DATA-AWARE TRAINING QUALITY MONITORING AND CERTIFICATION FOR RELIABLE DEEP LEARNING

[SUPPLEMENTARY MATERIAL]

Anonymous authors
Paper under double-blind review

A THE DECREASING BEHAVIOUR OF YES-0 BOUND

Theorem 1. *Let Ω be an activation function in a deep neural network. If Ω is applied in an element-wise manner and satisfies the following conditions:*

- 1-Lipschitz Condition*:*

$$\|\Omega(\mathbf{x}_1) - \Omega(\mathbf{x}_2)\| \leq \|\mathbf{x}_1 - \mathbf{x}_2\|, \quad \forall\, \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n, \tag{1}$$

- Projection Property*:*

$$\Omega(\mathbf{Y}) = \mathbf{Y}, \quad \text{if} \quad \mathbf{Y} \in H_\Omega, \tag{2}$$

then the YES-0 bound is monotonically decreasing with respect to the depth of the network. That is, for each layer k:

$$\|\mathbf{Y} - \mathbf{Y}_{k+1}\|_{\mathrm{F}}^2 \leq \|\mathbf{Y} - \mathbf{Y}_k\|_{\mathrm{F}}^2. \tag{3}$$

Proof. Following our formulations, the error at layer $(k-1)$ is

$$\mathbf{E}_{k-1} = \mathbf{Y} - \mathbf{Y}_k, \tag{4}$$

where $\mathbf{Y}$ is the target output, and $\mathbf{Y}_k$ is the network output after $(k-1)$ layers. At each layer, the network updates its output via:

$$\mathbf{Y}_{k+1} = \Omega(\mathbf{A}_k \mathbf{Y}_k), \tag{5}$$

with $\mathbf{A}_k$ representing the weight matrix associated with the YES-0 bound at layer k. The error at layer k is thus:

$$\mathbf{E}_k = \mathbf{Y} - \Omega(\mathbf{A}_k \mathbf{Y}_k). \tag{6}$$

By considering the 1-Lipschitz and projection properties of the activation function Ω, we have:

$$\begin{aligned} \|\mathbf{E}_k\|_{\mathrm{F}}^2 &= \|\mathbf{Y} - \Omega(\mathbf{A}_k \mathbf{Y}_k)\|_{\mathrm{F}}^2 \\ &= \|\Omega(\mathbf{Y}) - \Omega(\mathbf{A}_k \mathbf{Y}_k)\|_{\mathrm{F}}^2 \\ &\leq \|\mathbf{Y} - \mathbf{A}_k \mathbf{Y}_k\|_{\mathrm{F}}^2. \end{aligned} \tag{7}$$

Since $\mathbf{A}_k$ is the minimizer of the quadratic criterion $\|\mathbf{Y} - \mathbf{A}_k \mathbf{Y}_k\|_{\mathrm{F}}^2$, we have:

$$\|\mathbf{Y} - \mathbf{A}_k \mathbf{Y}_k\|_{\mathrm{F}}^2 \leq \|\mathbf{Y} - \mathbf{Y}_k\|_{\mathrm{F}}^2 = \|\mathbf{E}_{k-1}\|_{\mathrm{F}}^2. \tag{8}$$

Combining (8) with (7) completes the proof. □

In Fig. 1, we validate Theorem 1 by demonstrating the decay of the YES-0 bound across layers. This result is based on the phase retrieval model.

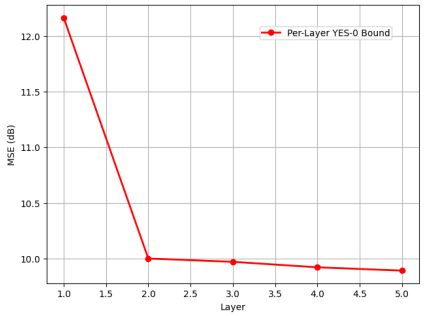


Figure 1: YES-0 decay with respect to the number of layers.

