ICML 2025

Quadratic Upper Bound for Boosting Robustness

Euijin You[1], Hyang-Won Lee[1*]

1 Department of Computer Science and Engineering, Konkuk University, Seoul, South Korea.
* Correspondence author: leehw@konkuk.ac.kr

Research Background

Adversarial Attack

$$\max_{\|\delta\|_p \leq \epsilon} \mathcal{L}(f_\theta(x+\delta), y)$$

- δ : imperceptible pixel-level **perturbation**

Adversarial Training (AT)

$$\min_{\theta} \max_{\|\delta\|_p \leq \epsilon} \mathcal{L}(f_\theta(x+\delta), y)$$

- Train a model to be robust against adversarial attacks
- Formulated as a **min-max optimization** problem
  - Inner maximization: find δ that maximizes the loss
  - Outer minimization: update model to minimize the worst-case loss

Research Background

Fast Adversarial Training (FAT)

- Time-consuming generation of training attacks through iterative updates
- FAT: Efficient single-step attacks with low-quality perturbations

  → Decreased model robustness

We propose a method that achieves improved robustness, even when the quality of perturbations generated during inner maximization is somewhat limited.

Proposed Method

Quadratic Upper Bound for AT

Lemma 1. *The AT loss function is upper-bounded as follows:*

$$\mathcal{L}(f(x+\delta)) \leq \mathcal{L}(f(x)) + (f(x+\delta) - f(x))^T \nabla_f \mathcal{L}(f(x)) + \frac{\|\boldsymbol{H}\|_2}{2} \|f(x+\delta) - f(x)\|_2^2, \quad (6)$$

where $\nabla_f \mathcal{L}$ is the gradient of the loss with respect to the logit f and $\|\boldsymbol{H}\|_2$ is the L_2 norm of the Hessian matrix of the loss with respect to the logit, evaluated at some point between $f(x)$ and $f(x+\delta)$.

Proposed Method

Quadratic Upper Bound Loss (QUB Loss)

Lemma 2. *We have* $||\boldsymbol{H}||_2 \leq \frac{1}{2}$.

The derivation of the bound is presented in Appendix C.

Based on Lemmas 1 and 2, the QUB loss is defined as

$$\mathcal{L}_{\text{QUB}} = \mathcal{L}(f(x)) + (f(x+\delta) - f(x))^T \nabla_f \mathcal{L}(f(x)) + \frac{1}{4}\|f(x+\delta) - f(x)\|_2^2. \quad (7)$$

Proposed Method

Interpretation of QUB Loss

$$\mathcal{L}_{\text{QUB}} = \mathcal{L}(f(x)) + (f(x+\delta) - f(x))^T \nabla_f \mathcal{L}(f(x)) + \frac{1}{4}\|f(x+\delta) - f(x)\|_2^2$$

First term

- Cross-entropy loss on clean samples enhancing standard accuracy

Third term

- Maintaining consistent model outputs before and after perturbation
- Securing robustness by preventing changing in results due to δ

Proposed Method

Interpretation of QUB Loss

$$\mathcal{L}_{\text{QUB}} = \mathcal{L}(f(x)) + \boxed{(f(x+\delta) - f(x))^T \nabla_f \mathcal{L}(f(x))} + \frac{1}{4}\|f(x+\delta) - f(x)\|_2^2$$

Second term

Approximation of the second term using the **chain rule**

$$(f(x+\delta) - f(x))^T \nabla_f \mathcal{L}(f(x)) \approx \delta^T \nabla_x \mathcal{L}(f(x)).$$

- The inner product between the δ and the loss gradient decreases when the two directions are **misaligned**
- Minimizing this term reduces the adversarial effect on the loss, thereby increasing robustness

Training Strategy

Algorithm 1 AT with Static QUB Loss

Input: network architecture f parameterized by θ, batch size B, batched training data $\{x_i, y_i\}_{i=1}^{B}$, training epoch T, perturbation generation method P

