LOAT: LATENT-ORDER ADVERSARIAL TRAINING FOR EFFICIENT AND TRANSFERABLE ROBUSTNESS

Anonymous authors
Paper under double-blind review

ABSTRACT

Adversarial training remains computationally prohibitive due to the uniform application of expensive PGD (projected gradient descent) attacks across all training samples. Although prior works identify "hard" samples deserving of more computational effort, such approaches require supervised definitions of difficulty and do not capture the complex dynamics of how neural networks naturally learn robust representations. We present Latent-Order Adversarial Training (LOAT), a novel unsupervised method that discovers the emergent structure in adversarial training. It clusters adversarial dynamics using multiple complementary feature views to cluster structural similarities and identify an adaptive path of compatible learned dynamics to more efficiently train sub-models via a generalized set of probabilistic choices. By combining the inherent descriptors in an evolutionary learning model, LOAT creates a global model to transfer a transition matrix T that captures empirical patterns of how training naturally flows between clusters. Experiments on CIFAR-10 demonstrate that this discovered structure can efficiently and adaptively allocate PGD steps per cluster, following the learned transition, reducing computational cost by 40-50% while maintaining comparable or better robustness. The transferable global structure of our algorithm contains learnable generalizable patterns independent of potentially biased human notions. LOAT shows that respecting intrinsic dynamics yields significant efficiency gains without sacrificing robustness.

1 INTRODUCTION

Adversarial training Madry et al. (2018), where a model is trained with not only standard training data, but also adversarial examples generated from an attack, is known to provide a robust defense against the challenge of correctly identifying perturbed image samples. While strong, this approach requires sufficiently large capacity and great computational cost, requiring multiple PGD (projected gradient descent) steps per sample per epoch, limiting the practical deployment due to time and space constraints. Many approaches have been developed to address this, categorically broken down into reducing per-sample cost through fewer PGD steps Shafahi et al. (2019), or focusing computation on targeted, important samples He et al. (2024). The natural assumption in both categories is that one knows *a priori* what makes a sample important for robust learning. Although intuitively, presumed hardness has no forced bearing or correlation, which is to say, there is no foreknowledge beyond heuristic reasoning to necessitate an assumed approach. Targeting samples with small margins or high loss and then allocate resources to them is, in essence, an ill-defined notion. This supervised definition of difficulty assumes that human-interpretable metrics capture what matters for robust learning. We however put forward the notion that unsupervised learning based on a multi-view of generalized statistics, geometry, confidence patterns, adversarial dynamics, gradient coherence, activation patterns, consistency metrics, and loss landscape, all weighted for learning output, can create a natural grouping without manual labeling, allowing for a discoverable order in curriculum or post-hoc learning during which a teacher can robustly impart cluster determinations and transitions, allowing students to be adaptive, predictive, and able to identify hidden organizational principles based on patterns which we as humans may lack the vocabulary to describe.

Current models label samples as difficult when accumulating more PGD (projected gradient descent) steps Madry et al. (2018); Zhang et al. (2019b), larger weights Balaji et al. (2019), or more frequent sampling Carmon et al. (2019), this, however, is a static approach; whereas our LOAT model recog-

nizes that a generalized and naturally emergent dynamic can better assess and categorize adversarial samples. We further note that difficulty levels may be periphery, subordinate, or inferior labels as compared to discoverable relationships (and orderings) which can benefit from interleaving, restructuring, or other potentially hidden paths.

Thus, we propose a fundamentally different perspective. Instead of imposing human notions of difficulty, we use LOAT to discover the natural organization converging to a robust model. After some amount of initial training epochs used to establish basic robust features, our unsupervised clustering reveals how the model has learned to organize the data space. Our novel transition matrix T, built from these converged patterns, captures a stable multi-view structure of the robust solution manifold wherein we track empirical flows between clusters, harnessing the latent grouping in a per cluster per epoch presentation. Thus, our key insight is that adversarial training naturally clusters based on compatible learning dynamics alongside a latent differentiability and ordering subscription of emergent patterns that can be used to improve efficiency without requiring supervised labels or a predetermined curriculum. This novel idea further provides a blueprint to accelerate future training via a teacher-student model that can be deployed to edge networks where much smaller compute power is available.

