Improving Black-box Adversarial Attacks with a Transfer-based Prior

Shuyu Cheng*, Yinpeng Dong*, Tianyu Pang,

Hang Su, Jun Zhu

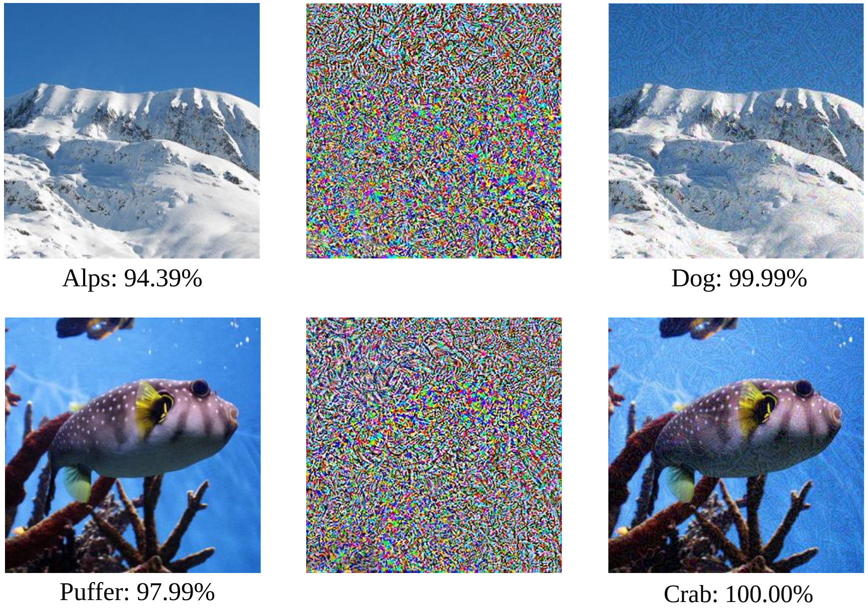
Dept. of Comp. Sci. and Tech., Tsinghua University

{chengsy18, dyp17, pty17}@mails.tsinghua.edu.cn,

{suhangss, dcszj}@mail.tsinghua.edu.cn

Background

- An adversarial example should be visually indistinguishable from the corresponding normal one, but yet are misclassified by the target model.



- **Adversarial attacks** find such examples.

Adversarial Attacks

- Goal: Given classifier $C(x)$ and input-label pair (x, y), find an adversarial example x^{adv} such that

$$C(x^{\text{adv}}) \neq y, \text{ s.t. } \|x^{\text{adv}} - x\|_p \leq \epsilon.$$

- x^{adv} can be generated by solving

$$x^{\text{adv}} = \underset{x':\|x'-x\|_p \leq \epsilon}{\arg\max} \; f(x', y)$$

- f is a loss function that we need to maximize in attacks. In untargeted attacks, it can be:
  - Cross entropy loss of the original label y
  - C&W loss $\max_{i \neq y} Z(x)_i - Z(x)_y$, $Z(x)$ is the logit
    - 0-surface is the decision boundary

White-box Attacks

- Projected gradient ascent (PGD)

$$x_{t+1}^{\mathrm{adv}} = \Pi_{B_p(x,\epsilon)}(x_t^{\mathrm{adv}} + \eta \cdot g_t)$$