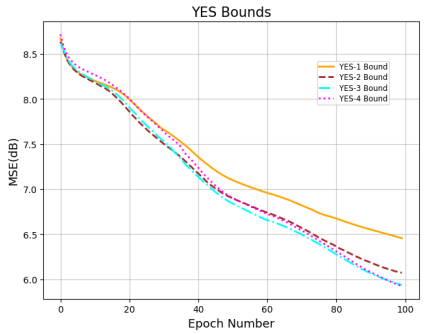


Figure 2: YES training bounds with varying degrees, without imposing monotonicity, are presented. As can be observed, increasing the degree of the bound does not necessarily improve it, as the bounds remain closely aligned with each other.

B Non-Decreasing Behaviour of YES-k bounds Without Monotonicity

YES training bounds with different degrees, without imposing monotonicity, are shown in Fig. 2 for the phase retrieval model. An interesting observation from this figure is that increasing the degree does not necessarily improve the YES bounds. In fact, all the bounds remain relatively close to each other.

Algorithm 1 Generation of the YES Training Bounds ($k \geq 1$).

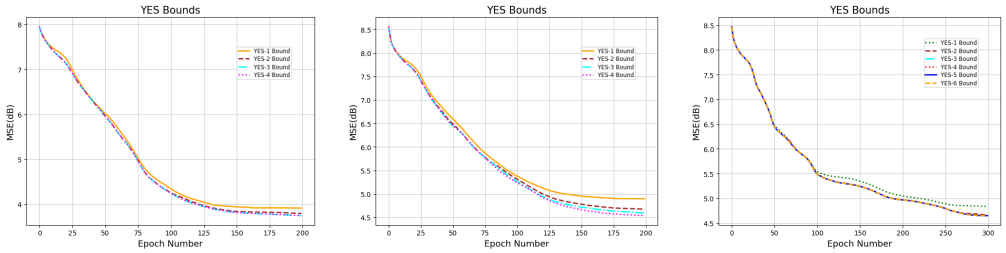
Input: $(\mathbf{X} \in \mathbb{R}^{n \times d}, \mathbf{Y} \in \mathbb{R}^{m \times d})$ are training data matrices with d denoting the number of training samples.
Output: YES training bounds.

$\mathcal{I} \leftarrow \{2, \cdots, K\}$.
$\mathbf{e} \leftarrow \mathbf{0}_{K-1}$.
for $k = 1 : (K-1)$ **do**
  $\mathcal{H} \leftarrow \text{combination}(\mathcal{I}, k)$ ▷ $\text{combination}(\mathcal{I}, k)$ is the combination operator that selects k items from the set $\mathcal{I}$.
  $\mathbf{u} \leftarrow \mathbf{0}_{|\mathcal{H}|}$ ▷ $\mathbf{0}_{|\mathcal{H}|}$ denotes a zero vector with the length $|\mathcal{H}|$.
  for $i = 0 : |\mathcal{H}| - 1$ **do**
    $\mathcal{H}_i \leftarrow \mathcal{H}[i]$ ▷ $\mathcal{H}[i]$ denotes the i-th combination item of $\mathcal{H}$.
    $l \leftarrow 0$.
    $\mathbf{Y}^\star \leftarrow [\,]$ ▷ $[\,]$ denotes an empty tensor.
    for $j = 0 : (k-1)$ **do**
      $\mathbf{Y}^\star \,.\,\text{append}(\text{model}_{\mathcal{H}_i}(\mathbf{X}))$ ▷ $\mathbf{Y}^\star \,.\,\text{append}(\mathbf{T})$ denotes appending the matrix $\mathbf{T}$ in an empty tensor $\mathbf{Y}^\star$, $\text{model}_{\mathcal{H}_i}(\mathbf{X})$ denotes the output of the training model at specific layers specified by the elements in $\mathcal{H}_i$.
    end for
    $\mathbf{Y}_t \leftarrow \mathbf{X}$.
    for $j = 0 : (k-1)$ **do**
      while $l \leq \mathcal{H}_i[j]$ **do**
        $\mathbf{A}_t \leftarrow \mathbf{Y}^\star[j]\mathbf{Y}_t^\dagger$.
        $\mathbf{Y}_t \leftarrow \Omega\left(\mathbf{A}_t\mathbf{Y}_t\right)$.
        $l \leftarrow l + 1$.
      end while
    end for
    for $z = 1 : K - l - 1$ **do**
      $\mathbf{A}_t \leftarrow \mathbf{Y}\mathbf{Y}_t^\dagger$.
      $\mathbf{Y}_t \leftarrow \Omega\left(\mathbf{A}_t\mathbf{Y}_t\right)$.
    end for
    $\mathbf{u}[i] \leftarrow \|\mathbf{Y} - \mathbf{Y}_t\|_{\text{F}}^2/d$.
  end for
  $\mathbf{e}[k-1] \leftarrow \min\ \mathbf{u}$.
end for
YES bound $\leftarrow \min\ \mathbf{e}$.
return YES bound

