An Intermediate-Level Attack Framework on the Basis of Linear Regression

Yiwen Guo[ID], Qizhang Li[ID], Wangmeng Zuo[ID], *Senior Member, IEEE*, and Hao Chen[ID]

Abstract—This article substantially extends our work published at ECCV (Li *et al.*, 2020), in which an intermediate-level attack was proposed to improve the transferability of some baseline adversarial examples. Specifically, we advocate a framework in which a direct linear mapping from the intermediate-level discrepancies (between adversarial features and benign features) to prediction loss of the adversarial example is established. By delving deep into the core components of such a framework, we show that a variety of linear regression models can all be considered in order to establish the mapping, the magnitude of the finally obtained intermediate-level adversarial discrepancy is correlated with the transferability, and further boost of the performance can be achieved by performing multiple runs of the baseline attack with random initialization. In addition, by leveraging these findings, we achieve new state-of-the-arts on transfer-based ℓ_∞ and ℓ_2 attacks. Our code is publicly available at https://github.com/qizhangli/ila-plus-plus-lr.

Index Terms—Deep neural networks, adversarial examples, adversarial transferability, generalization ability, robustness

✦

1 INTRODUCTION

T HE community has witnessed a great surge of studies on adversarial examples. It has been shown that, even in a black-box setting where no information about the victim machine learning model was available, an attacker is still capable of generating adversarial examples with reasonably high attack success rates. In general, adversarial attacks in black-box settings are performed via zeroth-order optimization [2], [3], [4], [5], [6], [7], or applying boundary search [8], [9], [10], [11], or based on the transferability of adversarial examples [1], [12], [13], [14], [15], [16], [17].

The transferability of adversarial examples has drawn great interest from researchers, owing to its essential role in generating a variety of black-box adversarial examples. The phenomenon shows that adversarial examples generated on one classification model (i.e., a source model) can successfully fool other classification models (i.e., the victim models) which probably have different architectures and different parameters. Much recent work has been published to shed light on the transferability of adversarial examples and to develop ways of improving it [1], [13], [14], [15], [16], [17].

- *Yiwen Guo is with the ByteDance AI Lab, Beijing 100084, China. E-mail: guoyiwen89@gmail.com.*
- *Qizhang Li is with the Tencent Security Big Data Lab, Shenzhen, Guang-dong 518010, China, and also with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China. E-mail: liqizhang95@gmail.com.*
- *Wangmeng Zuo is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China. E-mail: cswmzuo@gmail.com.*
- *Hao Chen is with the Department of Computer Science, University of California, Davis, CA 95616 USA. E-mail: chen@ucdavis.edu.*

In this paper, we also take a step towards explaining and exploiting the adversarial transferability.

We focus on adversarial examples crafted on the basis of images, which are used for attacking image classification systems or other related computer vision systems. We consider the scenario in which the attacker being given a normal image classification source model $f : \mathbb{R}^n \times \mathbb{C} \to \mathbb{R}$, which is a deep neural network (DNN) that maps any input image $\mathbf{x} \in \mathbb{R}^n$ along with its ground-truth label $y \in \mathbb{C}$ to evaluation loss, as a composition of a feature extractor g and a classifier h that evaluates the prediction made based on the extracted features, i.e., $f = h \circ g$ [1], [14], [17]. In general, g and h can be considered as two sub-nets of the original DNN model, consisting of its lower and higher layers, respectively. Note that g does not necessarily contains more layers than that of h, and we make the depth of g a hyperparameter that can be tuned.

In this paper, we concern a problem setting (illustrated in Fig. 1) in which some initial adversarial examples have already been crafted following a baseline method (e.g., I-FGSM) on a white-box source model, and we aim to enhance their transferability to unknown victim models. To achieve this, we advocate a framework where a linear mapping (denoted by h') is adopted instead of the original nonlinear one, i.e., h, for evaluating the middle-layer features extracted by g which is believed to be largely shared among different DNN models. Results of the baseline method are utilized to establish the linear mapping h', and the more transferable adversarial example is anticipated by maximizing the output of $f' = h' \circ g$.

What's New in Comparison to [1]. 1) By introducing the framework based on linear regression, we show that various models can be applied for obtaining the linear function h', including ridge regression, support vector regression [18], and ElasticNet [19]. 2) We provide ample evidence of why (and how) improved adversarial transferability are achieved in such a framework. 3) We show how to further boost the performance of adversarial transferability, in addition to the results in [1]. 4) New state-of-the-arts are achieved.
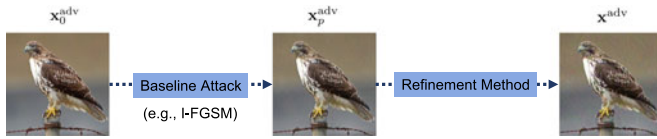
Fig. 1. An overview of the problem setting in [1], [14], and this article, in which a baseline attack was performed in advance.

2 RELATED WORK

The transferability of adversarial examples has been intensively studied, since it was first discovered [20]. Ensemble learning [13], [21] and random augmentation [22] are two popular strategies for improving the transferability. A series of recent methods also exploit the architecture of advanced DNNs. For instance, Zhou *et al.* [15] proposed TAP, in which the discrepancy between benign DNN features and their adversarial counterparts was maximized. Similarly, Inkawhich *et al.* [23] encouraged the intermediate-level representation of an adversarial example to be similar to that of a target example. Wu *et al.* [24] showed that, for models with skip connections, propagating less gradient through the main stream of a DNN led to more transferable adversarial examples. Guo *et al.* [17] advocated to perform backpropagation without caring about ReLU during backward pass. Other effective methods include [25], [26], [27], [28].

The most relevant work to ours is introduced by Huang *et al.* [29], since we both follow the problem setting in Fig. 1, and we will introduce and discuss it carefully in Section 3.1.

3 LINEARIZATION AND TRANSFERABILITY

It was originally proposed in our paper [1] that the transferability of an off-the-shelf multi-step attack (e.g., the iterative fast gradient sign method, I-FGSM [30]) can be enhanced by encouraging middle-layer features of the source model to be adversarial, in the sense of maximizing ridge regression loss. The work takes an attempt towards connecting the transferability and linear regression. In what follows, we will first introduce the method and then give more discussions and insights in an unified framework.

The concerned problem setting throughout this paper is that, some initial adversarial examples have already been crafted using a baseline method like I-FGSM, and we aim to develop a method to refine the adversarial examples and enhance their transferability across a set of unseen victim models. See Fig. 1. The transferability is experimentally assessed by evaluating the attack success rates on a variety of victim models, using adversarial examples generated on a single source model.[1]

3.1 Intermediate-Level Attacks

Suppose that I-FGSM [30] has already been performed on a source model as a baseline attack, as has been illustrated in Fig. 1, the goal of our work is to enhance the transferability of the I-FGSM results.

Given a benign example $\mathbf{x}$, the update rule of I-FGSM is: for the t-th step,

$$\mathbf{x}_{t+1}^{\mathrm{adv}} = \Pi_\Psi(\mathbf{x}_t^{\mathrm{adv}} + \alpha \cdot \mathrm{sgn}(\nabla f(\mathbf{x}_t^{\mathrm{adv}}, y))), \quad (1)$$

in which $\alpha > 0$ is the attack step size, $\mathbf{x}_0^{\mathrm{adv}} = \mathbf{x}$, and $\Pi_\Psi(\cdot)$ projects its input onto the set of valid images (i.e., Ψ). If in total p steps of the I-FGSM update were performed, we could collect p temporary results $\mathbf{x}_0^{\mathrm{adv}}, \ldots, \mathbf{x}_t^{\mathrm{adv}}, \ldots, \mathbf{x}_{p-1}^{\mathrm{adv}}$ together with the final baseline result $\mathbf{x}_p^{\mathrm{adv}}$. We also collected the value of their corresponding prediction loss (i.e., the cross-entropy loss evaluating their prediction correctness), denoted by $l_0, \ldots, l_t, \ldots, l_p$. The middle-layer features are represented as $\mathbf{h}_u^{\mathrm{adv}} = g(\mathbf{x}_u^{\mathrm{adv}})$, for $u = 0, \ldots, t, \ldots, p$.

