BOOSTING ADVERSARIAL ROBUSTNESS AND GENERALIZATION WITH DICTIONARY STRUCTURE

Anonymous authors
Paper under double-blind review

ABSTRACT

This work investigates a novel approach to boost adversarial robustness and generalization by incorporating structural prior into the design of deep learning models. Specifically, our study surprisingly reveals that existing dictionary learning-inspired convolutional neural networks (CNNs) are robust against random noise but remain highly vulnerable to adversarial attacks. To address this, we propose Elastic Dictionary Learning Networks (EDLNets), a novel ResNet architecture that significantly enhances adversarial robustness and generalization. Extensive and reliable experiments demonstrate consistent improvements in adversarial robustness across multiple datasets, backbone architectures, and threat models. To the best of our knowledge, this is the first work to discover and validate that dictionary structure can reliably enhance deep learning robustness under strong adaptive attacks, unveiling a promising direction for future research. Our implementation is available at `https://anonymous.4open.science/r/ElasticDL-6343`.

1 INTRODUCTION

In adversarial machine learning, most works have demonstrated their effectiveness in safeguarding deep neural networks by refining the geometry and landscape of parameter space, including robust training (Madry, 2017; Zhang et al., 2019; Gowal et al., 2021) and regularization (Cisse et al., 2017; Zheng et al., 2016). Especially in the visual domain, adversarial training methods powered by generative AI (Wang et al., 2023; Gowal et al., 2021) have achieved remarkable success and dominated the robustness leaderboard (Croce et al., 2020). However, their success increasingly relies on extensive synthetic training data crafted by generative models and increasing networks capacity, posing critical challenges to breaking through the plateau in adversarial robustness.

In fact, adversarially trained neural networks improve robustness and generalization largely by memorizing adversarial perturbations during training (Madry, 2017). While increasing the backbone model size can enhance memorization capacity, adversarial training methods often suffer from catastrophic *robust overfitting* problem (Rice et al., 2020). Existing methods aim to mitigate robust overfitting from various perspectives, including regularization (Andriushchenko & Flammarion, 2020; Qin et al., 2019; Sriramanan et al., 2020), data augmentation (DeVries, 2017; Zhang, 2017; Carmon et al., 2019; Zhai et al., 2019), and generative modeling (Wang et al., 2023; Gowal et al., 2021). However, all these methods contribute along a similar trajectory by refining the training strategy, making it difficult to achieve further groundbreaking advancements with the same network capacity.

In this work, we propose to explore an orthogonal direction for innovation upon the finding that most state-of-the-art deep learning approaches largely ignore the *dictionary structural prior* in signal representations. Several works (Papyan et al., 2017; Cazenavette et al., 2021; Mahdizadehaghdam et al., 2019; Li et al., 2022) have explored robust architectures based on sparse coding or dictionary learning priors, assuming that a signal can be sparsely represented as a linear superposition of atoms from a convolutional dictionary. This approach enables natural image patterns to be captured by the sparse code while effectively denoising corruptions. However, their effectiveness has only been validated in scenarios involving weak random corruptions and universal perturbations. Its full potential remains unexplored in the pursuit of stronger adversarial robustness and generalization.

To fill this research gap, we revisit dictionary learning in deep learning, which uncovers its inability in handling adaptive adversarial attacks, supported by both empirical evidence and theoretical reasoning. To overcome these limitations, we propose a novel elastic dictionary learning (Elastic DL)

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

framework that complements existing adversarial training methods to achieve superior robustness and generalization. Our contributions are as follows:

- We revisit convolutional dictionary learning in deep neural networks and reveal the inherent sensitivity of its ℓ_2-based reconstruction objective to adversarially induced outliers. To address this limitation, we introduce Elastic Dictionary Learning (Elastic DL) by incorporating ℓ_1-based reconstruction, a novel framework that achieves a favorable trade-off between standard accuracy and adversarial robustness.
- We develop an efficient reweighted iterative shrinkage thresholding algorithm (RISTA) to approximate the non-smooth Elastic DL objective with theoretical convergence guarantees. The algorithm can be seamlessly integrated into deep learning models as a replacement for conventional convolutional layers to enhance all convolutional architectures.
- Extensive experiments demonstrate that our proposed Elastic DL framework can significantly improve adversarial robustness and generalization. When combined with high-performing defenses listed on the RobustBench (Croce et al., 2020) leaderboard, Elastic DL improves upon their original performance across multiple perturbation budgets under both ℓ_∞- and ℓ_2-norm attacks.

2 Preliminary: Rethinking Convolution via Dictionary Learning

Convolution is typically treated as a conventional feed-forward operation for capturing localized patch information in images. Here, we instead revisit convolution from a dictionary learning perspective, inverse-modeling the output embedding as the sparse code from which the input can be reconstructed via a learned convolutional dictionary.

Notations. Let the input signal be denoted as $\boldsymbol{\xi} \in \mathbb{R}^{H\times W}$ and the convolution kernel as $\boldsymbol{\alpha} \in \mathbb{R}^{k\times k}$, where $k = 2k_0 + 1$. The *convolution* and the *transposed convolution* of $\boldsymbol{\xi}$ and $\boldsymbol{\alpha}$ are defined as:

$$(\boldsymbol{\alpha}\star\boldsymbol{\xi})[i,j] = \sum_{p=-k_0}^{k_0}\sum_{q=-k_0}^{k_0} \boldsymbol{\xi}[i+p, j+q]\cdot\boldsymbol{\alpha}[p,q], \quad (\boldsymbol{\alpha}*\boldsymbol{\xi})[i,j] = \sum_{p=-k_0}^{k_0}\sum_{q=-k_0}^{k_0} \boldsymbol{\xi}[i-p, j-q]\cdot\boldsymbol{\alpha}[p,q].$$

Let the C-channel input signal be denoted as $\boldsymbol{x} = \{\boldsymbol{\xi}_1, ..., \boldsymbol{\xi}_C\} \in \mathbb{R}^{H\times W\times C}$, and D-channel the output signal as $\boldsymbol{z} = \{\boldsymbol{\eta}_1, ..., \boldsymbol{\eta}_D\} \in \mathbb{R}^{H\times W\times D}$. The convolution operator $\mathcal{A}(\cdot)$ and its adjoint transposed convolution operator $\mathcal{A}^*$ are associated with kernel $\mathbf{A}$ as:

$$\mathcal{A}(\boldsymbol{x}) = \sum_{c=1}^{C} (\boldsymbol{\alpha}_{1c}\star\boldsymbol{\xi}_c, ..., \boldsymbol{\alpha}_{Dc}\star\boldsymbol{\xi}_c), \quad \mathcal{A}^*(\boldsymbol{z}) = \sum_{d=1}^{D} (\boldsymbol{\alpha}_{d1}*\boldsymbol{\eta}_d, ..., \boldsymbol{\alpha}_{dC}*\boldsymbol{\eta}_d),$$

where the associated kernel $\mathbf{A} = \{\boldsymbol{\alpha}_{dc}\}_{d\in[D],c\in[C]} \in \mathbb{R}^{D\times C\times k\times k}$. Here, H, W, C, D, and k represent the height, width, input dimension, output dimension, and kernel size, respectively.

Conventional convolutional learning. Convolutional architectures exploit the spatial structure of visual inputs through local receptive fields and shared feature detectors (Fukushima, 1980; LeCun et al., 1989). Local connectivity captures spatial patterns, while weight sharing enables feature detection across different locations. The convolutional operation is expressed as $\boldsymbol{z} = \mathcal{A}(\boldsymbol{x})$, where $\mathcal{A}$ is a learnable convolutional operator, $\boldsymbol{x}$ is the input, and $\boldsymbol{z}$ is the resulting feature representation. Although effective for feature extraction, this operation alone does not explicitly constrain how the extracted features reconstruct the input. Convolutional dictionary learning provides a complementary perspective by treating feature extraction as structured sparse inference.

Dictionary learning in convolution. Prior studies (Papyan et al., 2017; Cazenavette et al., 2021; Mahdizadehaghdam et al., 2019; Li et al., 2022) incorporate dictionary-learning priors into neural architectures to make their representations more interpretable. Under this formulation, an input signal $\boldsymbol{x} \in \mathbb{R}^{H\times W\times C}$ is approximated by a sparse combination of atoms from a convolutional dictionary $\mathbf{A}$, expressed as $\boldsymbol{x} \approx \mathcal{A}^*(\boldsymbol{z})$, where $\mathcal{A}^*$ denotes the operator associated with $\mathcal{A}$. Rather than obtaining features through a single convolution, the sparse representation is inferred by solving

$$\min_{\boldsymbol{z}} \ \|\boldsymbol{x} - \mathcal{A}^*(\boldsymbol{z})\|_2^2 + \lambda \|\boldsymbol{z}\|_1, \tag{1}$$

where $\lambda > 0$ controls the trade-off between reconstruction fidelity and code sparsity. The reconstruction term encourages consistency with the learned dictionary, while the sparsity penalty favors representations supported by a small number of dictionary atoms.

Robustness implication of sparse coding in vanilla dictionary learning. Prior studies have reported promising robustness of sparse codes against random corruptions and universal adversarial perturbations (Cazenavette et al., 2021; Mahdizadehaghdam et al., 2019; Li et al., 2022), suggesting that this structural prior can suppress perturbations poorly represented by the dictionary while preserving informative features. This behavior arises from the interplay of the two reconstruction terms: the ℓ_2-reconstruction term promotes recovering the original input $\boldsymbol{x}$ from the dictionary $\mathcal{A}$ and code $\boldsymbol{z}$, while the ℓ_1-sparsity term encourages $\boldsymbol{z}$ to retain only the most informative atoms. Together, they suppress unstructured perturbations and noise that the dictionary cannot compactly represent.