Output: Adversarially robust network f

for $t = 1$ **to** T **do**

  for $i = 1$ **to** B **do**

    $\delta = P(f, x_i, y_i)$

    Use Equation (7) to compute $\mathcal{L}_{\textbf{QUB}}$

    $\theta \leftarrow \theta - \nabla_\theta \mathcal{L}_{\text{QUB}}$

  end for

end for

QUB-Static

- Using **any existing metho**d for inner maximization (generate δ)
- Calculating loss with **QUB Loss** instead of Adversarial Training Loss

Training Strategy

Algorithm 2 AT w/ Decreasing Weight on QUB Loss

Input: network architecture f parameterized by θ, batch size B, batched training data $\{x_i, y_i\}_{i=1}^{B}$, training epoch T, perturbation generation method P

Output: Adversarially robust network f

for $t = 1$ **to** T **do**
  $\lambda_t = t/T$
  for $i = 1$ **to** B **do**
    $\delta = P(f, x_i, y_i)$
    $\mathcal{L}_{\text{AT}} = \mathcal{L}(f(x_i + \delta), y)$
    Use Equation (7) to compute $\mathcal{L}_{\text{QUB}}$
    $\mathcal{L}_{\text{total}} = (1 - \lambda_t) \cdot \mathcal{L}_{\text{QUB}} + \lambda_t \cdot \mathcal{L}_{\text{AT}}$
    $\theta \leftarrow \theta - \nabla_\theta \mathcal{L}_{\text{total}}$
  end for
end for

QUB-Decreasing

- Upper bound optimization focuses on worst case
  → often resulting in overly pessimistic training
- Can cause unnecessary trade-off with standard accuracy, even when robust is sufficient
- Proposed: QUB-decreasing scheduling (Start with QUB, then linearly decrease and transition to AT)

Experiments

Datasets: CIFAR-10, CIFAR-100, Tiny ImageNet

Models: ResNet-18, WRN-34-10, PreActResNet-18

Baselines:

- Iterative methods (PGD, TRADES)
- single-step methods (e.g., FGSM-RS, FGSM-CKPT, ELLE-A, etc.)

Evaluation: Standard Accuracy, Robust Accuracy, Dominant eigenvalue, Sparsity

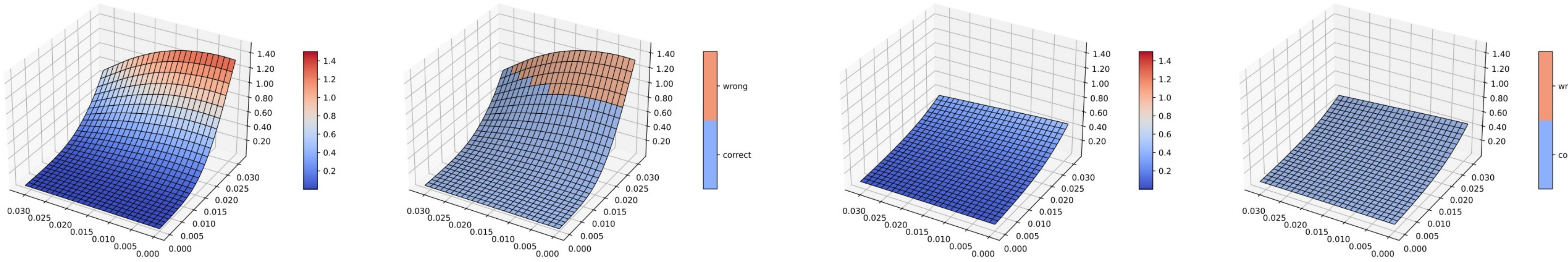
Experiments

Table 1. Test robustness (%) on the CIFAR-10 dataset using ResNet18 architecture. Number in bold indicates the best.