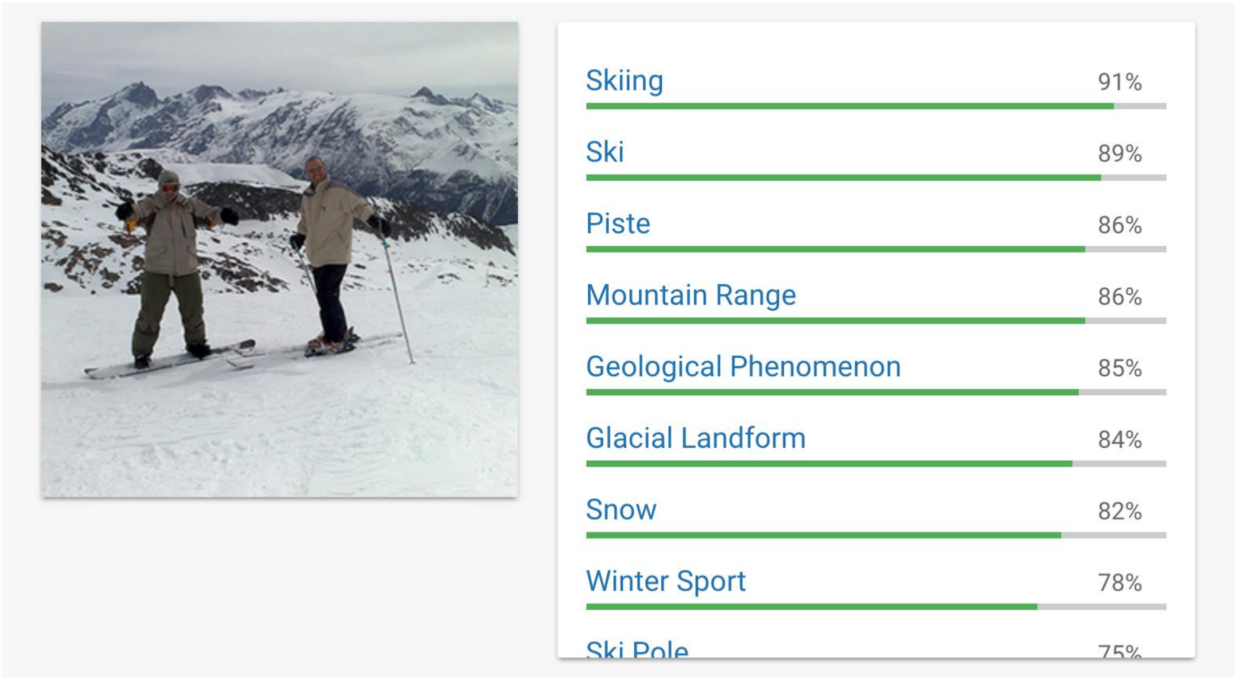
  - Π is the projection operation
  - $B_p(x,\epsilon)$ is the ℓ_p ball centered at x with radius ϵ
  - g_t is the normalized gradient under the ℓ_p norm
    - $p = 2$: $g_t = \frac{\nabla_x f(x_t^{\mathrm{adv}}, y)}{\|\nabla_x f(x_t^{\mathrm{adv}}, y)\|_2}$
    - $p = \infty$: $g_t = \mathrm{sign}(\nabla_x f(x_t^{\mathrm{adv}}, y))$
- Key: We need to know $\nabla_x f(x_t^{\mathrm{adv}}, y)$
  - In the following part, we omit the dependency w.r.t. y, write the objective as $f(x)$ and write the gradient as $\nabla f(x)$.

Black-box Attacks

- Transfer-based
  - Generate adversarial examples against white-box models, and leverage **transferability** for attacks
  - Require no knowledge of the target model, no queries
  - Need white-box models (datasets), assumes similarity
- Query-based
  - Get some information from the target model directly, through queries
    - **Score-based**
    - Decision-based
  - Goal: Improve success rate (e.g., success rate under 10000 queries) and save queries

Score-based Attacks

- Query loss function $f(x)$ given x



- We need to maximize $f(x)$ until attack succeeds.
- Gradient-based method: **Estimate** $\nabla f(x)$ **by queries**, and apply first-order optimization methods.

Random Gradient-Free (RGF) Method

- $\hat{g} = \frac{1}{q}\sum_{i=1}^{q} \hat{g}_i$, where $\hat{g}_i = \frac{f(x+\sigma u_i)-f(x)}{\sigma} \cdot u_i$
- $\{u_i\}_{i=1}^{q}$ are i.i.d. r.v. sampled from a distribution on R^D.
- In ordinary RGF method, u_i is sampled uniformly from the D-dimensional Euclidean hypersphere.
- $\hat{g}_i \approx u_i u_i^\top \nabla f(x)$ [1]
- Pros: Unbiased
- Cons: High variance
- How to improve: Incorporating informative priors
- Evaluation metric / Loss function: Something like MSE?

