Making targeted black-box evasion attacks effective and efficient

<u>Mika Juuti</u>

Aalto University /
University of Waterloo

Joint work with

Buse Atli
Aalto University

N. Asokan
University of Waterloo

12th ACM Workshop on Artificial Intelligence and Security, November 15th 2019, London, UK

Preliminaries

Black-box evasion attacks

Black-box attacks advancing rapidly [1,2]

… but efficiency depends on what is API

… whether is targeted attack

Many realistic APIs are restrictive

- Scores for a small subset of all classes
  - Partial Information
- Existing targeted attacks inefficient or ineffective

[1] Ilyas et al. *Black-box adversarial attacks with limited information and queries*. ICML'18.
[2] Co et al. *Procedural Noise Adversarial Examples for Black-Box Attacks on Deep Convolutional Networks.* ACM CCS 2019

Query-only methods: Natural Evolution Strategies

Case study on NES [1, 2]:

- Most effective query-only method for targeted adversarial example crafting for such partial information APIs
- Start / goal image distinction

For-loop with three phases:

- Increase pseudo-log-likelihood via NES
- Line search for decreasing perturbation
- Update or backtrack (reset search)

```
# MAIN LOOP
for i in range(max_iters):

    l, g = get_grad(adv, args.samples_per_draw, batch_size)

    proposed_adv = adv - is_targeted * current_lr * np.sign(g)
    prop_de = 0.0

    while current_lr >= args.min_lr:
        # PARTIAL INFORMATION ONLY
        if k < NUM_LABELS:
            proposed_epsilon = max(epsilon - prop_de, goal_epsilon)

        # GENERAL LINE SEARCH
        proposed_adv = adv - is_targeted * current_lr * np.sign(g)
        proposed_adv = np.clip(proposed_adv, lower, upper)

        if robust_in_top_k(target_class, proposed_adv, k):

            adv = proposed_adv
            epsilon = max(epsilon - prop_de/args.conservative, goal_epsilon)

        else:

            if prop_de < 2e-3:
                prop_de = 0
            current_lr = max_lr
            print("[log] backtracking eps to %3f" % (epsilon-prop_de,))
```

```
# GRADIENT ESTIMATION EVAL
def get_grad(pt, spd, bs):

        loss, dl_dx_ = sess.run([final_losses, grad_estimate], feed_dict)

        noise_pos = tf.random_normal((batch_per_gpu//2,) + initial_img.shape)
        noise = tf.concat([noise_pos, -noise_pos], axis=0)
        eval_points = x + args.sigma * noise
```

[1] Ilyas et al. *Black-box adversarial attacks with limited information and queries*. ICML'18.
[2] https://github.com/labsix/limited-blackbox-attacks/blob/master/attacks.py

Targeted attacks on restrictive APIs

Query-only methods:

- High effectiveness, any DNN attackable
- Inefficient: requires 1000s – 10,000s queries per sample on restrictive APIs

Transferability ensemble methods [1,2]

- Efficient: first query may already succeed
- Ineffective: success rate is low
- Case study on MIFGSM

What can the adversary do to make targeted evasion more efficient while retaining effectiveness?

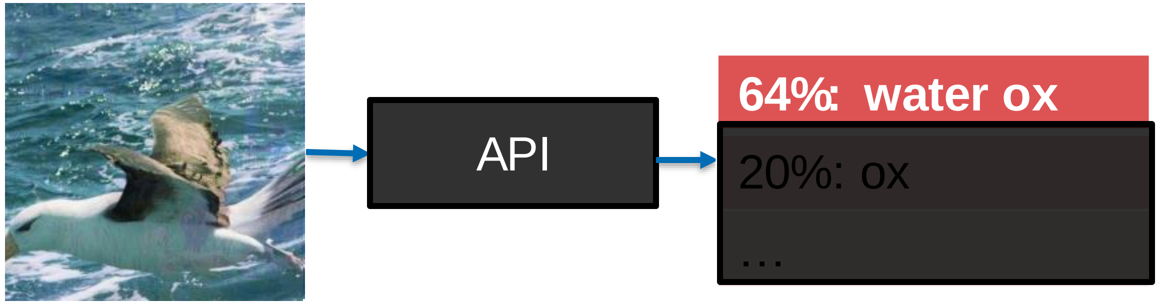
"We notice that targeted attacks have little transferability … it's hard … for the ImageNet dataset."