Optimization algorithms in solving vanilla dictionary learning. For a fixed dictionary $\mathcal{A}$, Eq. (1) is convex but non-smooth due to the ℓ_1 penalty, and several iterative algorithms have been proposed to solve it. The Iterative Shrinkage-Thresholding Algorithm alternates between a gradient-descent step on the ℓ_2-reconstruction term and a proximal step that enforces sparsity via the soft-thresholding operator (Daubechies et al., 2004). FISTA builds on ISTA by extrapolating between consecutive iterates before each gradient step, using momentum from past updates to guide the search direction (Beck & Teboulle, 2009). Alternatively, the Alternating Direction Method of Multipliers (ADMM) introduces an auxiliary variable to split the problem into two decoupled subproblems, solving the quadratic data-fidelity term efficiently in the frequency domain via the Sherman-Morrison formula and the ℓ_1 term through the same soft-thresholding step (Papyan et al., 2017; Boyd et al., 2011).

3 Elastic Dictionary Learning Neural Networks

In this section, we first propose a novel elastic dictionary learning (Elastic DL), which aims to achieve the best accuracy-robustness trade-off. We then develop a RISTA algorithm to efficiently solve elastic DL with rigorous theoretical guarantees, and unroll it as Elastic Dictionary Learning Networks (EDLNets). An overview of the proposed architecture is presented in Figure 1.

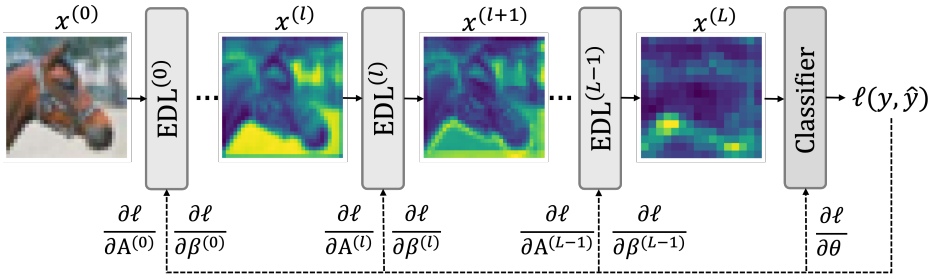


Figure 1: Overview of Elastic Dictionary Learning Networks (EDLNets). EDLNets are constructed by replacing the convolutional layers in conventional backbones (e.g., ResNets) with EDL layers obtained by unrolling the proposed efficient RISTA algorithm. Each EDL layer incorporates a dictionary-structure prior, assuming that the input signal $\boldsymbol{x}^{(l)}$ can be represented by a sparse code $\boldsymbol{x}^{(l+1)}$ using a small number of atoms from the dictionary $\mathbf{A}^{(l)}$. During backward propagation, the task loss is propagated through all unrolled RISTA iterations to jointly optimize the layer-wise dictionaries $\{\mathbf{A}^{(l)}\}_{l=0}^{L-1}$ and balancing parameters $\{\beta^{(l)}\}_{l=0}^{L-1}$, as well as the classifier parameters θ.

3.1 Elastic Convolutional Dictionary Learning

Vanilla dictionary learning is not truly robust. Despite the promising robustness of sparse coding in vanilla dictionary learning (Vanilla DL) reported in several works (Cazenavette et al., 2021; Mahdizadehaghdam et al., 2019; Li et al., 2022), such benefits do not necessarily extend to stronger adaptive attacks. To validate this, we conduct a preliminary experiment on CIFAR10 using a Vanilla DL variant of ResNet18 (Li et al., 2022), in which all convolutional layers are replaced with convolutional sparse coding layers based on Vanilla DL as formulated in Eq. (1). We evaluate this Vanilla DL network (with both a fixed λ and a tuned λ) under random impulse noise and an adaptive PGD adversarial attack (Madry, 2017) with budget $\frac{8}{255}$. As shown in Table 1, the Vanilla DL network improves upon the standard ResNet18 baseline in robustness against random noise,

Table 1: Preliminary study on CIFAR10 comparing standard ResNet18 with a Vanilla DL variant under varying levels of random noise and adaptive PGD attack ($\epsilon = \frac{8}{255}$). Vanilla DL exhibits strong robustness against weak random noise but is easily broken by the strong PGD attack.

Model \ Noise Level	L-1	L-2	L-3	L-4	L-5	PGD
Classic ConvNet (ResNet18)	81.44	57.23	48.32	32.49	16.98	**0.00**
Vanilla DL (Fixed $\lambda = 0.1$)	82.39	68.90	59.28	40.8	23.83	**0.01**
Vanilla DL (Tuned λ)	82.39	68.90	59.28	43.71	33.43	**0.13**

with a more pronounced improvement achieved by tuning the sparsity weight λ. However, it still suffers a sharp performance drop under the adaptive PGD attack, with accuracy falling to near zero. Detailed results across various noise levels and λ values are provided in Figure 11 in Appendix E.1. In fact, the ℓ_2-reconstruction term of Vanilla DL in Eq. (1) imposes a quadratic penalty $\|\cdot\|_2^2$ on the residual $\boldsymbol{x} - \mathcal{A}^*(\boldsymbol{z})$, making it highly sensitive to outliers introduced by high-level noise and adaptive attacks. The experimental results reveal that existing Vanilla DL is robust against random noise but can easily be compromised by adaptive attacks. Thus, a substantial gap remains in achieving truly robust dictionary learning in deep learning.

Elastic dictionary learning in convolution. Our design draws on a classical insight from robust statistics: the squared ℓ_2 loss used in least-squares estimation amplifies large residuals quadratically and is therefore highly sensitive to outliers, whereas the ℓ_1 loss underlying the Least Absolute Deviations (LAD) estimator grows only linearly with residual magnitude, making it far more robust to gross errors (Huber, 1992; Rousseeuw & Leroy, 2003). Motivated by this principle, we introduce an analogous elastic reconstruction penalty into convolutional dictionary learning, so that the resulting sparse code remains robust to adversarially-induced outliers in the reconstruction residual while preserving reconstruction fidelity on clean inputs. To achieve greater robustness while maintaining strong natural performance, we propose Elastic Dictionary Learning (Elastic DL), which combines squared ℓ_2 and ℓ_1 reconstruction penalties while retaining the sparse-coding prior:

$$\min_{\boldsymbol{z}} \ \mathcal{L}(\boldsymbol{z}) = \frac{\beta}{2} \left\|\boldsymbol{x} - \mathcal{A}^*(\boldsymbol{z})\right\|_2^2 + \frac{1-\beta}{2} \left\|\boldsymbol{x} - \mathcal{A}^*(\boldsymbol{z})\right\|_1 + \lambda \left\|\boldsymbol{z}\right\|_1, \tag{2}$$

where $\beta \in [0, 1]$ is a learnable parameter to control the contributions of the two reconstruction terms.

Why the elastic dictionary learning improves robustness. Eq. (2) admits a probabilistic reading via maximum likelihood estimation: the ℓ_2-reconstruction term implicitly assumes a light-tailed residual $\boldsymbol{x} - \mathcal{A}^*(\boldsymbol{z})$, which is reasonable for natural noise but lets a handful of adversarially large residuals dominate and distort $\boldsymbol{z}$, whereas the ℓ_1-reconstruction term corresponds to a heavy-tailed (Laplace) assumption that grows only linearly with residual magnitude, bounding the influence of such outliers. The two terms are thus complementary: ℓ_2 preserves reconstruction efficiency on clean inputs, while ℓ_1 suppresses adversarial outliers when present. The separate ℓ_1-sparsity penalty on $\boldsymbol{z}$ serves a different purpose, retaining only the most informative dictionary atoms and discarding noise-driven activations. Rather than fixing this trade-off a priori, we treat β as a learnable, layer-wise parameter optimized jointly with the dictionary during adversarial training, letting each layer adapt toward the reconstruction assumption that best balances accuracy and robustness.

Challenges in efficiently solving elastic dictionary learning. Despite its favorable robustness properties, the objective in Eq. (2) is considerably harder to optimize than its vanilla counterpart. In vanilla dictionary learning (Sec. 2), the reconstruction term in Eq. (1) is a smooth, convex quadratic, so the problem can be solved efficiently by taking a gradient step on this term and applying the soft-thresholding operator to enforce sparsity on $\boldsymbol{z}$. In elastic dictionary learning, however, the ℓ_1-reconstruction term introduces a second source of non-smoothness in addition to the ℓ_1-sparsity term, so the reconstruction term itself no longer admits a simple gradient. This precludes a direct extension of the standard ISTA-style update and calls for a dedicated algorithm capable of efficiently approximating the solution to Eq. (2). To the best of our knowledge, no prior work has proposed elastic convolutional dictionary learning and addressed how to unroll such a doubly non-smooth elastic dictionary learning objective into efficient, trainable neural network layers, leaving an important research gap that we address next.

3.2 Reweighted Iterative Shrinkage Thresholding (RISTA) Algorithm

To address the challenge posed above, we propose the Reweighted Iterative Shrinkage-Thresholding Algorithm (RISTA), which solves Eq. (2) by iteratively minimizing a convex *localized upper bound* of the original non-smooth objective, yielding a rigorous descent on the original loss at every step.

Localized upper bound. To address $\|\boldsymbol{x} - \mathcal{A}^*(\boldsymbol{z})\|_1$ term, we first propose a convex upper bound $\mathcal{U}(\boldsymbol{z}, \boldsymbol{z}_*)$ as an alternative in the following Lemma 3.1.