To achieve an adversarial example with improved transferability on the basis of I-FGSM [30], we proposed [1] to solve

$$\max_{\mathbf{\Delta_x}} (g(\mathbf{x} + \mathbf{\Delta_x}) - \mathbf{h}_0^{\mathrm{adv}})^T \mathbf{w}^*, \quad \mathrm{s.t.} \ (\mathbf{x} + \mathbf{\Delta_x}) \in \Psi. \quad (2)$$

In [1], $\mathbf{w}^* = (\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I}_m)^{-1}\mathbf{H}^T\mathbf{r}$ was obtained from a closed-form solution of a ridge regression problem, in which the t-th row of $\mathbf{H} \in \mathbb{R}^{(p+1) \times m}$ was $(\mathbf{h}_t^{\mathrm{adv}} - \mathbf{h}_0^{\mathrm{adv}})^T$ and the t-th entry of $\mathbf{r} \in \mathbb{R}^{p+1}$ was l_t. We showed in that paper that an approximation to $\mathbf{w}^*$ could be obtained using $\mathbf{w}^* \approx \mathbf{H}^T\mathbf{r}$, in order to gain higher computational efficiency (see Appendix A for a brief introduction of the approximation). That is, we can turn to solve

$$\max_{\mathbf{\Delta}_x} (g(\mathbf{x} + \mathbf{\Delta_x}) - \mathbf{h}_0^{\mathrm{adv}})^T \mathbf{H}^T\mathbf{r}, \quad \mathrm{s.t.} \ (\mathbf{x} + \mathbf{\Delta_x}) \in \Psi. \quad (3)$$

In a related paper, Huang *et al.* [14] also proposed to operate on the middle-layer features of the source model for gaining higher transferability, and their work inspired ours. Specifically, they proposed to solve

$$\max_{\mathbf{\Delta}_x} (g(\mathbf{x} + \mathbf{\Delta_x}) - \mathbf{h}_0^{\mathrm{adv}})^T (\mathbf{h}_p^{\mathrm{adv}} - \mathbf{h}_0^{\mathrm{adv}}),$$
$$\mathrm{s.t.} \ (\mathbf{x} + \mathbf{\Delta_x}) \in \Psi. \quad (4)$$

Apparently, their formulation in Eq. (4) can be regarded as a special case of ours in Eq. (3), in a single-step setting of I-FGSM with $p = 1$. Since both Huang *et al.*'s method and our method operate on the intermediate-level representations of DNNs, we will call them intermediate-level attack (ILA) and ILA++, respectively, in this paper.

3.2 Linear Regression for Obtaining h'

Our ILA++ [1] established a direct mapping from the intermediate layer of a DNN to the prediction loss, such that how much classification loss the intermediate-level discrepancies shall lead to can be reasonably evaluated. It shows considerably better performance in comparison with the baseline attack and ILA (see Table 1 and other experimental results in [1]).

The effectiveness of ILA++ can be explained from different perspectives. 1) ILA++ substitutes the original nonlinear function h with a linear one, and it is likely that a more linear model leads to improved transferability of adversarial examples [17]. 2) the optimization problem of ILA++ aims at maximizing the projection on a directional guide $\mathbf{H}^T\mathbf{r}$ which is a linear combination of many intermediate-level discrepancies (i.e., "perturbations" of features) obtained from the iterative baseline attack. Such a directional guide,

1. Regardless the success rate on the source model, which is generally near 100%

TABLE 1
Comparison of ILA++ to ILA and the Baseline Attack, i.e., I-FGSM

Method	ϵ	ResNet-50*	VGG-19 [31]	ResNet-152 [32]	Inception v3 [33]	DenseNet [34]	MobileNet v2 [35]	SENet [36]	ResNeXt [37]	WRN [38]	PNASNet [39]	MNASNet [40]	Average
I-FGSM	16/255	100.00%	61.50%	52.82%	30.86%	57.36%	58.92%	38.12%	48.88%	48.92%	28.92%	57.20%	48.35%
	8/255	100.00%	38.90%	29.36%	15.36%	34.86%	37.66%	17.76%	26.30%	26.26%	13.04%	35.08%	27.46%
	4/255	100.00%	18.86%	11.28%	6.66%	15.44%	18.36%	5.72%	9.58%	9.98%	4.14%	17.02%	11.70%
I-FGSM+ILA	16/255	99.96%	93.62%	91.50%	73.10%	92.46%	91.86%	84.22%	89.94%	89.30%	77.72%	90.78%	87.45%
	8/255	99.96%	72.86%	66.80%	38.58%	68.18%	69.64%	50.14%	62.56%	62.46%	40.80%	68.16%	60.02%
	4/255	99.96%	38.90%	29.42%	13.48%	34.04%	36.22%	18.62%	26.36%	27.42%	12.82%	35.88%	27.32%
I-FGSM+RR (ILA++)	16/255	99.96%	**94.24%**	**92.22%**	75.26%	**93.22%**	**92.16%**	85.74%	**90.94%**	**90.20%**	79.96%	91.72%	88.57%
	8/255	99.96%	75.42%	70.50%	41.90%	71.74%	72.54%	53.76%	66.32%	66.30%	44.78%	70.56%	63.38%
	4/255	99.94%	**42.76%**	**33.14%**	15.90%	38.10%	39.76%	21.06%	30.28%	31.02%	15.20%	38.72%	30.59%

*Success rate of transfer-based attacks on ImageNet is reported, with ℓ_∞ constraint in the untargeted setting. We use the symbol * to indicate when the source model is used as the target. Average is obtained on models that are different from the source model.*

i.e., $\mathbf{H}^T\mathbf{r}$, is basically a weighted sum of some baseline intermediate-level discrepancies. It leads to higher prediction loss and is thus more powerful.

We study whether other choices of $\mathbf{w}^*$ improve adversarial transferability. Still, we focus on a framework in which the refined adversarial examples are obtained via solving Eq. (2) using off-the-shelf constrained optimization methods (e.g., FGSM [41], I-FGSM [30], PGD [42]) and input gradient of $f' = h' \circ g$, rather than input gradient of $f = h \circ g$. These optimization methods are variants of gradient ascent and they can be performed just like in their original settings, except with a different objective function, i.e., Eq. (2).

We first compare linear regression models for obtaining $\mathbf{w}^*$ and learning the relationship between intermediate-level discrepancies and prediction loss. In particular, three regression models are considered, including ridge regression (RR, also known as Tikhonov regularization) as introduced in Section 3.1, ElasticNet [19], and support vector regression (SVR) [18]:

$$\text{RR}: \min_{\mathbf{w}} \sum_{t=0}^{p} z_t^2 + \lambda\|\mathbf{w}\|_2^2,$$

$$\text{ElasticNet}: \min_{\mathbf{w}} \sum_{t=0}^{p} z_t^2 + \lambda_1\|\mathbf{w}\|_1 + \lambda_2\|\mathbf{w}\|_2^2, \qquad (5)$$

$$\text{SVR}: \min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 \quad \text{s.t.} \ |z_t - b| < e,$$

in which $z_t = \mathbf{w}^T(\mathbf{h}_t^{\text{adv}} - \mathbf{h}_0^{\text{adv}}) - l_t$, and b is set to zero.

By solving different optimization problems in Eq. (5), we can obtain different $\mathbf{w}^*$ for Eq. (2). Based on them, different refined adversarial examples can eventually be obtained by solving Eq. (2) then. Other sparse coding algorithms than ElasticNet [19] are not considered, since a sparse $\mathbf{w}^*$ in fact restricts possible directions of perturbations in the feature space, and thus lead to worse attack success rates even on the source model. We show more about this in Tables 6 and 7 by varying the hyper-parameter λ_1 in Eq. (5).

Setup. An experiment on ImageNet was setup to compare the three options in Eq. (5). Similar to other experiments, we used a pre-trained ResNet-50 [32] as the source model and evaluated attack success rates of the generated adversarial examples on a variety of victim models. In our framework, a baseline attack I-FGSM was performed in advance, and then the obtained baseline result was enhanced by solving Eq. (2) using off-the-shelf methods (here, I-FGSM again), in which $\mathbf{w}^*$ was obtained from any of the regression models. The step size of I-FGSM is set to be $1/255$ and $\epsilon/5$ for ℓ_∞ and ℓ_2 attacks, respectively. We run 100 iterations for I-FGSM and we let $p = 10$ for attacks using temporary results from the baseline attack. See Section 5 for details.