C THE YES-k TRAINING BOUNDS ($k \geq 1$) WITH MONOTONICITY

In Algorithm 1, we reformulate the YES-k bounds for $k \geq 1$, incorporating a monotonic modification through the inclusion of YES-k subsets to ensure the bounds remain monotonic.

For the YES bounds with monotonicity, as illustrated in Fig. 3 with various initializations, it is evident that the YES bounds are closely grouped. We investigated this observation using fully-connected networks with both 5-layer and 7-layer architectures, conducted this experiment 1000 times, and consistently observed similar results. This observation suggests that we may leverage the advantages of higher-degree YES bounds by calculating only the first few YES-k bounds, which could be beneficial from a computational standpoint.



(a) (b) (c)

Figure 3: YES training bounds with varying degrees, this time incorporating monotonicity across different initializations, are presented. Similar to the non-monotonic case, increasing the degree of the bound does not necessarily enhance it, as the bounds remain close to each other.

D Test Results For Training Process Monitored By YES Clouds

Beyond the training process, it is insightful to investigate how test results behave as the training progresses through different regions of the color-coded clouds. To explore this, we present both training and test outcomes for the phase retrieval model in Fig. 4. As observed, when the training loss decreases in the red region, the test loss similarly declines. When training plateaus in the yellow region, the test loss also plateaus. Interestingly, upon entering the green region, the training loss initially exhibits fluctuations, likely due to the learning rate, before plateauing—a pattern mirrored in the test loss. However, after approximately 2000 epochs, the test loss increases.

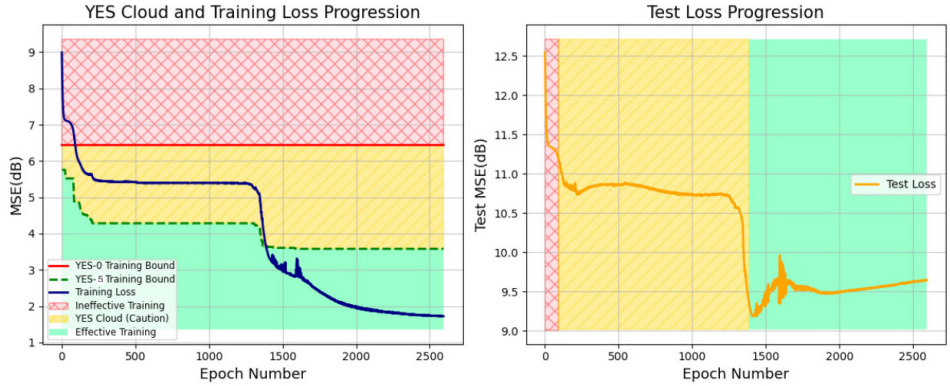


Figure 4: The YES bounds cloud for the training process is presented alongside the test stage for the same training results monitored by the YES training bounds.

