

ANITI

Communauté  
d'universités  
et établissements  
de Toulouse

U

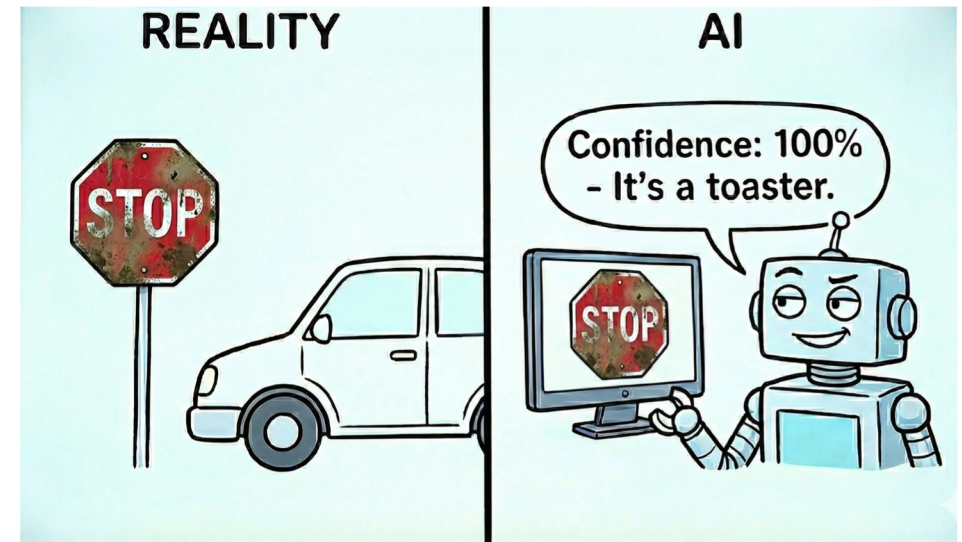


# LiST: Lipschitz Scaling Training for Robust and Calibrated Neural Networks

Arthur Chiron, Franck Mamalet, Thomas Massena, Mathieu Serrurier

# ANITI The Accuracy-Robustness-Calibration Trilemma

- Standard Neural Networks are extremely **accurate** but:
  - They are **brittle** <sup>[1,2]</sup>: low/specific noise can substantially alter their predictions.
  - They are often **over-confident** <sup>[3]</sup>, according high certainty even when they're wrong.



[1] Szegedy et al., Intriguing Properties of Robust Classification, arXiv 2014

[2] Goodfellow et al., Explaining and Harnessing Adversarial Examples, arXiv 2015

[3] Guo et al., On Calibration of Modern Neural Networks, ICML 2017

# ANITI Lipschitz-constrained Neural Networks (LNNs)

- Lipschitz Networks are **robust** but:
  - It is needed to **tune** <sup>[4]</sup> the constant  $L$  to **navigate** the Accuracy-Robustness trade-off.
  - The chosen value is usually **fixed**.
  - Their calibration properties remains **under-explored**.

Lipschitz Constant  $L$



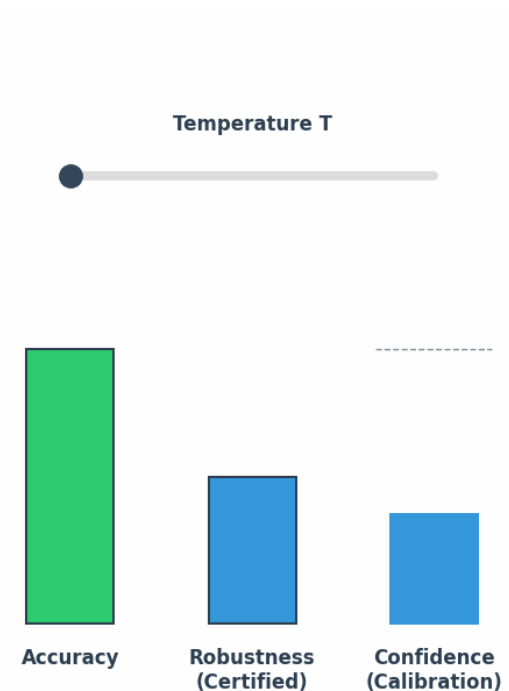
Accuracy

Robustness  
(Certified)

[4] Béthune et al., Pay Attention to your loss, understanding misconceptions about Lipschitz neural networks, NeurIPS 2022

# ANITI Calibration of Neural Networks

- Standard Network can be calibrated using Temperature Scaling <sup>[3]</sup>:
  - **Find  $T$**  such that  $f/T$  is **calibrated**.
  - $T$  is computed using a **calibration set**.
  - Post-hoc operation, the accuracy and the robustness of the model is **unchanged**.