[1] Liu et al. *Delving into transferable adversarial attacks.* ICLR'17
[2] Dong et al. *Boosting adversarial attacks with momentum.* CVPR'18 NIPS adversarial attack competition winners
[3] dongyp13/Targeted-Adversarial-Attack https://github.com/dongyp13/Targeted-Adversarial-Attack)

Adversary model

Minimum distance adversarial examples

- Up to 5% mod. (12.8/255) on L_∞-norm [1]
- Partial information on API outputs (label+prob.), API access black-box



Evaluation

- 100 images, adapted from [2]:
  - Also includes start images
- Evaluation on ImageNet classifiers:
  - ResNet-101/152, VGG16, Inception v3
- Realistically adversary has access to 10s of surrogate models

Goal	Tgt class
	*Water ox*
	*Brown bear*
	*French horn*
…	…

[1] Ilyas et al. *Black-box adversarial attacks with limited information and queries.* ICML'18.
[2] Liu et al. *Delving into transferable adversarial attacks.* ICLR'17 : sunblaze-ucb/transferability-advdnn-pub https://github.com/sunblaze-ucb/transferability-advdnn-pub/blob/master/data/image_label_target.csv

Results

Baseline results

	Ensemble transferability	Query-only (max 100,000 queries)
Inception v3	12% : 1	88%: 44,158
ResNet-101	47%: 1	89%: 32,864
VGG16	47%: 1	94%: 28,875
ResNet-152	58%: 1	91%: 34,689

Success rate: mean queries

Adversary has large ensemble with 10/11 components

- Targeted transferability between 47% and 58%

Transferability worst on Inception v3

- Resizing operation from 224 → 299 pixels, functions as a defense [1]

[1] Xie et al. *Black-box adversarial attacks with limited information and queries*. ICLR'18.

Basic agility

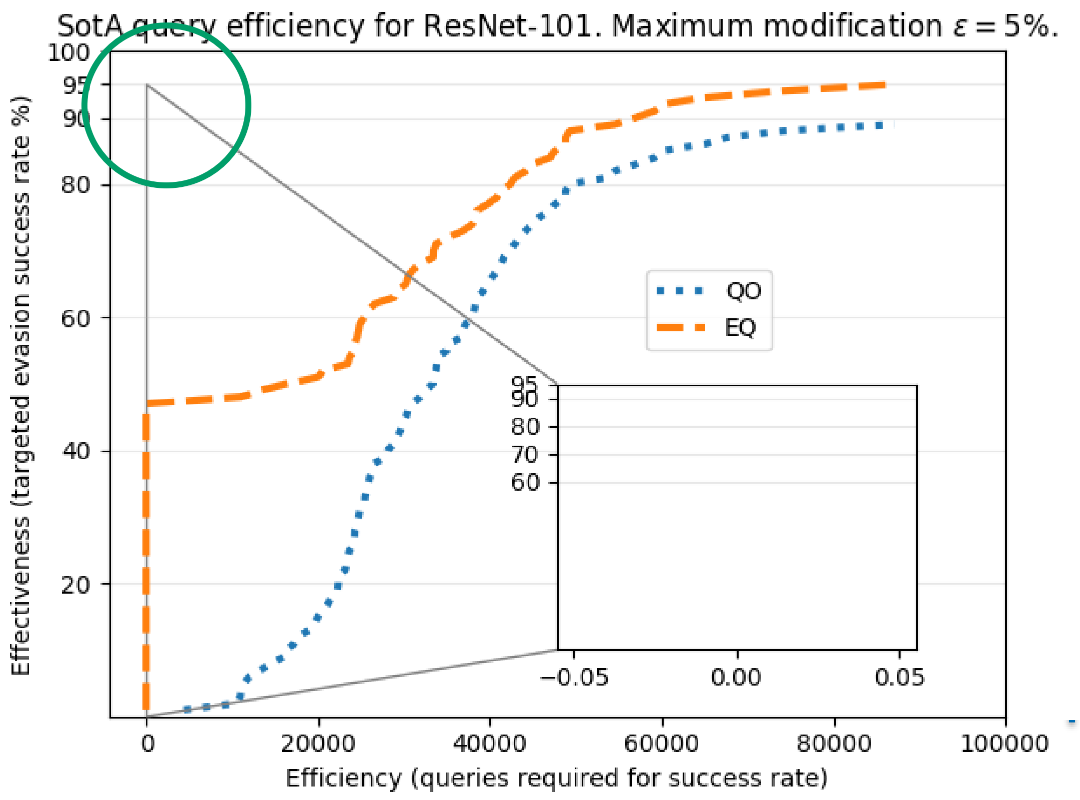
We investigate agile adversaries:

- Can combine methods to reduce queries

Basic agile adversary:

- Ensemble method, then query-only: EQ
- Improves efficiency and effectiveness

Agile adversary can improve efficiency / effectiveness



Improved efficiency / effectiveness

Can we improve efficiency / effectiveness trade-off
by designing a new type of attack?

1. **Take [1] work as a starting point, maintain start / goal image distinction as in [1]**
   - Benefits for effectiveness?
2. **Replace NES with ensemble-based gradient [2]**
   - NES [1] perturbation calculation requires ~ 100 queries per sample
3. **Avoid queries from line search**
   - Unnecessary if ensemble gradient close to API model's

PRISM: Partial Information Substitute Model Attack

[1] Ilyas et al. *Black-box adversarial attacks with limited information and queries*. ICML'18.
[2] Liu et al. *Delving into transferable adversarial attacks*. ICLR'17
[3] Dong et al. *Boosting adversarial attacks with momentum*. CVPR'18

PRISM: variants and performance

PRISM and PRISM$_R$

- Use all ensemble components or random subset for gradient calculation
- More effective than Ensemble alone
- Require more queries, but can increase effectiveness over regular ensemble-use

Table 5: Effectiveness of black-box evasion methods, *success rate* and *median* number of queries required for success.