Lemma 3.1. *Let $\mathcal{R}(\boldsymbol{z}) := \|\boldsymbol{x} - \mathcal{A}^*(\boldsymbol{z})\|_1$, and for any fixed point $\boldsymbol{z}_*$, $\mathcal{U}(\boldsymbol{z}, \boldsymbol{z}_*)$ is defined as*

$$\mathcal{U}(\boldsymbol{z}, \boldsymbol{z}_*) = \|\boldsymbol{w}^{1/2} \odot (\boldsymbol{x} - \mathcal{A}^*(\boldsymbol{z}))\|_2^2 + \frac{1}{2}\mathcal{R}(\boldsymbol{z}_*), \tag{3}$$

where $\boldsymbol{w} = \frac{1}{2|\boldsymbol{x}-\mathcal{A}^*(\boldsymbol{z}_*)|}$. *Then, for any* $\boldsymbol{z}$, *the following holds:*

$$(1)\, \mathcal{U}(\boldsymbol{z}, \boldsymbol{z}_*) \geq \mathcal{R}(\boldsymbol{z}), \quad (2)\, \mathcal{U}(\boldsymbol{z}_*, \boldsymbol{z}_*) = \mathcal{R}(\boldsymbol{z}_*).$$

Proof. Please refer to Appendix B.1. □

The statement (1) indicates that $\mathcal{U}(\boldsymbol{z}, \boldsymbol{z}_*)$ serves as an upper bound for $\mathcal{R}(\boldsymbol{z})$, while statement (2) demonstrates that $\mathcal{U}(\boldsymbol{z}, \boldsymbol{z}_*)$ equals $\mathcal{R}(\boldsymbol{z})$ at point $\boldsymbol{z}_*$. With fixed $\boldsymbol{z}_*$, the alternative objective $\mathcal{U}(\boldsymbol{z}, \boldsymbol{z}_*)$ in Eq. (3) is quadratic and can be efficiently optimized. Therefore, instead of minimizing the non-smooth $\mathcal{R}(z)$ directly, we can alternatively optimize the quadratic upper bound $\mathcal{U}(\boldsymbol{z}, \boldsymbol{z}_t)$ with gradient descent algorithm at iteration t.

Alternative objective. By Lemma 3.1, an alternative objective for Eq. (2) at each step t is:

$$\boldsymbol{z}_{t+1} = \arg\min_{\boldsymbol{z}} \hat{\mathcal{L}}(\boldsymbol{z}, \boldsymbol{z}_t) = \frac{\beta}{2}\|\boldsymbol{x} - \mathcal{A}^*(\boldsymbol{z})\|_2^2 + \frac{1-\beta}{2}\|\boldsymbol{w}_t^{1/2} \odot (\boldsymbol{x} - \mathcal{A}^*(\boldsymbol{z}))\|_2^2 + \lambda\|\boldsymbol{z}\|_1, \quad (4)$$

where $\boldsymbol{w}_t = \frac{1}{2|\boldsymbol{x}-\mathcal{A}^*(\boldsymbol{z}_t)|} \in \mathbb{R}^{H\times W\times C}$. Specifically, when $\boldsymbol{w}_t = \mathbf{1}$, the problem reduces to the formulation in Eq. (1). In practice, to avoid the singularity when the reconstruction residual approaches zero, we use an ϵ-stabilized form in our implementation: $\boldsymbol{w}_t = \frac{1}{2(|\boldsymbol{x}-\mathcal{A}^*(\boldsymbol{z})|+\epsilon)}$.

RISTA algorithmic iteration. Then, we can optimize the ℓ_1-regularized problem in Eq. (4) instead of original Eq. (2) by our reweighted iterative shrinkage thresholding algorithm (RISTA):

$$\boldsymbol{g}_t = \mathcal{A}\left((\beta\mathbf{1} + (1-\beta)\boldsymbol{w}_t) \odot (\boldsymbol{x} - \mathcal{A}^*(\boldsymbol{z}_t))\right), \quad \boldsymbol{z}_{t+1} = \mathcal{T}_{\lambda\eta}\left(\boldsymbol{z}_t + \eta \cdot \boldsymbol{g}_t\right), \quad (5)$$

where $\mathcal{T}_{\lambda\eta}(\boldsymbol{z}) = \text{sign}(\boldsymbol{z})\left(|\boldsymbol{z}| - \lambda\eta\right)_+$ represents the soft thresholding operator and η is the step size. The detailed derivation of Eq. (5) is provided in Appendix B.2. As a consequence of Lemma 3.1, we can conclude the iteration $\{\boldsymbol{z}_t\}_{t=0}^T$ obtained by Eq. (5) fulfill the loss descent of original $\mathcal{L}(\boldsymbol{z})$:

$$\mathcal{L}(\boldsymbol{z}_{t+1}) \leq \hat{\mathcal{L}}(\boldsymbol{z}_{t+1}, \boldsymbol{z}_t) \leq \hat{\mathcal{L}}(\boldsymbol{z}_t, \boldsymbol{z}_t) = \mathcal{L}(\boldsymbol{z}_t).$$

Here, the first inequality follows from the upper-bound property in Lemma 3.1 (1), the second holds because $\boldsymbol{z}_{t+1}$ is obtained by minimizing $\hat{\mathcal{L}}(\cdot, \boldsymbol{z}_t)$ over $\boldsymbol{z}$ and is therefore no larger than its value at any other point, including $\boldsymbol{z}_t$, and the final equality follows from the tightness property in Lemma 3.1 (2). This implies convergence of Eq. (2) can be achieved by optimizing the localized upper bound Eq. (4).

4 Experiment

In this section, we comprehensively evaluate the effectiveness of our proposed EDLNets under various experimental settings. Additionally, we provide several ablation studies to demonstrate the working mechanism of our approach.

4.1 Experimental Setting

Datasets. We conduct the experiments on several datasets including CIFAR10 (Krizhevsky et al., 2009), CIFAR100 (Krizhevsky et al., 2009) and Tiny-ImageNet (Le & Yang, 2015).

Backbone architectures. We select ResNets as the backbones, including ResNet10, ResNet18, ResNet34, and ResNet50 (He et al., 2016). Each of the convolutional layers in ResNets are replaced with our Elastic DL layer, resulting in the corresponding EDLNets. We use ResNet18 as the default backbone if not being specified.

Evaluation methods. We evaluate the performance of the models against various attacks, including FGSM (Goodfellow et al., 2014), PGD (Madry, 2017), C&W (Carlini & Wagner, 2017), AutoAttack (Croce & Hein, 2020), and SparseFool (Modas et al., 2019), covering budget measurements across ℓ_∞-norm, ℓ_2-norm, and ℓ_1-norm. For the PGD attack, we consider both ℓ_∞-norm and ℓ_2-norm, denoted as PGD-ℓ_∞ and PGD-ℓ_2, respectively. SparseFool uses the ℓ_1-norm. Unless otherwise specified, ℓ_∞ is used as the default measurement. We perform multiple robustness reliability tests, including *certifiable robustness* (Figure 4), *transferability analysis* (Figure 5), *zero-order gradient analysis* (Appendix E.3.4), and attack with BPDA (Appendix E.3.5).

Baselines. For robust overfitting mitigation, we include the baselines including regularization (ℓ_1, ℓ_2 regularizations and their combination), Cutout (DeVries, 2017), Mixup (Zhang, 2017), and early stopping (Rice et al., 2020). For adversarial training methods, we compare the baselines including

PGD-AT (Madry, 2017), TRADES (Zhang et al., 2019), MART (Wang et al., 2019), SAT (Huang et al., 2020), AWP (Wu et al., 2020), Consistency (Tack et al., 2022), DYNAT (Liu et al., 2024), PORT (Sehwag et al., 2021), and HAT (Rade & Moosavi-Dezfooli, 2022).

Hyperparameter setting. We train the baselines for 200 epochs with batch size 128, weight decay 2e-5, momentum 0.9, and an initial learning rate of 0.1 that is divided by 10 at the 100-th and 150-th epoch. The ϵ to avoid singularity is set as 1e-3. For our Elastic DL, we pretrain the Vanilla DL model for 150 epochs and then fine-tune the Elastic DL model for 50 epochs. We compare and discuss different training strategies in Appendix D.

4.2 Adversarial Robustness & Generalization

Table 2: Natural and robust performance of PGD-based adversarial training with different methods to mitigate the overfitting on CIFAR10. BEST represents the highest test accuracy achieved during training, while FINAL is the average accuracy over the last five epochs. DIFF, the difference between BEST and FINAL, measures the ability to mitigate overfitting. The best performance is highlighted in **bold**, while the second-best is underlined.

	Natural Acc.			Robust Acc.		
Method	Final	Best	Diff	Final	Best	Diff
Vanilla	78.98	79.90	0.92	44.90	48.01	3.11
ℓ_1 Reg.	64.84	65.71	0.87	40.94	41.97	1.03
ℓ_2 Reg.	78.88	79.39	0.51	42.73	48.26	5.53
ℓ_2 + ℓ_1 Reg.	66.86	67.62	0.76	42.53	43.33	0.80
Cutout	75.11	75.58	0.47	47.12	48.23	1.11
Mixup	69.64	72.05	2.41	46.10	48.53	2.43
Early Stopping	75.51	75.51	**0.00**	47.69	47.95	**0.26**
Vanilla DL	82.59	**83.27**	0.68	44.03	50.53	6.50
Elastic DL (Ours)	**83.01**	**83.27**	0.26	**54.94**	**55.66**	0.72

First, we validate the effectiveness of our approach in mitigating overfitting. Next, we comprehensively evaluate the robustness of adversarial training methods. Finally, we demonstrate that our approach surpasses the state-of-the-art methods on the leaderboard via structural priors.

Robust overfitting mitigation. To validate the effectiveness of incorporating structural priors, we compare our method with existing popular baselines on CIFAR10 in mitigating the *robust overfitting* problem in Table 2 and Figure 2. We leave the training curves of all the methods in Appendix E.2.1 and Appendix E.2.2 due to the space limit. From the results, we can make the following observations:

- From Table 2, we observe that our Elastic DL not only achieves a significant advantage in both absolute FINAL and BEST performance but also maintains a relatively small gap (DIFF) between them, indicating that incorporating the structural prior effectively guides adversarial training to achieve better robustness and generalization.
- From Figure 2, we observe that during the 100th to 200th epochs, the Vanilla DL model exhibits a severe *robust overfitting* phenomenon. By incorporating our Elastic DL structural prior at the 150th epoch, the test robustness improves substantially, highlighting the promising potential of the Elastic DL structural prior in overcoming the bottleneck of adversarial robustness and generalization.

