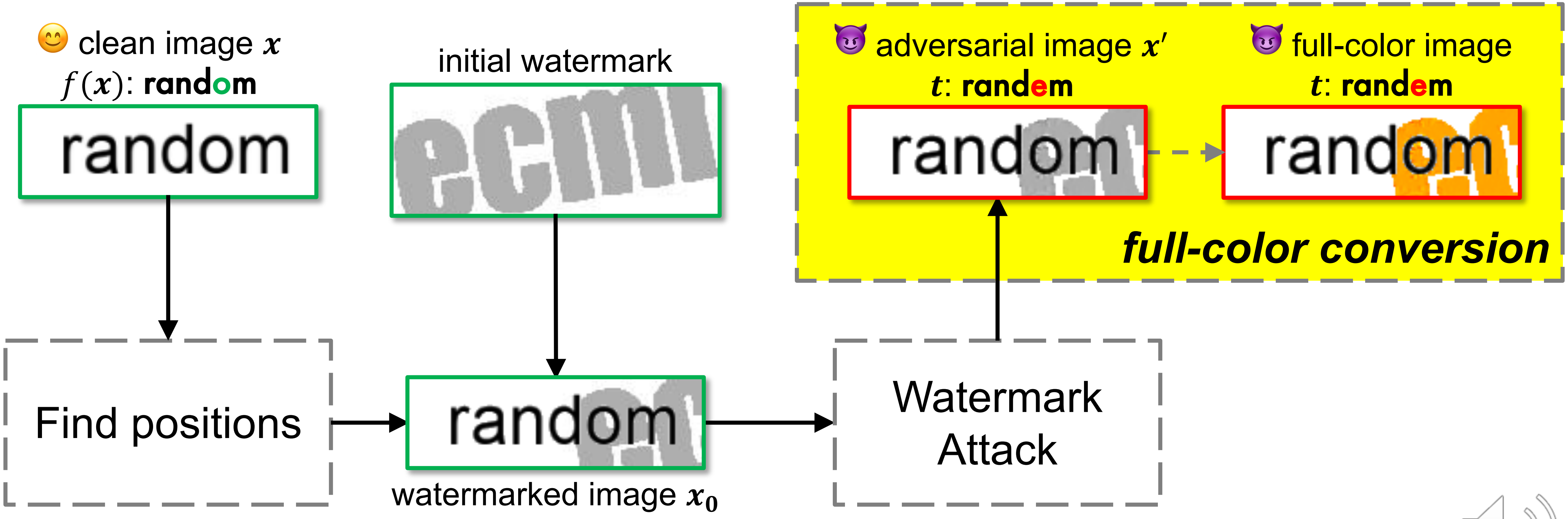
adversarial image

intermediate status

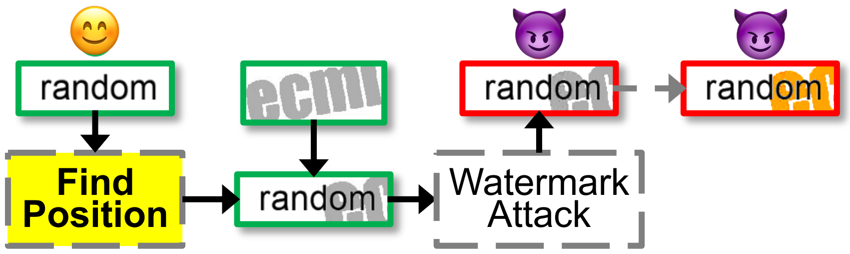


Attack Pipeline:
3*. Full-Color Conversion

harmless image
adversarial image
intermediate status



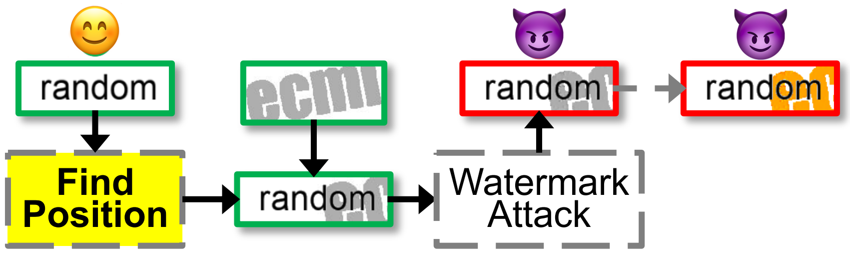
Critical Preprocess:
Find Positions

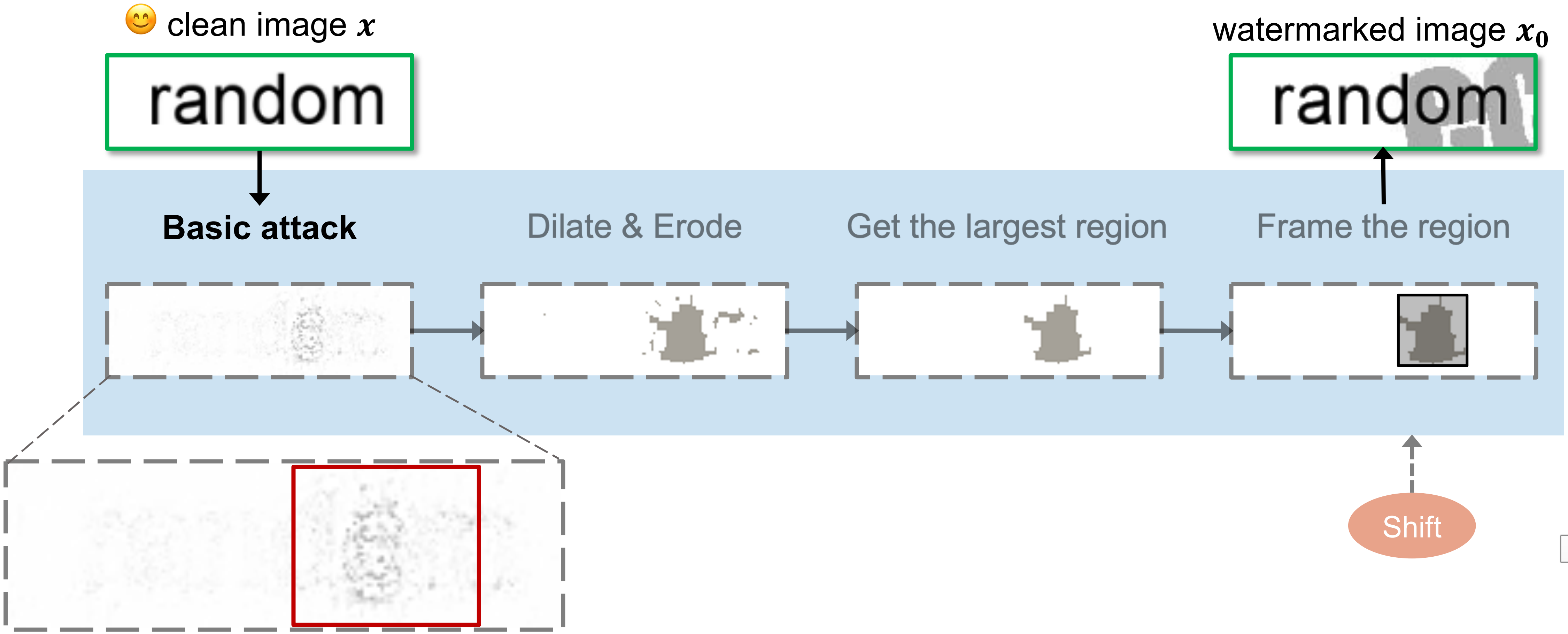


- **Basic Attack**
  - We find positions based on the noise of the basic attack.
  - We use *Momentum Iterative Method (MIM)* as the basic attack to find positions.

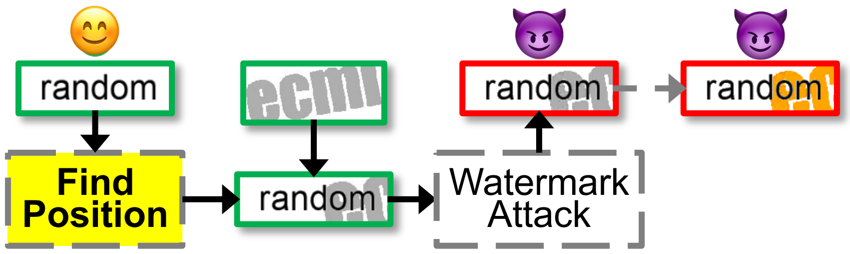
[Dong, Yinpeng et al. “Boosting Adversarial Attacks with Momentum.” *2018* CVPR]