2 Related Work

One of the most highly relevant topics to our model is that of curriculum training, an approach that systematically increases the difficulty of adversarial examples presented during the training process, where weak attacks mitigate catastrophic forgetting and help with generalization, building to stronger attacks in a learned fashion, Cai et al. (2018). Another important idea is that of adaptive early stopping Al-Rimy et al. (2023), wherein different heuristic approaches are used to identify cut-off points, with some form of customized budgets per sample Cheng et al. (2020) taking the form of early stopping based on misclassification Zhang et al. (2020) or based on gradient alignment Sitawarin et al. (2020).

State of the art models focus on hardness He et al. (2024) but often target some metric such as accuracy in exchange for efficiency, or the opposite, that of boosting speed with less robustness Goodfellow et al. (2015). The TRADES model Zhang et al. (2020) uses a theoretical upper bound minimization algorithm for adversarial training, a concept that many (including us) harness, with others Ding et al. (2018) noticing the importance of misclassified examples in training, adding probabilities of prediction as a way to smoothly combine samples Wang et al. (2020).

In terms of efficiency, the most comparable model to LOAT is Free-AT Shafahi et al. (2019). In Free-AT one does a forward pass on a clean example and a backward pass to get gradients, and then simultaneously updates both the model parameters and input perturbation. The free part is the reuse of the same gradient for both model updates and adversarial perturbation. It achieves similar robustness to standard PGD adversarial training while being roughly as fast as natural training. It was demonstrated similarly on CIFAR-10.

While these works (such as Customized Adversarial Training, and Free-AT) have similarities, we differentiate and build on these by discovering semantic structure, arguing that groups are not random, per-sample, or naturally classifiable by hardness. We instead explore conceptual dependencies and their orderings at the cluster level (a more generalizable approach to allow for robustness), identifying emergent structure to guide order, transfer, and efficiency without imposing a curriculum. We focus on an approach that is both highly effective and efficient, so much so that it outperforms even the highly efficient Free-AT technique.

3 Mathematical Background

Adversarial training can be presented as a min-max optimization problem, searching for a minimum worst-case loss within a perturbation set for each sample Madry et al. (2018).

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta\in\mathcal{B}(x,\epsilon)} \ell(x+\delta, y; \theta) \right]. \tag{1}$$

Here we have some model parameter θ, sampled training data (x, y) from the data $\mathcal{D}$ with a loss function $\ell(*)$ calculated with adversarial examples convolved (added) by δ. Here the perturbation set is referenced as

$$\mathcal{B}(x, \epsilon) = \{\delta \mid x + \delta \in [0, 1],\ \|\delta\|_p \leq \epsilon\}\,, \tag{2}$$

where ϵ is the maximum perturbation magnitude, $\|\delta\|$ the quantified size using some norm fit to a range $[0, 1]$ Zhao et al. (2024).

We note that Projected Gradient Descent (PGD) is a widely-accepted k-step maximization techinque

$$x'_{t+1} = \mathrm{Proj}_{\mathcal{B}(x,\epsilon)}\left(x'_t + \alpha \cdot \mathrm{sign}\big(\nabla_{x'_t}\, \ell(x'_t, y; \theta)\big)\right), \tag{3}$$

wherein x'_t denotes the adversarial example at iteration t, α is the step size, and $\ell(x'_t, y; \theta)$ is the loss function with respect to the true label y and model parameters θ. The term $\nabla_{x'_t}\ell(\cdot)$ is the gradient of the loss with respect to the input, and $\mathrm{sign}(\cdot)$ applies the element-wise sign. Finally, $\mathrm{Proj}_{\mathcal{B}(x,\epsilon)}$ projects the perturbed input back into the allowed ϵ-ball $\mathcal{B}(x, \epsilon)$ around the original input x, ensuring that perturbations remain within the maximum budget ϵ Bottou (2010); Madry et al. (2018). And while other methods such as Fast Gradient Sign Method (FGSM) and Carlini and Wagner (CW) exist, for both practical and instructive purposes, we focus on PGD, as the strength of the attack can vary, allowing for robust testing.