Adversarial training robustness. To validate the effectiveness of our Elastic DL, we select several existing popular adversarial defenses and report the experimental results of backbone ResNet18 under various attacks in Table 3. From the results we can make the following observations:

- Our HAT + Elastic DL significantly outperforms other methods across various attacks, achieving state-of-the-art performance among all baselines.
- Our Elastic DL is a robust architecture that is orthogonal to existing adversarial training methods and can be combined with them to further improve robustness.

Comparison with high-performing defenses. We examine whether Elastic DL improves the performance of HAT (Rade & Moosavi-Dezfooli, 2022) and PORT (Sehwag et al., 2021), two competitive defenses included in RobustBench (Croce et al., 2020). Using ResNet-18 on CIFAR-10, we evaluate their original and Elastic DL variants under ℓ_∞ and ℓ_2 attacks. As shown in Tables 4 and 5, adding Elastic DL improves robust accuracy in most evaluated settings, while the effect on natural accuracy varies by method.

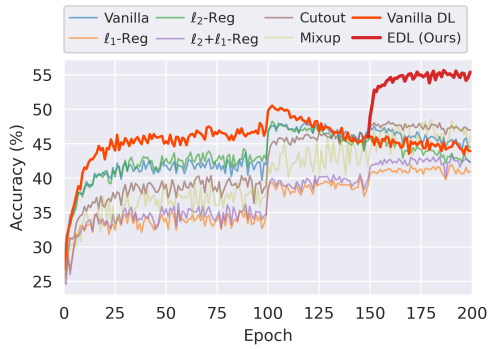


Figure 2: Test robust accuracy during the adversarial training. we pretrain the Vanilla DL model for 150 epochs and fine-tune the Elastic DL model starting from 150-th epoch. Our Elastic DL method can achieve the best robustness. adversarial robustness.

Table 3: Adversarial robustness on CIFAR10 with ResNet18 as backbone. The best performance is highlighted in **bold**.

METHOD	CLEAN	PGD	FGSM	C&W	AA
MART	79.03	48.90	60.86	45.92	43.88
SAT	83.28	43.57	50.13	47.47	39.72
AWP	81.20	51.60	55.30	48.00	46.90
CONSISTENCY	84.37	45.19	53.84	43.75	40.88
DYNAT	82.34	52.25	65.96	52.19	45.10
PGD-AT	80.90	44.35	58.41	46.72	42.14
+ VANILLA DL	83.28	45.64	53.88	41.22	43.70
+ ELASTIC (OURS)	83.57	53.22	69.35	60.80	52.90
TRADES-2.0	82.80	48.32	51.67	40.65	36.40
+ VANILLA DL	79.05	40.64	47.12	41.49	34.90
+ ELASTIC (OURS)	79.85	49.32	58.68	49.47	47.20
TRADES-0.2	85.74	32.63	44.26	26.70	19.00
+ VANILLA DL	82.55	25.37	44.48	30.30	15.30
+ ELASTIC (OURS)	84.75	33.61	57.86	40.68	28.10
PORT	84.59	58.62	62.64	58.12	55.14
+ VANILLA DL	82.35	56.40	60.68	56.77	54.00
+ ELASTIC (OURS)	82.76	59.00	68.54	60.92	56.30
HAT	85.95	56.29	61.17	49.52	53.16
+ VANILLA DL	86.42	57.79	62.67	51.61	54.30
+ ELASTIC (OURS)	**86.84**	**62.48**	**71.46**	**59.90**	**59.07**

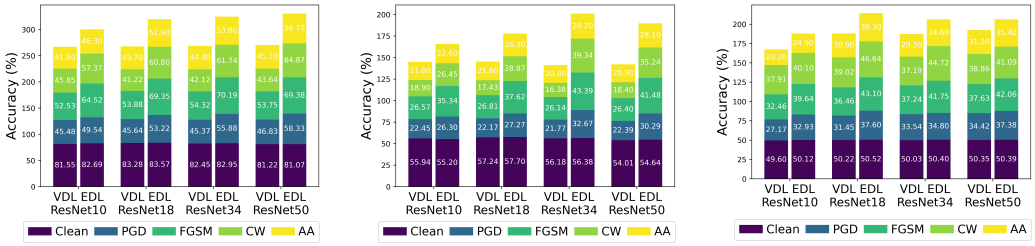
Table 4: Performance of ResNet18 on CIFAR10 under ℓ_∞-norm attack.

PERFORMANCE UNDER ℓ_∞-NORM ATTACK							
	CLEAN	PGD-ℓ_∞			AUTOATTACK-ℓ_∞		
BUDGET	0	$\frac{8}{255}$	$\frac{16}{255}$	$\frac{32}{255}$	$\frac{8}{255}$	$\frac{16}{255}$	$\frac{32}{255}$
PORT	84.59	58.62	27.49	5.79	55.14	17.80	0.30
+ VANILLA DL	82.35	56.40	27.30	6.38	54.00	20.40	0.60
+ ELASTIC (OURS)	82.76	59.00	36.53	12.17	56.30	24.60	1.70
HAT	85.95	56.29	25.82	6.09	53.16	17.20	0.60
+ VANILLA DL	86.42	57.79	26.08	6.07	54.30	17.56	0.63
+ ELASTIC (OURS)	**86.84**	**62.48**	**37.66**	**13.69**	**59.10**	**25.93**	**2.10**

Table 5: Performance of ResNet18 on CIFAR10 under ℓ_2-norm attack.

PERFORMANCE UNDER ℓ_2-NORM ATTACK							
	CLEAN	PGD-ℓ_2			AUTOATTACK-ℓ_2		
BUDGET	0	0.5	1.0	2.0	0.5	1.0	2.0
PORT	88.82	74.89	54.47	27.69	**73.80**	48.10	5.90
+ VANILLA DL	87.34	73.52	53.75	27.50	71.80	49.10	6.70
+ ELASTIC (OURS)	87.81	**75.56**	**56.76**	**31.44**	72.20	**52.40**	**11.10**
HAT	89.92	74.68	47.67	21.38	72.90	40.80	2.20
+ VANILLA DL	88.84	67.99	40.87	17.97	66.80	27.80	0.60
+ ELASTIC (OURS)	**89.95**	74.62	51.41	27.05	73.20	44.50	3.20

4.3 ABLATION STUDY



(a) CIFAR10 (b) CIFAR100 (c) Tiny-ImageNet

Figure 3: Adversarial robustness under various settings. Our Elastic DL outperforms Vanilla DL across various datasets (CIFAR10 / CIFAR100 / Tiny-ImageNet), backbones (ResNet10 / ResNet18 / ResNet34 / ResNet50) and attacks (PGD / FGSM / CW / AA).

Universality across datasets and backbones. To validate the consistent effectiveness of our proposed methods, we conduct comprehensive abation studies on the different backbones (ResNet10, ResNet18, ResNet34, ResNet50), datasets (CIFAR10, CIFAR100, Tiny-ImageNet). As demonstrated in the Figure 3, Table 7, 8 and 9 in Appendix E.3.1, our proposed Elastic DL exhibit excellent clean performance and robustness under various attacks.

Certifiable robustness. We also provide the results of certifiable robustness via randomized smoothing, which is a certified defense that can theoretically guarantee certified accuracy regardless of the evaluated attacks. The results in Figure 4 demonstrate that our Elastic DL delivers better certified robustness than the vanilla DL.

Transferability analysis. To validate the effectiveness of our method and strength of evaluated adaptive attack, we evaluate transfer attacks from multiple baselines, along with the adaptive attack for comparison. It can be observed from Figure 5 that the adaptive attack yields the strongest attack, thereby validating the effectiveness of our experimental evaluation.

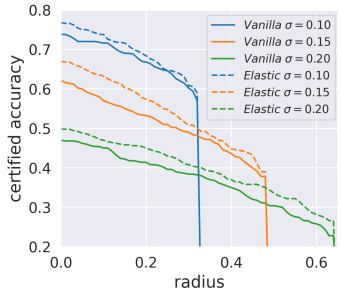


Figure 4: Certifiable robustness. Elastic DL delivers better certified robustness than vanilla DL.

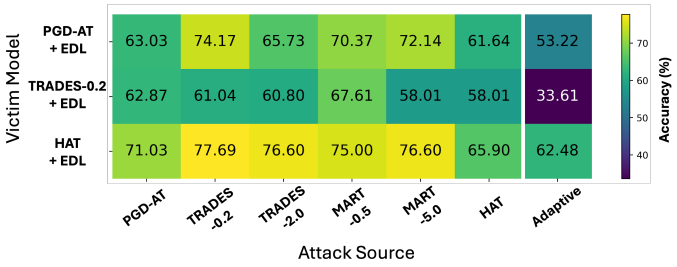


Figure 5: Transferability analysis. We evaluate the transfer attacks from multiple baselines, using the adaptive attack as a comparison, where the adaptive attack demonstrates the strongest performance.

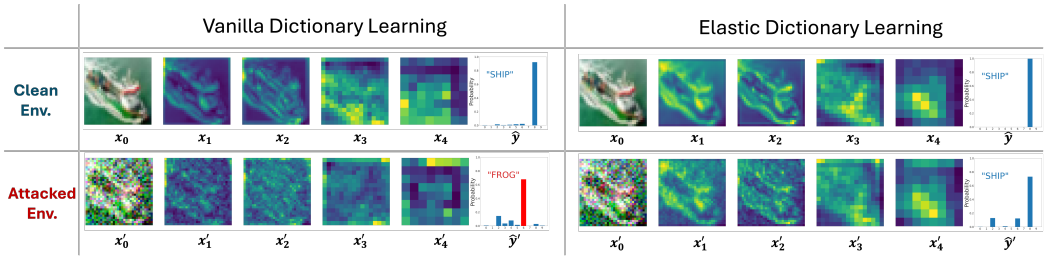


Figure 6: Hidden embedding visualization under clean and attacked scenarios. The difference between clean and attacked embeddings in Elastic DL is smaller compared to Vanilla DL, with this effect becoming more significant in deeper layers. Consequently, while an adversarial attack alters the Vanilla DL output from "*SHIP*" to "*FROG*", Elastic DL successfully preserves the correct prediction.