E IMAGE RECOVERY FROM CORRUPTED QUADRATIC MODEL

To elucidate the practical significance of the YES bounds and their associated cloud system, we further examine an image recovery task for an image degradation process characterized by the model:

$$\mathbf{b}_i = |\mathbf{A}\mathbf{x}_i|^2 + \mathbf{n}_i, \quad i \in [d], \tag{9}$$

where each $\mathbf{x}_i$ represents a distinct patch of the original image undergoing recovery. This model presents a complex degradation process that involves three primary challenges:

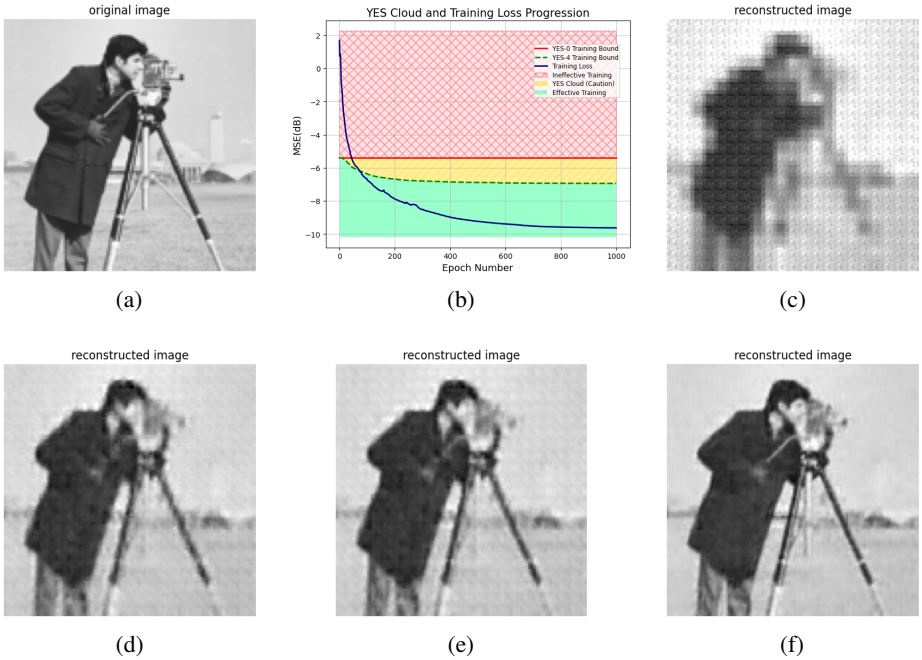


Figure 5: Monitoring the training process for the 5-layer fully-connected network used to reconstruct the cameraman image, as shown in (a), from the corrupted phase retrieval model, with the YES bounds cloud illustrated in (b). The quality of the reconstructed image is presented at different stages of training: (c) at the initial training loss, (d) as the training loss enters the cautionary yellow region, (e) when it reaches the green region, and (f) when the training loss converges to the final solution in the green region. As observed from the reconstructed images, the training performance improves progressively as the loss moves from the yellow region to the green region, achieving the best performance upon convergence. This demonstrates the effectiveness of the YES bounds cloud in monitoring the training process, even for tasks like image denoising.

1. Blurring Operation ($\mathbf{A}\mathbf{x}_i$): The matrix $\mathbf{A}$ applies a blurring operator to the image patch $\mathbf{x}_i$, necessitating deblurring techniques to counteract the smoothing effects.

2. Phase Loss ($|\cdot|^2$): The absolute value squared operation results in phase loss, requiring phase retrieval methods to restore essential phase information for accurate reconstruction.

3. Additive Noise ($\mathbf{n}_i$): The term $\mathbf{n}_i$ introduces additive noise, demanding denoising strategies to mitigate its adverse effects on image quality.

The performance of the proposed YES bounds and cloud system is illustrated in Fig. 5 and Fig. 6, which present the model's recovery performance under four distinct conditions:

- *Training Loss at Initial Value:* At the outset, the training loss is significantly higher than the YES bounds, indicating suboptimal performance. The recovery results at this stage exhibit pronounced blurring, substantial phase distortions, and noticeable noise artifacts, reflecting the model's nascent state.