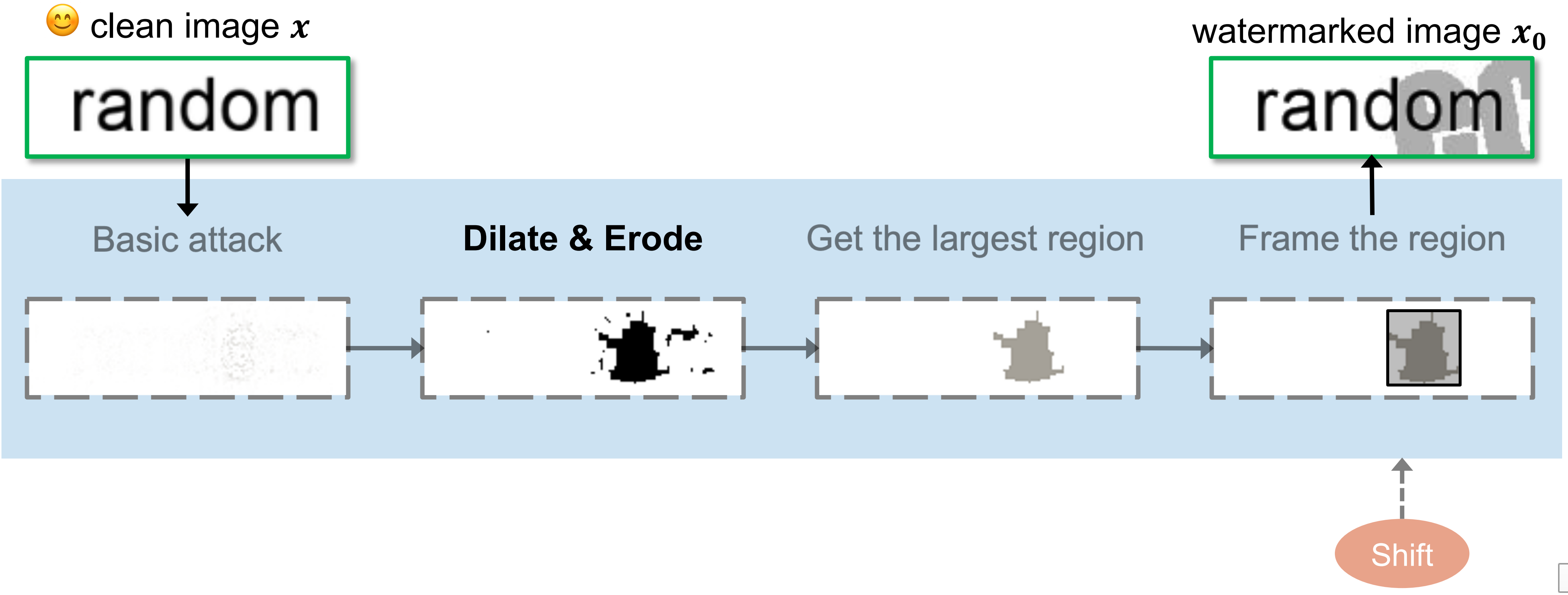
Find Positions:
1. Basic Attack



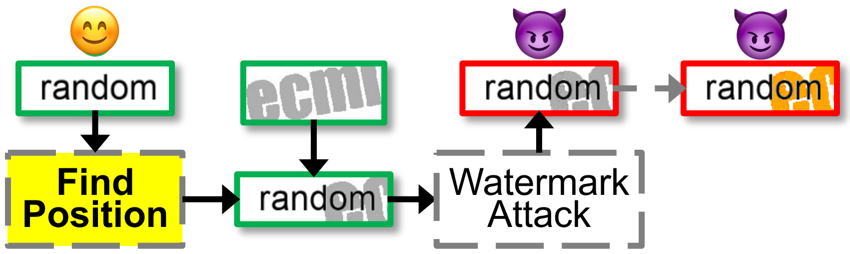


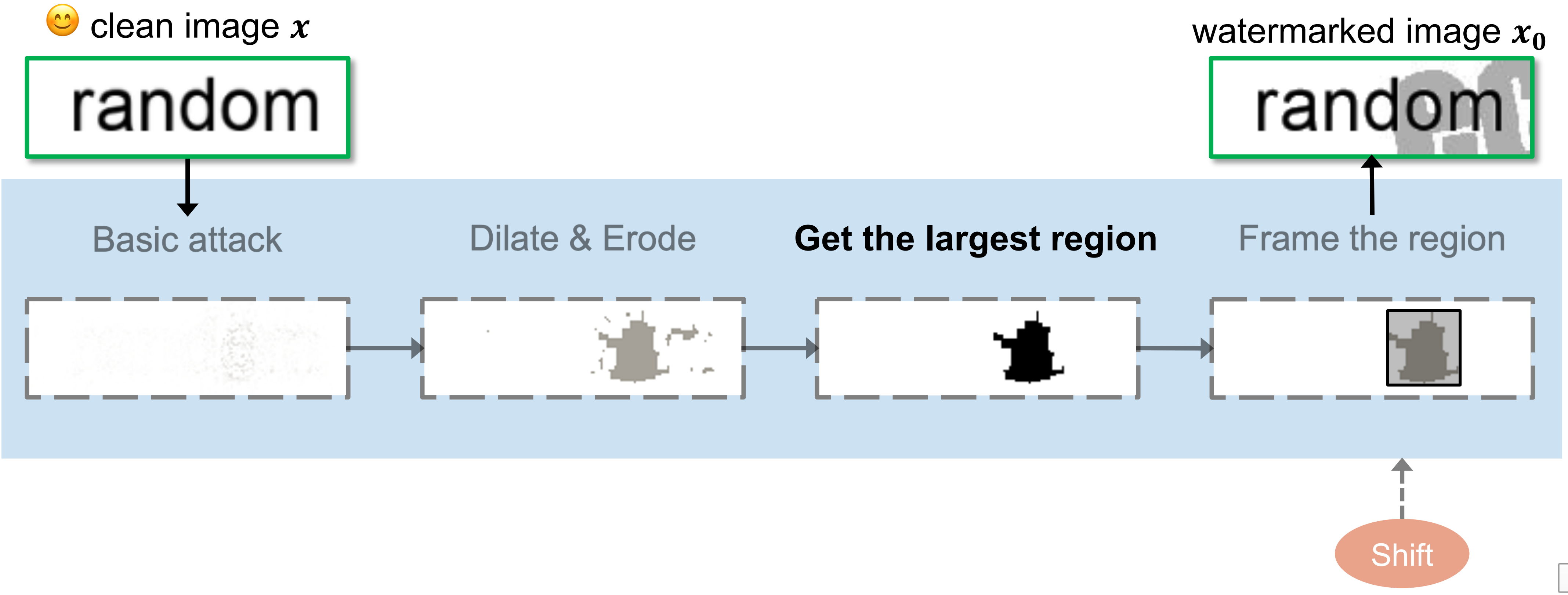
Find Positions:
2. Dilate & Erode



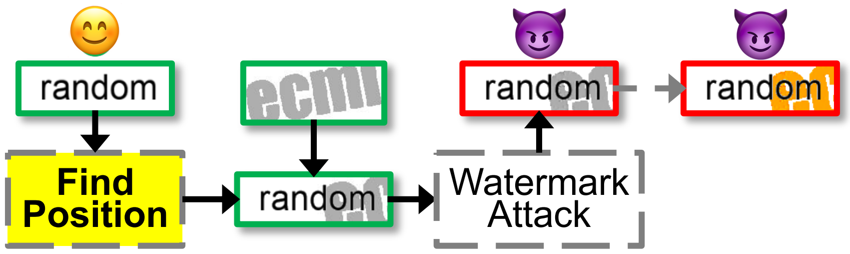


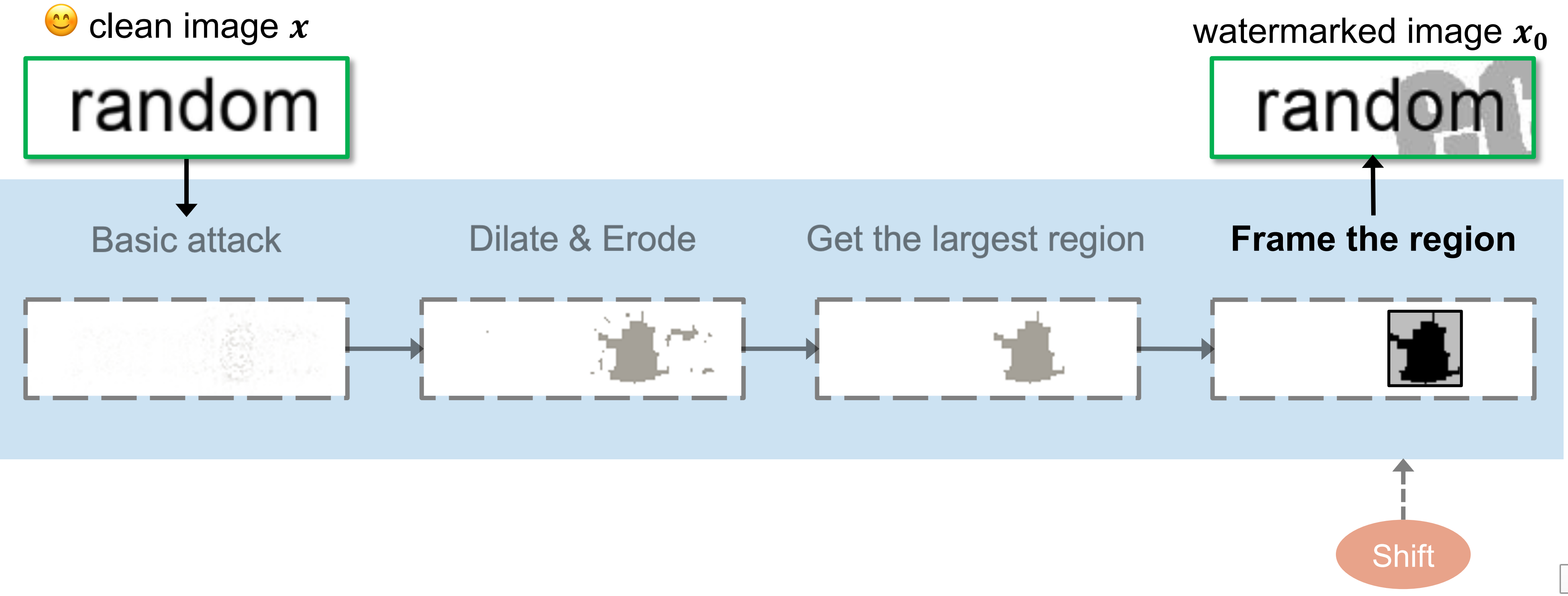
Find Positions:
3. Get the largest region



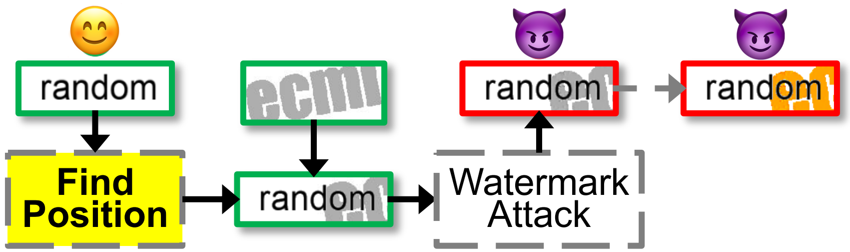


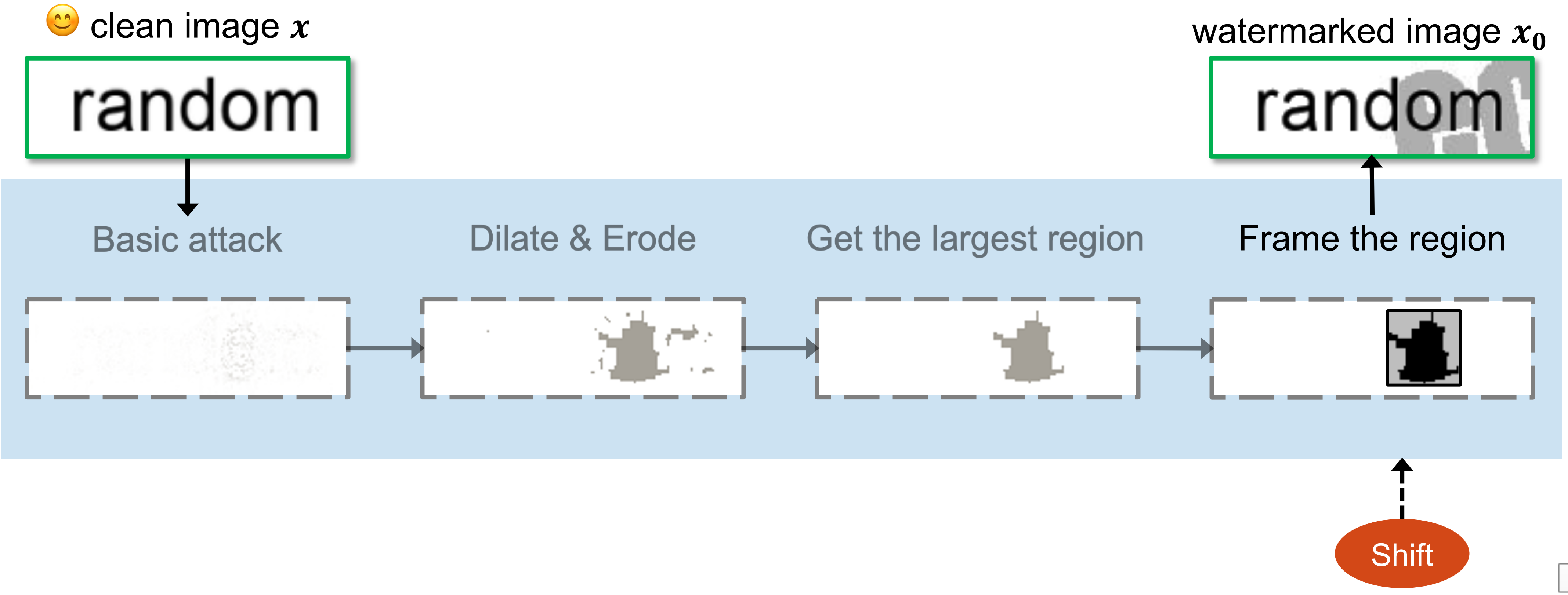
Find Positions:
4. Frame the region

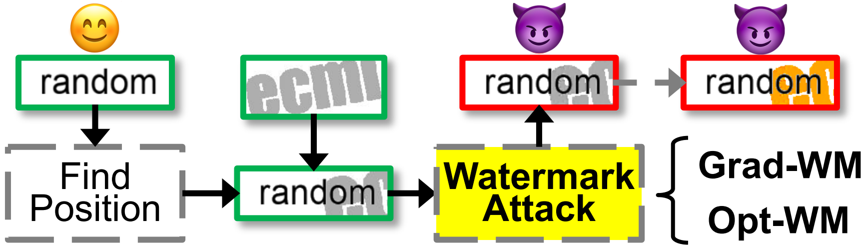




Find Positions:
4*. Add a Shift







Watermark Attack:
WM Attack = Basic Attack + WM

- **Basic Attacks: (traditional attacks)**
  - **Grad-Basic**: Gradient-based Basic Attack
    - Momentum Iterative Method (MIM) [Dong et al. 2018]
  - **Opt-Basic**: Optimization-based Basic Attack
    - OCR Attack [Song et al. 2018]
- **WM**: Watermark region allowed to add noise
- **Watermark Attacks: (our attacks)**
  - **Grad-WM**: Gradient-based Watermark Attack
    - **Grad-WM = Grad-Basic + WM**
  - **Opt-WM**: Optimization-based Watermark Attack
    - **Opt-WM = Opt-Basic + WM**