1. Assume f is differentiable and $\sigma \to 0$.

Gradient estimation framework

- Suppose we want to choose a best estimator in the set G of all possible gradient estimators, so we want to design a loss function for a gradient estimator.
- Our loss function for $\hat{g}$:

$$L(\hat{g}) = \min_{b \geq 0} \mathbb{E}\|\nabla f(x) - b\hat{g}\|_2^2$$

- Minimized mean square error w.r.t. the scale coefficient b
  - Usually the normalized gradient is used, hence the norm does not matter

Application to the RGF estimator

- For example, when $\hat{g}$ is an RGF estimator with u_i i.i.d. sampled from any distribution on the hypersphere:
- **Theorem 1.** Suppose $\|u_i\|_2 = 1$ in the RGF method. If f is differentiable at x, the loss of the RGF estimator $\hat{g}$ is

$$\lim_{\sigma \to 0} L(\hat{g}) =$$

$$\|\nabla f(x)\|_2^2 - \frac{\left(\nabla f(x)^\top \mathbf{C} \nabla f(x)\right)^2}{\left(1 - \frac{1}{q}\right) \nabla f(x)^\top \mathbf{C}^2 \nabla f(x) + \frac{1}{q} \nabla f(x)^\top \mathbf{C} \nabla f(x)}$$

where $\mathbf{C} = \mathbb{E}[u_i u_i^\top]$.

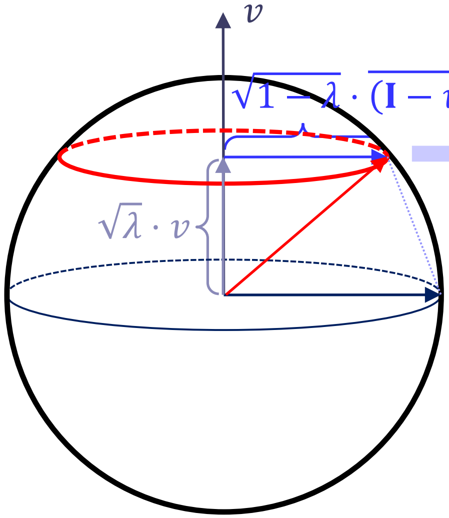
Prior-guided RGF (P-RGF) method

- For the ordinary RGF estimator, $\mathbf{C} = \frac{\mathbf{I}}{D}$. Any better one?
- Suppose we know v, the normalized ($\|v\|_2 = 1$) transfer gradient of a surrogated model. Then we can design $\mathbf{C}$ as

$$\mathbf{C} = \lambda v v^\top + \frac{1-\lambda}{D-1}(\mathbf{I} - v v^\top)$$

- which can be implemented by $u_i = \sqrt{\lambda} \cdot v + \sqrt{1-\lambda} \cdot \overline{(\mathbf{I} - v v^\top)\xi_i}$, where ξ_i is sampled uniformly from the unit hypersphere.

Prior-guided RGF (P-RGF) method



u_i is uniformly sampled from this red ring

$\overline{(\mathbf{I}-vv^\top)\xi_i}$ is uniformly sampled from the navy ring

Note: $(\mathbf{I}-vv^\top)$ is a projection matrix that remove the component of v

$$u_i = \sqrt{\lambda}\cdot v + \sqrt{1-\lambda}\cdot\overline{(\mathbf{I}-vv^\top)\xi_i}$$

$$\mathbb{E}[u_i u_i^\top] = \lambda vv^\top + \frac{1-\lambda}{D-1}(\mathbf{I}-vv^\top)$$

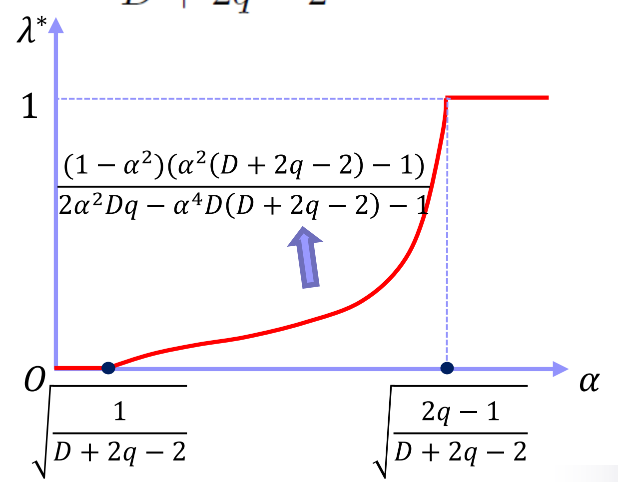
- $\lambda = \frac{1}{D} \approx 0$ is ordinary RGF estimator: **unbiased, high variance**
- $\lambda = 1$ corresponds to $u_i = v$, i.e. directly using the transfer gradient without queries: **highly biased, no variance**
- We need to find the optimal λ.