- *Training Loss at YES-0 (Top of the Cloud):* As training progresses, the loss approaches the YES-0 bound—the top of the cloud. At this juncture, the model achieves a moderate level of recovery, with reduced blurring and phase errors, alongside diminished noise. However, the performance remains below optimal, as indicated by the fact that the training loss has not yet breached the lower bounds of the cloud.
- *Training Loss at YES-*$(K-1)$ *(Bottom of the Cloud):* Further training brings the loss down to the YES-$(K-1)$ bound—the bottom of the cloud. The recovery results at this stage demonstrate significant improvements, with minimal blurring, accurate phase reconstruction, and negligible noise. This indicates that the model is nearing the performance limits as prescribed by the YES bounds.
- *Optimized Convergence:* Upon convergence, the training loss reaches its optimal value, falling within the YES bounds. The recovery results are exemplary, showcasing precise deblurring, flawless phase retrieval, and excellent noise suppression. This final stage confirms that the model has achieved a state of optimal performance, as validated by the YES bounds.

To numerically validate the training results monitored by the YES bounds cloud, we apply these bounds to a corrupted phase retrieval model using two different images: the 128×128 cameraman and the boat, where we consider the patch size of 8×8 of these images for the model in (9). These images help assess the effectiveness of the YES cloud across various regions and determine whether the training loss entering the green region can indeed lead to practical solutions for real-world tasks, such as image denoising. Figs. 5 and 6 (a) show the original image of the cameraman and the boat, respectively, (b) present the YES cloud used for tracking the training loss with the YES bounds, (c) illustrate the initial results when the training loss is in the red region, (d) show the output as the loss enters the cautionary yellow region, (e) depict the outcomes when the training loss reaches the green region, indicating effective training according to the YES bounds, and finally, (f) shows the point at which the training loss converges. As illustrated in Fig. 5(c) and Fig. 6(c), the quality of the reconstructed images at the initial stage of training is poor, with noticeable noise, blurring, and patch artifacts. However, in Fig. 5(d) and Fig. 6(d), these imperfections are reduced compared to the earlier stage. Interestingly, by the time the training loss enters the green region, as shown in Fig. 5(e) and Fig. 6(e), the background of the images appears much clearer, with a significant reduction in noise and blurring artifacts. Finally, once the training fully converges in the green region, we observe a well-reconstructed input version, with most distortions effectively removed.

Note that while dropping below the YES cloud signals entering into the effective training regime, it is certainly not recommended to stop the training once this occurs. In fact, it would be reasonable to continue the training, e.g, as long as the rate of decrease in the loss is satisfactorily large. However, we show in this example that one may expect satisfactory performance even if they stop the training prematurely in the green. Based on this, we recommend the following criteria for stopping the training: 1) in the green, 2) the rate of change in network weights is sufficiently low.