Our loss model function, with sample x, true label y, and adversarial example x' is generated using a cluster-adaptive budget for cluster c with difficulty score D_c. The LOAT objective is

$$\mathcal{L}_{\text{LOAT}} = \underbrace{\alpha_c\, \mathrm{CE}\big(f_\theta(x), y\big) + \beta_c\, \mathrm{CE}\big(f_\theta(x'), y\big)}_{\text{(1) Robust risk}} \tag{4}$$

$$+ \underbrace{\lambda_{\text{trans}}\, \mathrm{R}\big(T, c, p_\theta(x')\big)}_{\text{(2) Transition regularizer}}. \tag{5}$$

where $f_\theta(x)$ is the model with parameter θ, $p_\theta(x) = \mathrm{softmax}(f_\theta(x))$ is the predictive distribution, $\mathrm{CE}(\cdot, \cdot)$ is the cross-entropy loss, x' is the adversarial example of x, generated with a cluster-specific number of PGD steps proportional to difficulty D_c, α_c, β_c are cluster-adaptive weights to balance clean vs. adversarial loss, T is the learned transition matrix between clusters, $R(T, c, p_\theta(x'))$ is the generic transition-based regularizer that encourages consistency with the structure encoded in T, and λ_{trans} is the weight on the transition regularizer. We note that the form of $R(\cdot)$ can vary. In our implementation, transition structure is enforced implicitly via cluster-aware sampling and adaptive attack budgets. The cluster-adaptive attacks and early stopping naturally lead to efficiency in the sampling. Other choices (e.g. divergence penalties such as KL) could also be used, but we emphasize that R is a general placeholder for any transition-consistency mechanism.

We further note that the knowledge-distillation term

$$\lambda_{\text{KD}}(D_c)\, \mathrm{KL}(p_T(\cdot|x'), p_\theta(x')) \tag{6}$$

can be added if a teacher distribution p_T is available.

Our main goal in the student-teacher AT model is to transfer learning and measure it by computational efficiency. We define an efficiency score as

$$\mathcal{E} \;=\; \frac{\text{Robust Accuracy (\%)}}{\text{PGD Calls (in millions)}}\,, \tag{7}$$

where robust accuracy is evaluated under a fixed adversarial budget, and PGD calls denote the total number of inner attack steps used during training. This metric normalizes robustness by computational effort, enabling direct comparison across methods with different attack step allocations.

4 METHODOLOGY

Our approach consists of two main phases: (1) teacher training for discovery, and (2) student training with the transferred curriculum. In LOAT, the teacher finds the latent structure, while the student inherits a compact recipe for efficient and robust training.

4.1 PHASE 1: TEACHER TRAINING WITH DISCOVERY

We initialize a teacher model f_T and train it with adversarial examples generated by projected gradient descent (PGD). For inputs x_i and perturbation radius ϵ, with step size α and maximum budget $S_{\max}$:

$$x^{adv} = \text{PGD}(x_i; \epsilon, \alpha, S_{\max}). \tag{8}$$

The teacher is initialized with several known optimal algorithms and parameters thereof. Our model uses the TRADES objective Zhang et al. (2019b), beginning with a SimCLR encoder for feature stabilization during pre-training, followed by an adversarially-trained autoencoder A_ϕ to produce robust latent embeddings. We use these embeddings to provide a stable basis for profiling and reduce noise in feature clustering.

Our multi-view feature extraction consists of several parts, each well known but heretofore not integrated together in an unsupervised model. For each batch B_i, we compute complementary feature sets of the (1) statistics (entropy distributions, adversarial vulnerability metrics, and gradient norms under weak perturbations), (2) geometry features (Bag-of-Embeddings, sliced Wasserstein distances to prototypes, low-rank covariance spectra, FFT-based frequency signatures, and Gram matrix eigenvalues from intermediate layers), (3) confidence patterns (prediction stability and entropy under noise), (4) adversarial dynamics (response curves across multiple ϵ values), (5) gradient coherence (gradient alignment and diversity metrics), (6) activation patterns (layer-wise activation statistics), (7) consistency metrics (prediction variance under input perturbations), and (8) loss landscape (local loss geometry through directional sampling).

We take these features and input them into a model to assess optimal learning weights via differential evolution optimization to learn continuous weights $[0, 1]$ for each feature where clustering quality takes into account a weighted combination of silhouette scores (cluster separation), Calinski-Harabasz index (between/within variance ratio), diversity metric (inter-cluster distinction), learning gradient potential (trainability differences), and robustness variance.

$$\max_{\mathbf{w}} \quad \sum_{v=1}^{V} w_v \, q_v \tag{9}$$

$$\text{s.t.} \quad \sum_{v=1}^{V} w_v = 1, \quad w_v \geq 0, \tag{10}$$

where q_v is the quality metric (e.g., silhouette, Calinski–Harabasz, etc.) for view v, and w_v are continuous weights optimized via differential evolution.