Results. Tables 2 and 3 summarize the attack success rates using different linear regression models, in ℓ_∞ and ℓ_2 settings, respectively. It can be seen that all the three linear regression methods led to improved attack transferability. Tables 6 and 7 report the performance of I-FGSM+ElasticNet with λ_1 ranging from 0.05 to 1.0, and, in Tables 2 and 3, we provide detailed results with $\lambda_1 = 0.05$. Apparently, smaller λ_1 leads to higher success rates using I-FGSM+ElasticNet, yet we did not test with further smaller λ_1, since solution to the ElasticNet optimization became inaccurate, as suggested by the Python `sklearn` package. As $\lambda_1 \to 0$,

TABLE 2
Comparison of Different Linear Regression Models in ℓ_∞ Attack Settings

Method	ϵ	ResNet-50*	VGG-19 [31]	ResNet-152 [32]	Inception v3 [33]	DenseNet [34]	MobileNet v2 [35]	SENet [36]	ResNeXt [37]	WRN [38]	PNASNet [39]	MNASNet [40]	Average
I-FGSM	16/255	100.00%	61.50%	52.82%	30.86%	57.36%	58.92%	38.12%	48.88%	48.92%	28.92%	57.20%	48.35%
	8/255	100.00%	38.90%	29.36%	15.36%	34.86%	37.66%	17.76%	26.30%	26.26%	13.04%	35.08%	27.46%
	4/255	100.00%	18.86%	11.28%	6.66%	15.44%	18.36%	5.72%	9.58%	9.98%	4.14%	17.02%	11.70%
I-FGSM+RR	16/255	99.96%	**94.24%**	92.22%	75.26%	**93.22%**	**92.16%**	**85.74%**	**90.94%**	**90.20%**	79.96%	91.72%	88.57%
	8/255	99.96%	75.42%	70.50%	41.90%	71.74%	72.54%	53.76%	66.32%	66.30%	44.78%	70.56%	63.38%
	4/255	99.94%	42.76%	33.14%	15.90%	38.10%	39.76%	21.06%	30.28%	31.02%	15.20%	38.72%	30.59%
I-FGSM+ElasticNet	16/255	99.90%	89.76%	84.82%	63.76%	86.48%	86.84%	76.58%	82.84%	82.72%	67.14%	86.68%	80.76%
	8/255	99.92%	64.62%	54.44%	29.70%	57.46%	60.30%	41.08%	51.06%	51.22%	30.88%	59.94%	50.07%
	4/255	99.90%	31.66%	21.64%	10.62%	26.02%	29.14%	13.58%	19.40%	20.52%	9.40%	28.50%	21.05%
I-FGSM+SVR	16/255	99.96%	94.18%	**92.24%**	**76.18%**	93.18%	92.06%	85.70%	90.92%	89.96%	**80.02%**	**92.10%**	**88.65%**
	8/255	99.96%	**75.96%**	**71.00%**	**42.96%**	72.16%	**72.94%**	**54.28%**	**66.86%**	**66.90%**	**45.52%**	**71.52%**	**64.01%**
	4/255	99.92%	**43.80%**	**34.64%**	**16.50%**	38.24%	41.38%	21.88%	31.70%	32.32%	15.86%	39.62%	31.59%

*The success rate of transfer-based attacks on ImageNet is evaluated, with different ℓ_∞ constraints in the untargeted setting. We use the symbol * to indicate when the source model is used as the target. Average is calculated on the models different from the source model.*

TABLE 3
Comparison of Different Linear Regression Models in ℓ_2 Attack Settings

Method	ϵ	ResNet-50*	VGG-19 [31]	ResNet-152 [32]	Inception v3 [33]	DenseNet [34]	MobileNet v2 [35]	SENet [36]	ResNeXt [37]	WRN [38]	PNASNet [39]	MNASNet [40]	Average
I-FGSM	15	99.96%	78.64%	74.94%	50.02%	76.36%	75.20%	60.18%	71.16%	71.20%	51.20%	73.80%	68.27%
	10	99.94%	64.48%	58.60%	33.64%	59.94%	60.92%	42.20%	53.98%	53.76%	33.94%	59.12%	52.06%
	5	99.98%	36.96%	28.12%	14.68%	31.70%	34.94%	18.02%	25.40%	25.82%	12.92%	33.08%	26.16%
I-FGSM+RR	15	99.92%	93.82%	**91.32%**	75.16%	92.32%	**91.74%**	83.28%	**89.56%**	89.28%	**78.98%**	91.14%	87.66%
	10	99.90%	84.42%	80.96%	56.72%	81.54%	**82.08%**	66.58%	77.92%	77.26%	59.28%	**81.06%**	74.78%
	5	99.90%	59.00%	51.76%	27.42%	54.06%	55.52%	35.76%	47.58%	47.82%	29.84%	53.82%	46.26%
I-FGSM +ElasticNet	15	99.92%	87.42%	82.22%	61.58%	84.42%	85.08%	71.16%	79.46%	79.34%	63.68%	84.60%	77.90%
	10	99.88%	75.48%	66.82%	42.96%	69.08%	71.44%	52.42%	63.18%	63.60%	43.90%	71.14%	62.00%
	5	99.74%	46.84%	38.38%	20.70%	41.66%	44.72%	26.14%	34.52%	35.22%	20.78%	43.46%	35.24%
I-FGSM+SVR	15	99.92%	**94.06%**	91.22%	**76.16%**	92.38%	**91.74%**	83.48%	89.42%	**89.44%**	78.94%	**91.24%**	87.81%
	10	99.90%	**84.98%**	**81.64%**	**58.36%**	**82.28%**	81.90%	**66.74%**	**78.40%**	**77.88%**	**59.94%**	80.96%	75.31%
	5	99.90%	**59.08%**	**53.16%**	**29.06%**	**55.08%**	**56.26%**	**36.70%**	**48.30%**	**48.80%**	**30.88%**	**55.16%**	47.25%

*The success rate of transfer-based attacks on ImageNet is evaluated, with different ℓ_2 constraints in the untargeted setting. We use the symbol * to indicate when the source model is used as the target. Average is calculated on the models different from the source model.*

ElasticNet boils down to ridge regression, and we shall observe similar performance on I-FGSM+RR and I-FGSM +ElasticNet when λ_1 approaches 0. λ and C were set to $1e10$ and $1e-10$, respectively, for obtaining the I-FGSM+RR and I-FGSM+SVR results in Tables 2 and 3. Note that C is inversely proportional to the strength of the SVR regularization, and λ regularizes RR. How λ and C affect the performance of I-FGSM+RR and I-FGSM+SVR will be discussed in the following subsection.

3.3 Maximizing Feature Distortion

Solving the optimization problem in Eq. (2) indicates that: 1) maximizing the middle-layer distortion (i.e., the magnitude of the intermediate-level feature discrepancy $g(\mathbf{x} + \Delta_\mathbf{x}) - \mathbf{h}_0^{\mathrm{adv}}$) invoked by the adversarial example $\mathbf{x} + \Delta_\mathbf{x}$, and 2) maximizing the cosine similarity between $g(\mathbf{x} + \Delta_\mathbf{x}) - \mathbf{h}_0^{\mathrm{adv}}$ and a reasonably chosen directional guide (i.e., $\mathbf{w}^*/\|\mathbf{w}^*\|$). It is of yet unclear whether both the two factors are essential for improving the transferability. In this subsection and the following subsection, we aim to shed more light on it. We will first show how the magnitude of the intermediate-level discrepancy (i.e., $\|g(\mathbf{x} + \Delta_\mathbf{x}) - \mathbf{h}_0^{\mathrm{adv}}\|$) affects transferability of the refined adversarial examples and attack success rates.

We show in Fig. 3 the magnitude of intermediate-level discrepancies obtained using the four methods (i.e., I-FGSM, I-FGSM+RR, I-FGSM+ElasticNet, and I-FGSM+SVR), together with the obtained average attack success rates. The error bar in each subplot has been uniformly scaled to make the illustration clearer, considering that both the magnitude and the attack success rate vary drastically across victim models. As can be observed, the magnitude of intermediate-level discrepancies is positively correlated with the average success rate and the transferability.