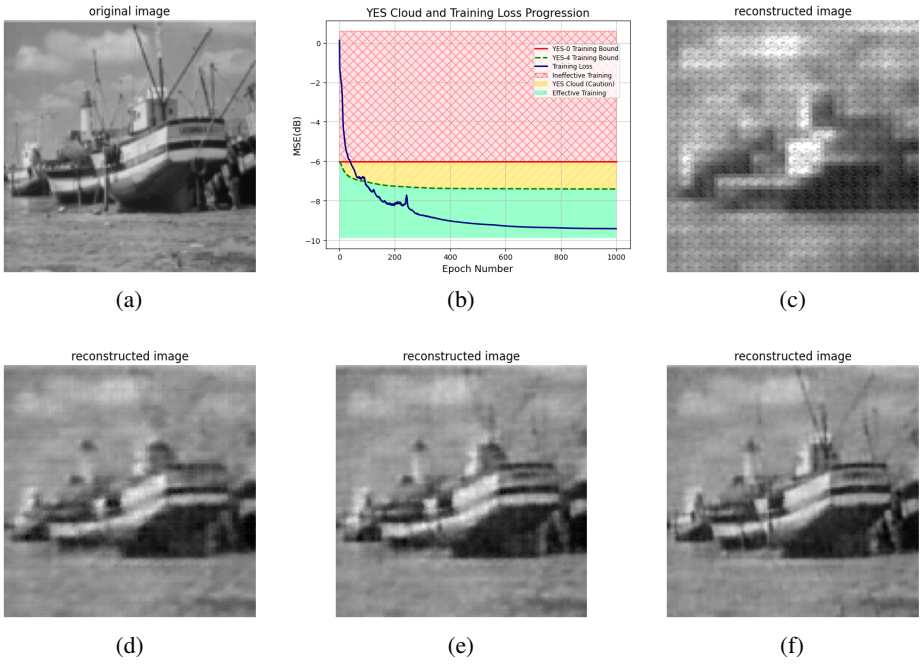


Figure 6: The training process of the 5-layer fully-connected network, which was utilized to reconstruct the boat image from a corrupted phase retrieval model, was closely monitored. The YES bounds cloud, depicted in (b), guided the process. The progression of the image quality through various training stages is showcased: (c) at the initial training loss, (d) as the loss enters the cautionary yellow region, (e) upon reaching the green region, and (f) when the loss stabilizes in the green region, indicating the final solution. The reconstructed images clearly show that as the training loss transitions from yellow to green, the performance steadily improves, culminating in optimal performance at convergence.

F MNIST CLASSIFICATION

To further assess the performance of our YES bounds in practical scenarios, we conducted experiments using the MNIST dataset, which was designed for classification tasks. We worked with 5000 training and 5000 test samples. Each image, representing a digit $i \in \{0, \cdots, 9\}$, was encoded by generating a zero matrix with the same dimension as the input image with a single 1 placed at $(i+1, i+1)$. A 5-layer fully connected network was trained with SGD, using an initial learning rate η_0 and a decay factor of 0.7 every 50 epochs. The classification was performed by minimizing the MSE between model outputs and encoded images, with the success rate determined by accurate classifications over the entire dataset.

As shown in Fig. 7(a), with a learning rate of $1e-4$, the training loss struggles to move beyond the caution region and remains close to the bottom of the YES clouds. In terms of success rates, Fig. 7(b) displays the training process, while Fig. 7(c) presents the test stage. Although the performance appears satisfactory, the YES cloud suggests that the model's solution is akin to a linear projection, indicating suboptimal training parameters. Adjusting these parameters could lead to improved model performance.

In Fig. 7(d), we apply a learning rate of $5e-4$ for the solver. In this case, the training loss reaches the green region after approximately 30 epochs. The success rate for the training results, shown in Fig. 7(e), indicates that when the training loss enters the yellow region, the success rate is 85 percent. Once it enters the green region, the success rate increases to 95 percent, and at the convergence point, we achieve 100 percent performance. For the test results depicted in Fig. 7(f), the loss reaches the yellow region at 83 percent, and upon entering the green region, the success rate becomes 92 percent. At the convergence rate, the test results reach 95 percent. As discussed earlier, when the training loss plateaus in the green region, the model's solution can be a strong candidate for optimality.