Hidden embedding visualization. We also conduct visualization analyses on the hidden embedding to obtain better insight into the effectiveness of our proposed Elastic DL. We begin by quantifying the relative difference between clean embeddings ($\boldsymbol{x}_i$) and attacked embeddings ($\boldsymbol{x}_i'$) across all layers, as in Figure 17 in Appendix E.3.6. Additionally, we visualize one instance in Figure 6, with more examples provided in Appendix E.3.6. The results in Figure 17 show that Elastic DL has smaller embedding difference across layers, indicating that our proposed Elastic DL architecture indeed mitigates the impact of the adversarial perturbation. Moreover, as demonstrated in the example in Figure 6, the presence of adversarial perturbations can disrupt the hidden embedding patterns, leading to incorrect predictions in Vanilla DL. In contrast, our Elastic DL appears to lessen the effects of such perturbations and maintain predicting groundtruth label. From the figures, we can also clearly tell that the difference between clean attacked embeddings of Vanilla DL is much more significant than in Elastic DL.

Different attack measurements. In addition to ℓ_∞-norm attack (PGD-ℓ_∞), we also validate the consistent effectiveness of our Elastic DL with ℓ_2-norm (PGD-ℓ_2) and ℓ_1-norm (SparseFool) attacks in the Figure 16 in Appendix E.3.3.

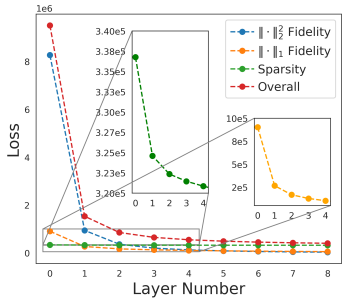


Figure 7: Algorithm convergence. RISTA algorithm achieves fast convergence within just three steps.

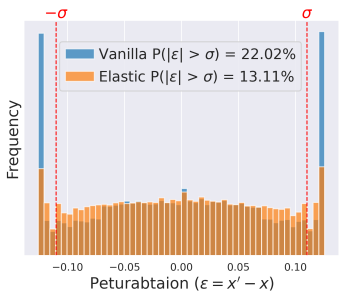


Figure 8: Attack behaviors. The attacker tends to attack Vanilla DL model by introducing outlying values.

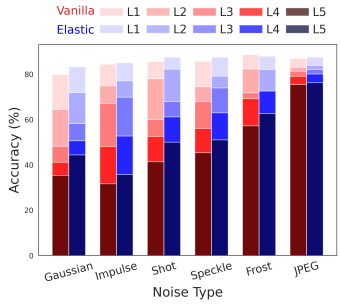


Figure 9: Out-of-distribution robustness. Our Elastic DL also demonstrates excellent out-of-distribution robustness.

Convergence. To validate the effectiveness of our RISTA iterations, we plot the loss descent curves of overall objective Eq.(2) along with the individual terms ($\|\boldsymbol{x} - \mathcal{A}^*(\boldsymbol{z})\|_2^2$, $\|\boldsymbol{x} - \mathcal{A}^*(\boldsymbol{z})\|_1$ and $\|\boldsymbol{z}\|_1$) in Figure 7, which shows that RISTA converges rapidly within first three steps.

Attack behaviors. To investigate the attack behaviors, we apply the PGD attack to both models and visualize the resulting perturbations in Figure 8. It can be observed that, in the Vanilla DL model, the adversarial attack introduces substantial outlying noise, which can be largely mitigated by our Elastic DL method.

Out-of-distribution Robustness. Beyond in-distribution robustness, we further validate the advantage of our proposed Elastic DL structure by evaluating the out-of-distribution performance of Vanilla DL and Elastic DL. The results in Figure 9 demonstrate the superiority of our Elastic DL over the Vanilla DL under various types of out-of-distribution noise.

Computational cost analysis. Table 6 compares the computational cost of Vanilla DL and Elastic DL on CIFAR10 with ResNet18. Both models share identical parameters (11.174M), FLOPs (4.776G), and checkpoint size (42.71MB), since Elastic DL adds only the learnable weight β and reuses the same dictionary operators as Vanilla DL. Under PGD-7 adversarial training, Vanilla DL trains for 200 epochs (13.8 GPU hours), while Elastic DL's pretraining–finetuning strategy (150 epochs Vanilla DL + 50 epochs Elastic DL) raises this to 15.2 GPU hours, only about 10% more. Thus, Elastic DL's robustness gains come at a modest computational overhead.

Table 6: Computational cost comparison.

METHOD	TRAIN (H)	PARAMS (M)	FLOPS (G)	SIZE (MB)
VANILLA DL	13.8	11.174	4.776	42.71
ELASTIC DL	15.2	11.174	4.776	42.71

5 RELATED WORKS

Robust overfitting. Overfitting in adversarially trained deep networks has been shown to significantly harm test robustness Rice et al. (2020). To address the issue of severe robust overfitting, several efforts have been made from various perspectives. For instance, Dropout Srivastava et al. (2014) is a widely used regularization method that randomly disables units and connections during training to mitigate overfitting. Regularization techniques Andriushchenko & Flammarion (2020); Qin et al. (2019); Sriramanan et al. (2020) have also proven effective in preventing overfitting by penalizing the complexity of model parameters. Data augmentation is another common approach for reducing overfitting in deep network training Schmidt et al. (2018), with methods including Cutout DeVries (2017), Mixup Zhang (2017), semi-supervised learning techniques Carmon et al. (2019); Zhai et al. (2019), and generative modeling Wang et al. (2023); Gowal et al. (2021) being particularly notable. Additionally, early stopping Rice et al. (2020) has demonstrated great effectiveness in achieving optimal robust performance during adversarial training. However, existing methods have yet to fully realize the potential of structural priors for improving adversarial robustness and generalization.

Dictionary learning prior in deep learning. Dictionary learning has been well-studied and widely applied in signal and image processing Olshausen & Field (1996); Wright et al. (2008); Wright & Ma (2010); Zhao et al. (2011); Yang et al. (2011); Lu et al. (2013); Chen & Wu (2013); Jiang et al. (2015); Yang et al. (2011), based on the assumption that an input signal can be represented by a few atoms from a dictionary. Building on this foundation, Papyan et al. (2017); Cazenavette et al. (2021); Mahdizadehaghdam et al. (2019); Li et al. (2022) successfully incorporated dictionary learning into deep learning to interpret or replace the "black-box" nature of neural networks. While these methods have demonstrated promising generalization and robustness against random noise and universal attacks Li et al. (2022); Mahdizadehaghdam et al. (2019), their practical benefits for improving robustness under adaptive attacks are yet to be thoroughly investigated.

6 CONCLUSION

This paper proposes an orthogonal direction to break through the current plateau of adversarial robustness. We begin by revisiting dictionary learning in deep learning and reveal its limitations in dealing with outliers and its vulnerability to adaptive attacks. As a countermeasure, we propose a novel elastic dictionary learning approach along with an efficient RISTA algorithm unrolled as an convolutional neural layers. Our comprehensive experiments demonstrate that our method achieves remarkable robustness, surpassing state-of-the-art baselines available on the robustness leaderboard. To the best of our knowledge, this is the first work to discover and validate that structural prior can reliably enhance adversarial robustness and generalization, unveiling a promising direction for future research.

AI USE STATEMENT

In this work, we used generative AI tools solely to assist with polishing the writing of the manuscript, including improving grammar, clarity, and sentence-level phrasing, and refining section transitions. We have not used generative AI tools to generate research ideas, design experiments, implement code, run or analyze experiments, or produce any of the theoretical or empirical results presented in this paper; the Elastic Dictionary Learning formulation, the RISTA algorithm, the EDLNet architecture, and all experimental findings are the original work of the authors. All AI-assisted writing suggestions were reviewed by the authors for factual accuracy and consistency with the underlying technical content before inclusion in the manuscript. We take responsibility for the final content of this work, including all text, claims, and artifacts, regardless of any assistance from generative AI in its wording.

ETHICS STATEMENT

This work does not involve human subjects and does not introduce any new datasets. All experiments are conducted on standard, publicly available benchmark image classification datasets (CIFAR10, CIFAR100, and Tiny-ImageNet) that are widely used in the adversarial robustness literature and do not contain personally identifiable or sensitive information. Our contribution aims to improve the adversarial robustness and generalization of deep neural networks, which we believe has a positive societal impact by making machine learning systems more reliable and secure against malicious perturbations. We are not aware of any conflicts of interest, fairness or discrimination concerns, or dual-use risks associated with this research.

REPRODUCIBILITY STATEMENT

We have taken several steps to facilitate the reproducibility of our results. The complete derivation of the RISTA algorithm, including the localized upper bound construction and the proof of its descent guarantee (Lemma 3.1), is provided in Appendix B. The full forward pass of EDLNet is specified in Algorithm 1, and training details, including hyperparameter settings and the pretraining/fine-tuning strategy used for the Vanilla DL and Elastic DL models, are described in Section 4.1 and Appendix D. All experiments are conducted on publicly available benchmark datasets (CIFAR10, CIFAR100, and Tiny-ImageNet), with evaluation protocols and attack budgets detailed in Section 4.1. Anonymized source code for EDLNet and the RISTA algorithm is available at `https://anonymous.4open.science/r/ElasticDL-6343` to facilitate reproduction of our results.

REFERENCES

Sravanti Addepalli, Samyak Jain, Gaurang Sriramanan, and R Venkatesh Babu. Scaling adversarial training to large perturbation bounds. In *European Conference on Computer Vision*, pp. 301–316. Springer, 2022.

Moustafa Alzantot, Yash Sharma, Supriyo Chakraborty, Huan Zhang, Cho-Jui Hsieh, and Mani B Srivastava. Genattack: Practical black-box attacks with gradient-free optimization. In *Proceedings of the genetic and evolutionary computation conference*, pp. 1111–1119, 2019.

Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. *Advances in Neural Information Processing Systems*, 33:16048–16059, 2020.

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European conference on computer vision*, pp. 484–501. Springer, 2020.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. Ieee, 2017.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. *Advances in neural information processing systems*, 32, 2019.

George Cazenavette, Calvin Murdock, and Simon Lucey. Architectural adversarial robustness: The case for deep pursuit. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7150–7158, 2021.

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pp. 15–26, 2017.

Zhuoyuan Chen and Ying Wu. Robust dictionary learning by error source decomposition. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2216–2223, 2013.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International conference on machine learning*, pp. 854–863. PMLR, 2017.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216. PMLR, 2020.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(11):1413–1457, 2004.

Terrance DeVries. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy A Mann. Improving robustness using generated data. *Advances in Neural Information Processing Systems*, 34:4218–4233, 2021.

Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Chih-Hui Ho and Nuno Vasconcelos. Disco: Adversarial defense with local implicit functions. *Advances in Neural Information Processing Systems*, 35:23818–23837, 2022.

Lang Huang, Chao Zhang, and Hongyang Zhang. Self-adaptive training: beyond empirical risk minimization. *Advances in neural information processing systems*, 33:19365–19376, 2020.

Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pp. 492–518. Springer, 1992.

Wenhao Jiang, Feiping Nie, and Heng Huang. Robust dictionary learning with capped l1-norm. In *Twenty-fourth international joint conference on artificial intelligence*, 2015.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

Boqi Li and Weiwei Liu. Wat: improve the worst-class robustness in adversarial training. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 14982–14990, 2023.

Mingyang Li, Pengyuan Zhai, Shengbang Tong, Xingjian Gao, Shao-Lun Huang, Zhihui Zhu, Chong You, Yi Ma, et al. Revisiting sparse convolutional model for visual recognition. *Advances in Neural Information Processing Systems*, 35:10492–10504, 2022.

Zhenyu Liu, Haoran Duan, Huizhi Liang, Yang Long, Vaclav Snasel, Guiseppe Nicosia, Rajiv Ranjan, and Varun Ojha. Dynamic label adversarial training for deep learning robustness against adversarial attacks. *arXiv preprint arXiv:2408.13102*, 2024.

Cewu Lu, Jiaping Shi, and Jiaya Jia. Online robust dictionary learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 415–422, 2013.

Aleksander Madry. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Shahin Mahdizadehaghdam, Ashkan Panahi, Hamid Krim, and Liyi Dai. Deep dictionary learning: A parametric network approach. *IEEE Transactions on Image Processing*, 28(10):4790–4802, 2019.

Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*, 2017.

Apostolos Modas, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Sparsefool: a few pixels make a big difference. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9087–9096, 2019.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2574–2582, 2016.

Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. *arXiv preprint arXiv:2205.07460*, 2022.

Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pp. 506–519, 2017.

Vardan Papyan, Yaniv Romano, and Michael Elad. Convolutional neural networks analyzed via convolutional sparse coding. *Journal of Machine Learning Research*, 18(83):1–52, 2017.

Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. Adversarial robustness through local linearization. *Advances in neural information processing systems*, 32, 2019.

Rahul Rade and Seyed-Mohsen Moosavi-Dezfooli. Reducing excessive margin to achieve a better accuracy vs. robustness trade-off. In *International Conference on Learning Representations*, 2022.

Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International conference on machine learning*, pp. 8093–8104. PMLR, 2020.

Peter J Rousseeuw and Annick M Leroy. *Robust regression and outlier detection*. John wiley & sons, 2003.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. *Advances in neural information processing systems*, 31, 2018.

Vikash Sehwag, Saeed Mahloujifar, Tinashe Handina, Sihui Dai, Chong Xiang, Mung Chiang, and Prateek Mittal. Robust learning meets generative models: Can proxy distributions improve adversarial robustness? *arXiv preprint arXiv:2104.09425*, 2021.

Changhao Shi, Chester Holtz, and Gal Mishne. Online adversarial purification based on self-supervision. *arXiv preprint arXiv:2101.09387*, 2021.

Gaurang Sriramanan, Sravanti Addepalli, Arya Baburaj, et al. Guided adversarial attack for evaluating and enhancing adversarial defenses. *Advances in Neural Information Processing Systems*, 33: 20297–20308, 2020.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Jihoon Tack, Sihyun Yu, Jongheon Jeong, Minseon Kim, Sung Ju Hwang, and Jinwoo Shin. Consistency regularization for adversarial robustness. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 8414–8422, 2022.

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International conference on learning representations*, 2019.

Zekai Wang, Tianyu Pang, Chao Du, Min Lin, Weiwei Liu, and Shuicheng Yan. Better diffusion models further improve adversarial training. In *International Conference on Machine Learning*, pp. 36246–36263. PMLR, 2023.

John Wright and Yi Ma. Dense error correction via l1-minimization. *IEEE Trans. Inf. Theor.*, 56 (7):3540–3560, July 2010. ISSN 0018-9448. doi: 10.1109/TIT.2010.2048473. URL `https://doi.org/10.1109/TIT.2010.2048473`.

John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2): 210–227, 2008.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in neural information processing systems*, 33:2958–2969, 2020.

Meng Yang, Lei Zhang, Jian Yang, and David Zhang. Robust sparse coding for face recognition. In *CVPR 2011*, pp. 625–632. IEEE, 2011.

Jongmin Yoon, Sung Ju Hwang, and Juho Lee. Adversarial purification with score-based generative models. In *International Conference on Machine Learning*, pp. 12062–12072. PMLR, 2021.

Runtian Zhai, Tianle Cai, Di He, Chen Dan, Kun He, John Hopcroft, and Liwei Wang. Adversarially robust generalization just requires more unlabeled data. *arXiv preprint arXiv:1906.00555*, 2019.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pp. 7472–7482. PMLR, 2019.

Hongyi Zhang. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Cong Zhao, Xiaogang Wang, and Wai-Kuen Cham. Background subtraction via robust dictionary learning. *EURASIP Journal on Image and Video Processing*, 2011:1–12, 2011.

Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the robustness of deep neural networks via stability training. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pp. 4480–4488, 2016.

A Overview of Elastic Dictionary Learning

Overview of Elastic DL neural networks. Having derived RISTA as an efficient solver for the elastic dictionary learning objective in Eq. (2), we now turn to how this algorithm is used to construct a trainable neural architecture. We present the complete forward pass of an Elastic Dictionary Learning Network (EDLNet) in Algorithm 1. Specifically, we replace each conventional convolutional layer with an Elastic Dictionary Learning (EDL) layer, whose underlying optimization problem is approximately solved by unrolling T iterations of the proposed RISTA algorithm. The resulting architecture can be trained end-to-end, jointly optimizing the dictionary operators $\{\mathcal{A}^{(l)}\}_{l=0}^{L-1}$, layer-wise balancing weights $\{\beta^{(l)}\}_{l=0}^{L-1}$, and classifier parameters θ.

Algorithm 1 Forward Pass of an Elastic DL Network

Input: Image $\boldsymbol{x}$
Initialize: $\boldsymbol{x}^{(0)} \leftarrow \boldsymbol{x}$
for $l = 0$ **to** $L-1$ **do**
  $\boldsymbol{z}_0^{(l)} \leftarrow \mathcal{A}^{(l)}\big(\boldsymbol{x}^{(l)}\big)$
  for $t = 0$ **to** $T-1$ **do**
    $\boldsymbol{r}_t^{(l)} \leftarrow \boldsymbol{x}^{(l)} - \mathcal{A}^{(l)*}\Big(\boldsymbol{z}_t^{(l)}\Big), \boldsymbol{w}_t^{(l)} \leftarrow \frac{1}{2\left(\left|\boldsymbol{r}_t^{(l)}\right|+\epsilon\right)}$
    $\boldsymbol{g}_t^{(l)} \leftarrow \mathcal{A}^{(l)}\left((\beta^{(l)}\mathbf{1} + (1-\beta^{(l)})\boldsymbol{w}_t^{(l)}) \odot \boldsymbol{r}_t^{(l)}\right)$
    $\boldsymbol{z}_{t+1}^{(l)} \leftarrow \mathcal{T}_{\lambda\eta}\Big(\boldsymbol{z}_t^{(l)} + \eta \cdot \boldsymbol{g}_t^{(l)}\Big)$
  end for
  $\boldsymbol{x}^{(l+1)} \leftarrow \boldsymbol{z}_T^{(l)}$
end for
Output: $\hat{y} \leftarrow \mathrm{MLP}_\theta\big(\boldsymbol{x}^{(L)}\big)$

B THEORETICAL PROOF

B.1 PROOF OF LEMMA 3.1

Proof. Since $\sqrt{a} \leq \frac{a}{2\sqrt{b}} + \frac{\sqrt{b}}{2}$ and equality holds when $a = b$, substituting $a = (\boldsymbol{x}[i,j,c] - \mathcal{A}^*(\boldsymbol{z})[i,j,c])^2$ and $b = (\boldsymbol{x}[i,j,c] - \mathcal{A}^*(\boldsymbol{z}_*)[i,j,c])^2$ gives

$$|\boldsymbol{x}[i,j,c] - \mathcal{A}^*(\boldsymbol{z})[i,j,c]| \tag{6}$$

$$\leq \frac{1}{2} \cdot \frac{1}{|\boldsymbol{x}[i,j,c] - \mathcal{A}^*(\boldsymbol{z}_*)[i,j,c]|} \cdot (\boldsymbol{x}[i,j,c] - \mathcal{A}^*(\boldsymbol{z})[i,j,c])^2 + \frac{1}{2}|\boldsymbol{x}[i,j,c] - \mathcal{A}^*(\boldsymbol{z}_*)[i,j,c]| \tag{7}$$

$$= \boldsymbol{w}[i,j,c] \cdot (\boldsymbol{x}[i,j,c] - \mathcal{A}^*(\boldsymbol{z})[i,j,c])^2 + \frac{1}{2}|\boldsymbol{x}[i,j,c] - \mathcal{A}^*(\boldsymbol{z}_*)[i,j,c]| \tag{8}$$