Method	Step	SA	PGD10	PGD20	PGD50-10	AA	Time (h)
no AT	-	**94.64**	0.00	0.00	0.00	0.00	0.57
NuAT	1	82.99	51.40	50.33	49.60	47.70	1.36
GAT	1	81.64	**54.78**	**53.87**	53.30	47.96	1.45
TRADES	10	82.11	54.25	53.39	52.77	**50.16**	3.50
Free-AT	1	75.99	45.32	44.74	44.27	41.38	0.3
+ QUB-static	1	72.98	46.72	46.19	45.89	42.82	0.56
+ QUB-decreasing	1	76.10	45.58	44.89	44.35	41.60	0.56
FGSM-RS	1	84.32	47.28	45.60	44.66	43.34	0.86
+ QUB-static	1	71.13	42.96	42.19	41.54	38.48	1.16
+ QUB-decreasing	1	72.90	43.85	42.96	42.52	39.31	1.16
FGSM-CKPT	1	90.02	41.19	38.81	37.42	37.22	1.05
+ QUB-static	1	87.63	45.41	43.78	42.54	41.53	1.35
+ QUB-decreasing	1	88.56	43.87	41.88	40.70	39.85	1.35
FGSM-GA	1	82.93	49.89	48.53	47.74	45.75	3.02
+ QUB-static	1	79.75	52.24	51.33	50.82	47.33	3.27
+ QUB-decreasing	1	81.83	50.88	49.83	49.07	46.74	3.27
FGSM-PGI(MEP)	1	81.48	53.43	52.47	51.75	48.41	0.89
+ QUB-static	1	80.45	53.99	53.16	52.43	48.35	1.19
+ QUB-decreasing	1	81.56	53.95	52.99	52.24	48.58	1.19
N-FGSM	1	81.21	49.12	48.02	47.36	45.17	0.58
+ QUB-static	1	80.76	51.19	50.24	49.60	47.00	0.70
+ QUB-decreasing	1	80.77	50.30	49.35	48.70	46.60	0.70
FGSM-UAP	1	81.62	53.38	52.59	51.83	47.75	1.18
+ QUB-static	1	79.70	54.25	53.51	52.77	47.76	1.49
+ QUB-decreasing	1	80.54	54.07	53.32	52.43	47.80	1.49
ELLE-A	1	82.14	47.91	46.39	45.57	43.52	0.97
+ QUB-static	1	77.60	50.20	49.44	48.86	45.51	1.21
+ QUB-decreasing	1	80.96	49.70	48.62	47.88	45.55	1.21
PGD-AT	10	81.53	52.99	52.30	51.82	48.33	2.34
+ QUB-static	10	80.24	54.58	**53.87**	**53.39**	49.91	2.64
+ QUB-decreasing	10	82.78	53.33	52.31	51.58	49.02	2.64

- Failure to prevent catastrophic overfitting in FGSM-RS
- Consistent **performance gains** with QUB across methods (except FGSM-RS)
- QUB-static: Clear SA trade-offs
- QUB-decreasing: Reduced trade-offs + SA improvements (achieving **superior balance**)

Loss Landscape Visualization



(a) (b)

Figure 1. Loss landscape for a specific sample: (a) model trained with FGSM-CKPT and (b) with FGSM-CKPT + QUB. The left side shows colors based on the loss value, and the right side shows colors based on prediction accuracy.

- **Flatter** loss landscape–less sensitivity to perturbations
- Improved defense over **a wider region**

For full results, please refer to the paper.

Conclusion

- **Convexity-based robust loss**: Introduced a novel loss function leveraging convexity to enhance adversarial robustness

- **QUB minimization**: Replaced standard AT loss with the quadratic upper bound (QUB) of cross-entropy loss for optimization

- **Seamless FAT integration**: Demonstrated compatibility with existing Adversarial Training frameworks

- **Empirical validation**: Achieved enhanced robustness across diverse experimental setups and evaluation metrics

ICML 2025

Quadratic Upper Bound for Boosting Robustness

Thank you for listening!

Presenter: Euijin You, yuj0508@konkuk.ac.kr
Corresponding Author: Prof. Hyang-Won Lee, leehw@konkuk.ac.kr

Connected Intelligence LAB