Solving for the optimal λ

- Let $\alpha = v^\top \overline{\nabla f(x)}$ where $\overline{\nabla f(x)}$ is the l_2 normalization of the true gradient $\nabla f(x)$.
  - α denotes the usefulness of the prior v
- By our gradient estimation framework, the optimal λ is

$$\lambda^* = \begin{cases} 0 & \text{if } \alpha^2 \leq \dfrac{1}{D+2q-2} \\ \dfrac{(1-\alpha^2)(\alpha^2(D+2q-2)-1)}{2\alpha^2 Dq - \alpha^4 D(D+2q-2) - 1} & \text{if } \dfrac{1}{D+2q-2} < \alpha^2 < \dfrac{2q-1}{D+2q-2} \\ 1 & \text{if } \alpha^2 \geq \dfrac{2q-1}{D+2q-2} \end{cases}$$

solved by minimizing $L(\hat{g})$.



Estimating α

- The ground truth value of $\alpha = \overline{\nabla f(x)}^\top v$ is not accessible, which needs an estimation[1].
- Note that $\alpha = \frac{\nabla f(x)^\top v}{\|\nabla f(x)\|_2}$. The numerator is easy to estimate by finite difference. Hence the key problem is to estimate $\|\nabla f(x)\|_2$.
  - Finite difference: $\nabla f(x)^\top v \approx \frac{f(x+\sigma v)-f(x)}{\sigma}$.
- <u>Good thing: A scalar is much easier to estimate than a vector!</u>

1. $\overline{x}$ denotes the ℓ_2 normalization of x in this work.

Norm estimation: The framework

- Suppose by S queries, we can get $\nabla f(x)^T w_1, \dots, \nabla f(x)^T w_S$ by finite difference, and $\|w_i\| = 1$.
- If we have a S-variable function g such that
$$g(ax_1, ax_2, \dots, ax_n) = a^r g(x_1, x_2, \dots, x_n)$$
- Then
$$g(w_1^\top \nabla f(x), \dots, w_S^\top \nabla f(x))$$
$$= \|\nabla f(x)\|_2^r \cdot g\left(w_1^\top \overline{\nabla f(x)}, \dots, w_S^\top \overline{\nabla f(x)}\right)$$
- Hence

Each $w_s^\top \nabla f(x)$ can be estimated by finite difference!

$$\frac{g\left(w_1^\top \nabla f(x), \dots, w_S^\top \nabla f(x)\right)}{\mathbb{E}\left[g\left(w_1^\top \overline{\nabla f(x)}, \dots, w_S^\top \overline{\nabla f(x)}\right)\right]}$$

The expectation can be computed when each w_s is uniformly distributed on the sphere!

is an unbiased estimator of $\|\nabla f(x)\|_2^r$.

Norm estimation

- Here, we choose

$$g(z_1, z_2, \ldots, z_S) = \frac{1}{S}\sum_{s=1}^{S} z_s^2$$

when $r = 2$.

- Then the estimator of $\|\nabla f(x)\|_2$ is

$$\|\nabla f(x)\|_2 \approx \sqrt{\frac{D}{S}\sum_{s=1}^{S}(w_s^\top \nabla f(x))^2}$$

where $\{w_s\}_{s=1}^{S}$ is i.i.d. uniformly sampled from the unit hypersphere.

Summary of the P-RGF method

Algorithm 1 Prior-guided random gradient-free (P-RGF) method

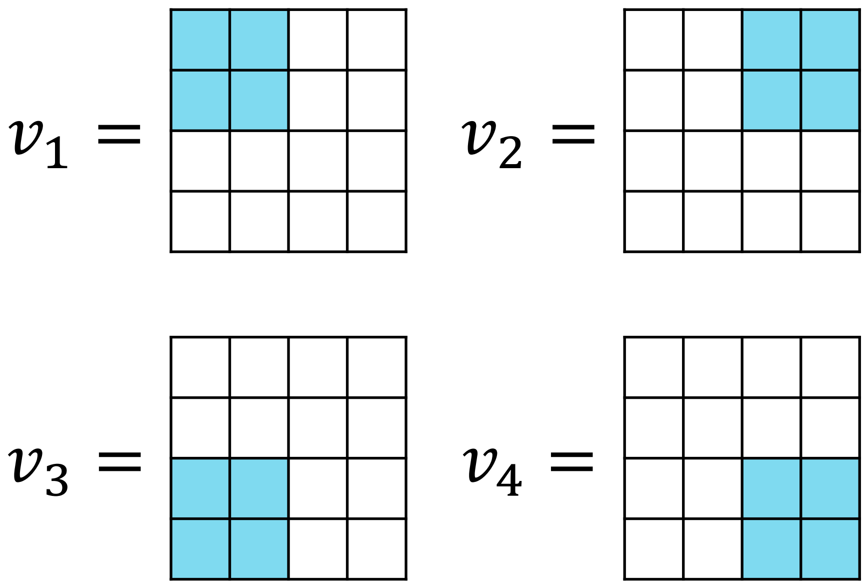
Input: The black-box model f; input x and label y; the normalized transfer gradient v; sampling variance σ; number of queries q; input dimension D.