- cross entropy ℓ
- ϵ-bounded noise
- step size α
- decay factor μ

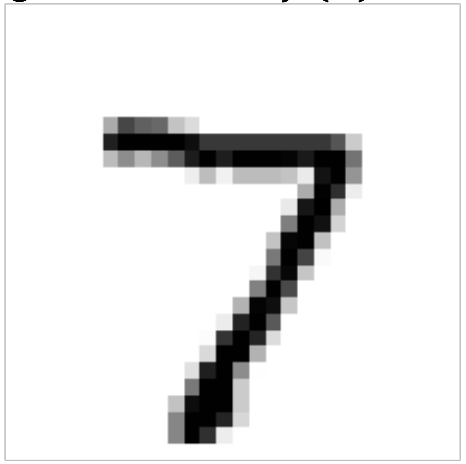
Gradient-based Basic Attack:

Grad-Basic

❌ **background pollution**

❌ **sequential labeling task**

ground-truth $f(\boldsymbol{x})$: **7**



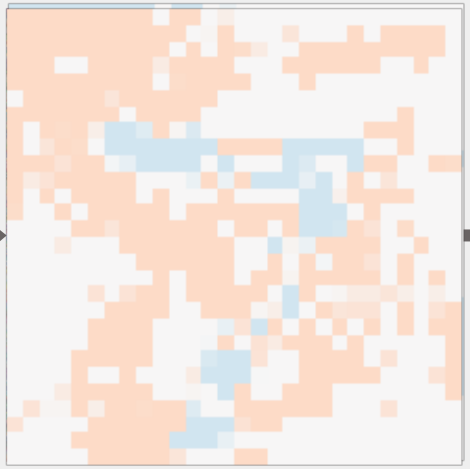
clean image $\boldsymbol{x}$

initialization: $\boldsymbol{x}_0' = \boldsymbol{x}; \boldsymbol{g}_0 = \boldsymbol{0};$

iteration i

model f

x_i'



saliency map $\nabla_x \ell(\boldsymbol{x}_i', \boldsymbol{t})$

$$\boldsymbol{g}_{i+1} = \mu \cdot \boldsymbol{g}_i + \frac{\nabla_x \ell(\boldsymbol{x}_i', \boldsymbol{t})}{\|\nabla_x \ell(\boldsymbol{x}_i', \boldsymbol{t})\|_1}$$

$\boldsymbol{g}_{i+1}$

$\boldsymbol{g}_i$

target output $\boldsymbol{t}$: **8**

adversarial image $\boldsymbol{x}_{i+1}'$

$$\boldsymbol{x}_{i+1}' = \boldsymbol{x}_i' + clip_\epsilon\left(\alpha \cdot \frac{\boldsymbol{g}_{i+1}}{\|\boldsymbol{g}_{i+1}\|_p}\right)$$

new iteration

L_p-norm:

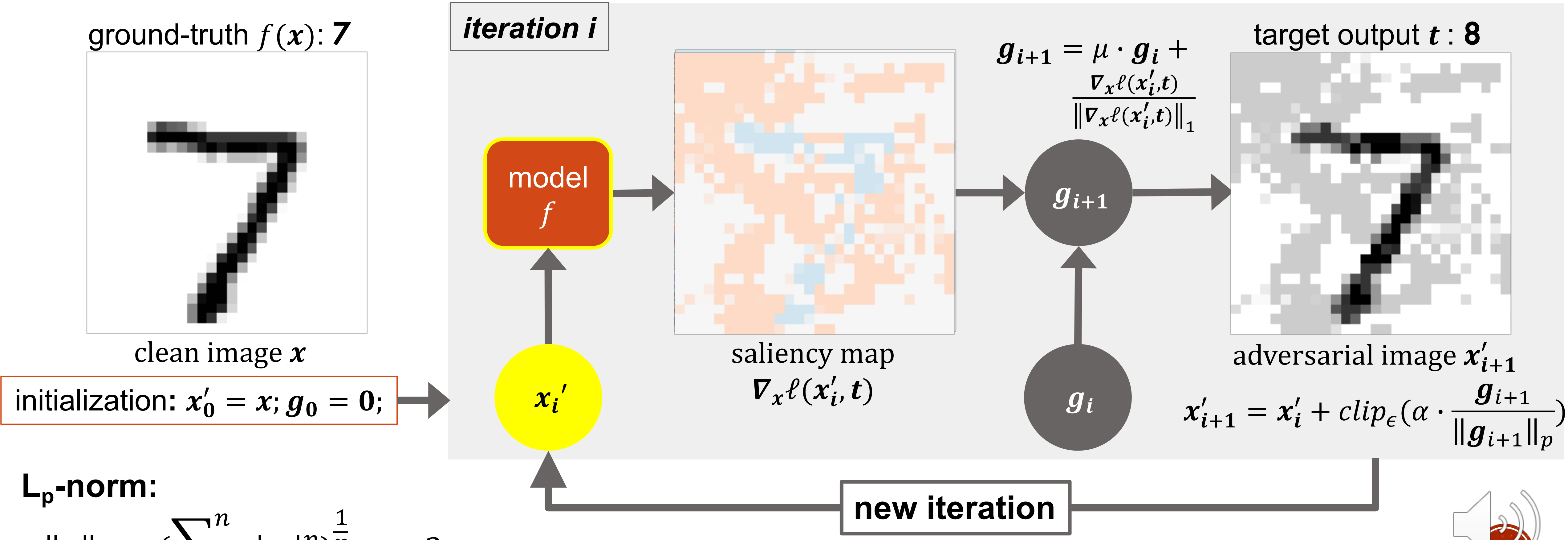
$$\|\boldsymbol{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}, p = 2, \infty$$

[Dong, Yinpeng et al. “Boosting Adversarial Attacks with Momentum.” *2018* CVPR]

- cross entropy ℓ
- ϵ-bounded noise
- step size α
- decay factor μ

Gradient-based Basic Attack:
Grad-Basic

✗ **background pollution**
✗ **sequential labeling task**



L_p-norm:

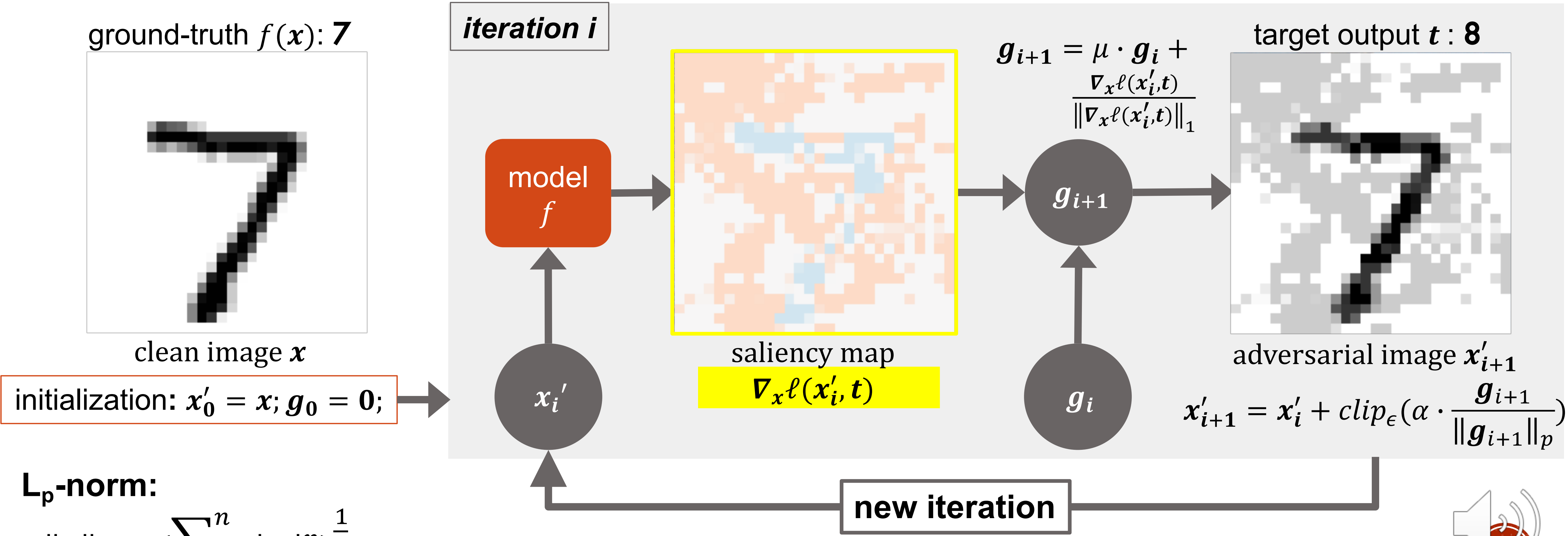
$$\|\boldsymbol{x}\|_p = (\sum_{i=1}^{n} |x_i|^p)^{\frac{1}{p}}, p = 2, \infty$$

[Dong, Yinpeng et al. "Boosting Adversarial Attacks with Momentum." *2018* CVPR]