Summing both sides over (i,j,c), we obtain

$$\begin{aligned}
\mathcal{R}(\boldsymbol{z}) = \|\boldsymbol{x} - \mathcal{A}^*(\boldsymbol{z})\|_1 &= \sum_{i,j,c} |\boldsymbol{x}[i,j,c] - \mathcal{A}^*(\boldsymbol{z})[i,j,c]| \\
&\leq \sum_{i,j,c} \boldsymbol{w}[i,j,c] \cdot (\boldsymbol{x}[i,j,c] - \mathcal{A}^*(\boldsymbol{z})[i,j,c])^2 + \frac{1}{2} \sum_{i,j,c} |\boldsymbol{x}[i,j,c] - \mathcal{A}^*(\boldsymbol{z}_*)[i,j,c]| \\
&= \|\boldsymbol{w}^{1/2} \odot (\boldsymbol{x} - \mathcal{A}^*(\boldsymbol{z}))\|_2^2 + \frac{1}{2}\mathcal{R}(\boldsymbol{z}_*) \\
&= \mathcal{U}(\boldsymbol{z}, \boldsymbol{z}_*)
\end{aligned}$$

and equality holds at $a = b$ (i.e., $\boldsymbol{z} = \boldsymbol{z}_*$):

$$\mathcal{U}(\boldsymbol{z}_*, \boldsymbol{z}_*) = \mathcal{R}(\boldsymbol{z}_*). \tag{9}$$

□

B.2 PROOF OF ALGORITHM ITERATION IN EQ. (5)

Here, we derive the algorithm for Elastic dictionary learning (Elastic DL).

Proof. For the convex objective

$$f(\boldsymbol{z}) = \frac{\beta}{2}\|\boldsymbol{x} - \mathcal{A}^*(\boldsymbol{z})\|_2^2 + \frac{1-\beta}{2}\|(\boldsymbol{w}^{(t)})^{1/2} \odot (\boldsymbol{x} - \mathcal{A}^*(\boldsymbol{z}))\|_2^2,$$

we can reach the optimum via first-order gradient descent:

$$\boldsymbol{z}_{t+1} = \boldsymbol{z}_t - t\nabla f(\boldsymbol{z}_t),$$

or equivalently,

$$\boldsymbol{z}_{t+1} = \arg\min_{\boldsymbol{z}} \{f(\boldsymbol{z}_t) + \langle \boldsymbol{z} - \boldsymbol{z}_t, \nabla f(\boldsymbol{z}_t)\rangle + \frac{1}{2t}\|\boldsymbol{z} - \boldsymbol{z}_t\|^2\}.$$

For the corresponding ℓ_1-regularized problem

$$\min_{\boldsymbol{z}} f(\boldsymbol{z}) + \lambda\|\boldsymbol{z}\|_1,$$

we have

$$\begin{aligned}
\boldsymbol{z}_{t+1} &= \arg\min_{\boldsymbol{z}} \{f(\boldsymbol{z}_t) + \langle \boldsymbol{z} - \boldsymbol{z}_t, \nabla f(\boldsymbol{z}_t)\rangle + \frac{1}{2t}\|\boldsymbol{z} - \boldsymbol{z}_t\|^2 + \lambda\|\boldsymbol{z}\|_1\} \\
&= \arg\min_{\boldsymbol{z}} \{\frac{1}{2t}\|\boldsymbol{z} - (\boldsymbol{z}_t - t\nabla f(\boldsymbol{z}_t))\|^2 + \lambda\|\boldsymbol{z}\|_1\} \\
&= \arg\min_{\boldsymbol{z}} \{g(\boldsymbol{z}) := \frac{1}{2t}\|\boldsymbol{z} - \boldsymbol{y}\|^2 + \lambda\|\boldsymbol{z}\|_1\} \quad (\boldsymbol{y} = \boldsymbol{z}_t - t\nabla f(\boldsymbol{z}_t))
\end{aligned}$$

The optimality condition is:

$$
\begin{aligned}
& 0 \in \partial_{\boldsymbol{z}} g(\boldsymbol{z}^*) = \frac{1}{t}(\boldsymbol{z}^* - \boldsymbol{y}) + \lambda \text{sign}(\boldsymbol{z}^*) \\
\Leftrightarrow \quad & \boldsymbol{y} \in \boldsymbol{z}^* + \lambda t \text{sign}(\boldsymbol{z}^*) \\
\Leftrightarrow \quad & \boldsymbol{y} \in (\text{Id} + \lambda t \text{sign}(\cdot))(\boldsymbol{z}^*) \\
\Leftrightarrow \quad & \boldsymbol{z}^* = \mathcal{T}_{\lambda t}(\boldsymbol{y}) := (\text{Id} + \lambda t \text{sign}(\cdot))^{-1}(\boldsymbol{y}) = \text{sign}(\boldsymbol{y})\left(|\boldsymbol{y}| - \lambda t\right)_+ .
\end{aligned}
$$

Since

$$
\begin{aligned}
\nabla f(\boldsymbol{z}_t) &= -\beta \mathcal{A}(\boldsymbol{x} - \mathcal{A}^*(\boldsymbol{z}_t)) - (1-\beta)\mathcal{A}\left(\boldsymbol{w}^{(t)} \odot (\boldsymbol{x} - \mathcal{A}^*(\boldsymbol{z}_t))\right) \\
&= -\mathcal{A}\left(\left(\beta \mathbf{1} + (1-\beta)\boldsymbol{w}^{(t)}\right) \odot (\boldsymbol{x} - \mathcal{A}^*(\boldsymbol{z}_t))\right),
\end{aligned}
$$

we have

$$
\begin{aligned}
\boldsymbol{z}_{t+1} = \boldsymbol{z}^* &= \mathcal{T}_{\lambda t}(\boldsymbol{y}) = \mathcal{T}_{\lambda t}\left(\boldsymbol{z}_t - t \cdot \nabla f(\boldsymbol{z}_t)\right) \\
&= \mathcal{T}_{\lambda t}\left(\boldsymbol{z}_t + t \cdot \mathcal{A}\left(\left(\beta \mathbf{1} + (1-\beta)\boldsymbol{w}^{(t)}\right) \odot (\boldsymbol{x} - \mathcal{A}^*(\boldsymbol{z}_t))\right)\right).
\end{aligned}
$$

□

C Related Works

C.1 Adversarial Attacks

Adversarial attacks are typically classified into two main categories: *white-box* and *black-box* attacks. In white-box attacks, the attacker has full knowledge of the target neural network, including its architecture, parameters, and gradients. Common examples of white-box attacks include gradient-based methods such as FGSM (Goodfellow et al., 2014), DeepFool (Moosavi-Dezfooli et al., 2016), PGD (Madry, 2017), and the C&W attack (Carlini & Wagner, 2017). In contrast, black-box attacks operate under limited information, where the attacker can only interact with the model through its input-output behavior without direct access to internal details. Examples of black-box methods include surrogate model-based approaches (Papernot et al., 2017), zeroth-order optimization techniques (Chen et al., 2017), and query-based methods (Andriushchenko et al., 2020; Alzantot et al., 2019).

Below we list details of the attacks used in the main paper:

- Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014): FGSM is one of the earliest and most widely used adversarial attack methods. It generates adversarial examples using the gradient of the loss function with respect to the input to craft small but purposeful perturbations that lead the model to make incorrect predictions.
- Projected Gradient Descent (PGD) (Madry, 2017): PGD is an iterative and more robust extension of FGSM. It repeatedly applies small perturbations within a defined range (an ϵ-ball) to maximize the model's loss, and is widely considered a strong adversary for evaluating model robustness.
- Carlini & Wagner Attack (C&W) (Carlini & Wagner, 2017): This attack crafts adversarial examples by optimizing a custom loss function designed to minimize perturbation magnitude while ensuring the resulting sample is misclassified.
- AutoAttack (Croce & Hein, 2020): AutoAttack is an ensemble of adversarial attack methods that automatically evaluates model robustness. It combines several attacks to provide a strong, reliable benchmark without manual tuning.
- SparseFool (Modas et al., 2019): SparseFool is a sparse adversarial attack that perturbs only a few pixels of the input image, highlighting how minimal changes can significantly alter model predictions.

C.2 Adversarial Defenses

Significant efforts have been devoted to enhancing model robustness through a variety of strategies, including detection techniques (Metzen et al., 2017; Feinman et al., 2017; Grosse et al., 2017; Sehwag et al., 2021; Rade & Moosavi-Dezfooli, 2022; Addepalli et al., 2022), purification-based approaches (Ho & Vasconcelos, 2022; Nie et al., 2022; Shi et al., 2021; Yoon et al., 2021), robust training methods (Madry, 2017; Zhang et al., 2019; Gowal et al., 2021; Li & Liu, 2023), and regularization-based techniques (Cisse et al., 2017; Zheng et al., 2016). Among these, adversarial training-based methods (Sehwag et al., 2021; Rade & Moosavi-Dezfooli, 2022; Addepalli et al., 2022) have proven highly effective against adaptive adversarial attacks, consistently leading the robustness leaderboard (RobustBench) (Croce et al., 2020). Despite their success, most existing methods rely heavily on extensive synthetic training data generated by advanced generative models, larger network architectures, and empirically driven training strategies. These dependencies pose substantial challenges to advancing beyond the current plateau in adversarial robustness. In this work, we introduce an elastic dictionary framework that incorporates structural priors into model design. This approach is fully orthogonal to existing methods and offers a complementary pathway to further enhance robustness when integrated with current techniques.

Below we list details of the adversarial training baselines used in the main paper:

- PGD-AT (Madry, 2017): Projected Gradient Descent Adversarial Training (PGD-AT) is a fundamental adversarial training approach that enhances model robustness by iteratively generating adversarial examples using PGD and training the model on them.
- TRADES (Zhang et al., 2019): TRADES (Tradeoff-inspired Adversarial Defense via Surrogate Loss Minimization) balances robustness and accuracy by introducing a regularization term that penalizes the discrepancy between natural and adversarial predictions.