Some of the hyper-parameters in fact make considerable impact on the intermediate-level distortions, e.g., λ in the formulation of ridge regression, λ_1 and λ_2 in ElasticNet, and C in SVR. We have discussed λ_1 in Tables 6 and 7, and we know the other regularization hyper-parameter (i.e., λ_2) in ElasticNet plays the same role as λ in ridge regression. Therefore, here we mostly report the results of I-FGSM+RR and I-FGSM+SVR by varying λ and C, respectively. We can observe in Fig. 4 that, when larger intermediate-level distortions were obtained (by tuning the hyper-parameters), higher attack success rates were obtained, which also shows the positive correlation between the transferability of adversarial examples and the magnitude of the intermediate-level discrepancies. Yet, Fig. 4 also shows that sensitivity of the attack performance to the hyper-parameters (λ and C) only exists in a limited region, indicating that the power of $\mathbf{w}^*$ for achieving large distortion and high transferability does not always change consistently with λ or C.
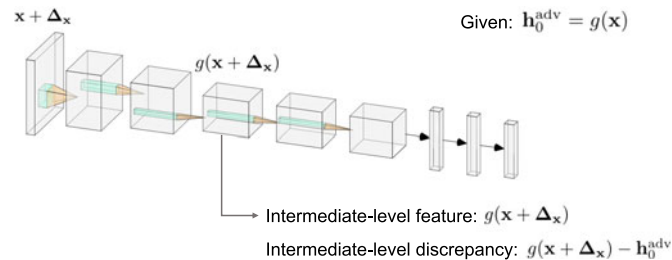

Fig. 2. Illustration of how the intermediate-level discrepancy is calculated given an adversarial input $\mathbf{x} + \Delta_\mathbf{x}$ and its benign counterpart $\mathbf{x}$. We will discuss how 1) the magnitude of the intermediate-level discrepancy (i.e., $\|g(\mathbf{x} + \Delta_\mathbf{x}) - \mathbf{h}_0^{\mathrm{adv}}\|$) and 2) the directional guide for optimizing $\Delta_\mathbf{x}$ would affect the obtained transferability.
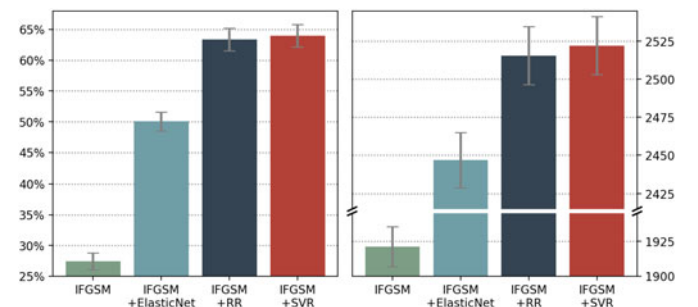

Fig. 3. On the left is the average success rate of the four methods, and on the right the average ℓ_2 magnitude of the intermediate-level discrepancies. Apparently, they are positively correlated, and larger intermediate-level distortions lead to higher attack success rates on the victim models.

(a) λ in I-FGSM+RR
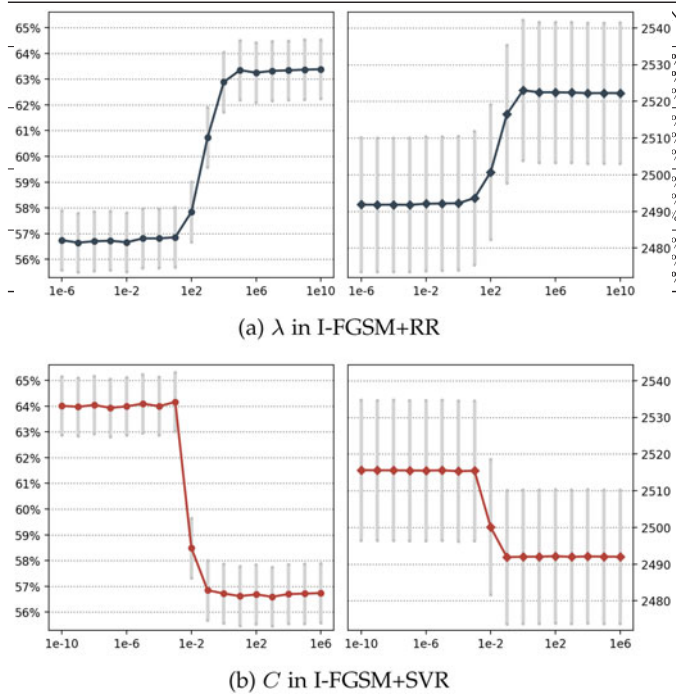

(b) C in I-FGSM+SVR

Fig. 4. How the hyper-parameters of the linear regression methods affect the average success rate and middle-layer distortion (i.e., the magnitude of intermediate-level discrepancies). On the left subplots, we report the success rates, and on the right subplots, we report the magnitudes. $\epsilon = 8/255$.

We further test how these methods perform when the intermediate-level feature distortion is not encouraged to be enlarged in linear regression. We consider linear mappings between the *normalized* intermediate-level discrepancies and the prediction loss of the source model. With such a mapping (denoted by $\tilde{\mathbf{w}}^*$ here), we can rewrite the optimization problem as:

$$\max_{\boldsymbol{\Delta_x}} \left(\frac{g(\mathbf{x} + \boldsymbol{\Delta_x}) - \mathbf{h}_0^{\mathrm{adv}}}{\|g(\mathbf{x} + \boldsymbol{\Delta_x}) - \mathbf{h}_0^{\mathrm{adv}}\|_2} \right)^T \tilde{\mathbf{w}}^*, \ \ \mathrm{s.t.} \ (\mathbf{x} + \boldsymbol{\Delta_x}) \in \Psi, \qquad (6)$$

in which the magnitude of $g(\mathbf{x} + \boldsymbol{\Delta_x}) - \mathbf{h}_0^{\mathrm{adv}}$ is not encouraged to be maximized in obtaining the final refined adversarial example.

For solving the optimization problem in Eq. (6), attacks can similarly be performed, just like for solving Eq. (2).

Tables 4 and 5 demonstrate that such "normalized" attacks fail to achieve satisfactory results on the ImageNet victim models. In the ℓ_∞ settings, I-FGSM+RR shows an average success rate of 29.17%, 22.63%, and 11.39%, under $\epsilon = 16/255$, $8/255$, and $4/255$, respectively. The obtained attack success rates are generally no higher than the baseline I-FGSM results under $\epsilon = 16/255$ and $8/255$ and far lower than those obtained when no normalization is enforced. Similar observations can be made on the ℓ_2 attack settings. We thus know from the experimental results that the magnitude of the intermediate-level discrepancies is an essential factor.

3.4 Direction Matters as Well

In Section 3.3, we have demonstrated that the magnitude of intermediate-level discrepancies matters in improving the transferability of adversarial examples. In this subsection, we will show that the choice of directional guide also affects the final attack success rates.

We have shown several different options for obtaining $\mathbf{w}^*$. One can see that, by tuning C, I-FGSM+SVR outperforms I-FGSM+RR, as demonstrated in Table 2 and Fig. 4. We can also observe that I-FGSM+RR finally did not lead to larger intermediate-level discrepancies than I-FGSM+SVR, implying that the choice of the directional guide (or say the direction of the feature perturbation) also matters.

We further try using random and less adversarial vectors as directional guides. We shall stick with the objective function in (2), except that now we are free to use any random method for obtaining $\mathbf{w}^*$. Several different ways can be tested, to achieve the goal. First, we can use random perturbations (rather than the baseline adversarial perturbations) to get a set of inputs. That is, we can add $\boldsymbol{\Delta}_0, \dots, \boldsymbol{\Delta}_{p-1}$, in which $\boldsymbol{\Delta}_t \sim \mathcal{N}(0, \Sigma)$, $\forall 0 \le t \le p - 1$, to $\mathbf{x}$ and get p inputs $\mathbf{x} + \boldsymbol{\Delta}_0, \dots, \mathbf{x} + \boldsymbol{\Delta}_{p-1}$. Given these inputs, the intermediate-level discrepancies and prediction loss on the source model can then be computed, and we can further obtain a linear regression model using for example RR, ElasticNet, and SVR. An alternative method is to incorporate randomness on the hidden layer. That is, we may first compute $\mathbf{h}_0^{\mathrm{adv}} = g(\mathbf{x})$ and then add random perturbations $\boldsymbol{\Delta}_0', \dots, \boldsymbol{\Delta}_{p-1}'$, in which $\boldsymbol{\Delta}_t' \sim \mathcal{N}(0, \Sigma')$, $\forall 0 \le t \le p-1$, to $\mathbf{h}_0^{\mathrm{adv}}$ and evaluate the prediction loss on the basis of the perturbed features. The