Gradient-based Basic Attack:
Grad-Basic

background pollution
sequential labeling task

- cross entropy ℓ
- ϵ-bounded noise
- step size α
- decay factor μ



L_p-norm:

$$\|\boldsymbol{x}\|_p = (\sum_{i=1}^{n} |x_i|^p)^{\frac{1}{p}}, p = 2, \infty$$

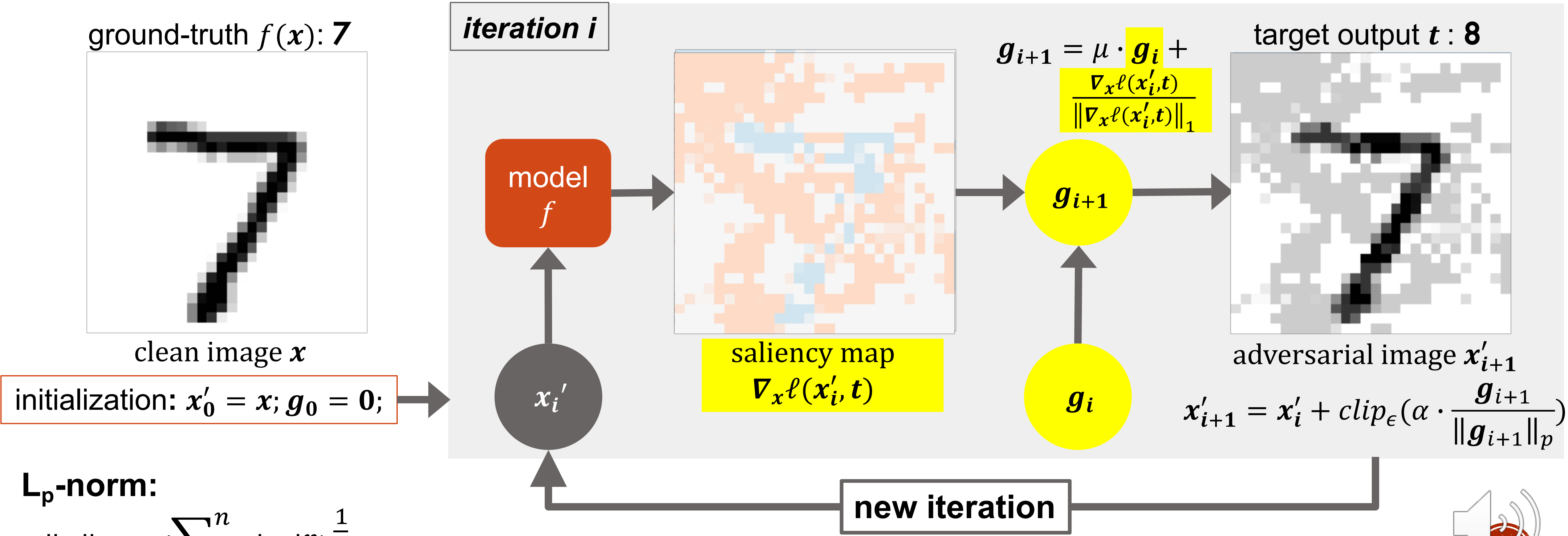
[Dong, Yinpeng et al. "Boosting Adversarial Attacks with Momentum." *2018* CVPR]

Gradient-based Basic Attack:

Grad-Basic

- cross entropy ℓ
- ϵ-bounded noise
- step size α
- decay factor μ

background pollution
sequential labeling task



$\mathrm{L_p}$-norm:

$$\|\boldsymbol{x}\|_p = (\sum_{i=1}^{n} |x_i|^p)^{\frac{1}{p}}, p = 2, \infty$$

[Dong, Yinpeng et al. “Boosting Adversarial Attacks with Momentum.” *2018* CVPR]

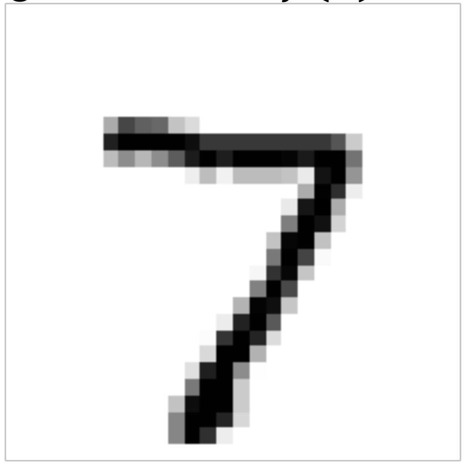
- cross entropy ℓ
- ϵ-bounded noise
- step size α
- decay factor μ

Gradient-based Basic Attack:

Grad-Basic

background pollution
sequential labeling task

ground-truth $f(\boldsymbol{x})$: **7**



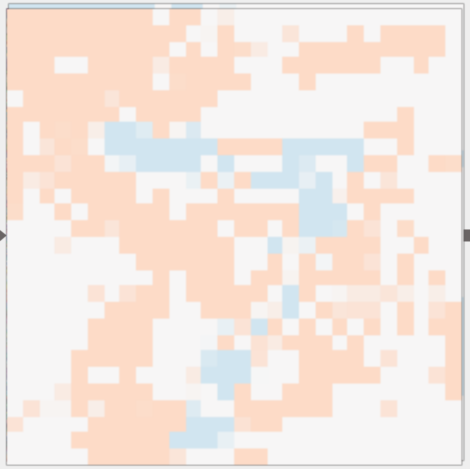
clean image $\boldsymbol{x}$

initialization: $\boldsymbol{x}_0' = \boldsymbol{x}; \boldsymbol{g}_0 = \boldsymbol{0};$

iteration i

model f

$\boldsymbol{x}_i'$



saliency map
$\nabla_x \ell(\boldsymbol{x}_i', \boldsymbol{t})$

$$\boldsymbol{g}_{i+1} = \mu \cdot \boldsymbol{g}_i + \frac{\nabla_x \ell(\boldsymbol{x}_i', \boldsymbol{t})}{\|\nabla_x \ell(\boldsymbol{x}_i', \boldsymbol{t})\|_1}$$

$\boldsymbol{g}_{i+1}$

$\boldsymbol{g}_i$

target output $\boldsymbol{t}$: **8**

adversarial image $\boldsymbol{x}_{i+1}'$

$$\boldsymbol{x}_{i+1}' = \boldsymbol{x}_i' + clip_\epsilon\left(\alpha \cdot \frac{\boldsymbol{g}_{i+1}}{\|\boldsymbol{g}_{i+1}\|_p}\right)$$

new iteration

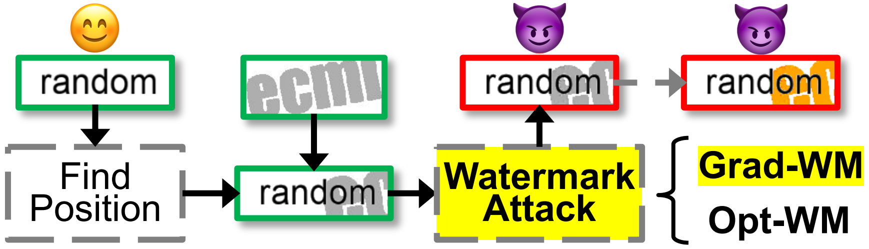
L_p-norm:

$$\|\boldsymbol{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}, p = 2, \infty$$

[Dong, Yinpeng et al. “Boosting Adversarial Attacks with Momentum.” *2018* CVPR]

Watermark Attack:

Grad-WM=Grad-Basic+WM



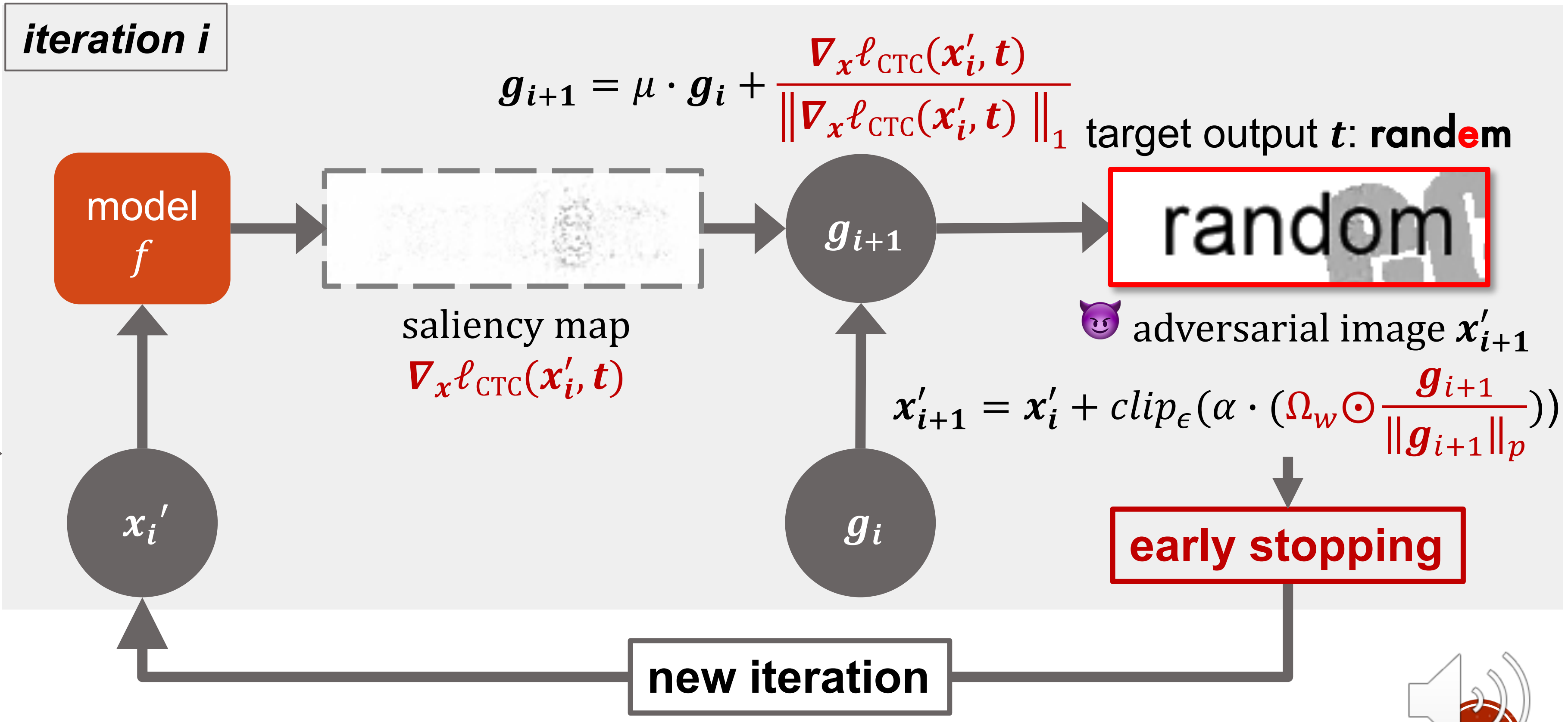
sequential labeling task
background pollution

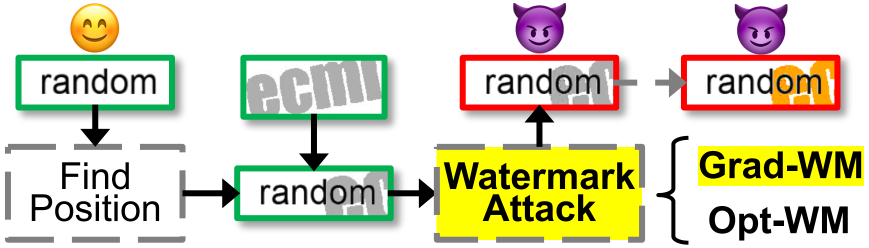
- grayscale β
- watermark mask Ω_w
- text mask Ω_t

ground-truth $f(x)$: **random**

😊 clean image x

initialization: $g_0 = 0$;
$x'_0 = (1 - \Omega_w \odot \overline{\Omega_t}) \odot x + \beta \cdot \Omega_w \odot \overline{\Omega_t}$;





Watermark Attack:

Grad-WM=Grad-Basic+WM

✅ sequential labeling task

✅ background pollution

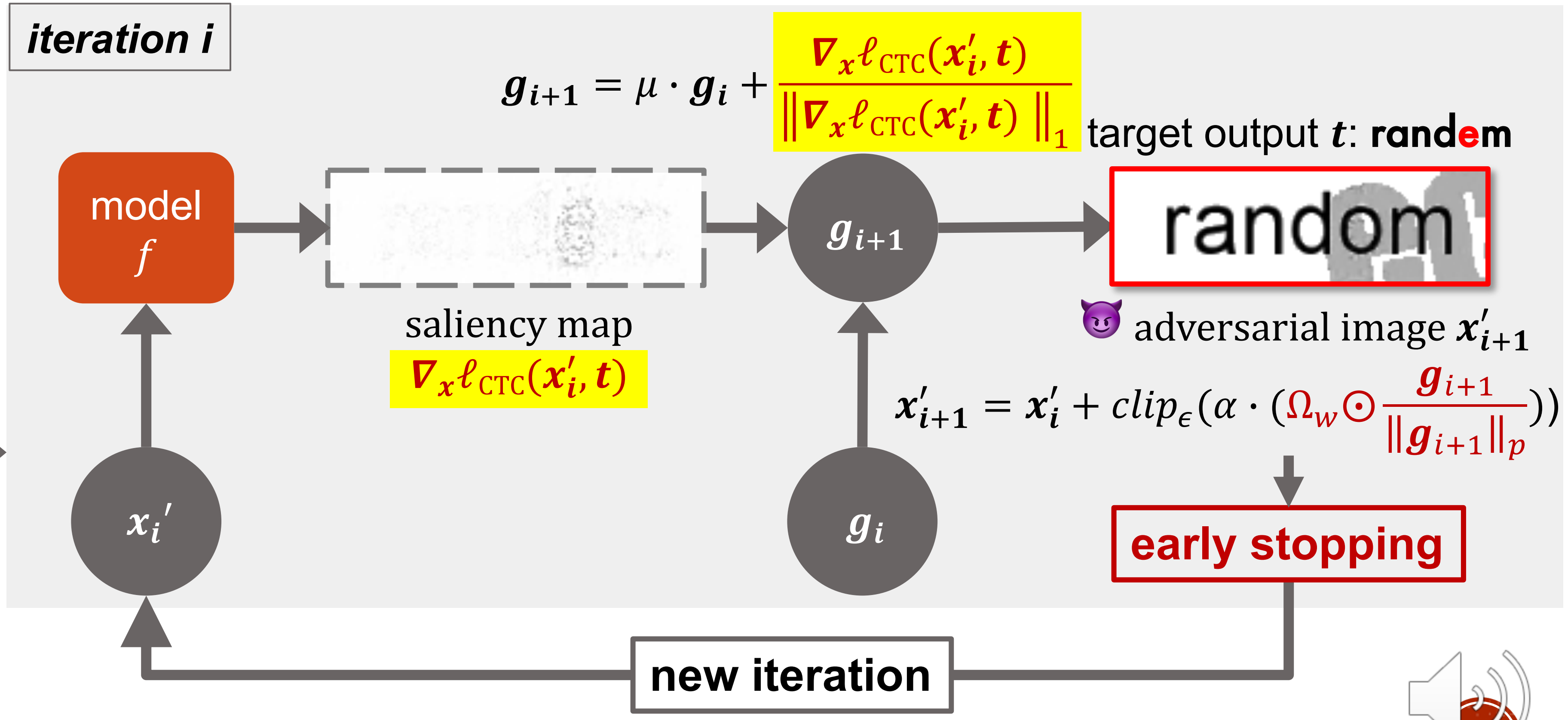
- grayscale β
- watermark mask Ω_w
- text mask Ω_t

ground-truth $f(x)$: **random**

😊 clean image x

initialization: $g_0 = 0$;

$x'_0 = (1 - \Omega_w \odot \overline{\Omega_t}) \odot x + \beta \cdot \Omega_w \odot \overline{\Omega_t}$;



Watermark Attack:

Grad-WM=Grad-Basic+WM

random → Find Position → random → Watermark Attack → random → random

Grad-WM
Opt-WM

sequential labeling task
background pollution

- grayscale β
- watermark mask Ω_w
- text mask Ω_t

ground-truth $f(x)$: **random**

clean image x

initialization: $g_0 = 0$;
$x'_0 = (1 - \Omega_w \odot \overline{\Omega_t}) \odot x + \beta \cdot \Omega_w \odot \overline{\Omega_t}$;

iteration i

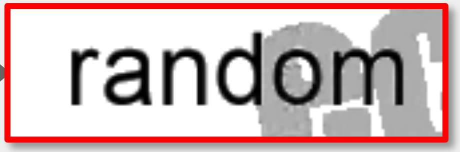
$$g_{i+1} = \mu \cdot g_i + \frac{\nabla_x \ell_{\mathrm{CTC}}(x'_i, t)}{\| \nabla_x \ell_{\mathrm{CTC}}(x'_i, t) \|_1}$$

target output t: **randem**

model f

saliency map
$\nabla_x \ell_{\mathrm{CTC}}(x'_i, t)$

g_{i+1}



adversarial image x'_{i+1}

$$x'_{i+1} = x'_i + clip_\epsilon\left(\alpha \cdot \left(\Omega_w \odot \frac{g_{i+1}}{\| g_{i+1} \|_p}\right)\right)$$

$x_i{}'$

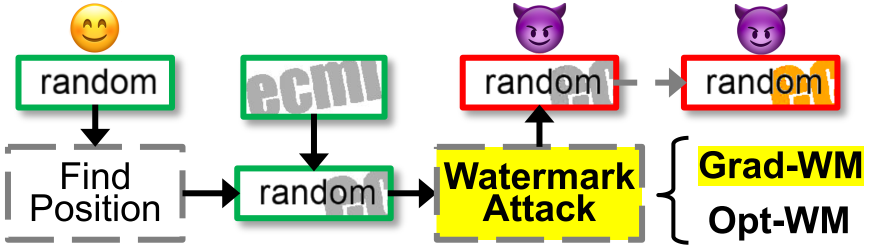
g_i

early stopping

new iteration

Watermark Attack:

Grad-WM=Grad-Basic+WM



sequential labeling task

background pollution

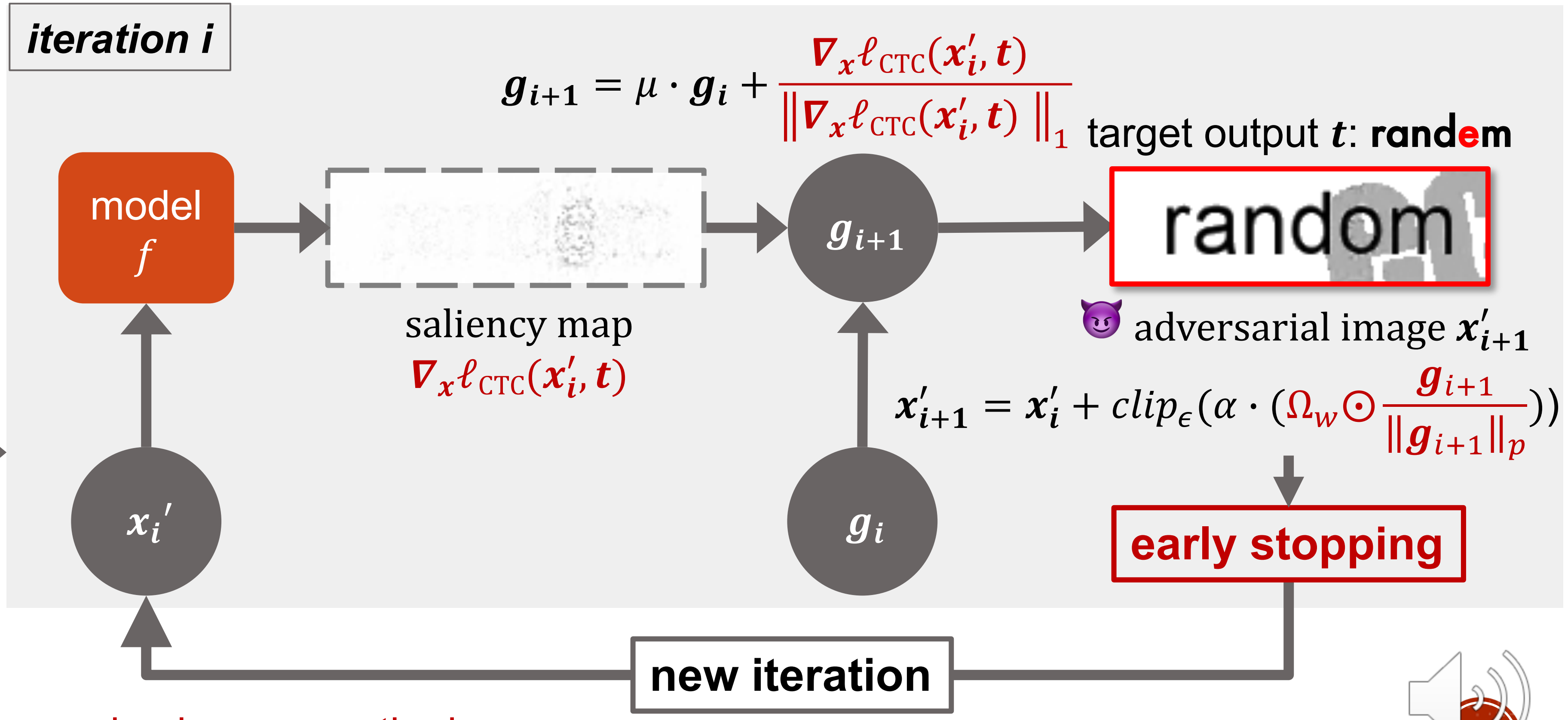
- grayscale β
- watermark mask Ω_w
- text mask Ω_t

ground-truth $f(x)$: **random**

clean image $\boldsymbol{x}$

initialization: $\boldsymbol{g_0} = \boldsymbol{0}$;

$\boldsymbol{x}_0' = (1 - \Omega_w \odot \overline{\Omega_t}) \odot \boldsymbol{x} + \beta \cdot \Omega_w \odot \overline{\Omega_t}$;

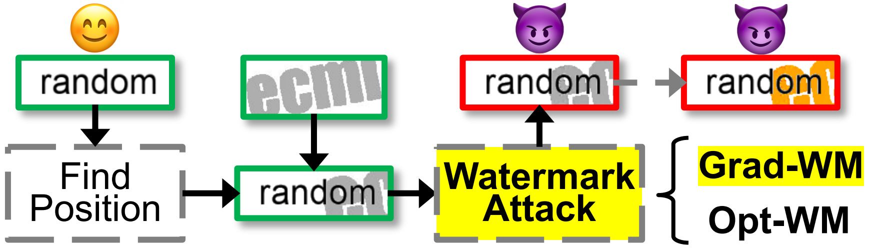


spread noise over entire image

Ω_w-bounded natural noise

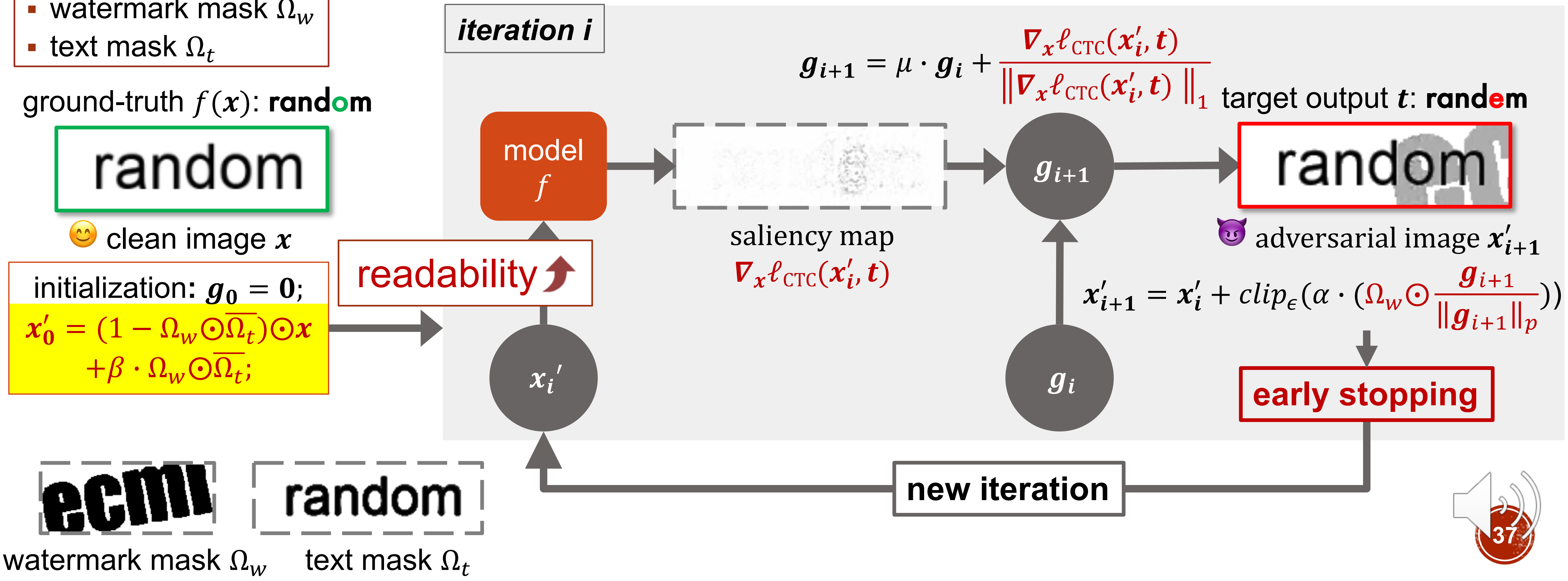
Watermark Attack:

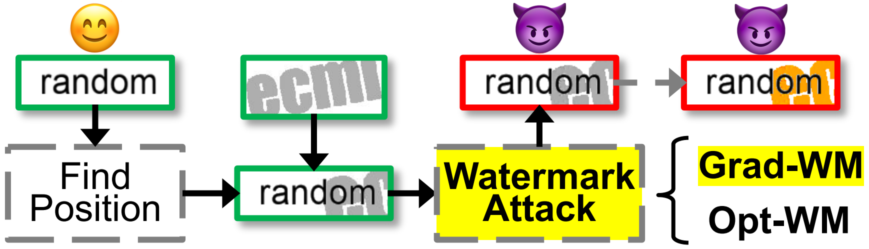
Grad-WM=Grad-Basic+WM



sequential labeling task
background pollution

- grayscale β
- watermark mask Ω_w
- text mask Ω_t





Watermark Attack:
Grad-WM=Grad-Basic+WM

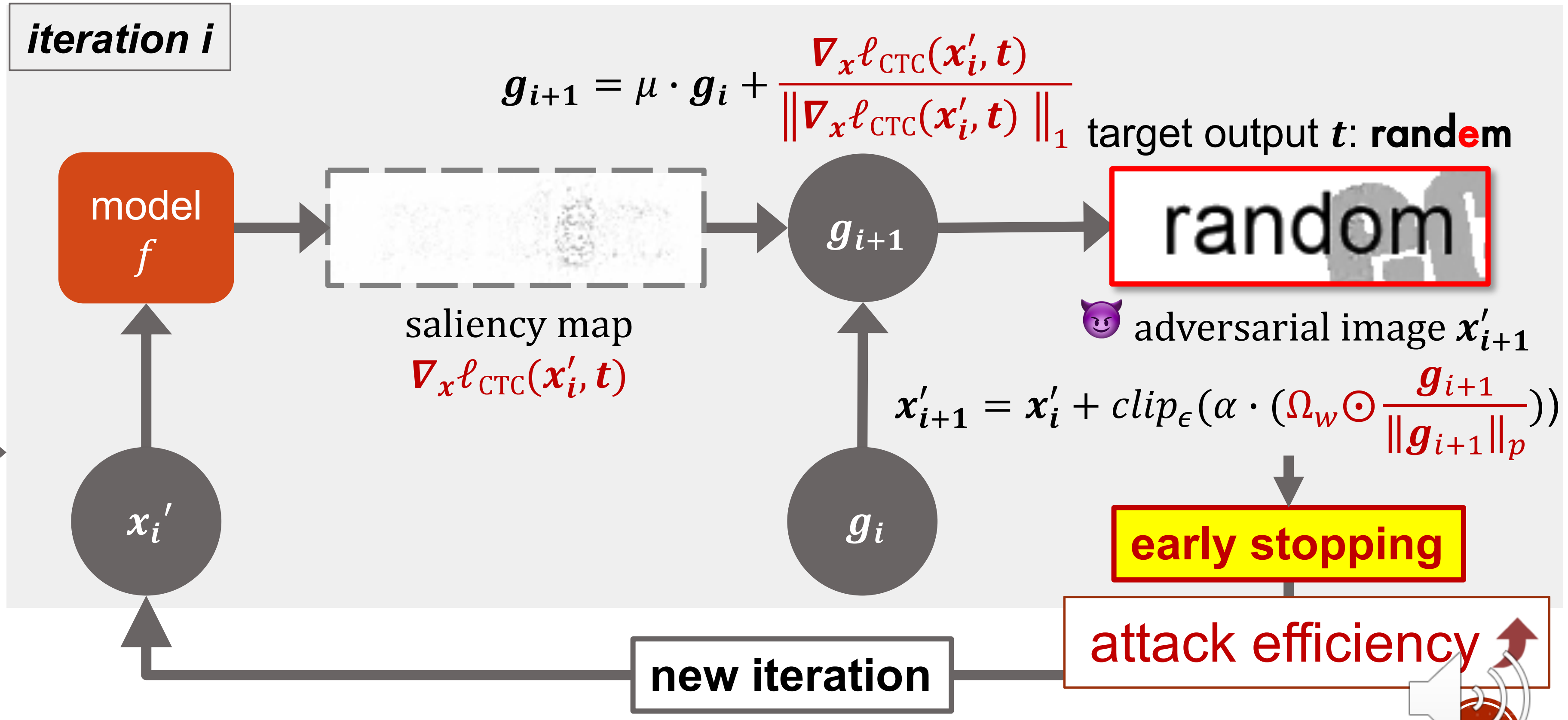
sequential labeling task
background pollution

- grayscale β
- watermark mask Ω_w
- text mask Ω_t

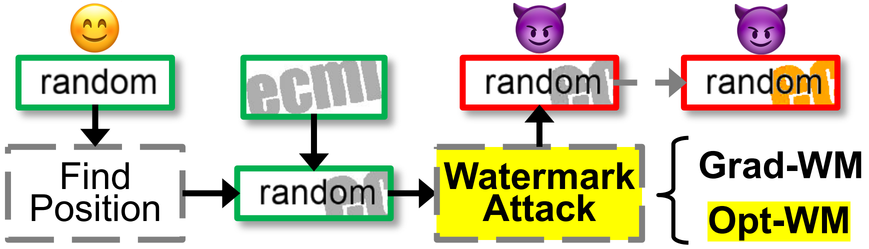
ground-truth $f(x)$: **random**

clean image $\boldsymbol{x}$

initialization: $\boldsymbol{g_0} = \boldsymbol{0}$;
$\boldsymbol{x}'_0 = (1 - \Omega_w \odot \overline{\Omega_t}) \odot \boldsymbol{x} + \beta \cdot \Omega_w \odot \overline{\Omega_t}$;



Watermark Attack:
Opt-WM=Opt-Basic+WM



✅ sequential labeling task
✅ background pollution

- **Opt-Basic:**

$$\min_{\boldsymbol{w}} \; c \cdot \ell_{CTC}\left(\frac{\tanh(\boldsymbol{w})+1}{2}, \boldsymbol{t}\right) + \left\|\frac{\tanh(\boldsymbol{w})+1}{2} - \boldsymbol{x}\right\|_2^2$$

- **Opt-WM:**
  - 1. Separate the *perturbation term* $\boldsymbol{w}$:

$$\min_{\boldsymbol{w}} \; c \cdot \ell_{CTC}\left(\frac{\tanh(\boldsymbol{w}+\boldsymbol{x})+1}{2}, \boldsymbol{t}\right) + \left\|\frac{\tanh(\boldsymbol{w}+\boldsymbol{x})+1}{2} - \boldsymbol{x}\right\|_2^2$$

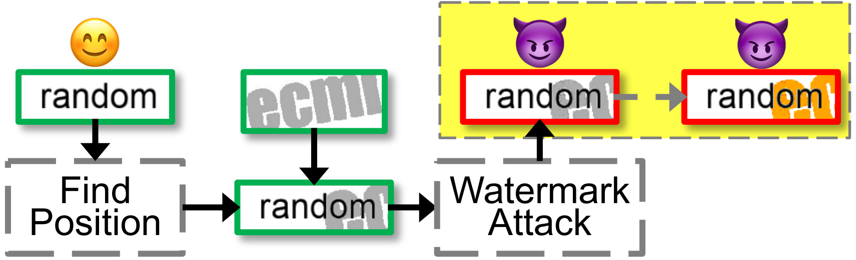
  - 2. Introduce the *watermark mask* Ω_w:

$$\min_{\boldsymbol{w}} \; c \cdot \ell_{CTC}\left(\frac{\tanh(\Omega_w \odot \boldsymbol{w}+\boldsymbol{x})+1}{2}, \boldsymbol{t}\right) + \left\|\frac{\tanh(\Omega_w \odot \boldsymbol{w}+\boldsymbol{x})+1}{2} - \boldsymbol{x}\right\|_2^2$$

amazon  vs.  amazon

Opt-Basic (traditional) **Opt-WM (ours)**

Improving Readability: Full-Color Conversion



- Grayscale watermark 👉 Full-color watermark
- Given grayscale value *Gray*, fix *R* value and *B* value, we can calculate the left *G* value by the *ITU-R 601-2 luma transform:*

$$\mathrm{Gray} = \mathrm{R} * 0.299 + \mathrm{G} * 0.587 + \mathrm{B} * 0.114$$

Experiment Settings:
Threat Model

- **Calamari OCR**
  - a open-source OCR model
  - 2 convolutional layers, 2 pooling layers, a LSTM layer
  - trained by CTC in Tensorflow

Calamari OCR

https://github.com/Calamari-OCR/calamari

Experiment Settings:
Data Generation ---- IMDB

- **Printed-text images (100% accuracy)**
  - 5 fonts: *Courier, Georgia, Helvetica, Times, Arial* (font size:32 px)

    - 1092 word images
    - 1479 sentence images
    - 97 paragraph images

Experiment Settings:
Data Generation ---- IMDB

- **Attack pairs**
  - **Letter-Level Attacks** (word images)
    - Difficulty: **Easy** Case / **Random** Case / **Hard** Case (Replace)

pants


pacts


pasts

    - Operation: **Replace** Case / **Insert** Case / **Delete** Case

pants


partis


pars (parts)

  - **Word-Level Attacks** (word / sentence / paragraph images) (Replace)

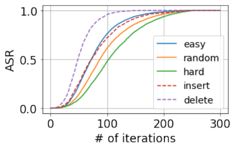
taupe


Tale one did exactly that.