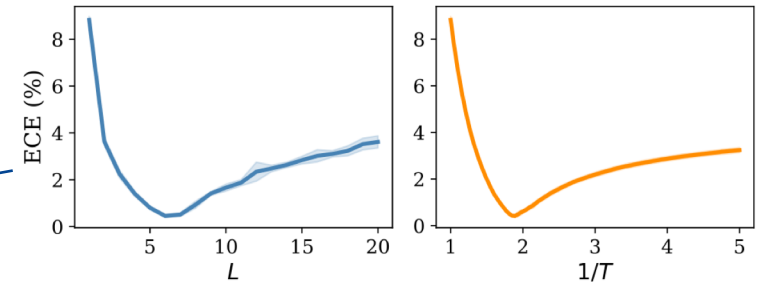
[3] Guo et al., On Calibration of Modern Neural Networks, ICML 2017

# ANITI LiST: Lipschitz Scaling Training

1. We find that tuning  $T$  is **functionally equivalent** to tuning  $L$ .

# ANITI LiST: Lipschitz Scaling Training

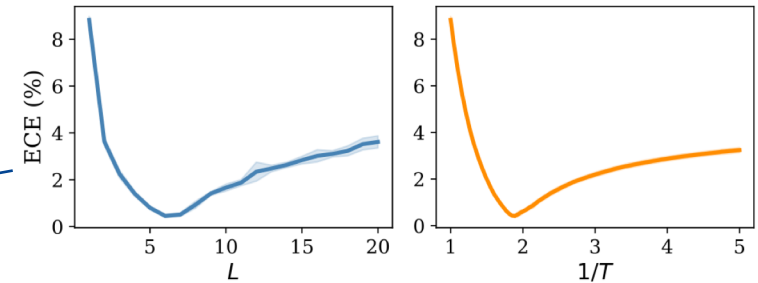
1. We find that tuning  $T$  is **functionally equivalent** to tuning  $L$ .



# ANITI LiST: Lipschitz Scaling Training

1. We find that tuning  $T$  is **functionally equivalent** to tuning  $L$ .
2. We use this duality to **dynamically tune  $L$  (LiST)** by doing TS at each epoch, using the rule:

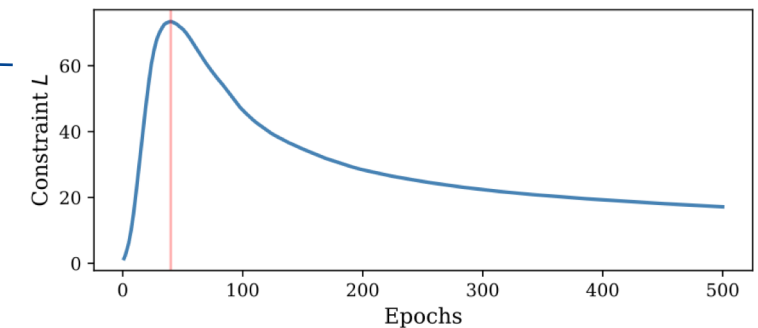
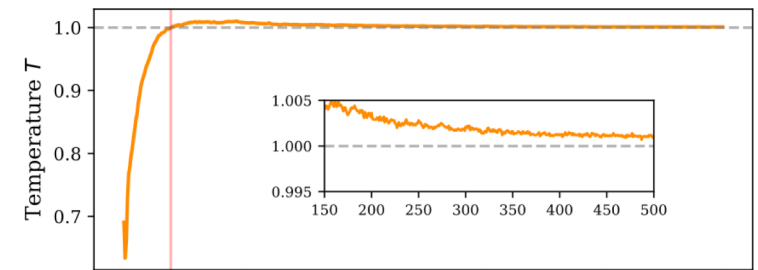
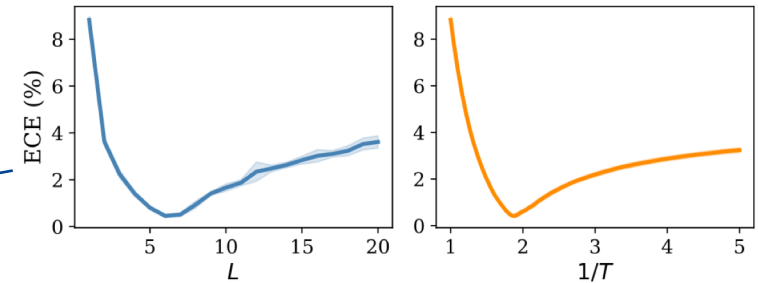
$$L_{e+1} = \frac{L_e}{T}.$$



# ANITI LiST: Lipschitz Scaling Training

1. We find that tuning  $T$  is **functionally equivalent** to tuning  $L$ .
2. We use this duality to **dynamically tune  $L$  (LiST)** by doing TS at each epoch, using the rule:

$$L_{e+1} = \frac{L_e}{T}.$$





# ANITI Conclusion

- Our algorithm enables to train Lipschitz networks that are:

**Accurate      Robust      Calibrated**

- Submission is coming...
- Come talk w/ us at the Poster Session !



### Motivations & Background

Reliable deployment requires two properties often treated separately:

- **Robustness:** Stability against adversarial perturbations.
- **Calibration:** Reliable confidence scores.

The **Accuracy-Robustness Trade-off:** For a Lipschitz Network, the **Certified Radius**  $R(x)$  is lower-bounded by the margin divided by the Lipschitz constant  $L$  (Tsuzuku et al., 2018):

$$R(x) \geq \frac{\gamma - \max_{y \neq x} \xi_y}{\sqrt{2}L}$$

The **Dilemma:** Maximizing robustness requires minimizing  $L$ , but a static low  $L$  restricts expressivity and hurts accuracy.