In short, the optimization discovers which feature combinations create the most learnable distinctions. For epochs 1-15 the teacher continually improves the model, and for epochs 16-30 the teacher tracks transitions between clusters, building a T matrix:

$$T_{ij} = \frac{C_{ij}}{\sum_j C_{ij}}, \tag{11}$$

where C_{ij} counts empirical transitions from cluster i to cluster j; rows are normalized to probabilities, with $T[i, j]$ representing the pedagogical value of teaching cluster i before cluster j, capturing the natural learning progression, prerequisite relationships, and synergistic cluster pairs. We use our

clusters to target specific patterns that can be learned, latent presentations in the data that is not categorized in preemptive notions, along with potential orderings that provide more efficient learning for edge computation.

Thus, the LOAT teacher distillation outputs feature weights, the T transition matrix, difficulty profiles for adaptive training, and proven paths that consistently improved learning.

4.2 Phase 2: Student Training with Transferred Curriculum

The student (an edge machine) receives the teacher's recipe to assign new batches to clusters without re-discovery, using the learned feature weights and distilled information to adaptively train against adversarial samples. It uses different transition strengths to adapt, requiring less PGD steps for stronger conceptual paths, with savings potentially ranging from $\tilde{3}0\%$-60% while maintaining robustness, creating greater efficiency for learned patterns. Student learning uses "difficulty" profiles D_c which are updated with an exponential moving average of robust error and PGD usage, guiding adaptive attack budgets per cluster.

$$D_c^{(t)} = \beta\, D_c^{(t-1)} + (1-\beta)\left(\hat{e}_c^{(t)} + \lambda\, \hat{s}_c^{(t)}\right), \tag{12}$$

where $D_c^{(t)}$ is the updated difficulty for cluster c at epoch t, $\hat{e}_c^{(t)}$ is the robust error rate, $\hat{s}_c^{(t)}$ is the normalized average PGD steps, and $\beta \in [0, 1]$ is the smoothing factor.

In order to account for variation, the student also employs UCB (upper confidence bound) reward, where cluster selection is balanced in exploration and exploitation targeting efficiency as robust accuracy per PGD call.

$$R_b = \frac{\text{Acc}_{\text{robust}}(b)}{\text{PGD}_{\text{calls}}(b)}, \tag{13}$$

UCB is a calculated value that guides the agent's decision-making by combining the estimated average reward of an action with an exploration bonus, where $\text{Acc}_{\text{robust}}(b)$ is robust accuracy on batch b, and $\text{PGD}_{\text{calls}}(b)$ is the total attack calls used, with UCB updates:

$$UCB_c = \hat{\mu}_c + \alpha\sqrt{\frac{\ln t}{n_c}}, \tag{14}$$

where $\hat{\mu}_c$ is the running mean reward for cluster c, n_c is the visit count, and t is the global timestep.

We note that teacher learning can occur at different stages, either as a continuously learned curriculum or as a post-hoc analysis only starting at later epochs, with approaches varying based on the dataset. Thus we present our novel unsupervised discovery model wherein we have a plethora of weighted metrics to classify without assumptions. LOAT is multi-scale and adaptive, able to identify patterns in adversarial samples, creating an online continuously refined curriculum that is learned during training wherein the teacher transfers the latent-ordering of knowledge to the student, allowing for an efficient presentation in a data-driven discovery of scaffolding perspectives to create the natural grouping of datasets.

4.3 Time and Space Complexity

Teacher Discovery. Clustering has complexity

$$O(N \cdot d \cdot K_f), \tag{15}$$

where N is the number of samples, d is feature dimension, and K_f is the number of clusters.

Student Training. Comparable to standard AT but with reduced PGD steps:

$$O\left(E_s \cdot N \cdot \overline{K(c)}\right), \tag{16}$$

where $\overline{K(c)} \ll S_{\max}$ is the expected PGD steps per cluster.