TABLE 4
Evaluation of Normalized ℓ_∞ Attacks

Method	ϵ	ResNet-50*	VGG-19 [31]	ResNet-152 [32]	Inception v3 [33]	DenseNet [34]	MobileNet v2 [35]	SENet [36]	ResNeXt [37]	WRN [38]	PNASNet [39]	MNASNet [40]	Average
I-FGSM	16/255	100.00%	**61.50%**	**52.82%**	30.86%	57.36%	58.92%	38.12%	48.88%	48.92%	28.92%	57.20%	**48.35%**
	8/255	100.00%	38.90%	29.36%	15.36%	34.86%	37.66%	17.76%	26.30%	26.26%	13.04%	35.08%	27.46%
	4/255	100.00%	18.86%	11.28%	6.66%	15.44%	18.36%	5.72%	9.58%	9.98%	4.14%	17.02%	11.70%
I-FGSM+RR (normalized)	16/255	99.94%	41.48%	34.52%	20.70%	40.66%	40.76%	20.20%	30.32%	31.74%	15.98%	38.46%	31.48%
	8/255	99.94%	35.24%	27.94%	15.90%	33.72%	35.32%	14.78%	24.44%	25.04%	11.50%	32.70%	25.66%
	4/255	99.92%	21.48%	14.76%	8.90%	19.10%	21.60%	**7.18%**	12.82%	12.86%	**5.68%**	19.44%	14.38%
I-FGSM+ElasticNet (normalized)	16/255	99.78%	24.36%	16.26%	11.06%	21.64%	24.94%	9.06%	14.12%	15.14%	7.40%	23.66%	16.76%
	8/255	99.62%	19.00%	11.58%	7.86%	15.84%	19.46%	6.22%	10.04%	10.46%	4.92%	18.70%	12.41%
	4/255	98.26%	10.46%	5.66%	4.78%	8.62%	12.30%	2.96%	5.00%	5.06%	2.36%	10.90%	6.81%
I-FGSM+SVR (normalized)	16/255	99.94%	39.86%	33.66%	20.38%	39.40%	39.72%	18.84%	29.72%	30.46%	15.46%	37.28%	30.48%
	8/255	99.94%	35.16%	27.00%	**16.24%**	33.04%	34.32%	14.56%	23.94%	24.68%	11.26%	31.80%	25.20%
	4/255	99.92%	**21.60%**	**15.12%**	9.42%	**19.66%**	**22.14%**	7.08%	**13.08%**	**13.12%**	5.36%	**20.26%**	**14.68%**

The attack success rate on ImageNet in the untargeted setting are reported. The symbol * is used to indicate when the source model is used as the target. It can be obviously seen that the performance of these linear regression models degraded significantly in comparison with the results in Table 2.

TABLE 5
Evaluation of Normalized ℓ_2 Attacks

Method	ϵ	ResNet-50*	VGG-19 [31]	ResNet-152 [32]	Inception v3 [33]	DenseNet [34]	MobileNet v2 [35]	SENet [36]	ResNeXt [37]	WRN [38]	PNASNet [39]	MNASNet [40]	Average
I-FGSM	15	99.96%	**78.64%**	74.94%	**50.02%**	**76.36%**	**75.20%**	60.18%	71.16%	71.20%	51.20%	73.80%	**68.27%**
	10	99.94%	**64.48%**	58.60%	33.64%	**59.94%**	60.92%	42.20%	53.98%	53.76%	33.94%	59.12%	**52.06%**
	5	99.98%	36.96%	28.12%	14.68%	31.70%	34.94%	18.02%	25.40%	25.82%	12.92%	33.08%	26.16%
I-FGSM+RR (normalized)	15	99.92%	72.46%	70.86%	49.46%	73.32%	71.30%	51.44%	65.04%	66.22%	44.96%	70.30%	63.54%
	10	99.90%	59.70%	57.20%	**36.76%**	60.46%	59.06%	37.92%	51.22%	51.52%	33.30%	57.94%	50.51%
	5	99.92%	**39.52%**	**33.10%**	19.70%	**38.20%**	**39.08%**	19.28%	30.08%	30.04%	15.92%	**37.14%**	**30.21%**
I-FGSM+ElasticNet (normalized)	15	99.88%	51.54%	42.72%	29.66%	47.20%	49.94%	30.50%	38.88%	39.46%	25.06%	50.44%	40.54%
	10	99.84%	39.44%	31.16%	20.30%	35.42%	38.80%	20.96%	28.82%	29.32%	17.68%	39.48%	30.14%
	5	98.86%	22.96%	17.10%	10.92%	20.08%	23.82%	9.84%	15.18%	14.76%	8.26%	22.42%	16.53%
I-FGSM+SVR (normalized)	15	99.92%	70.08%	68.86%	48.80%	70.86%	69.14%	49.80%	62.46%	63.62%	43.66%	68.24%	61.55%
	10	99.90%	58.10%	55.12%	36.46%	58.60%	57.76%	35.96%	49.74%	49.74%	32.10%	56.40%	49.00%
	5	99.92%	38.70%	32.30%	**19.76%**	37.44%	38.36%	19.10%	29.80%	29.54%	15.44%	36.40%	29.68%

*The attack success rates on ImageNet in the untargeted setting are reported. The symbol * is used to indicate when the source model is used as the target. It can be obviously seen that the performance of these linear regression models degraded significantly in comparison with the results in Table 3.*

linear regression model is then similarly obtained and Eq. (2) is to be solved with such random $\mathbf{w}^*$.

Tables 8 and 9 summarize the attack success rates using the two sorts of random directions, respectively. It can be seen that, since the random directions did not lead to high prediction error even on the ResNet-50 source model, the final attack success rates on the concerned victim models are also unsatisfactory. We would like to further mention that the magnitude of intermediate-level discrepancies are, nevertheless, high. More specifically, for random directional guides, average magnitude of the obtained intermediate-level discrepancies can achieve 2077.92, yet the final average success rate on the victim models is at most 21.25% under $\epsilon = 8/255$, which is even worse than the I-FGSM baseline (i.e., 27.46% with 1921.30). That is, given random (and thus non-disruptive) directional guides, it is challenging to learn adversarial examples that could achieve satisfactory attack success rates on the source model even with more significant intermediate-level perturbations. This can be ascribed to the high dimensionality of the intermediate-level feature space, and we therefore conclude that the direction matters as well.

Comparing the two sorts of random directional guides, we also conclude that random directions in the input space lead to slightly more disruptive attacks in contrast to that derived from random vectors in the feature space. We believe this is unsurprising, as random vectors in the feature space do not necessarily lie on the image manifold and can be misleading for guiding intermediate-level attacks.

4 TO FURTHER BOOST THE PERFORMANCE

We conclude from Section 3 that: 1) more transferable adversarial examples can be obtained by taking advantage of some linear regression models learned from the temporary results $\mathbf{x}_0^{\mathrm{adv}}, \ldots, \mathbf{x}_t^{\mathrm{adv}}, \ldots, \mathbf{x}_{p-1}^{\mathrm{adv}}$ of a multi-step baseline attack, and 2) it is challenging to learn a disruptive adversarial example given only random intermediate-level feature discrepancies and their corresponding prediction loss. Nevertheless, for a p-step baseline attack, the method is only capable of collecting $p + 1$ samples to train a linear regression model, let alone later results (with relatively large t) of the baseline attack are all very similar and thus less informative, as can be seen in Fig. 5. As such, one plausible way of further improving performance in our framework is to apply more than one baseline search and collect more diverse directional guides, if computational complexity of the baseline attack is not a primary concern. Once $\mathbf{w}^*$ is obtained, the objective in Eq. (2) is still utilized to
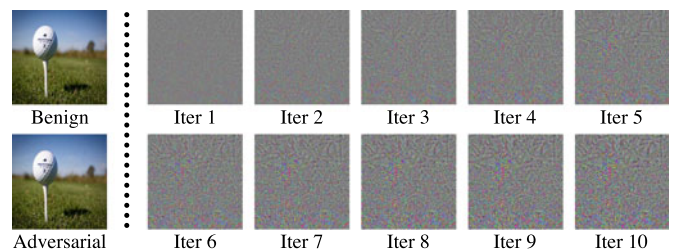

Fig. 5. Visualization of the obtained adversarial perturbations using I-FGSM. It can be seen that perturbations at later iterations are all very similar. $\epsilon = 8/255$.