Output: Estimate of the gradient $\nabla f(x)$.

```
Estimate the cosine similarity α = v^⊤ \overline{∇f(x)} (detailed in Sec. 3.3);
Calculate λ* according to Eq. (12) given α, q, and D;
if λ* = 1 then
    return v;
end if
ĝ ← 0;
for i = 1 to q do
    Sample ξ_i from the uniform distribution on the D-dimensional unit hypersphere;
    u_i = √λ* · v + √(1 − λ*) · \overline{(I − vv^⊤)ξ_i};
    ĝ ← ĝ + (f(x + σu_i, y) − f(x, y)) / σ · u_i;
end for
return ∇f(x) ← (1/q) ĝ.
```

Incorporating data-dependent prior

- Restrict the adversarial perturbations to lie in a d-dimensional linear subspace spanned by $\{v_1, v_2, \dots, v_d\}$
- For example, for $4 \times 4 \times 1$ images, $D = 16, d = 4$, we choose the subspace to be "in lower resolution":



Incorporating data-dependent prior

- To perform the RGF method incorporating data-dependent prior, we need to set

$$\mathbf{C} = \frac{1}{d}\sum_{i=1}^{d} v_i v_i^\top$$

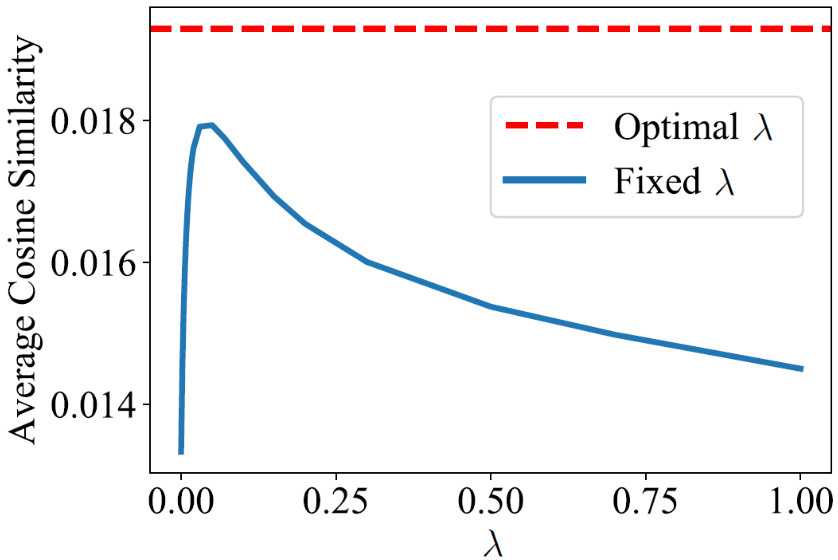
- To further incorporate the transfer-based prior, we can set

$$\mathbf{C} = \lambda v v^\top + \frac{1-\lambda}{d}\sum_{i=1}^{d} v_i v_i^\top$$

- Similarly we can obtain the optimal λ.