Evaluation Metrics

image quality

- **Perturbation Level**
  - **MSE**： mean-square error $\quad \mathrm{MSE} = \frac{1}{|\boldsymbol{x}|}(\boldsymbol{x} - \boldsymbol{x}')^2$
  - PSNR： peak-signal-to-noise ratio $\quad \mathrm{PSNR} = 10\log(\frac{\mathrm{D}^2}{\mathrm{MSE}})$
  - SSIM： structural similarity index
- **Success Rate** ✅
  - ASR： targeted attack success rate 💯 $\quad \mathrm{ASR} = \frac{\#(f(\boldsymbol{x}')=\boldsymbol{t})}{\#(\boldsymbol{x})}$
- **Attack Efficiency** ⚡
  - **$\mathbf{I_{avg}}$** ： average iterations of successful attacks $\quad \mathrm{I_{avg}} = \frac{\sum_i^n \#(\mathrm{iteration}_i)}{n}$
  - **ASR-Iter**: plot the change of ASR along with increasing iterations



- MSE: mean-square error
- I_{avg} : average iterations of successful attacks
- ASR: targeted attack success rate 💯

Basic Attack vs. FAWA:
Grad-Basic vs. Grad-WM

		replacement						insertion		deletion	
		easy		random		hard					
		MSE	I_{avg}	MSE	I_{avg}	MSE	I_{avg}	MSE	I_{avg}	MSE	I_{avg}
Grad-Basic	Courier	10.5	59	14.0	74	17.0	70	11.6	50	3.2	21
	Georgia	27.4	43	32.8	99	37.3	104	22.1	83	17.3	55
	Helvetica	27.0	51	33.6	113	38.6	113	23.0	70	16.7	43
	Times	26.4	62	31.5	85	35.8	109	20.3	98	17.2	68
	Arial	29.8	51	36.7	73	42.5	66	24.3	88	19.2	59
example		parts		parts		parts		parts		parts	
Grad-WM	Courier	2.8	30	3.6	18	4.3	27	3.6	21	0.7	8
	Georgia	7.8	15	8.9	33	9.8	30	5.1	39	3.5	21
	Helvetica	8.4	9	10.0	52	11.2	52	6.3	23	3.7	19
	Times	7.3	15	8.3	20	9.3	34	4.5	7	3.4	21
	Arial	9.4	13	11.1	14	12.7	25	6.2	33	4.4	20
example		parts		parts		parts		parts		parts	
target output		pants		pacts		pasts		partis		pars	

letter-level attack in word images

- **ASR: 100% in default**

- MSE: mean-square error
- I_{avg} : average iterations of successful attacks
- ASR: targeted attack success rate 💯

FAWA:
Lower Perturbation Level

		replacement						insertion		deletion	
		easy		random		hard					
		MSE	I_{avg}	MSE	I_{avg}	MSE	I_{avg}	MSE	I_{avg}	MSE	I_{avg}
Grad-Basic	Courier	10.5	59	14.0	74	17.0	70	11.6	50	3.2	21
	Georgia	27.4	43	32.8	99	37.3	104	22.1	83	17.3	55
	Helvetica	27.0	51	33.6	113	38.6	113	23.0	70	16.7	43
	Times	26.4	62	31.5	85	35.8	109	20.3	98	17.2	68
	Arial	29.8	51	36.7	73	42.5	66	24.3	88	19.2	59
example		parts		parts		parts		parts		parts	
Grad-WM	Courier	2.8	30	3.6	18	4.3	27	3.6	21	0.7	8
	Georgia	7.8	15	8.9	33	9.8	30	5.1	39	3.5	21
	Helvetica	8.4	9	10.0	52	11.2	52	6.3	23	3.7	19
	Times	7.3	15	8.3	20	9.3	34	4.5	7	3.4	21
	Arial	9.4	13	11.1	14	12.7	25	6.2	33	4.4	20
example		parts		parts		parts		parts		parts	
target output		pants		pacts		pasts		partis		pars	

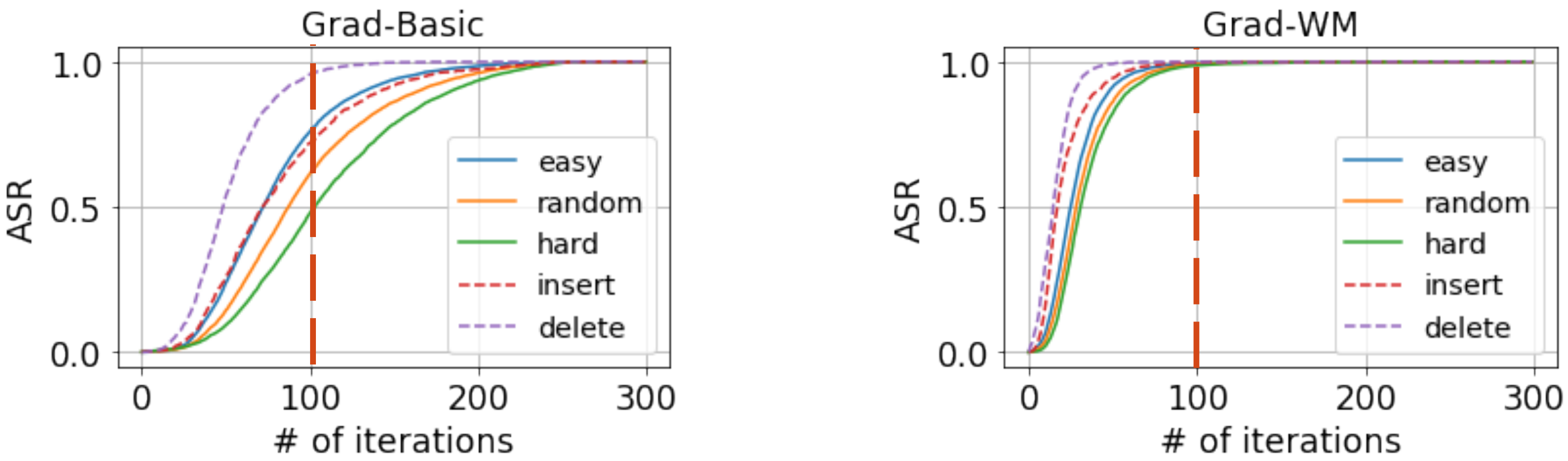
letter-level attack in word images

- **74% less noise** (MSE) on average

FAWA: Faster Attack Speed

- MSE: mean-square error
- I_{avg} : average iterations of successful attacks
- ASR: targeted attack success rate 💯

- **67% fewer iterations** (I_{avg}) on average
- A sharper slope indicates faster attack speed in the figure.



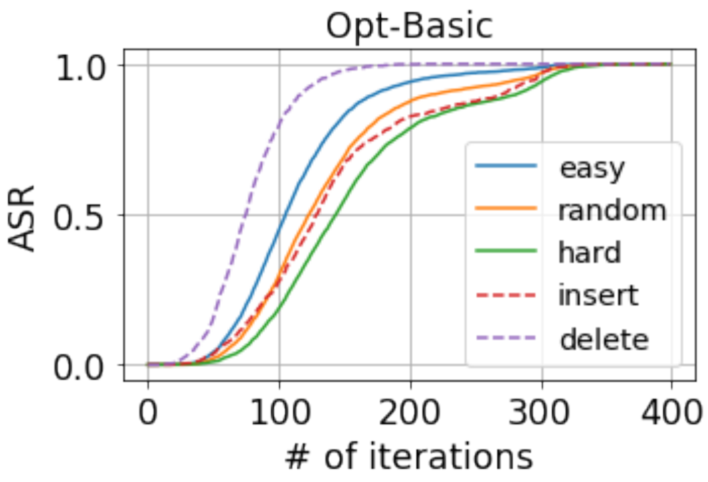
Basic Attack vs. FAWA:
Opt-Basic vs. Opt-WM

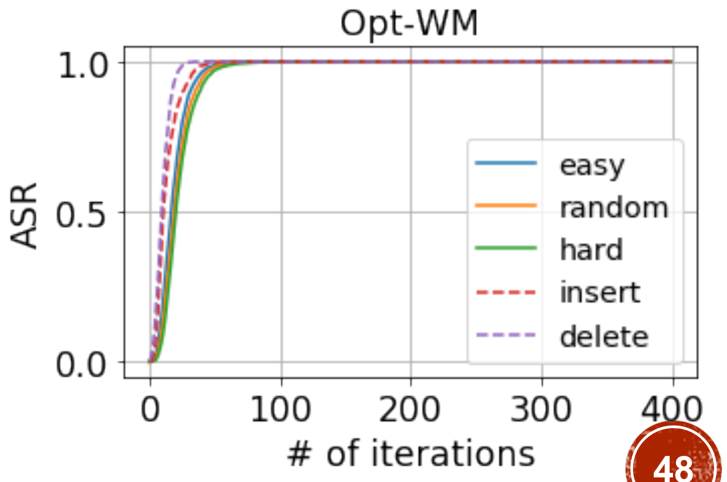
- MSE: mean-square error
- I_{avg} : average iterations of successful attacks

- **44% less noise**
- **88% fewer iterations**

		replacement						insertion		deletion	
		easy		random		hard					
		MSE	I_{avg}	MSE	I_{avg}	MSE	I_{avg}	MSE	I_{avg}	MSE	I_{avg}
Opt-Basic	Courier	25.4	266	30.3	313	36.7	321	25.4	309	13.6	43
	Georgia	52.0	292	59.4	318	67.5	328	41.6	337	45.0	169
	Helvetica	52.1	301	60.2	328	68.2	340	47.1	321	45.0	178
	Times	49.9	294	56.1	324	61.6	345	41.7	314	44.3	172
	Arial	56.3	304	65.3	327	73.8	341	48.3	324	51.0	176
example		parts		parts		parts		parts		parts	
Opt-WM	Courier	16.7	116	20.1	96	20.4	95	31.1	29	3.2	13
	Georgia	31.6	30	35.1	32	38.3	37	21.7	12	16.2	9
	Helvetica	33.3	31	37.0	42	38.8	53	25.1	13	16.5	9
	Times	30.3	22	33.9	26	35.9	36	19.2	11	15.4	8
	Arial	37.2	30	40.4	45	43.6	50	25.4	16	19.4	10
example		parts		parts		parts		parts		parts	
target output		pants		pacts		pasts		partis		pars	

letter-level attack in word images

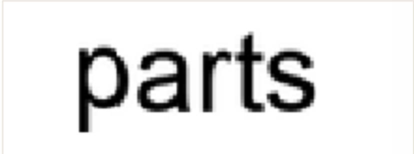
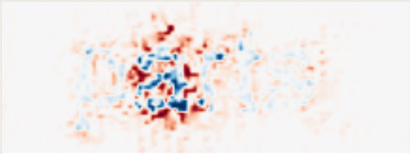
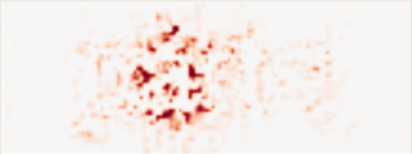
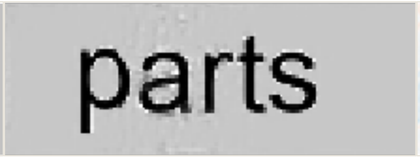
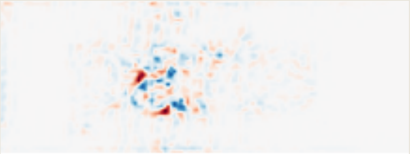
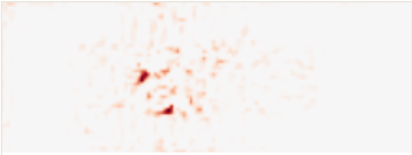
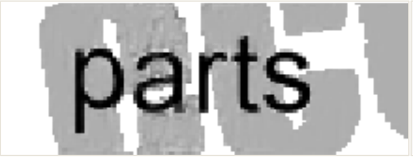
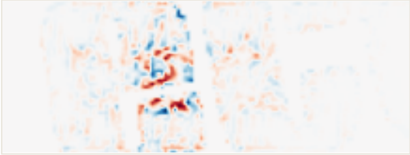
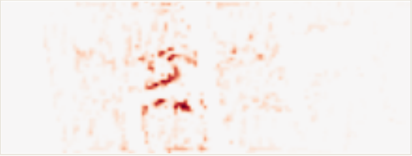




