Appendix: Boosting Adversarial Transferability via Negative Hessian Trace Regularization

Yunfei Long [1], Zilin Tian[1], Liguo Zhang[1*] , Huosheng Xu[1]
[1]Harbin Engineering University
{2016064109, tzl, zhangliguo, xuhuosheng}@hrbeu.edu.cn

1. The Hessian trace at Loss Maxima

Given that the loss function $\mathcal{L}(x)$ is second-order differentiable. If x^* is a local maximum point, then for sufficiently small perturbations υ we have

$$\mathcal{L}(x^* + \upsilon) \leq \mathcal{L}(x^*). \tag{1}$$

Using Taylor expansion (ignoring higher-order terms), we have

$$\mathcal{L}(x^* + \upsilon) \approx L(x^*) + \tfrac{1}{2}\upsilon^T \mathcal{H}(x^*)\upsilon, \tag{2}$$

where $\mathcal{H}(x) = \nabla_x \nabla_x \mathcal{L}(x)$ denote the Hessian matrix of $\mathcal{L}(x)$ with respect to x.

Therefore, for all sufficiently small perturbations υ, we must have

$$\operatorname{tr}(\mathcal{H}(x^*)) = \mathbb{E}_\upsilon \left[\upsilon^T \mathcal{H}(x^*)\, \upsilon\right] \leq 0. \tag{3}$$

Further, consider $\mathcal{L}(x)$ is Lipschitz continuous, which means that its Hessian matrix $\mathcal{H}(x)$ is continuous with respect to x. Let $\mathcal{F}(x) = \operatorname{tr}(\mathcal{H}(x))$, then $\mathcal{F}(x)$ is also a continuous function. It is known that at the local maximum point x^*,

$$\mathcal{F}(x) \leq 0. \tag{4}$$

By the definition of continuity, there exists a neighborhood $\mathcal{B}_\epsilon$ for this point, where ϵ is the radius. Such that when $\|x - x^*\| < \epsilon$),

$$|\mathcal{F}(x) - \mathcal{F}(\theta^*)| < \epsilon, \tag{5}$$

since $\mathcal{F}(x^*) < 0$ (so $|\mathcal{F}(x^*)| > 0$), we set $\epsilon = |\mathcal{F}(x^*)|$ and get:

$$|\mathcal{F}(x) - \mathcal{F}(x^*)| < |\mathcal{F}(x^*)| \tag{6}$$

thus ensuring that

$$\mathcal{F}(x) < 0.$$

Therefore, we theoretically prove that points near the local maximum x^* also satisfy the Hessian trace to be negative.

* Corresponding author.

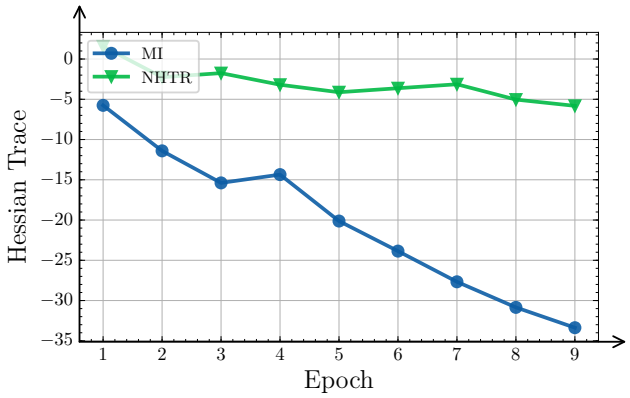

Figure 1. As adversarial examples approach the maximum loss, their Hessian traces remain consistently negative and gradually decrease, aligning with our theoretical analysis.

Adversarial examples are expected to converge toward the local loss maxima. We empirically observe that in the iterative optimization process of adversarial examples toward loss maxima, the trace of the total Hessian in the neighborhood near adversarial is consistently negative, as shown in Figure 1. It is evident that the Hessian trace of the adversarial sample generated by the MI method decreases at a faster rate, suggesting that it converges to a sharp maxima. In contrast, while the Hessian trace of the adversarial examples generated by our NHTR method remains negative, it does not exhibit a sharp decline, indicating that these examples reside in a flat region.