	Up to 1000 queries			
	Ensemble	PRISM	PRISM$_R$	Query-only
IncV3	26%: 2	69%: 11	75%: 14	0%: -
RN101	83%: 1	88%: 8	93%: 12	0%: -
VGG16	82%: 1	89%: 10	90%: 13	0%: -
RN152	84%: 1	95%: 8	96%: 11	0%: -

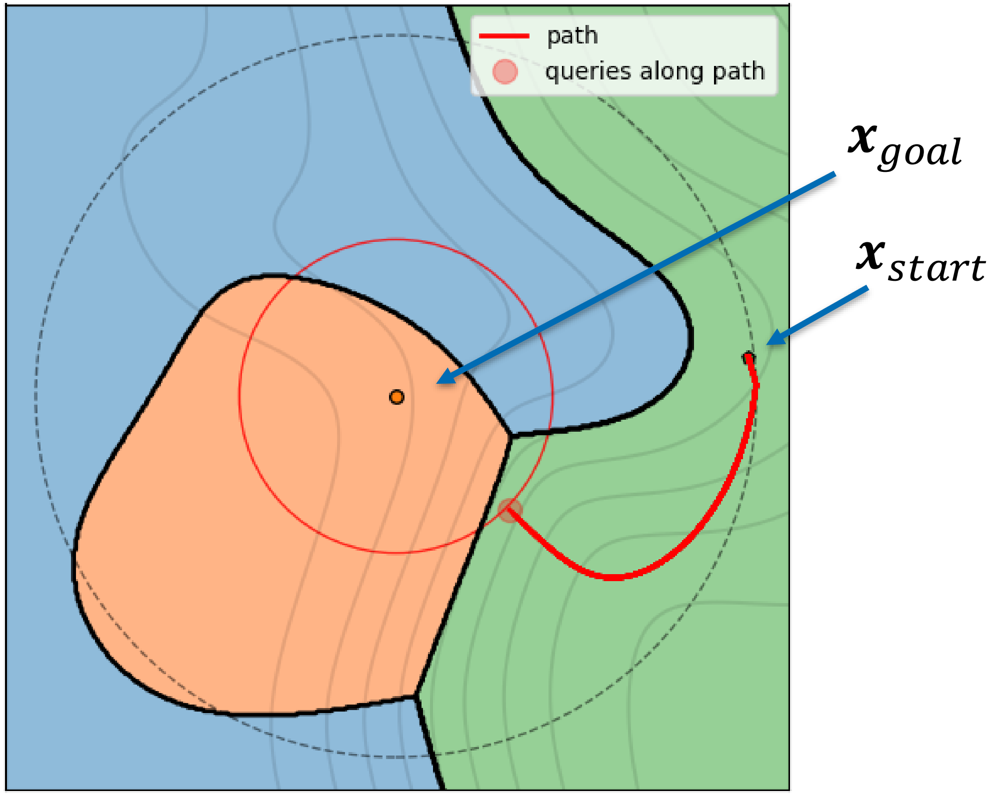


Illustration on PRISM

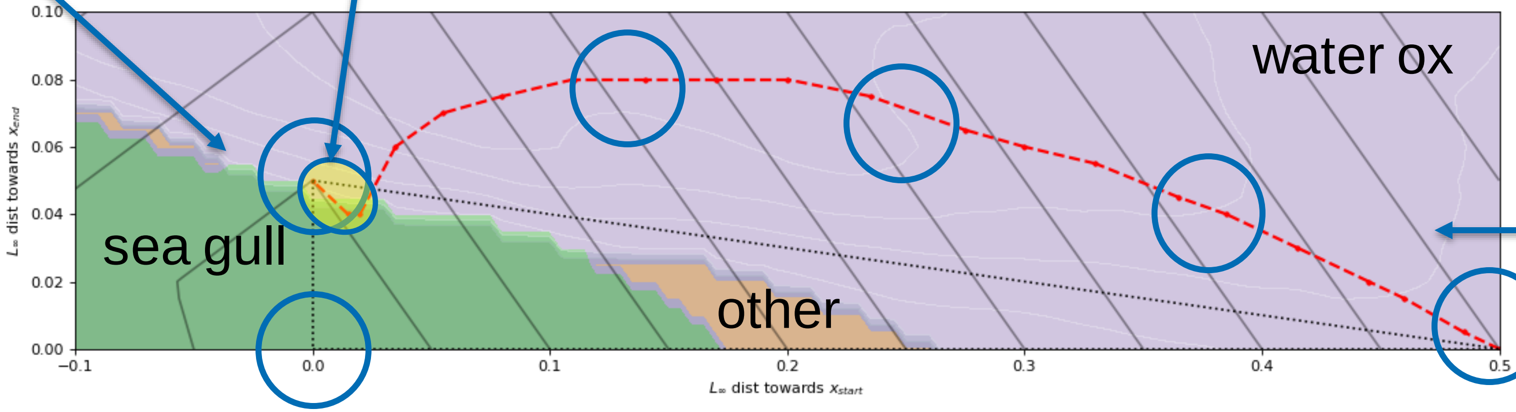
Start from same-class start image

Illustration on PRISM trajectory on ResNet-101

RN101 log-likelihood

Typical query region

L_∞ to x_{adv}



L_∞ contour

L_∞-distance to x_{start}

x_{goal} x_{adv} x_{start}

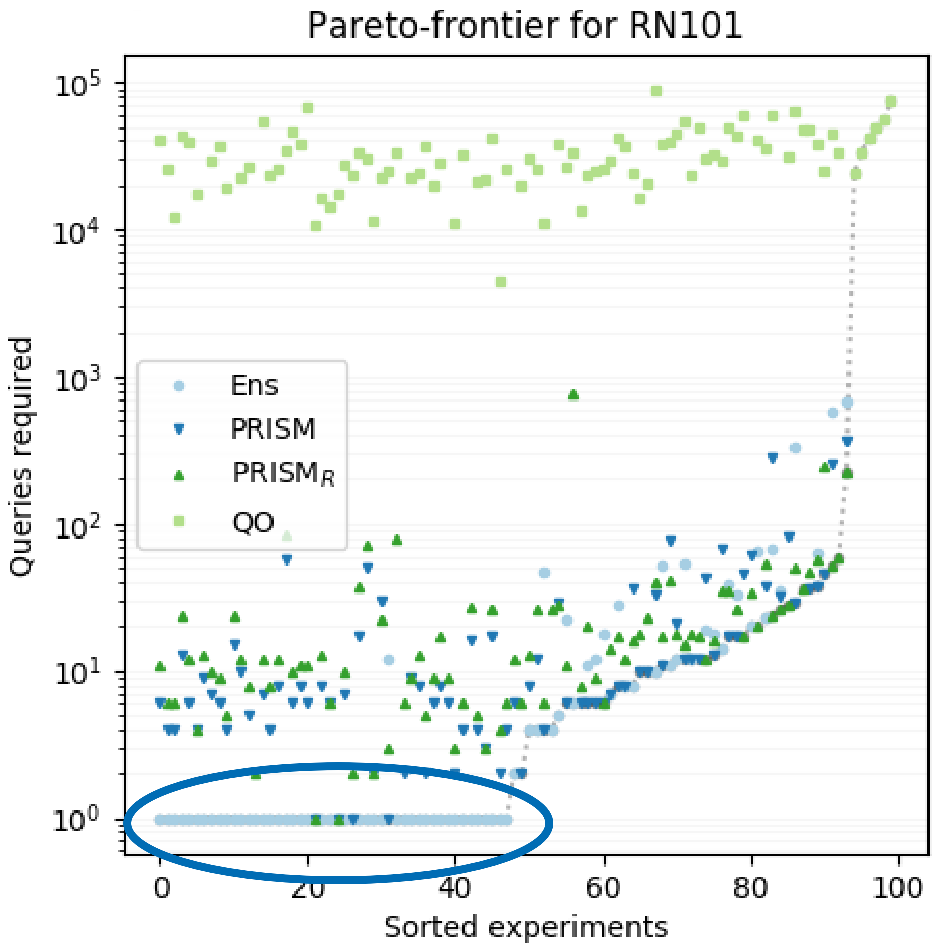
Localized perturbations

Pareto-efficiency

Given the same task, which methods are most efficient?

Upwards trend:

- **Some experiments are harder than others**
  - Larger number of min-queries to succeed
- **Transferability works in many cases**
  - Similarity between surrogates and victim (next slide)



Methods trade off efficiency for effectiveness

Impact of ensemble size

Number of components.	1	2	3	4	5	6	7	8	9	10
Target model					IncV3					
added model to ens.	DN201	RN101	RN50	DN169	DN121	RN34	RN18	VGG11	SN1.1	SN1.0
