A. Theoretical Analysis of AT-PR Algorithm Convergence

As discussed in Sec. 3.2, our algorithm's effectiveness relies on two convergence assumptions: 1) Multi-Start PGD with random hyperparameters may find all local optima in Step 1; 2) Boundary search with adaptive hyperparameter may detect the local boundary in Step 2. We now provide their *proof sketch*, which will be included with more details in the main paper.

Convergence of Multi-Start PGD. It is known from the original PGD paper [36] that under *standard* assumptions of differentiability, Lipschitz continuity, and convex feasible set, a single PGD run converges to a local stationary point of $L(x+\delta;\theta)$ (the local "loss landscape"). Assume L contains K distinct local maxima $\{\delta_k^*\}_{k=1}^K$ corresponding to adversarial regions of varying sizes, with associated attraction basins $\{\mathcal{A}_k\}_{k=1}^K$. Let P_k denote the probability that a random PGD starting point $\delta_0 \sim \text{Uniform}(\mathcal{B}(x,\gamma))$ lies within $\mathcal{A}_k$. Then, the probability that after N random initializations of PGD runs with varying step size, iterations, at least one PGD run finds a perturbation in $\mathcal{A}_k$ is:

$$\mathbb{P}(\text{at least one PGD finds } \delta_k^*) = 1-(1-P_k)^N. \tag{5}$$

Since the events are independent, the probability that all K local optima are found after N runs is:

$$\mathbb{P}(\text{find all}) = \prod_{k=1}^{K}\left[1-(1-P_k)^N\right]. \tag{6}$$

This probability approaches 1 as $N \to \infty$.

Convergence of Boundary Search. The convergence of the boundary search step is already established in [9], where it is shown that under *standard* assumptions of local Lipschitz continuity and non-vanishing gradients near the decision boundary, it converges to an ϵ-approximate boundary point in $O(1/\epsilon)$ iterations. We apply this search directly, on the set of adversarial examples (AEs) obtained from Step 1.

B. Hyperparameter Selection and Implementation Details

B.1. Experiment Setup

All experiments are conducted on one NVIDIA GeForce RTX 4090 GPU, Python 3.11, PyTorch 2.3.1. For the CIFAR-10, CIFAR-100, and SVHN datasets, we independently train ResNet-18, and WideResNet-50-2 on each dataset, and additionally include a Vision Transformer (ViT) for CIFAR-10. For the TinyImageNet dataset, we train ResNet-18 and ResNet-34 using the same training configuration as for CIFAR-10. All models are trained using stochastic gradient descent (SGD) with a momentum of 0.9 [45] and a weight decay coefficient of 5.0×10^{-4}. Training is performed for 200 epochs with an initial learning rate of 0.01, which is decayed by a factor of 10 at epochs 60, 120, and 150.

B.2. Training Algorithms Hyperparameter Setting

- **FGSM.** We use a one-step gradient attack with $\gamma = 8/255$, applying FGSM at each training step for AT.
- **PGD.** We set the perturbation radius to $\gamma = 8/255$ for all dataset. During training, we performed 10 steps of projected gradient descent attack, using a step size of $\alpha = 2/255$ for CIFAR-10, CIFAR-100, and TinyImageNet, and a step size of $\alpha = 1.25/255$ for SVHN.
- **TRADES.** We used the same step size and number of steps as described above for PGD. Additionally, we applied a weight of $\lambda = 6.0$ for all datasets, following the approach in [66].
- **MART.** We used the same step size and number of steps as described above for PGD. Additionally, we applied a weight of $\lambda = 5.0$ for all datasets, following the approach in [57].
- **ALP.** Follow the original work [29], we set $\lambda = 1$ for all datasets, except $\lambda = 0.5$ for SVHN.
- **CLP.** Following the same setting as ALP, we also set $\lambda = 1$ for all datasets, except $\lambda = 0.3$ for SVHN.
- **AT-PR.** For Algo. 1 Step 1, we apply PGD attacks to generate a diverse set of AEs. We set the size of AE candidate sets as $N = 10$, which our experiments show is sufficient to capture this diversity. We sample step sizes from $\alpha_{\text{min}} = 0.004$ to $\alpha_{\text{max}} = 0.01$, and attack steps from $step_{\text{min}} = 7$ to $step_{\text{max}} = 12$ with $\gamma = 8/255$. For the boundary search in Steps 2 and 3, we set the maximum number of iterations to $C = 20$.

C. Sketch Analysis of Generalization Errors

Following the recent **PAC-Bayes analysis for AR** [54], the generalization in PR can be bounded as:

$$R_{\text{PR}} \leq \hat{R}_{\text{PR}} + \text{TV}(\Pi\|\Delta) + \sqrt{\frac{\text{KL}(Q||P)+\log(1/\delta)}{2n}}, \tag{7}$$

where R_{PR} is the expected PR error (measured over the true data distribution), $\hat{R}_{\text{PR}}$ is the empirical PR error (measured over the training data), Q and P are the posterior and prior distributions of the DL model weights (measured by KL divergence). TV is the *total variation* of Π and Δ which are distributions on inputs and perturbation norm-balls, i.e., $\mathcal{X}\times\mathcal{Y}\times\mathcal{B}$. Δ is the true natural perturbation distribution (same to the one used in PR evaluation); and Π denotes the AE distribution generated by the training procedure. Our AT-PR, by focusing on the "widest" adversarial regions rather than only "peaky" worst-case regions, leading to a Π that better aligns with Δ. This reduces the total variation TV term and results in a tighter generalization bound compared to AT-WCR.