TABLE 6
Average Success Rate of I-FGSM+ElasticNet on Models *Excluding ResNet-50* in the Untargeted ℓ_∞ Setting

Method	ϵ	$\lambda_1 = 0.05$	$\lambda_1 = 0.1$	$\lambda_1 = 0.5$	$\lambda_1 = 1.0$
I-FGSM+ElasticNet	16/255	80.76%	78.06%	69.69%	65.24%
	8/255	50.07%	46.45%	36.40%	31.97%
	4/255	21.05%	19.31%	14.04%	11.87%

TABLE 7
Success Rate of I-FGSM+ElasticNet on the Source Model, *i.e., ResNet-50*, in the Untargeted ℓ_∞ Setting

Method	ϵ	$\lambda_1 = 0.05$	$\lambda_1 = 0.1$	$\lambda_1 = 0.5$	$\lambda_1 = 1.0$
I-FGSM+ElasticNet	16/255	99.90%	99.92%	99.78%	99.62%
	8/255	99.92%	99.90%	99.36%	98.16%
	4/255	99.90%	99.80%	97.48%	92.92%

TABLE 8
Evaluation of Random Directional Guides in the Input Space

Method	ϵ	ResNet-50*	VGG-19 [31]	ResNet-152 [32]	Inception v3 [33]	DenseNet [34]	MobileNet v2 [35]	SENet [36]	ResNeXt [37]	WRN [38]	PNASNet [39]	MNASNet [40]	Average
I-FGSM	16/255	100.00%	61.50%	52.82%	30.86%	57.36%	58.92%	38.12%	48.88%	48.92%	28.92%	57.20%	48.35%
	8/255	100.00%	38.90%	29.36%	15.36%	34.86%	37.66%	17.76%	26.30%	26.26%	13.04%	35.08%	27.46%
	4/255	100.00%	18.86%	11.28%	6.66%	15.44%	18.36%	5.72%	9.58%	9.98%	4.14%	17.02%	11.70%
Rand+RR	16/255	97.82%	66.26%	50.80%	31.44%	61.50%	76.50%	41.60%	45.50%	50.50%	17.74%	69.56%	51.14%
	8/255	90.94%	35.66%	18.96%	9.96%	27.94%	36.06%	12.98%	16.32%	19.86%	5.18%	29.58%	21.25%
	4/255	62.86%	12.72%	5.30%	4.20%	9.22%	11.20%	3.52%	4.64%	5.98%	1.54%	9.16%	6.75%
Rand+ElasticNet	16/255	98.56%	64.14%	44.60%	21.86%	47.30%	64.02%	40.10%	43.44%	48.34%	19.08%	63.20%	45.61%
	8/255	93.22%	26.58%	13.26%	7.78%	17.00%	24.86%	9.96%	10.90%	14.30%	4.12%	23.56%	15.23%
	4/255	59.42%	7.84%	2.96%	3.84%	5.34%	8.60%	2.06%	2.62%	3.28%	1.18%	7.78%	4.55%
Rand+SVR	16/255	97.76%	66.52%	50.84%	31.68%	61.82%	76.04%	41.54%	45.84%	50.30%	17.66%	69.74%	51.20%
	8/255	91.02%	35.32%	18.92%	10.06%	27.84%	36.06%	12.86%	16.52%	19.74%	5.40%	29.44%	21.22%
	4/255	62.86%	12.62%	4.92%	4.46%	9.24%	11.28%	3.24%	4.36%	5.74%	1.52%	8.80%	6.62%

The attack success rates on ImageNet are reported, with ℓ_∞ constraints in the untargeted setting. The symbol * is used to indicate when the source model is used as the target. We see that random directional guides lead to poor transferability in comparison to the results in Table 2.

TABLE 9
Evaluation of Random Directional Guides Generated Directly in the Feature Space (Indicated as Rand†)

Method	ϵ	ResNet-50*	VGG-19 [31]	ResNet-152 [32]	Inception v3 [33]	DenseNet [34]	MobileNet v2 [35]	SENet [36]	ResNeXt [37]	WRN [38]	PNASNet [39]	MNASNet [40]	Average
I-FGSM	16/255	100.00%	61.50%	52.82%	30.86%	57.36%	58.92%	38.12%	48.88%	48.92%	28.92%	57.20%	48.35%
	8/255	100.00%	38.90%	29.36%	15.36%	34.86%	37.66%	17.76%	26.30%	26.26%	13.04%	35.08%	27.46%
	4/255	100.00%	18.86%	11.28%	6.66%	15.44%	18.36%	5.72%	9.58%	9.98%	4.14%	17.02%	11.70%
Rand†+RR	16/255	92.58%	53.46%	31.72%	22.52%	35.48%	48.10%	30.34%	30.00%	35.20%	15.74%	47.16%	34.97%
	8/255	68.44%	21.24%	9.46%	2.40%	12.64%	20.20%	8.52%	9.02%	10.48%	3.76%	19.48%	12.26%
	4/255	24.40%	7.04%	2.40%	3.32%	4.38%	7.90%	2.22%	2.70%	3.16%	0.98%	7.40%	4.15%
Rand†+ElasticNet	16/255	57.28%	38.56%	29.88%	19.78%	31.06%	39.40%	27.12%	28.70%	29.90%	15.06%	38.36%	29.78%
	8/255	45.14%	18.42%	10.44%	7.52%	12.08%	18.46%	8.20%	9.26%	10.22%	3.66%	17.72%	11.60%
	4/255	24.78%	6.64%	2.60%	3.50%	4.70%	8.06%	1.88%	2.52%	2.76%	1.22%	7.40%	4.13%
Rand†+SVR	16/255	92.58%	53.40%	31.64%	22.54%	35.24%	47.54%	30.78%	30.24%	35.26%	15.86%	47.08%	34.96%
	8/255	68.52%	20.56%	9.52%	7.38%	11.90%	20.16%	8.78%	8.98%	10.78%	3.64%	19.52%	12.12%
	4/255	24.64%	6.96%	2.46%	3.26%	4.26%	8.16%	2.12%	2.78%	2.92%	1.20%	7.46%	4.16%

The attack success rates on ImageNet are reported, with ℓ_∞ constraints in the untargeted setting. The symbol * is used to indicate when the source model is used as the target. We see that random directional guides lead to poor transferability.

TABLE 10
Further Performance Gain Can Be Achieved by Performing More than One Run of a Baseline Method and Using a More Powerful Baseline Attack, e.g., LinBP

Method	ϵ	ResNet-50*	VGG-19 [31]	ResNet-152 [32]	Inception v3 [33]	DenseNet [34]	MobileNet v2 [35]	SENet [36]	ResNeXt [37]	WRN [38]	PNASNet [39]	MNASNet [40]	Average
LinBP	16/255	100.00%	93.04%	89.42%	60.72%	90.34%	88.90%	77.26%	84.38%	85.06%	63.78%	87.78%	82.07%
	8/255	100.00%	71.72%	58.58%	29.54%	63.42%	64.42%	41.36%	51.22%	54.62%	29.98%	62.52%	52.74%
	4/255	99.98%	36.28%	22.48%	10.60%	28.72%	31.80%	13.40%	18.12%	20.26%	8.74%	30.06%	22.05%
LinBP (×10) +RR	16/255	100.00%	97.80%	96.28%	85.22%	97.48%	97.40%	91.26%	95.18%	94.78%	87.28%	96.74%	93.94%
	8/255	100.00%	86.50%	78.26%	49.38%	81.86%	82.94%	61.86%	73.18%	74.16%	51.92%	81.56%	72.16%
	4/255	99.92%	52.32%	36.94%	17.22%	44.50%	46.70%	23.52%	32.18%	34.12%	16.66%	45.76%	34.99%
LinBP (×10) +ElasticNet	16/255	100.00%	94.04%	88.02%	68.42%	90.82%	91.62%	80.52%	85.20%	85.68%	70.48%	90.54%	84.53%
	8/255	99.94%	71.88%	56.40%	30.74%	61.74%	65.76%	42.92%	51.82%	52.98%	30.42%	63.98%	52.86%
	4/255	99.32%	35.30%	22.02%	10.70%	27.22%	31.48%	14.00%	18.64%	20.52%	8.82%	31.54%	22.02%
LinBP (×10) +SVR	16/255	100.00%	97.54%	95.50%	84.48%	96.94%	96.86%	90.28%	94.62%	94.34%	86.22%	96.40%	93.32%
	8/255	100.00%	85.70%	77.40%	49.24%	81.58%	82.58%	61.40%	72.20%	73.34%	51.38%	81.20%	71.60%
	4/255	99.88%	51.88%	37.16%	17.32%	44.26%	46.60%	23.62%	31.78%	34.18%	16.44%	46.20%	34.94%

The attack success rates on ImageNet are reported, with ℓ_∞ constraints in the untargeted setting. The symbol * is used to indicate when the source model is used as the target.

encourage maximal intermediate-level distortion with these directional guides.