ENSEMBLE (1 query)	2% : 1	4%: 1	5%: 1	6%: 1	6%: 1	8%: 1	10%: 1	12%: 1	12%: 1	12%: 1
ENSEMBLE (up to 1000 queries)	4% : 9	6%: 1	7%: 1	10%: 1	13%: 2	14%: 1	18%: 1	24%: 1	23%: 1	26%: 2
PRISM (up to 1000 queries)	2%: 89	6%: 60	12%: 28	16%: 27	26%: 14	41%: 15	54%: 12	60% 10	62%: 9	69%: 11
Target model					RN101					
added model to ens.	DN201	**RN50**	DN169	DN121	**RN34**	VGG16	**RN18**	VGG11	SN1.1	SN1.0
ENSEMBLE (1 query)	4%: 1	11%: 1	14%: 1	21%: 1	34%: 1	34%: 1	39%: 1	45%: 1	44%: 1	47%: 1
ENSEMBLE (up to 1000 queries)	7%: 1	**29%: 8**	35%: 3	43%: 2	59%: 1	62%: 1	74%: 1	76%: 1	78%: 1	83%: 1
PRISM (up to 1000 queries)	7%: 84	**34%: 29**	49%: 26	56%: 16	**69%: 14**	69%: 12	**83%: 8**	85%: 8	87%: 9	88%: 8
Target model					VGG16					
added model to ens.	DN201	RN101	RN50	DN169	DN121	RN34	RN18	**VGG11**	SN1.1	SN1.0
ENSEMBLE (1 query)	1%: 1	3%: 1	3%: 1	6%: 1	13%: 1	13%: 1	21%: 1	**41%: 1**	42%: 1	47%: 1
ENSEMBLE (up to 1000 queries)	6%: 11	9%: 87	13%: 17	20%: 10	26%: 1	34%: 5	44%: 2	75%: 1	80%: 1	82%: 1
PRISM (up to 1000 queries)	3%: 248	7%: 77	15%: 52	27%: 34	32%: 22	38%: 21	54%: 17	**85%: 12**	86%: 11	86%: 10