- MSE: mean-square error

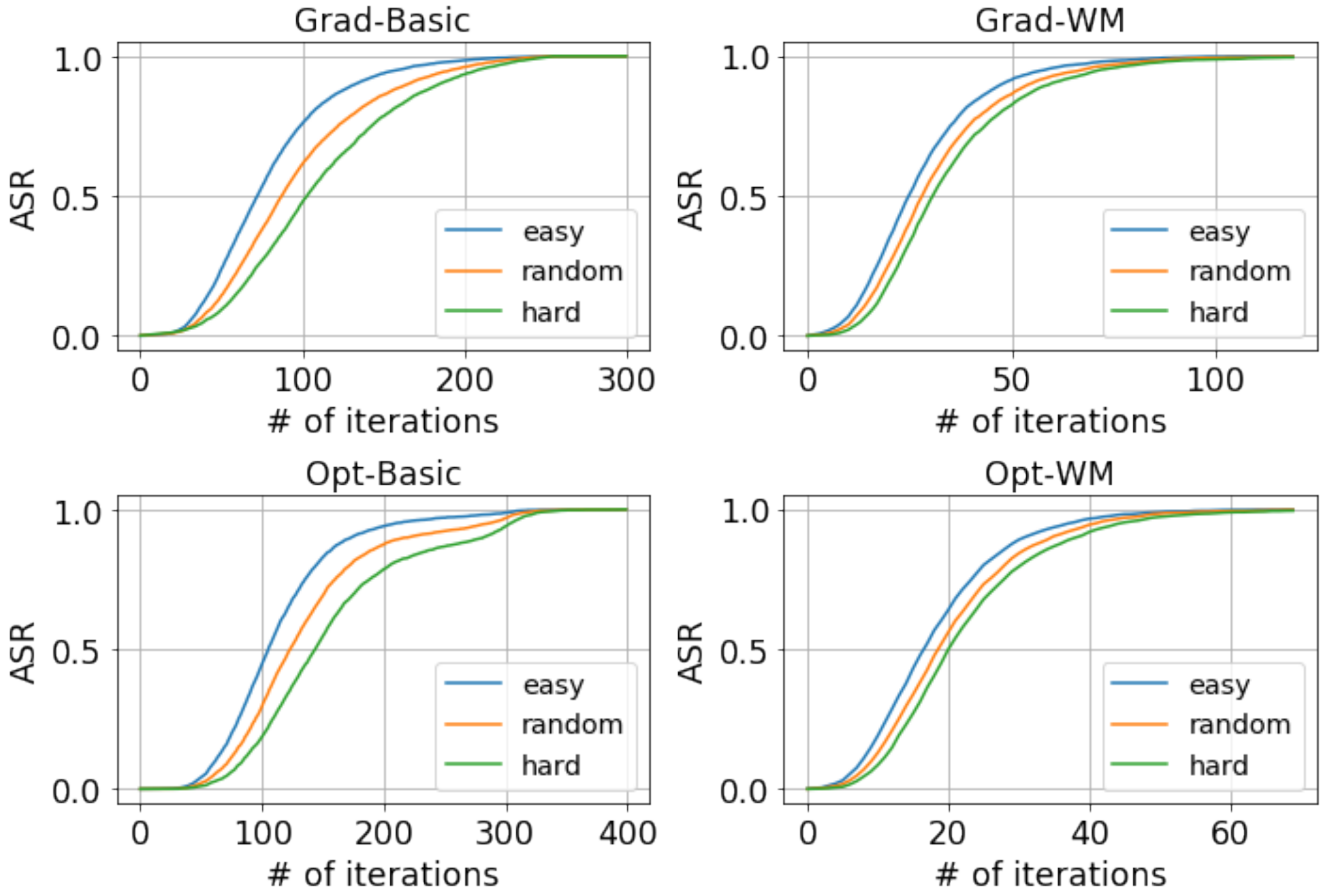
Visualization Explaination: Saliency Map

- The first to the last line are **clean**, **gray** and **watermark** backgrounds.
- **Reduced contrast** is beneficial to reduce noise.

clean images	adversarial images	saliency map	MSE	saliency map+	MSE+	saliency map-	MSE-
			55.82		34.77		21.05
			15.47		6.92		8.55
			24.97		14.13		10.84

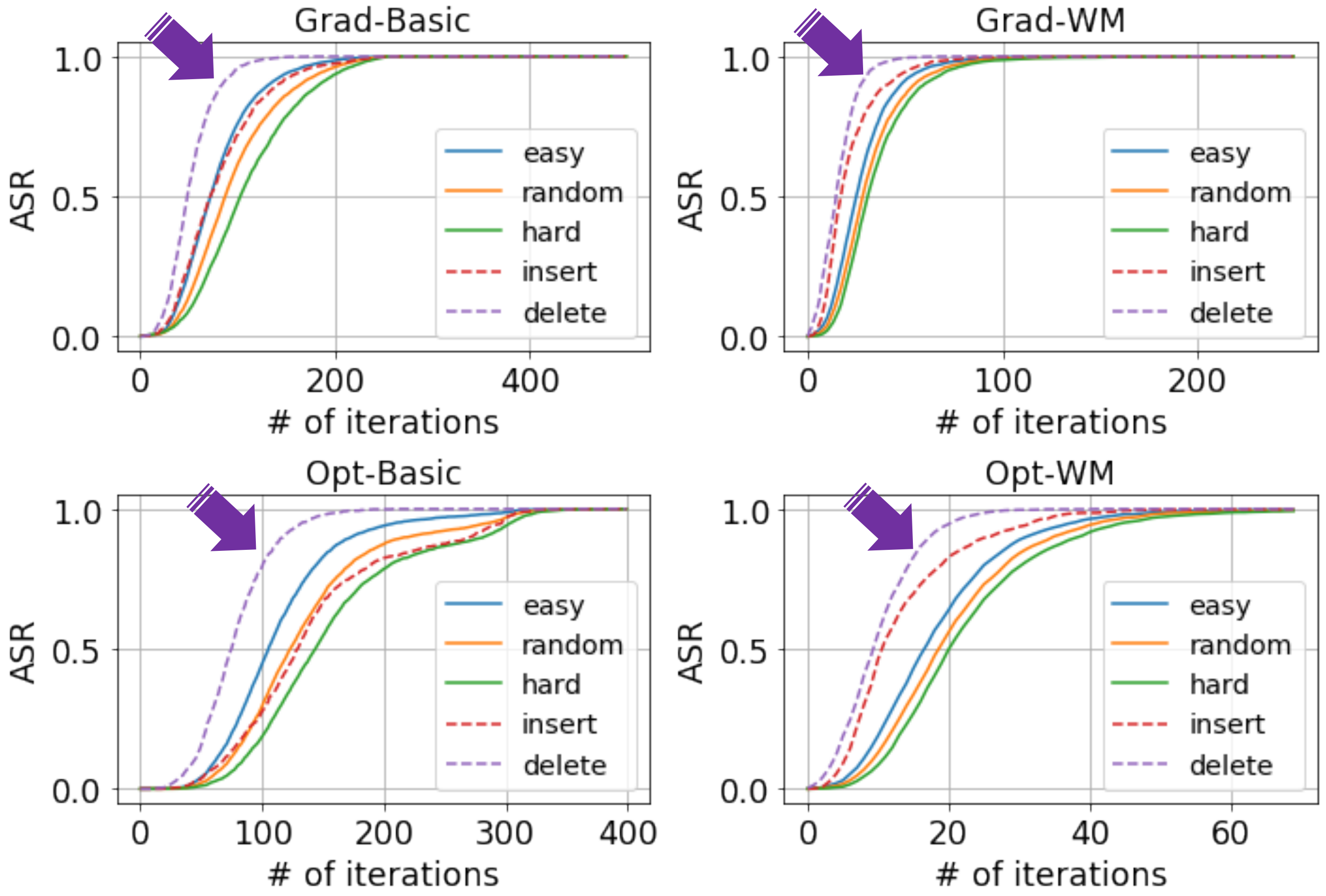
The target output is "ports".

Attack Difficulty:
Easy < Random < Hard



letter-level attack in word images with Arial font

Attack Difficulty:
Delete < Insert & Replace



letter-level attack in word images with Arial font

Word-Level Attacks:
WM is More Natural

<u>taupe</u>

Grad-Basic:

Grad-WM:

<u>Tale</u> one did exactly that.

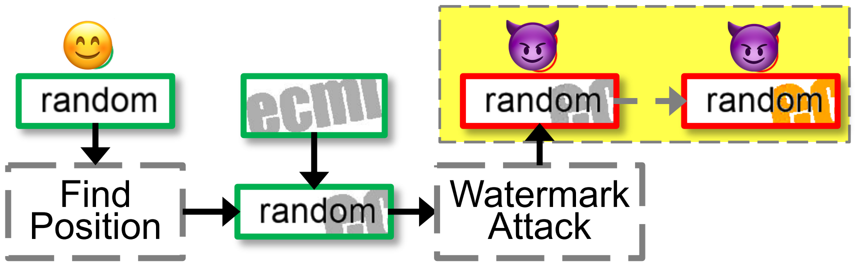
Grad-Basic:

Grad-WM:

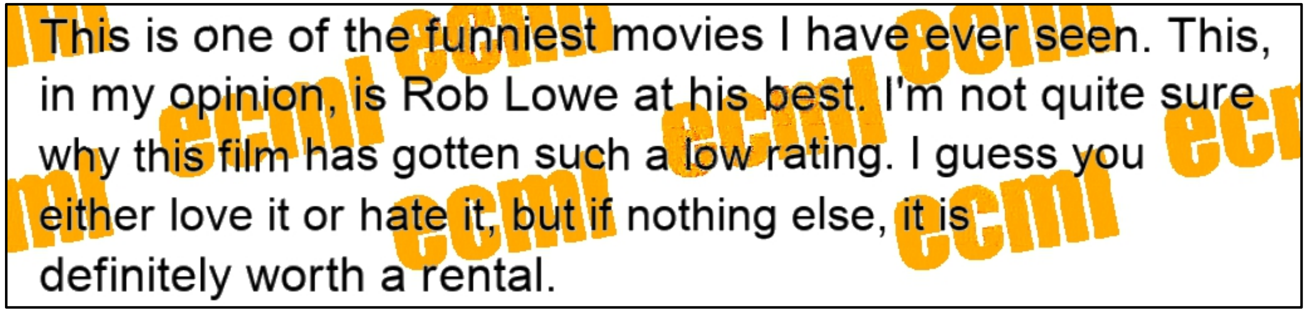
WM attack:

- **56% lower noise**
- **50% less iterations**

Improve Readability: Full-Color Watermarks



Input Image: positive



OCR Output: negative

This is one of the <u>scariest</u> movies I have ever seen. This, in my opinion, is Rob Lowe at his <u>worst</u>. I'm not quite sure why this film has gotten such a <u>high</u> rating. I guess you either love it or hate it, but if nothing else, it is <u>not</u> definitely worth a rental.

Other Colored Adversarial Examples:

Conclusion

- We propose **fast adversarial watermark attacks (*FAWA*)** on sequence-based **OCR** models.
  - Sequential labeling task——CTC loss
  - Background pollution——watermark
  - **Natural** watermark-style noise.
  - **Lower** perturbation level.
  - **Faster** attack speed.

WATERMARK

THANK YOU

Lu Chen[1], Jiao Sun[2], Wei Xu[1]

[1]IIIS, Tsinghua University

[2]CS, University of Southern California

清華大學
Tsinghua University

交叉信息研究院
Institute for Interdisciplinary
Information Sciences