2. Additional Experiments

2.1. Black-box Attacks on the CIFAR-10

We compare the attacking performance of the proposed Negative Hessian Trace Regularization (NHTR) with state-of-the-art (SOTA) attacking methods on the CIFAR-10 dataset. We consider normally trained models, including VGG-16 (VGG) [4], Inception-v3 (Inc-v3) [5], ResNet-50 (Res-50) [1], MobileNet (Mobile) [3], and Densenet [2] as

Attack	Res-18⇒						Res-50⇒					
	Res-50	VGG	Inc-v3	Mobile	Densenet	Avg.	Res-18	VGG	Inc-v3	Mobile	Densenet	Avg.
MI	65.9	60.9	57.7	61.8	60.7	61.4	66.4	63.2	66.6	69.9	71.2	67.5
NI	60.1	56.3	51.2	56.5	54.9	55.8	58.2	57.3	60.4	66.1	65.0	61.4
PI	67.8	68.7	62.8	69.4	65.1	66.8	64.4	62.3	66.8	70.9	68.8	66.6
VMI	80.1	77.4	74.0	76.5	77.5	77.1	64.7	66.3	70.3	73.8	69.9	69.0
VNI	69.7	70.6	64.4	69.7	68.9	68.7	77.5	77.5	78.0	81.9	81.1	79.2
EMI	81.8	86.3	79.5	83.1	80.8	82.3	79.4	82.0	83.0	87.8	82.7	83.0
RAP	78.8	81.5	76.1	80.3	78.5	79.0	69.2	72.5	73.8	79.8	75.6	74.2
PGN	84.3	89.7	84.3	88.1	87.3	86.7	88.1	90.2	89.4	94.7	90.0	90.5
NHTR	89.5	93.9	88.8	93.1	91.8	91.4	90.2	93.4	93.0	96.6	92.8	93.2

Table 1. The black-box attack success rates (%) of various gradient optimization methods with Res-18 and Res-50 as surrogate models. The dataset is CIFAR-10.

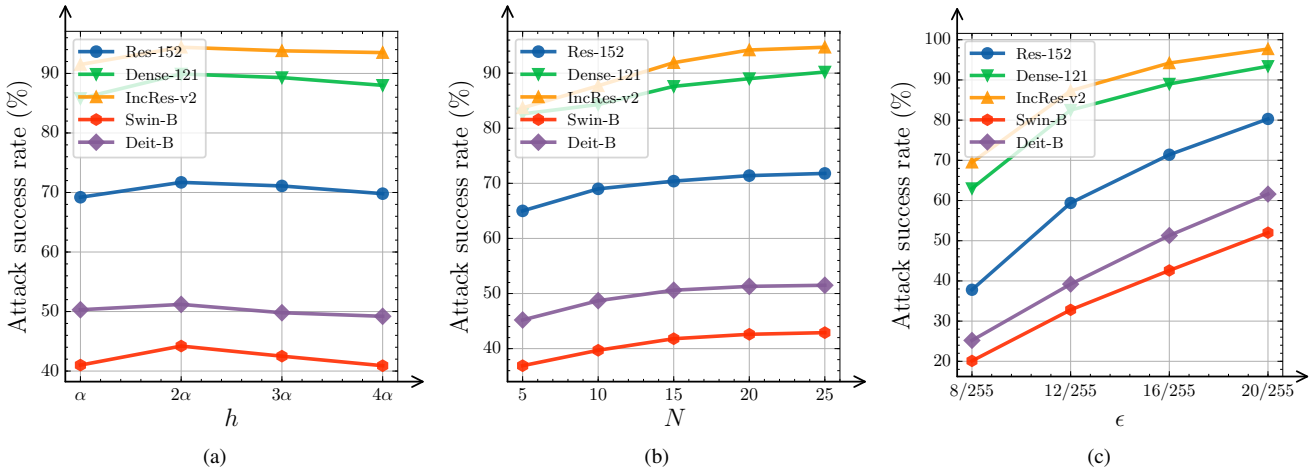

Figure 2. Ablation study. (a-c) illustrate the impact of the hyper-parameters on the transferability of adversarial examples: the discretization step size h, the number of samples N from neighborhood of adversarial examples, and the maximum perturbation ϵ.