Dominance

Given number of minimum queries, can we prescribe when methods perform better than others?

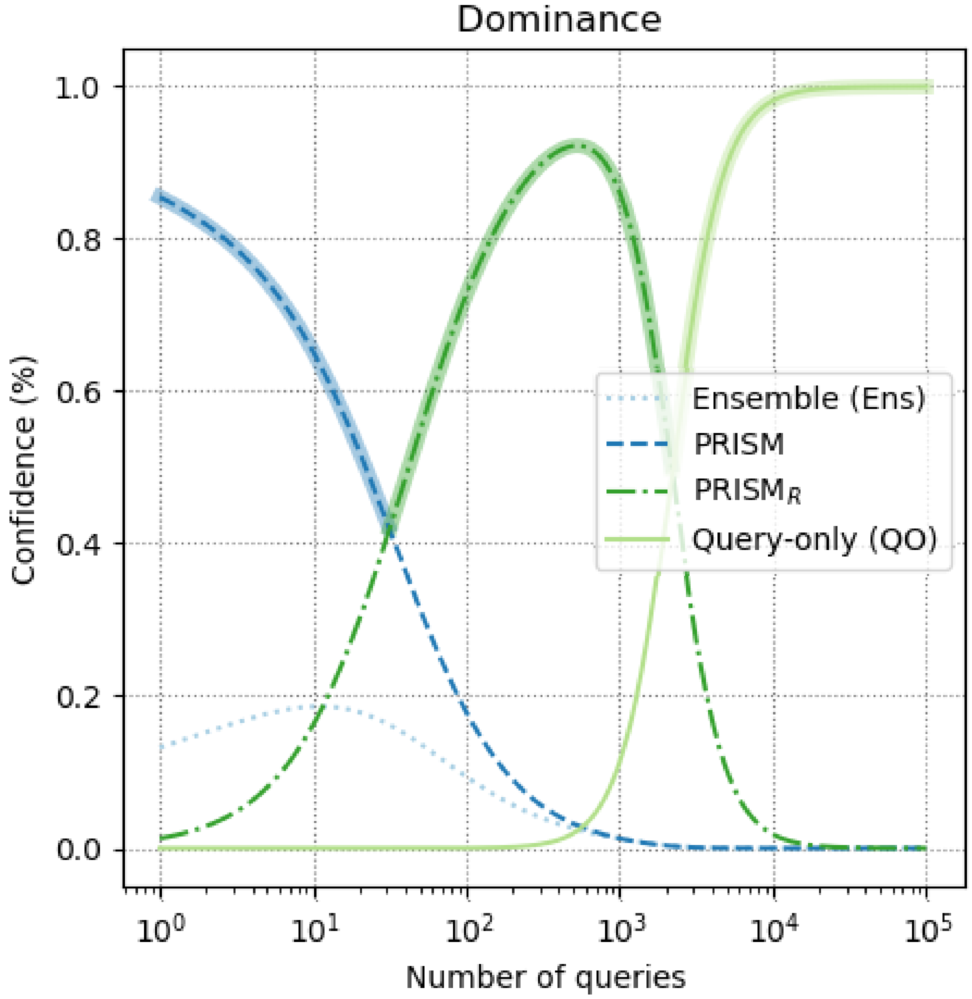
- Enables efficient strategy determination

Example efficient strategy:

- Ens: 0—1
- PRISM: 1—50
- PRISM$_R$: 50 – 3000
- Query-only: 3000+

>> EPP$_R$Q

Superimposed results for IncV3/ResNet-101/VGG16

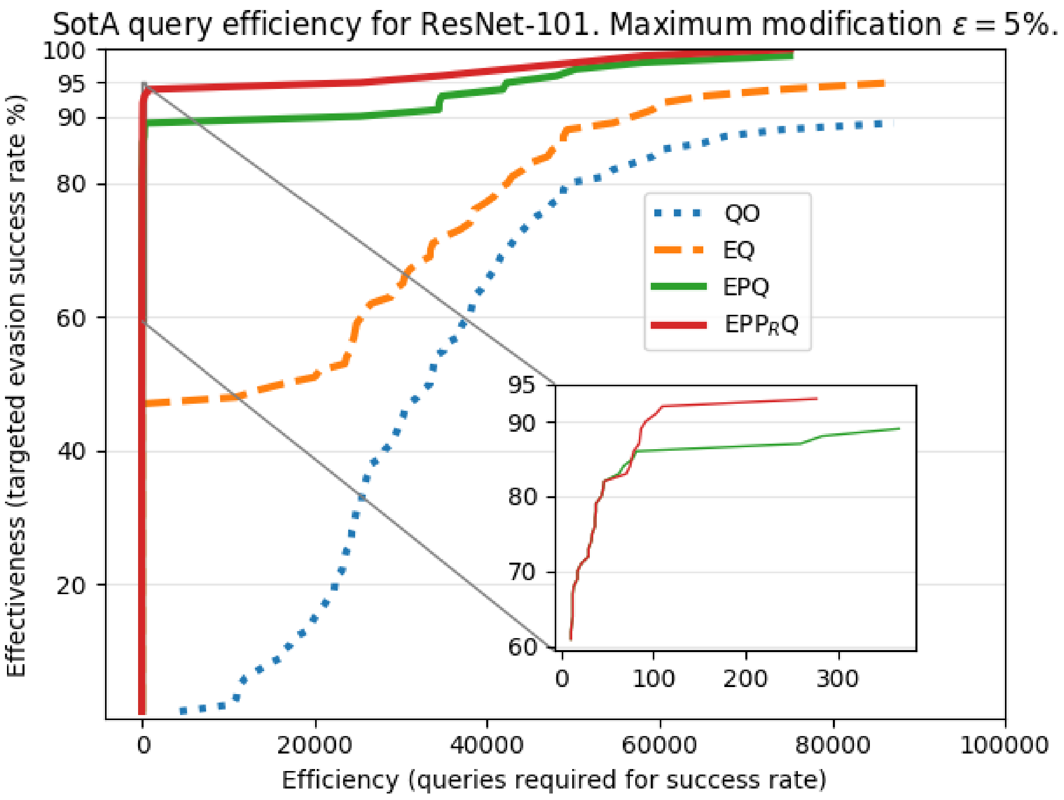


Fully agile attacker

Fully agile adversary EPP_RQ:

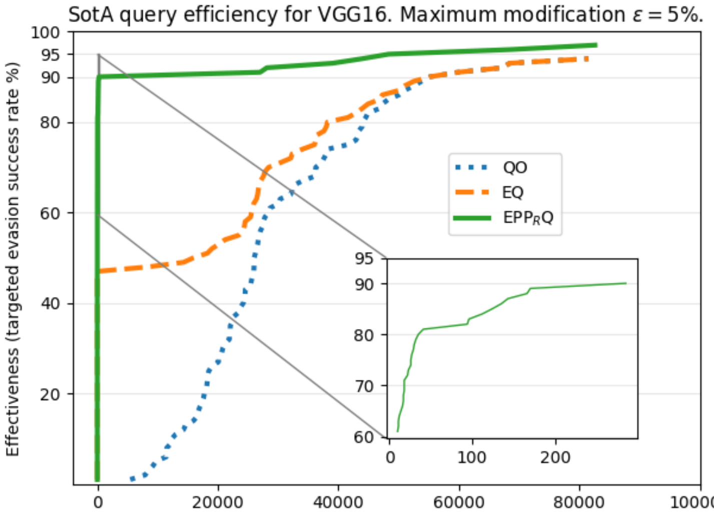
- Effectiveness: +3% to +13%
- Query-efficiency: 1.97x to 24.4x less (average)

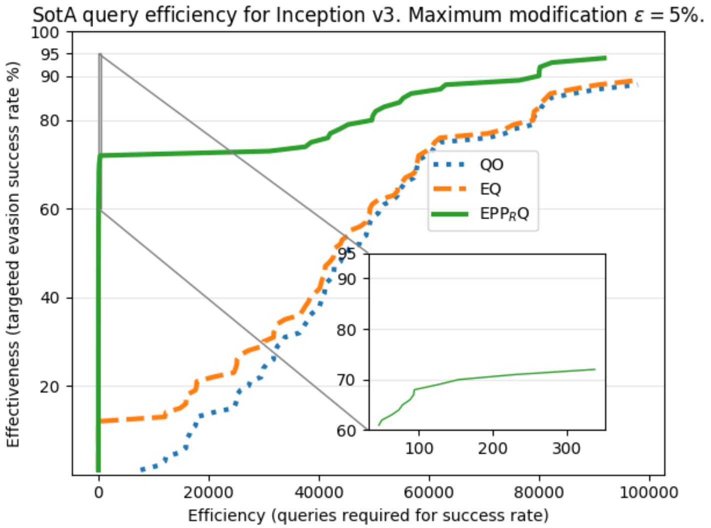
	Alternative strategies Success rate: average qs to reach		
Victim	**Fully agile EPP_RQ**	**Basic agile EQ**	**Baseline QO**
ResNet-101	**100%: 1171**	95%: 6.55x	94%: 11.4x
ResNet-152	**100%: 3005**	95%: 10.4x	91%: 24.4x
VGG16	**97%: 3359**	94%: 4.98x	94%: 8.26x
Inception v3	**94%: 13219**	89%: 1.97x	88%: 2.27x