5 Numerical Analysis and Results

Teacher Model We trained the LOAT teacher model on CIFAR-10 with a ResNet-18 backbone, a standard test data set for adversarial training. We used 30 epochs to learn, a batch size of 128, discovery intervals of 10 epochs, 5 clusters, a TRADES beta of 6.0, a simCLR of 50 epochs, an autoencoder trainer of 20 epochs, and during cluster discovery we profiled varying degrees of PGD steps (2-30) to establish difficulty fingerprints. In the training phase, we used a fixed PGD of 10 steps, in the evaluation phase we used 20 steps with 2 restarts, and for the profiling (discovery) phase we tested with [2,3,5,7,10,15,20,30] to characterize discovery and create the proper clusters.

The matrix was built in epochs 15-30, with a low entropy of 0.154, indicating that there are structured patterns. Our teacher had a test clean accuracy of 0.798, a test robustness of 0.462, and an efficiency score (which is intentionally low for the model building) of 0.003. 12.78 million PGD calls were used in training and the transition matrix had the strongest paths for self-reference (e.g. $0 \rightarrow 0, 1 \rightarrow 1$, etc.), respectively, 0.793, 0.823, 0.806, 0.799, and 0.790, showing that the clusters were indeed learned and different.

The final cluster difficulties were 0: 0.455, 1: 0.465, 2: 0.458, 3: 0.435, 4: 0.443. This indicates that the difficulty associated with the clusters was well distributed, suggesting good conceptual grouping, with no outliers to the data set.

We reference Table 1, showing that as the model matured, the best combination score went down minimally, with epoch 30 unreported in the logs and epoch 10 outperforming the others. The most important factors for CIFAR-10 were consistency, loss landscape, adaptive dynamics, and confidence, with moderate weights being statistics, geometry, and activations; while gradient coherence (in epoch 10) had little relevance.

Student Models Our student model aimed for fast and effective computation that could be deployed at the edge. We used 10 epochs, looking for efficiency from our models more than any other metric. We tested (1) full adaptive/curriculum-based LOAT with no resampling, preferring looping in the same cluster with an 80% probability, but switching clusters once samples were finished, (2) the same as above but using resampling within the clusters, (3) LOAT without curriculum learning (i.e. uniform choice of cluster), and (4) a baseline of no LOAT.

We also tested comparable state-of-the-art models such as CAT (Customized Adversarial Training) and found that despite being an excellent model, it took longer to run than even our teacher, was not transferable to the edge, and had $> 3.5M$ PGD calls (on a comparative basis) versus the LOAT student which took $\approx$ 25 minutes on a NVIDIA GeForce RTX 4070 Ti Super GPU, had $< 2M$ PGD calls, and slightly better robustness. In general, models such as CAT, TRADES and others have understandably less efficiency at 10 epochs, with comparable or less robustness ($\approx 20\%$) and millions more PGD calls ($\approx 4M$) Liu et al. (2023), versus LOAT which uses a student-teacher distillation, able to efficiently learn with less than half of the PGD calls.

In terms of efficiency, Fast-AT has the best raw compute efficiency for state-of-the-art algorithms, but lower and often less stable robustness since Fast-AT is known to suffer from catastrophic overfitting unless carefully tuned Zhao et al. (2023) versus LOAT, which is not as efficient as Fast-AT but has much better robustness-to-cost ratio and stability. Thus, due to its instability, we did not run comparable 10-epoch studies on Fast-AT, as it is known to be unreliable. However, we did compare our model to Free-AT, as described earlier and shown in Table 2.

Our baseline of 10 epochs gave a model with slightly higher robustness (at 0.368) but used $> 4.7M$ PGD calls to achieve this, giving an efficiency score of 0.007, as seen in Table 2. We note that at least three random initializations were used for each case and that LOAT with no-resampling vs Free-AT had a t-statistic of 10.39, a p-value of .0000297 (highly significant), and Cohen's d of 5.61 (extremely large positive effect size). This shows that LOAT is more effective than current methods.

While LOAT without resampling performed comparable to a uniform student baseline (suggesting that clusters might not provide strong signal), our ablation analysis reveals the opposite. By systematically removing individual clusters, we found that excluding any one cluster consistently produced more efficient models than those in Table 3. We see from Table 4 that removal of any cluster was beneficial for the model and that cluster 0 as compared to cluster 1, etc. each had a high significance (with respect to Cohen's d and p-values), showing these differences are fundamental, not noise.