Performance of gradient estimation

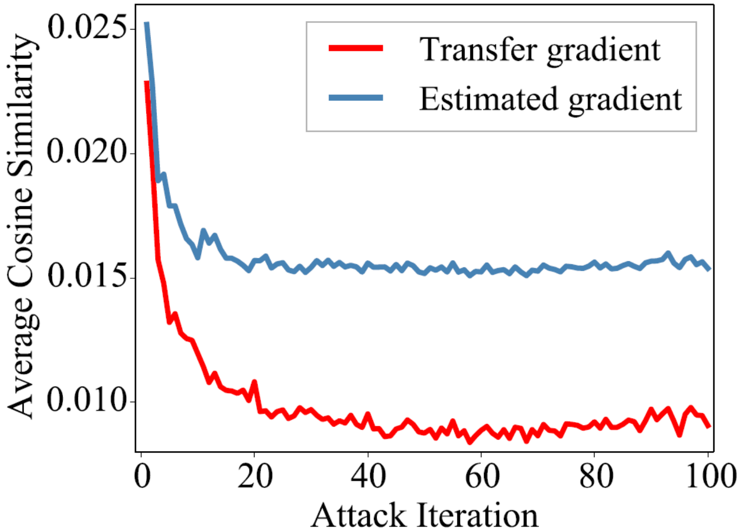
- Average cosine similarity between the gradient estimate and the true gradient:



- which shows the effectiveness of the derived optimal λ (i.e., λ^*) for gradient estimation compared with any fixed $\lambda \in [0,1]$

Performance of gradient estimation

- Cosine similarity (averaged over all images) between the gradient estimate and the true gradient w.r.t. attack iterations:



- The transfer gradient is more useful at the beginning and less useful later
  - Showing the advantage of using adaptive λ^*

Gradient averaging

- Alternative method of biased sampling
  - Also integrate the transfer-based prior into the query-based algorithm

$$\hat{g} = (1-\mu)v + \mu\overline{\hat{g}^U}$$

  - $\overline{\hat{g}^U}$ is the normalized **ordinary** RGF estimator with $\mathbf{C} = \frac{\mathbf{I}}{d}$
  - The optimal coefficient μ^* can be derived by the gradient estimation framework too.

Results of black-box attacks on normal models

Methods	Inception-v3		VGG-16		ResNet-50	
	ASR	AVG. Q	ASR	AVG. Q	ASR	AVG. Q
NES	95.5%	1718	98.7%	1081	98.4%	969
$\text{Bandits}_{\text{T}}$	92.4%	1560	94.0%	584	96.2%	1076
$\text{Bandits}_{\text{TD}}$	97.2%	874	94.9%	278	96.8%	512
AutoZoom	85.4%	2443	96.2%	1589	94.8%	2065
RGF	97.7%	1309	99.8%	935	99.5%	809
P-RGF ($\lambda = 0.5$)	96.5%	1119	97.3%	1075	98.3%	990
P-RGF (λ^*)	**98.1%**	745	**99.8%**	521	**99.6%**	452
Averaging ($\mu = 0.5$)	96.9%	1140	94.6%	2143	96.3%	2257
Averaging (μ^*)	97.9%	**735**	**99.8%**	**516**	99.5%	**446**
RGF_{D}	99.1%	910	**100.0%**	464	**99.8%**	521
P-RGF_{D} ($\lambda = 0.5$)	98.2%	1047	99.3%	917	99.3%	893
P-RGF_{D} (λ^*)	99.1%	649	99.7%	370	99.6%	**352**
$\text{Averaging}_{\text{D}}$ ($\mu = 0.5$)	**99.2%**	768	99.9%	900	99.2%	1177
$\text{Averaging}_{\text{D}}$ (μ^*)	**99.2%**	**644**	99.8%	**366**	99.5%	355

- ASR: Attack Success Rate (#queries is under 10,000); AVG. Q: Average #queries over successful attacks.
- Methods with the subscript "D" refers to the data-dependent version of the P-RGF method.

Results on defensive models

Methods	JPEG Compression		Randomization		Guided Denoiser	
	ASR	AVG. Q	ASR	AVG. Q	ASR	AVG. Q
NES	47.3%	3114	23.2%	3632	48.0%	3633
SPSA	40.0%	2744	9.6%	3256	46.0%	3526
RGF	41.5%	3126	19.5%	3259	50.3%	3569
P-RGF	61.4%	2419	60.4%	2153	51.4%	2858
Averaging	**69.4%**	**2134**	**72.8%**	**1739**	**66.6%**	**2441**
RGF_D	70.4%	2828	54.9%	2819	83.7%	2230
P-RGF_D	**81.1%**	2120	**82.3%**	1816	**89.6%**	1784
$\text{Averaging}_\text{D}$	80.6%	**2087**	77.4%	**1700**	87.2%	**1777**

- ASR: Attack Success Rate (#queries is under 10,000); AVG. Q: Average #queries over successful attacks.
- Methods with the subscript “D” refers to the data-dependent version of the P-RGF method.

Thanks!