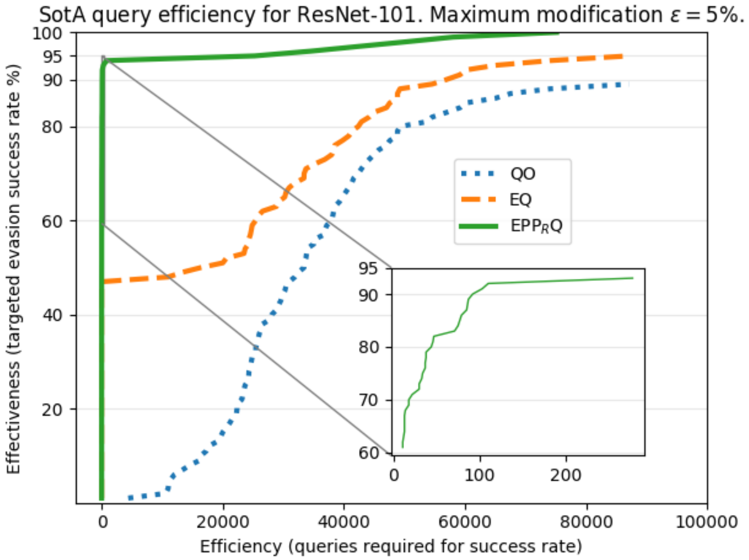


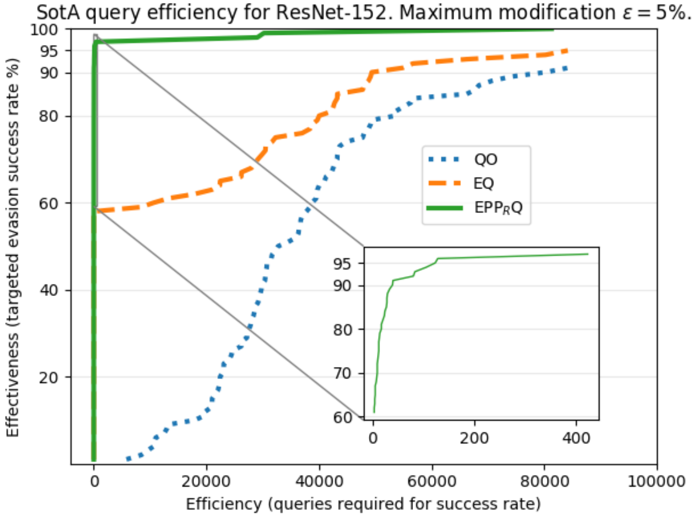
Different victim APIs (ImageNet)

- Most efficient when surrogate models available of similar architecture
  - ResNet-101 and ResNet-152
- Typically 2—3 orders faster than query-only alone