Table 1: Optimized feature weights across epochs.

Feature	Epoch 1	Epoch 10	Epoch 20
stats	0.899	0.638	0.157
geom	0.001	0.468	0.069
confidence	0.421	0.747	0.517
adv_dynamics	0.321	0.785	0.007
grad_coherence	0.544	0.007	0.463
activations	0.371	0.430	0.163
consistency	0.130	0.798	0.914
loss_landscape	0.087	0.786	0.457
Best combo score	**0.643**	**0.687**	**0.668**

This demonstrates that clusters encode meaningful structure and removing them yields measurably stronger and more efficient models. That is, while some clusters capture useful structure, others introduce negative transfer, likely due to conflicting gradient signals or optimization conflicts. The ability to identify and down-weight such clusters is precisely the strength of our approach. In other words, the clusters are meaningful in that they reveal that not all training examples contribute equally to adversarial robustness, and removing the harmful subsets directly improves both robustness and efficiency.

Table 2: Average performance across methods (CIFAR-10, $\epsilon = 8/255$).

Method	Clean Acc	Robust Acc	Training Calls	Efficiency
Baseline	0.6649	0.3661	4,747,804	0.00772
Free AT (m=4)	0.8083	0.2997	2,350,000	0.01278
Uniform	0.5916	0.2924	1,879,579	**0.01555**
LOAT (no reuse)	0.6012	0.3015	1,933,981	**0.01559**
LOAT (reuse)	0.6146	0.3109	2,249,795	0.01383

Note: Both Uniform (LOAT without curriculum) and LOAT (no reuse) achieve statistically significant efficiency over the baseline, Free-AT, and LOAT (reuse).

Table 3: Efficiency results across methods (CIFAR-10, $\epsilon = 8/255$). Higher is better.

Method	Mean	Std Dev	N	95% CI
Baseline	0.00772	0.00003	3	[0.00768, 0.00775]
Free AT (m=4)	0.01278	0.00009	3	[0.01268, 0.01288]
Uniform	0.01555	0.00013	3	[0.01540, 0.01569]
LOAT (no reuse)	0.01559	0.00070	7	[0.01507, 0.01611]
LOAT (reuse)	0.01383	0.00012	3	[0.01370, 0.01396]

6 Review and Summary

In this work, we introduced Latent-Order Adversarial Training (LOAT), a novel unsupervised approach to adversarial training that discovers emergent structure in the data and adapts attack budgets accordingly. Our experiments showed that CIFAR-10 naturally clusters into five stable groups, clearly differentiated. Our structure was robust enough to guide efficient training and to export the learned order to a student model. Unlike curriculum learning methods that require predefined hardness labels, LOAT learns directly from inherent features, discovering a landscape weighted via an evolutionary algorithm. We showed that compared to the baseline (which used double the PGD calls) and compared to state-of-the-art methods such as Free-AT (which had less efficiency), via the T matrix, our transferable model preserved robustness while reducing computational overhead.

We note that while five clusters provided meaningful differentiation, our tests indicated that three clusters did not. Future work could extend this to seven or more clusters to capture more subtle

Table 4: Cluster ablation results. Reported are means with 95% confidence intervals (CI) for efficiency. Removing different clusters yields distinct efficiency and robustness profiles, indicating that clusters encode meaningful structure.

Removed	Clean Acc	Robust Acc	Training Calls	Efficiency
0	0.576	0.278	1.47M	0.0189 [0.0185, 0.0193]
1	0.542	0.249	1.28M	0.0195 [0.0185, 0.0205]
2	0.569	0.277	1.60M	0.0173 [0.0170, 0.0176]
3	0.571	0.277	1.58M	0.0175 [0.0174, 0.0176]
4	0.591	0.289	1.77M	0.0163 [0.0163, 0.0164]

Algorithm 1 LOAT: Teacher Discovery and Student Transfer (Concise)