Our first attempt is to perform several runs of the PGD baseline attack [42], which adopts a random perturbation (on the benign example) ahead of performing the standard I-FGSM. Benefiting from the random perturbations, we can expect different temporary results for different runs of the baseline attack. An experiment was conducted to evaluate the effectiveness of this simple idea. Specifically, we performed 10 runs of PGD used RR, ElasticNet, and SVR to established the linear mapping. We obtained an average success rate of 66.14% on ImageNet with SVR under $\epsilon = 8/255$, outperforming the result in Table 2 significantly.

How the performance of our methods varies with the number of PGD runs is illustrated in Fig. 6. It can be seen that, for RR and SVR, increasing the number of PGD runs improves the average attack success rate. While for Elastic-Net which is a sparse coding method, the result is unstable. For ℓ_∞ attacks with other ϵ settings (i.e., $16/255$ and $4/255$) and ℓ_2 attacks, similar observations can be made, i.e., the best results is obtained with SVR, and RR achieved the second best.

More powerful baseline attack also leads to more transferable adversarial examples after refinement in our framework. Therefore, it is natural to try more advanced methods than I-FGSM and PGD. We have tested with MI-FGSM [43],

TABLE 11
Further Performance Gain Can Be Achieved by Performing More than One Run of a Baseline Method
and Using a More Powerful Baseline Attack, e.g., LinBP

Method	ϵ	ResNet-50*	VGG-19 [31]	ResNet-152 [32]	Inception v3 [33]	DenseNet [34]	MobileNet v2 [35]	SENet [36]	ResNeXt [37]	WRN [38]	PNASNet [39]	MNASNet [40]	Average
LinBP	15	100.00%	96.98%	95.80%	79.96%	96.16%	94.98%	89.42%	93.96%	93.62%	82.16%	94.34%	91.74%
	10	100.00%	89.88%	84.00%	56.16%	85.34%	84.72%	69.46%	79.08%	79.76%	58.44%	82.82%	76.97%
	5	100.00%	61.36%	45.70%	21.50%	52.02%	52.32%	30.30%	39.26%	42.02%	22.26%	49.50%	41.62%
LinBP ($\times$10) +RR	15	100.00%	**98.10%**	**96.62%**	**88.28%**	**97.82%**	**97.40%**	**92.18%**	**95.38%**	**95.58%**	**89.88%**	97.06%	**94.83%**
	10	100.00%	**93.76%**	**89.76%**	**68.66%**	**92.00%**	92.32%	**78.30%**	**86.70%**	86.64%	**71.44%**	**91.10%**	**85.07%**
	5	99.94%	**72.08%**	**58.66%**	**31.52%**	**64.18%**	66.38%	41.46%	**52.44%**	54.92%	33.02%	65.22%	**53.99%**
LinBP ($\times$10) +ElasticNet	15	99.96%	94.12%	89.80%	73.04%	92.12%	92.30%	80.12%	86.82%	87.12%	74.72%	91.48%	86.16%
	10	99.96%	84.78%	74.92%	51.22%	78.84%	81.34%	59.84%	70.28%	70.70%	50.94%	80.14%	70.30%
	5	99.04%	54.26%	40.42%	21.00%	46.40%	50.96%	28.52%	36.04%	37.12%	20.78%	49.76%	38.53%
LinBP ($\times$10) +SVR	15	100.00%	98.04%	96.40%	87.36%	97.52%	97.34%	91.92%	95.16%	95.50%	89.12%	**97.10%**	94.55%
	10	100.00%	93.44%	89.02%	68.60%	91.34%	**92.34%**	77.48%	85.92%	**86.70%**	70.52%	91.02%	84.64%
	5	99.90%	71.28%	58.20%	31.74%	63.94%	66.70%	41.52%	52.32%	54.40%	32.86%	64.94%	53.79%

*The attack success rates on ImageNet are reported, with ℓ_2 constraints in the untargeted setting. The symbol * is used to indicate when the source model is used as the target.*

TAP [15], and LinBP [17]. Together with random perturbations beforehand, each of these methods can be performed more than once for collecting diverse directional guides. Results showed that LinBP is the best in this regard, assisting RR in our framework to achieve an average attack success rate of 94.83%, 72.16%, and 53.99% under $\epsilon = 16/255$, $8/255$, and $4/255$, respectively. For comparison, our ILA++ in its default setting achieves 88.57%, 63.38%, and 30.59%, respectively (as reported in Table 1). Fig. 7, Table 10, and Table 11 report more detailed results. Due to the space limit of the paper, we omit results with MI-FGSM and TAP in this paper, and it is worthy noting that different baseline methods might favor different linear regression models, cf. Fig. 6 (in which +SVR is the best) and Fig. 7 (in which +RR is the best).

5 DETAILED EXPERIMENTAL SETTINGS

In this section, we will introduce detailed experimental settings for all the experiments in this paper.

5.1 General Settings

All the experiments were performed on the same set of ImageNet [44] models, i.e., ResNet-50 [32], VGG-19 [31], ResNet-152 [32], Inception v3 [33], DenseNet [34], MobileNet v2 [35], SENet [36], ResNeXt [37], WRN [38], PNASNet [39], and MNASNet [40]. These models are popularly used and can achieve reasonably high prediction accuracy on the ImageNet validation set: 76.15% (ResNet-50), 74.24% (VGG-19), 78.31% (ResNet-152), 77.45% (Inception v3),

74.65% (DenseNet), 71.18% (MobileNet v2), 81.30% (SENet), 79.31% (ResNeXt), 78.84% (WRN), 82.74% (PNASNet), and 73.51% (MNASNet). To keep in line with the work in [1], we chose ResNet-50 as the source model. Some victim models have different pre-processing pipelines, and we followed their official settings. For instance, for DenseNet, we resized its input images to 256×256 and then cropped them to 224×224 at center.

We randomly sampled 5,000 test images that can be correctly classified by all the victim models from the ImageNet validation set for evaluating attacks. Specifically, these images were from 500 (randomly chosen) classes, and we had 10 images per class.

For the constraint on perturbations and step size, we tested $\epsilon = 16/255, 8/255, 4/255$ with a common step size of $1/255$ for ℓ_∞ attacks, and we tested $\epsilon = 15.0, 10.0, 5.0$ using $\epsilon/5$ as the step size for ℓ_2 attacks. We run 100 iterations for I-FGSM and LinBP. We let $p = 10$ for attacks using temporary results from the baseline attack (in Section 3). Then, with 10 runs in Section 4, we have $10 \times 10 = 100$ samples for establishing a linear mapping. After updating the adversarial examples at each attack iteration, they were clipped to the range of [0.0, 1.0] to keep the validity of being images.

Since the concerned attacks utilize intermediate-level representations, the choice of hidden layer should have an impact on the performance of the attacks. Unless otherwise clarified, we always chose the first block of the third meta-block of ResNet-50. For LinBP, the last six building blocks in ResNet-50 were modified to be more linear during backpropagation.
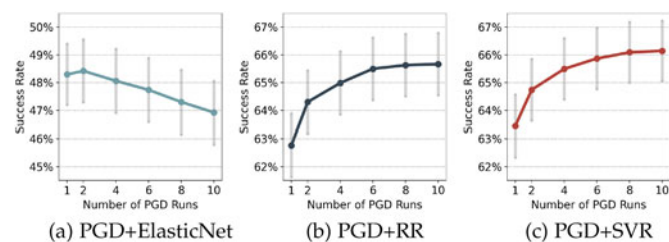

Fig. 6. How the number of PGD runs affects the average attack success rate in our framework. $\epsilon = 8/255$.
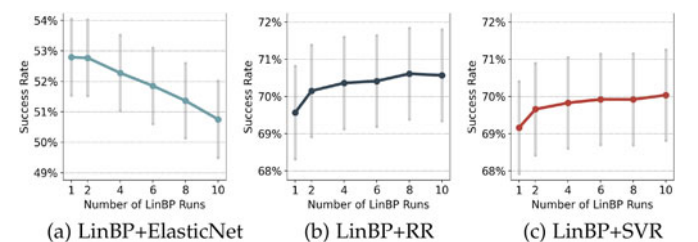

Fig. 7. How the number of LinBP runs affects the average attack success rate in our framework. $\epsilon = 8/255$.