black-box target models and select ResNet-18 (Res-18) as the surrogate model. The maximum perturbation ϵ is set to $8/255$, the number T of iterations to 10, and the step size to $\alpha = \epsilon/T$. The results presented in Table 1 clearly indicate that the proposed NHTR enhances attack transferability on the CIFAR-10 dataset. For example, when using the Res-18 model as the surrogate model, NHTR achieves an attack success rate of 93.9% on VGG, which is 16.5% higher than that of VMI and 33% higher than MI. This demonstrates the strong applicability of NHTR across various datasets and supports the notion that adversarial examples situated in flat local regions tend to exhibit improved transferability across diverse models.

2.2. Additional Ablation Study

1. **The discretization step size h.** We approximate the Hessian trace based on the finite differences. h denotes the discretization step size. We examine the impact of h on the transferability of adversarial examples and report attack success rates as h increases. As shown in Figure 2 (a), attack success rates peak at $r = 2 \cdot \alpha$.
2. **The sampling numbers N.** We investigate the impact of the sampling number N from the neighborhood around adversarial examples on transferability. We increase N from 5 to 25. The results are shown in Figure 2 (b). We observe that as the sampling number N increases from 5 to 20, the attack success rates rise significantly, from 20 to 25, the success rates exhibit only minor fluctuations. To strike a balance between transferability and computational overhead, we set N = 20.
3. **The maximum perturbation ϵ.** The impact of perturbation magnitude ϵ on the attack success rates of the proposed NHTR is illustrated in Figure 2 (c). We observe that a larger perturbation results in higher attack success rates. To balance the success rates of attacks and the imperceptibility of adversarial examples, we finally set the perturbation size to $16/255$ in our experiments.

2.3. Attack Multi-modal Large Language Model

To further demonstrate the effectiveness of adversarial examples generated by NHTR, we conducted experiments targeting large visual-language models, specifically ChatGPT. Using ViT-B as the surrogate model and same settings in the experiment, we crafted adversarial examples that impacted the model. As illustrated in Figure 3, ChatGPT erroneously identified the number of birds in the image as five.

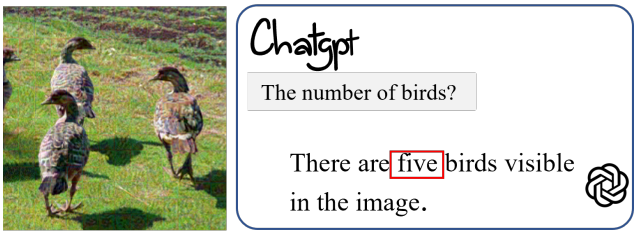

Figure 3. The performance of adversarial examples generated by NHTR misleading ChatGPT.

3. Core Code

We provide the core implementation of the Negative Hessian Trace Regularization method, developed using the PyTorch library, as shown in Figure 4.

```
def calculate_gradient(self, data, delta, label):
    g = 0
    for _ in range(self.num_neighbor):
        # randomly sampling form neighborhood
        adv_near = data + delta + torch.zeros_like(delta).uniform_(-self.xi,  *args: self.xi).to(self.device)
        logit = self.model(adv_near)
        loss1 = self.get_ce_loss(logit, label)
        # Calculate the gradient of the adv_near
        g_ = self.get_grad(loss1, delta)
        # Calculate the worst-case perturbations
        x_down = self.transform(adv_near + self.h*self.alpha * (-g_ / (torch.abs(g_).mean(dim=(1, 2, 3), keepdim=True))))
        logits = self.model(x_down)
        loss2 = self.get_ce_loss(logits, label)
        g_down = self.get_grad(loss2, delta)
        # Calculate the best-case perturbations
        x_up = self.transform(adv_near + self.h*self.alpha * (g_ / (torch.abs(g_).mean(dim=(1, 2, 3), keepdim=True))))
        logits = self.model(x_up)
        loss3 = self.get_ce_loss(logits, label)
        g_up = self.get_grad(loss3, delta)
        # Calculate the weighted gradient
        g += self._delta * g_ + (1-self._delta)/2 * (g_up + g_down)
    return g / self.num_neighbor
```

Figure 4. The core code of NHTR.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[2] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[3] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[4] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[5] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.