1: **Teacher (warmup).** Train the teacher with SimCLR and an autoencoder. Log how many attack steps each sample actually needed and the resulting robust errors.
2: **Teacher (periodic discovery).** At a regular interval:
   1. Encode a snapshot of the training data with a small adversarially-trained autoencoder to get robust embeddings.
   2. Build the feature views per sample.
   3. Use evolutionary methods to weight views and create clusters.
   4. Update a transition matrix that counts how batches move between clusters from the previous snapshot to the current one.
   5. For each cluster, update a difficulty score with an exponential moving average that combines recent robust errors and typical PGD steps actually used.
3: **Teacher (recipe).** Save a compact recipe: cluster centroids and normalizers, the transition matrix, the latest per-cluster difficulties, and the set of uncertain samples.
4: **Student (initialize).** Load the recipe. Set up a simple UCB (upper-confidence) chooser over clusters to balance exploration and exploitation during training.
5: **Student (training loop).** For each pass over the data:
   1. Pick the next cluster with the UCB chooser; prefer the transition suggested by the matrix from the most recent cluster.
   2. Draw a batch from that cluster. Set an attack budget per batch based on the cluster difficulty (e.g., small budget for "easy," medium for "moderate," larger for "hard," largest for "uncertain"). Always keep per-sample early stopping.
   3. Generate adversarial examples with the chosen budget and train the student (e.g., TRADES or cross-entropy on the adversarial batch).
   4. Compute a simple efficiency reward (robust accuracy achieved per total PGD calls for this batch). Update the UCB statistics.
   5. Refresh the cluster's difficulty score with an exponential moving average using the latest robust errors and median step usage.
6: **Output.** The trained student and the (optionally updated) recipe.

dynamics. Furthermore, our study focused on CIFAR-10, which is standard in adversarial training research and offers clear comparability to prior work. However, evaluating LOAT on more diverse datasets (e.g., CIFAR-100, Tiny-ImageNet, ImageNet-subsets) would further validate its generality. We evaluated robustness using PGD-20 with random restarts, a strong and widely adopted protocol that provides a fast, repeatable proxy for adversarial strength. Our study focused on robustness per unit of training compute, so using PGD-20 consistently across all methods allows us to compare efficiency at scale. To mitigate the risk of overestimation, we varied PGD settings (steps/restarts), confirmed monotonic success curves, and checked that no method exhibited signs of gradient obfuscation. We also ran AutoAttack on a small number of representative checkpoints as a verification (results not tabulated), which confirmed that PGD-20 captures the same relative trends. Because our goal is not to establish state-of-the-art absolute robustness but to measure efficiency trade-offs, we report PGD-based results for the full experimental grid.

In summary, our novel approach combines unsupervised discovery with adaptive efficiency. LOAT offers a middle ground between heavy PGD-based adversarial training and highly efficient but unstable Fast/Free-AT methods. Its emphasis on robust efficiency makes it a promising candidate for deployment in real-world applications where both adversarial robustness and computational feasibility are critical.

REFERENCES

B. A. S. Al-Rimy, F. Saeed, M. Al-Sarem, A. M. Albarrak, and S. N. Qasem. An adaptive early stopping technique for densenet169-based knee osteoarthritis detection model. *Diagnostics (Basel)*, 13(11):1903, May 2023. doi: 10.3390/diagnostics13111903.

Yogesh Balaji, Tom Goldstein, and Judy Hoffman. Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets. *arXiv preprint*, 2019.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010: 19th International Conference on Computational Statistics*, pp. 177–186, Paris, France, 2010. Springer.

Q. Cai, C. Liu, and D. Song. Curriculum adversarial training. *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 3740–3747, 2018. doi: 10.24963/ijcai.2018/520.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, and John C. Duchi. Unlabeled data improves adversarial robustness. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pp. 11192–11203, 2019.

Minhao Cheng, Qi Lei, Pin-Yu Chen, Inderjit Dhillon, and Cho-Jui Hsieh. Cat: Customized adversarial training for improved robustness, 2020.

Pablo de Jorge Aranda, Amirhossein Bibi, Riccardo Volpi, Amartya Sanyal, Philip Torr, Guillem Rogez, and Puneet Dokania. Make some noise: Reliable and efficient single-step adversarial training. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 12881–12893, 2022.

Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. Max-margin adversarial (mma) training: Direct input space margin maximization through adversarial training. *arXiv preprint arXiv:1812.02637*, 2018.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

Ham et al. Robust distillation for adversarial training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Shiyuan He, Jiwei Wei, Chaoning Zhang, Xing Xu, Jingkuan Song, Yang Yang, and Heng Tao Shen. Boosting adversarial training with hardness-guided attack strategy. *IEEE Transactions on Multimedia*, 26:7748–7760, 2024. doi: 10.1109/TMM.2024.3371211.