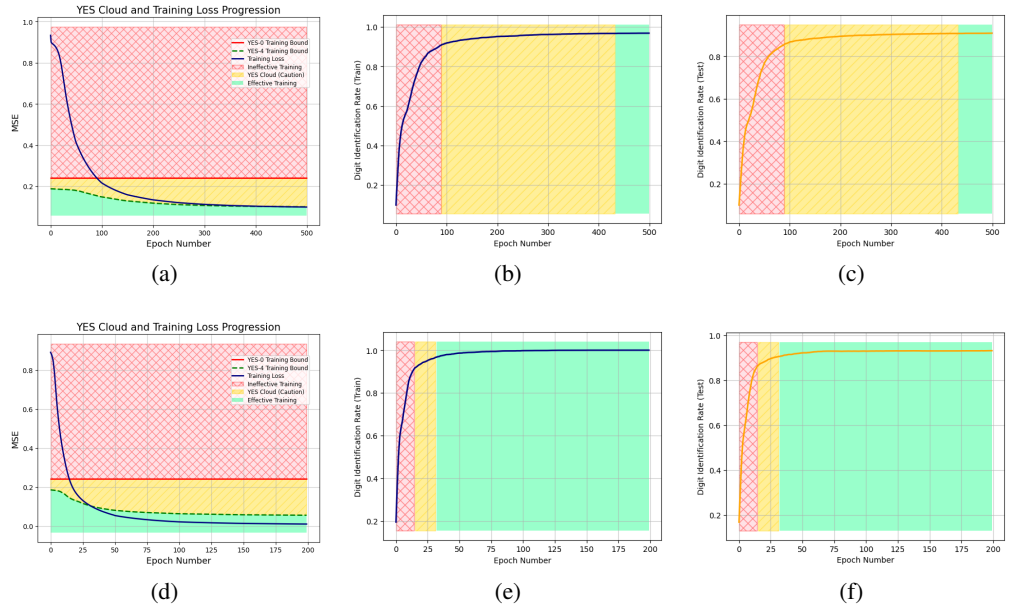


Figure 7: The YES bounds cloud for the training process on the MNIST dataset is presented alongside the success rates for both the training and testing stages. Figs. (a) and (d) show the YES clouds for solvers with different learning rates: (a) $1e-4$ and (b) $5e-4$. Figs. (b) and (c) display the success rates during the training and testing phases, respectively, within the color-coded YES cloud regions. These figures demonstrate how effectively the YES bounds monitor solver performance using a learning rate of $1e-4$. Figs. (e) and (f) illustrate the success rates within the YES cloud regions for the solver using a learning rate of $5e-4$.

Figs. 7(e) and (f) illustrate the model's effectiveness, achieving a 100% success rate in training and 95% in testing. This demonstrates the model's high accuracy and generalization, indicating that it is well-tuned to the task at hand.

G BEYOND CERTIFICATION: CAN WE GUIDE THE TRAINING PROCESS?

It is our understanding that the YES bounds and their associated clouds provide not only a mathematical framework for certifying AI training but may also aid practitioners as a method to guide the training process. To harness this dual utility without compromising the integrity of the certification process, it is imperative to maintain a clear information barrier between training and certification. This separation ensures that the training algorithm does not gain access to the YES bound data or the certification network weights. Allowing such access would undermine the certification's purpose by enabling the training process to exploit the bound information, leading to several adverse consequences:

- *Loss of Randomization Benefits:* Randomization, particularly during initialization and throughout training, plays a crucial role in escaping local minima and ensuring robust convergence. If the training process can access YES bound data and network weights, it may inadvertently eliminate these randomization benefits, resulting in deterministic and potentially suboptimal training trajectories.
- *Faulty Optimization Directions:* The training algorithm might adopt step directions that do not align with the true optimization landscape. Since there is no guarantee that the certification network weights resemble the optimum, leveraging these weights could steer the training process in misleading directions, ultimately degrading the quality of the trained model.
- *Obsolescence of the Certification Test:* The primary purpose of the certification test is to provide a reliable bound on the network's training performance. If the training process can consistently operate at or below this bound by utilizing certification data, the test will be rendered ineffective.

To mitigate the risks associated with direct access to certification data, it is essential to devise mechanisms that allow the training process to benefit from the YES bounds without exposing the certification test itself. One effective strategy is to share only the *test results*, such as those visualized through the YES cloud, rather than the underlying certification data or network weights. This approach provides the training algorithm with a lower bound on the distance between the current loss and the optimal loss without revealing any specific information about the certification criteria. Specifically, the distance of the current loss from the bottom of the YES cloud serves as a valuable indicator for adjusting the learning rate:

$$d_k = \max\{\mathcal{L}_k - \mathcal{L}_{\text{YES}}, 0\}, \tag{10}$$

where $\mathcal{L}_k$ is the current loss at epoch k, and $\mathcal{L}_{\text{YES}}$ represents the lower bound provided by the YES cloud. This distance d_k may inform the training process on how large of a step size could be chosen to make meaningful progress, ensuring that the learning rate adapts dynamically based on the proximity to the optimal loss. A natural implementation of this guidance mechanism involves defining an adaptive learning rate that incorporates both the traditional vanishing component and an additional term based on the distance d_k to the YES bounds.