**Goal:** LIST eliminates the manual tuning of  $L$  by using calibration as a dynamic proxy for the optimal trade-off.

### The Lipschitz-Temperature Duality

We identify a structural link between Post-hoc Temperature Scaling ( $T$ ) and the Lipschitz constant ( $L$ ). Given a network  $f(x)$ , scaling the logits by  $T$  is functionally equivalent to scaling the global Lipschitz constant:

$$\text{Calibrated Output: } \hat{y} = \sigma\left(\frac{f(x)}{T}\right) \iff \text{New Lipschitz Constant: } L' = \frac{L}{T}$$

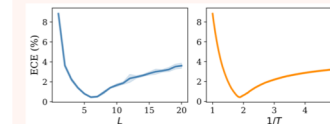


Figure 1. Empirical evidence: Varying the training constraint  $L$  produces an Error Calibration profile identical to varying post-hoc Temperature  $T$ .

**Pareto Selector Hypothesis:** The "calibrated" Lipschitz constant ( $L'$  where  $T \approx 1$ ) is not arbitrary. It corresponds to a principled equilibrium on the Pareto front, balancing under-fitting (low  $L$ ) and instability (high  $L$ ).

### LIST: The Algorithm

LIST automates the search for  $L'$  using a dynamic feedback loop.

**Mechanism (Iterative Update):** At each epoch, we measure the optimal temperature  $T^*$  on a validation set:

1. **Optimize Weights:** Minimize Cross-Entropy under current  $L_t$ .
2. **Feedback Control:**
  - If  $T^* > 1$  (Over-confident)  $\rightarrow$  **Tighten** ( $L_{t+1} = L_t/T^*$ ).
  - If  $T^* < 1$  (Under-confident)  $\rightarrow$  **Relax** ( $L_{t+1} = L_t/T^*$ ).

### Implementation: Adaptive Constraint Distribution

How do we enforce the dynamic global constraint  $L_t$ ?

We do not constrain layers uniformly. Instead, we distribute the total Lipschitz budget  $L_t$  across layers using a learnable weight vector  $w$ :

$$\phi_k(x) = \sigma_k \cdot g_k(x) \text{ where } \sigma_k = L_t^{\alpha_k}, \quad \alpha = \text{softmax}(w)$$

- $g_k$ : Normalized base layer (1-Lipschitz).
- $\sigma_k$ : Learned scaling factor.

This allows the network to automatically allocate **expansive** dynamics ( $\alpha_k > 1$ ) to critical layers while keeping others **contractive**, provided  $\sum \alpha_k = L_t$ .



arthur.chiron@irit.fr

ANITI Days

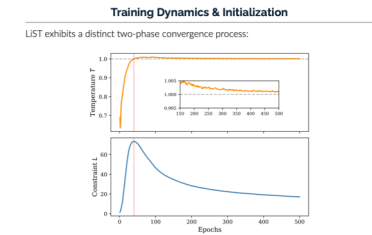


Figure 2. Top: Temperature converges to 1. Bottom: The Lipschitz constant  $L$  evolves dynamically.

- **Phase I (Relaxation):** Initial under-confidence triggers a rapid increase in  $L$ , allowing feature learning.
- **Phase II (Contraction):** As the model learns, LIST tightens  $L$  to maximize robustness while maintaining calibration.

**Initialization insensitivity:** Regardless of the initial  $L_0$  (spanning orders of magnitude), the feedback loop acts as a global attractor, driving the network to the same structural equilibrium.

### Preliminary Results: Pareto Efficiency

Note: Large-scale benchmarks (ImageNet) are currently being finalized. LIST targets the optimal trade-off between Clean Accuracy and Robustness (AutoAttack).

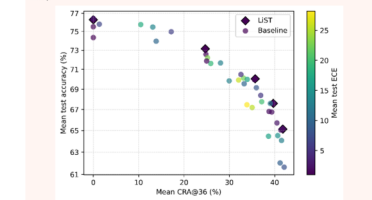


Figure 3. Pareto Analysis on CIFAR-10. Fixed-L baselines (circles) vs. LIST (diamond). Color indicates Calibration Error (ECE).

- **Automatic Selection:** LIST (diamond) naturally lands on the Pareto front.
- **Low ECE:** The method achieves minimal Expected Calibration Error compared to aggressive static constraints.
- **No Hyperparameters:** Unlike baselines, LIST does not require selecting a specific  $L$  a priori.

### Conclusion

- We introduced **Lipschitz Scaling Training (LIST)**, a method that:
1. **Unifies** Calibration and Certified Robustness objectives.
  2. **Automates** the tuning of the Lipschitz constant via a temperature feedback loop.
  3. **Converges** to a structural equilibrium that is both robust and calibrated out-of-the-box.

### References

- Coz et al., On Calibration of Modern Neural Networks, ICML 2017.
- Tsuzuku et al., Lipschitz Margin Training, NeurIPS 2018.