Binghui Li and Yuanzhi Li. Adversarial training can provably improve robustness: Theoretical analysis of feature learning process under structured data, 2025.

Xingbin Liu, Huafeng Kuang, Xianming Lin, Yongjian Wu, and Rongrong Ji. Cat:collaborative adversarial training, 2023. URL https://arxiv.org/abs/2303.14922.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

Leslie Rice, Eric Wong, and J. Zico Kolter. Overfitting in adversarially robust deep learning, 2020. URL https://arxiv.org/abs/2002.11569.

Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S. Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *arXiv preprint*, (arXiv:1904.12843), 2019.

N. D. Singh, Francesco Croce, and Matthias Hein. Revisiting adversarial training for imagenet: Architectures, training and generalization across threat models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2024.

Chawin Sitawarin, Supriyo Chakraborty, and David Wagner. Improving adversarial robustness through progressive hardening. *arXiv preprint*, (arXiv:2003.09347), 2020.

Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. On the convergence and robustness of adversarial training. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 6586–6595, 2019.

Yisen Wang, Difan Zou, Xingjun Ma, James Bailey, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=rklOg6EFwS.

Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training, 2020. URL https://arxiv.org/abs/2001.03994.

Cihang Xie, Mingxing Tan, Boqing Gong, Alan Yuille, and Quoc V. Le. Smooth adversarial training. *arXiv preprint arXiv:2006.14536*, 2020.

Dongxian Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, and Bin Dong. You only propagate once: Accelerating adversarial training via maximal principle. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 227–238, 2019a.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97, pp. 7472–7482. PMLR, 2019b.

Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger, 2020.

Mengnan Zhao, Lihe Zhang, Yuqiu Kong, and Baocai Yin. Fast adversarial training with smooth convergence, 2023. URL https://arxiv.org/abs/2308.12857.

Mengnan Zhao, Lihe Zhang, Jingwen Ye, Huchuan Lu, Baocai Yin, and Xinchao Wang. Adversarial training: A survey, 2024.

Haizhong Zheng, Ziqi Zhang, Juncheng Gu, Honglak Lee, and Atul Prakash. Efficient adversarial training with transferable adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Zhu et al. Fine-tuned adversarial training for robust generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

A AI Assistance Disclosure

We used AI tools to assist in the code generation, table building, and polishing of the writing. All choices, designs points, and final claims were made and verified by the authors. The authors take full responsibility for the paper's content, including any errors, and affirm that this does not diminish the originality of the paper.

B Pairwise Comparisons of Our Ablation Study

We further analyzed efficiency differences across clusters using pairwise statistical tests. Table 5 reports Cohen's d and p-values for all comparisons. The results demonstrate that the clusters are not interchangeable. That is, removing different clusters yields fundamentally different efficiency outcomes. Most comparisons show very large effect sizes (Cohen's $d > 5$) and are statistically significant ($p < 0.05$), confirming that the efficiency distributions are well separated. For example, removing cluster 4 produces the lowest efficiency (0.0163) and is significantly different from all other cluster removals (e.g., $d = 14.3$ vs. cluster 3, $p < 0.001$). By contrast, removing clusters 0 or 1 yields the highest efficiencies (≈ 0.0189–0.0195), significantly outperforming removals such as cluster 2 or 4. This validates our claim that clusters encode meaningful structure and that not all training examples contribute equally to adversarial robustness.

Table 5: Pairwise comparisons of cluster ablation efficiency. Reported are Cohen's d and p-values. Large effect sizes and low p-values indicate that clusters represent distinct groups.

Comparison	Cohen's d	p-value
no cluster0 vs no cluster1	-0.94	0.345
no cluster0 vs no cluster2	5.24	0.0042
no cluster0 vs no cluster3	5.29	0.0156
no cluster0 vs no cluster4	10.07	0.0053
no cluster1 vs no cluster2	3.42	0.0404
no cluster1 vs no cluster3	3.16	0.0583
no cluster1 vs no cluster4	5.03	0.0249
no cluster2 vs no cluster3	-1.22	0.245
no cluster2 vs no cluster4	5.11	0.0191
no cluster3 vs no cluster4	14.29	0.00028