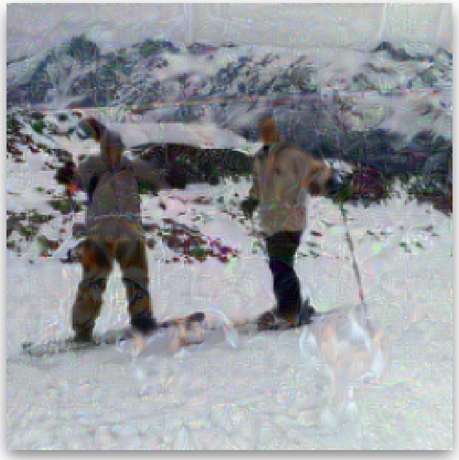




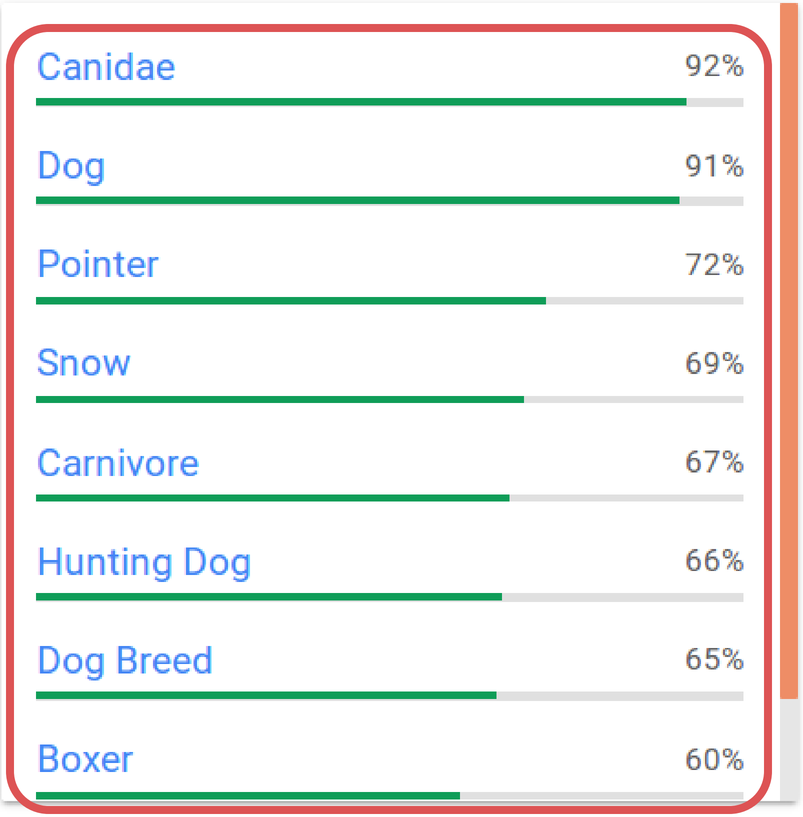
Case study: realistic APIs

PRISM / PRISM$_R$ effective against real APIs

- Reduce number of queries for one example from ~20,000 [1] to ~400—1000
- Example as in [1]



390.png

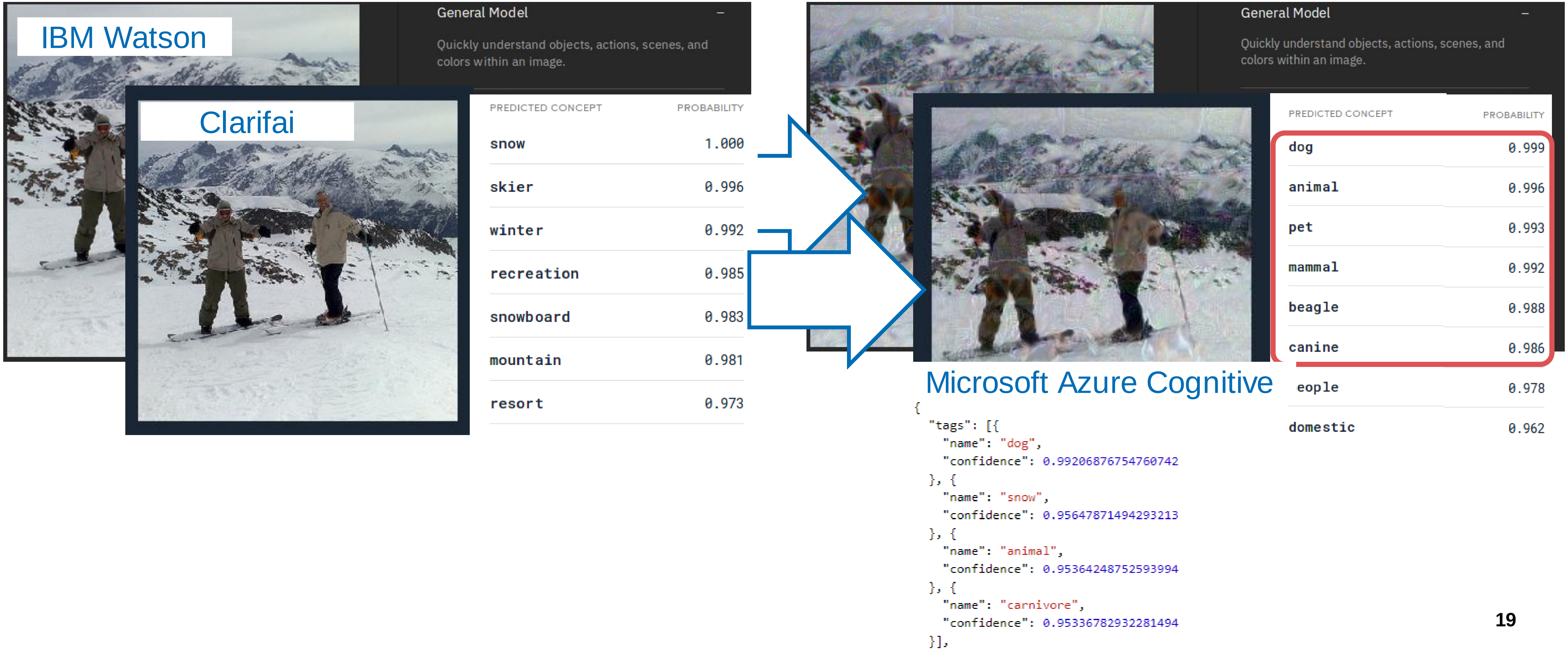


- Demo:

[1] Ilyas et al. *Black-box adversarial attacks with limited information and queries*. ICML'18.

Realistic APIs

Same PRISM / PRISM$_R$ examples transfer across all tested APIs



Conclusion

What can the adversary do to make targeted evasion more efficient while retaining effectiveness?

Combine availability of large ensembles + partial-information access to victim API (**PRISM**)

and

analyze and switch through different methods (**adversary agility**)

→ find adversarial examples **efficiently** and **effectively**

Mika Juuti
mika.juuti@uwaterloo.ca

Juuti et al. Making targeted black-box evasion attack effective and efficient. AISec'19. https://arxiv.org/abs/1906.03397