6 CONCLUSION

We have demonstrated that the transferability of adversarial examples can be substantially improved by establishing a linear mapping directly from some middle-layer features (of a source DNN) to its prediction loss. Various linear regression models can thus be chosen for achieving the goal, and we have shown the effectiveness using RR, ElasticNet, and SVR. We have carefully analyzed core components of such a framework and shown that, given powerful directional guides, the magnitude of intermediate-level feature discrepancies is correlated with the transferability. Using random directional guides or normalizing feature discrepancies as in Eq. (6) shall lead to poor attack performance. Given all these facts, we have demonstrated that further improved attack performance can be obtained via collecting more diverse yet still powerful directional guides. New state-of-the-arts have been achieved.

ACKNOWLEDGMENTS

REFERENCES

[1] Q. Li, Y. Guo, and H. Chen, "Yet another intermediate-leve attack," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 241–257.

[2] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, 2017, pp. 15–26.

[3] C.-C. Tu *et al.*, "AutoZOOM: Autoencoder-based zeroth order optimization method for attacking black-box neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 742–749.

[4] A. Nitin Bhagoji, W. He, B. Li, and D. Song, "Practical black-box attacks on deep neural networks using efficient query mechanisms," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 158–174.

[5] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2137–2146.

[6] A. Ilyas, L. Engstrom, and A. Madry, "Prior convictions: Black-box adversarial attacks with bandits and priors," in *Proc. Int. Conf. Learn. Representations*, 2019.

[7] C. Guo, J. R. Gardner, Y. You, A. G. Wilson, and K. Q. Weinberger, "Simple black-box adversarial attacks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2484–2493.

[8] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," 2017, *arXiv: 1712.04248*.

[9] F. Croce and M. Hein, "Minimally distorted adversarial examples with a fast adaptive boundary attack," in *Proc. Int. Conf. on Mach. Learn.*, 2020, pp. 2196–2205.

[10] J. Chen, M. I. Jordan, and M. J. Wainwright, "HopSkipJumpAttack: A query-efficient decision-based attack," in *Proc. IEEE Symp. Secur. Privacy*, 2020, pp. 1277–1294.

[11] Z. Yan, Y. Guo, J. Liang, and C. Zhang, "Policy-driven attack: Learning to query for hard-label black-box adversarial examples," in *Proc. Int. Conf. Learn. Representations*, 2021.

[12] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. Asia Conf. on Comput. Commun. Secur.*, 2017, pp. 506–519.

[13] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," in *Proc. Int. Conf. Learn. Representations*, 2017.

[14] Z. Huang and T. Zhang, "Black-box adversarial attack with transferable model-based embedding," in *Proc. Int. Conf. Learn. Representations*, 2020.

[15] W. Zhou *et al.*, "Transferable adversarial perturbations," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 452–467.

[16] Z. Yan, Y. Guo, and C. Zhang, "Subspace attack: Exploiting promising subspaces for query-efficient black-box attacks," in *Proc. Adv. Neural Informat. Process. Syst.*, 2019, pp. 3825–3834.

[17] Y. Guo, Q. Li, and H. Chen, "Backpropagating linearly improves transferability of adversarial examples," in *Proc. Adv. Neural Informat. Process. Syst.*, 2020, pp. 85–95.

[18] J. Platt *et al.*, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Adv. Large Margin Classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[19] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Statist. Soc., Ser. Methodol.*, vol. 67, no. 2, pp. 301–320, 2005.

[20] C. Szegedy *et al.*, "Intriguing properties of neural networks," in *Proc. Int. Conf. Learn. Representations*, 2014.

[21] Y. Dong, T. Pang, H. Su, and J. Zhu, "Evading defenses to transferable adversarial examples by translation-invariant attacks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4312–4321.

[22] C. Xie *et al.*, "Improving transferability of adversarial examples with input diversity," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2725–2734.

[23] N. Inkawhich, W. Wen, H. H. Li, and Y. Chen, "Feature space perturbations yield more transferable adversarial examples," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7059–7067.

[24] D. Wu, Y. Wang, S.-T. Xia, J. Bailey, and X. Ma, "Rethinking the security of skip connections in resnet-like neural networks," in *Proc. Int. Conf. Learn. Representations*, 2020.

[25] X. Wang, J. Ren, S. Lin, X. Zhu, Y. Wang, and Q. Zhang, "A unified approach to interpreting and boosting adversarial transferability," in *Proc. Int. Conf. Learn. Representations*, 2020.

[26] Z. Wang, H. Guo, Z. Zhang, W. Liu, Z. Qin, and K. Ren, "Feature importance-aware transferable adversarial attacks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 7639–7648.

[27] Y. Zhu, J. Sun, and Z. Li, "Rethinking adversarial transferability from a data distribution perspective," in *Proc. Int. Conf. Learn. Representations*, 2021.

[28] J. Springer, M. Mitchell, and G. Kenyon, "A little robustness goes a long way: Leveraging robust features for targeted transfer attacks," in *Proc. Adv. Neural Informat. Process. Syst.*, 2021, pp. 759–9773.

[29] Q. Huang, I. Katsman, H. He, Z. Gu, S. Belongie, and S.-N. Lim, "Enhancing adversarial example transferability with an intermediate level attack," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4732–4741.

[30] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *Proc. Int. Conf. Learn. Representations*, 2017.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[33] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.

[34] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269.

[35] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.

[36] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.

[37] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5987–5995.

[38] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vis. Conf.*, 2016.

[39] C. Liu *et al.*, "Progressive neural architecture search," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 19–35.

[40] M. Tan *et al.*, "MnasNet: Platform-aware neural architecture search for mobile," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2815–2823.

[41] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Representations*, 2015.

[42] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Representations*, 2018.

[43] Y. Dong *et al.*, "Boosting adversarial attacks with momentum," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9185–9193.

[44] O. Russakovsky *et al.*, "Imagenet large scale visual recognition challenge," *Proc. Int. J. Comput. Vis.*, vol. 115, pp. 211–252, 2015.
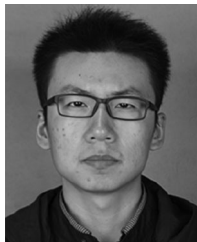
Yiwen Guo received the BE degree from Wuhan University, in 2011, and the PhD degree from Tsinghua University, in 2016. He was a research scientist with ByteDance AI Lab, Beijing. Prior to this, he was a staff research scientist with Intel Labs China. His current research interests include computer vision, pattern recognition, and machine learning.

Qizhang Li received the BE and ME degrees from Harbin Engineering University, Harbin, China, 2018 and 2021, respectively. He is currently working toward the PhD degree in computer science and technology with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China. His current research interests include computer vision and machine learning.
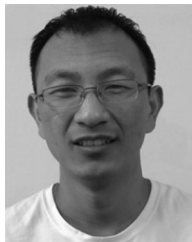
Wangmeng Zuo (Senior Member, IEEE) received the PhD degree in computer application technology from the Harbin Institute of Technology, Harbin, China, in 2007. He is currently a professor with the School of Computer Science and Technology, Harbin Institute of Technology. His current research interests include image enhancement and restoration, image and face editing, object detection, visual tracking, and image classification. He has published more than 100 papers in top tier academic journals and conferences. He has served as a Tutorial Organizer in ECCV 2016, an associate editor of the *IEEE Transactions Pattern Analysis and Machine Intelligence*, *The Visual Computers*, and *Journal of Electronic Imaging*.
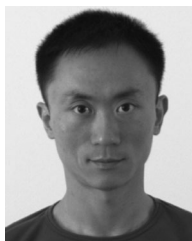
Hao Chen received BS and MS degress from Southeast University, and the PhD degree from the Computer Science Division, University of California, Berkeley. He is currently a professor with the Department of Computer Science, University of California, Davis. His research interests include computer security, machine learning, and mobile computing.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.