- MART (Wang et al., 2019): Misclassification-Aware Adversarial Training (MART) improves robustness by assigning higher weights to misclassified examples, emphasizing correctly classified samples' robustness.
- SAT (Huang et al., 2020): Self-Adaptive Training (SAT) refines adversarial training by adjusting the training process based on the model's confidence, mitigating the effects of incorrect labels and improving generalization.
- AWP (Wu et al., 2020): Adversarial Weight Perturbation (AWP) enhances robustness by perturbing model parameters within a constrained space to improve worst-case performance against adversarial attacks.
- Consistency (Tack et al., 2022): Consistency training leverages perturbation-invariant representations to enhance robustness by enforcing consistent predictions across different transformations of the input.
- DYNAT (Liu et al., 2024): Dynamic Adversarial Training (DYNAT) adapts training strategies dynamically based on model performance, balancing robustness and generalization efficiency.
- PORT (Sehwag et al., 2021): Proxy Distribution-based Robust Training (PORT) leverages data from proxy distributions, such as those generated by advanced generative models, to enhance adversarial robustness. By formally analyzing robustness transfer and optimizing training, PORT demonstrates significant improvements in robustness under various threat models.
- HAT (Rade & Moosavi-Dezfooli, 2022): Helper-based Adversarial Training (HAT) mitigates the accuracy-robustness trade-off by incorporating additional incorrectly labeled examples during training, reducing excessive margin changes along certain adversarial directions and achieving a better trade-off than existing methods.

D TRAINING STRATEGY

To identify a more effective training strategy, we evaluate two common training paradigms:

- **Training from scratch.** We initialize $\beta = 0.5$ in Elastic DL and train the model from scratch with a decaying learning-rate schedule.
- **Pretraining–Finetuning.** We first train Vanilla DL for 150 epochs until performance stabilizes. We then replace the forward module with Elastic DL, initialize $\beta = 0.5$, and finetune the model for an additional 50 epochs.

Figure 10 compares the performance of these two strategies. We observe that training Elastic DL from scratch leads to inferior performance. At the early stage of training, both the backbone parameters and the layer-wise balancing parameters β are optimized jointly under a relatively large learning rate, which makes optimization unstable and significantly affects final model quality. In particular, this early-stage instability degrades clean accuracy relative to Vanilla DL; since clean accuracy often underpins strong robust accuracy, this drop also hurts final robustness.

Therefore, in our main experiments, we adopt the **Pretraining–Finetuning** strategy.

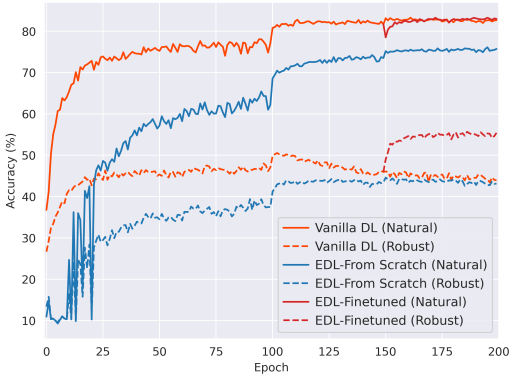


Figure 10: Comparison of training-from-scratch and pretraining–finetuning strategies for Elastic DL. Pretraining–finetuning achieves more stable optimization and better final performance.

E ADDITIONAL EXPERIMENTS

E.1 PRELIMINARY STUDIES

Preliminary study on Vanilla DL robustness. We evaluate Vanilla DL (with both fixed and tuned λ) under random impulse noise and adaptive PGD adversarial attacks (Madry, 2017). As illustrated in Figure 11, Vanilla DL's accuracy declines as the noise level increases and its distribution becomes more heavy-tailed. While tuning the sparsity weight λ improves resilience to random noise, models with any λ suffer a sharp performance drop under the adaptive PGD attack, with accuracy nearing zero.

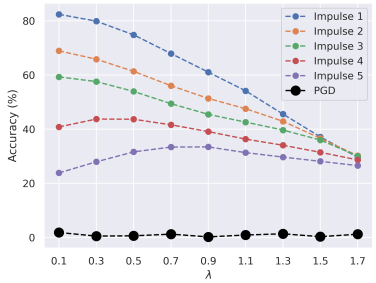


Figure 11: Performance of the Vanilla DL variant of ResNet18 under random impulse noise at different levels.

E.2 Adversarial Training Curves

E.2.1 Training Curves of Each Method

Training curve of our Elastic DL. From Figure 12, we observe that during epochs 100–150, the Vanilla DL model exhibits a severe *robust overfitting* phenomenon: training performance improves while test robust accuracy drops significantly. After incorporating our Elastic DL structural prior at epoch 150, both training and testing robustness improve substantially. Although natural performance dips slightly during the initial switching period, it recovers quickly within a few epochs. This highlights the promising potential of the Elastic DL structural prior for breaking through the bottleneck of adversarial robustness and generalization.

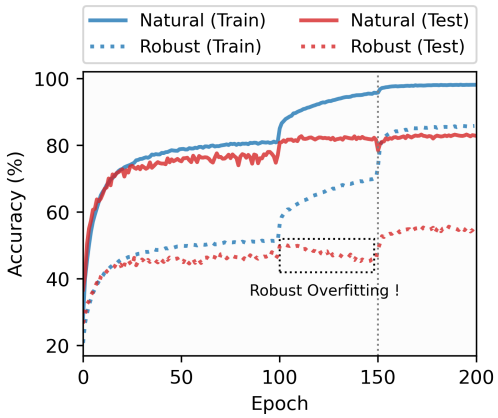
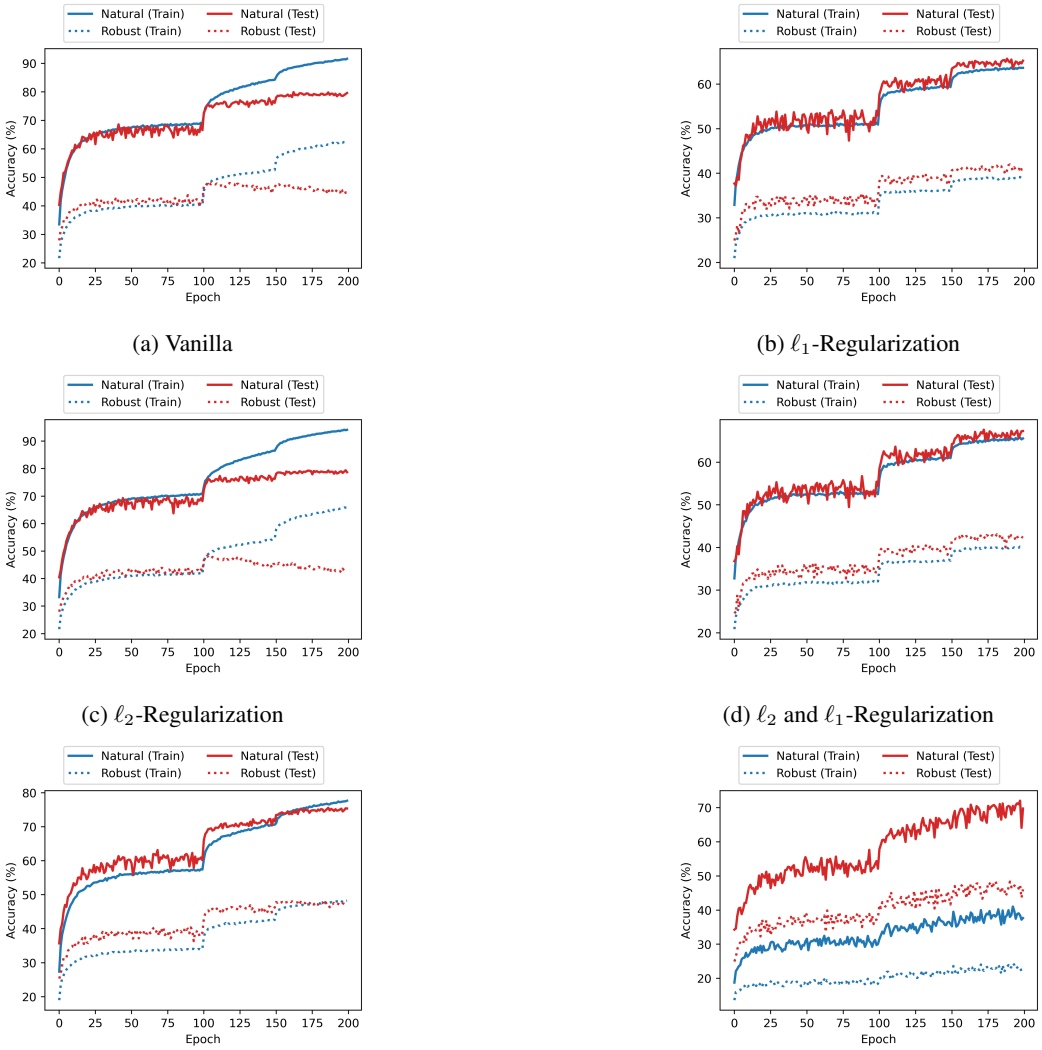


Figure 12: Adversarial training curve of our Elastic DL. During epochs 100–150, the model experiences a catastrophic *robust overfitting* problem. Introducing the Elastic DL structural prior at epoch 150 and fine-tuning effectively mitigates overfitting and yields significantly improved robustness and generalization.

Training curves of baseline methods. We track the training curves of the baselines, including regularization (ℓ_1, ℓ_2, and their combination), Cutout (DeVries, 2017), and Mixup (Zhang, 2017), in Figure 13.



(a) Vanilla

(b) ℓ_1-Regularization

(c) ℓ_2-Regularization

(d) ℓ_2 and ℓ_1-Regularization

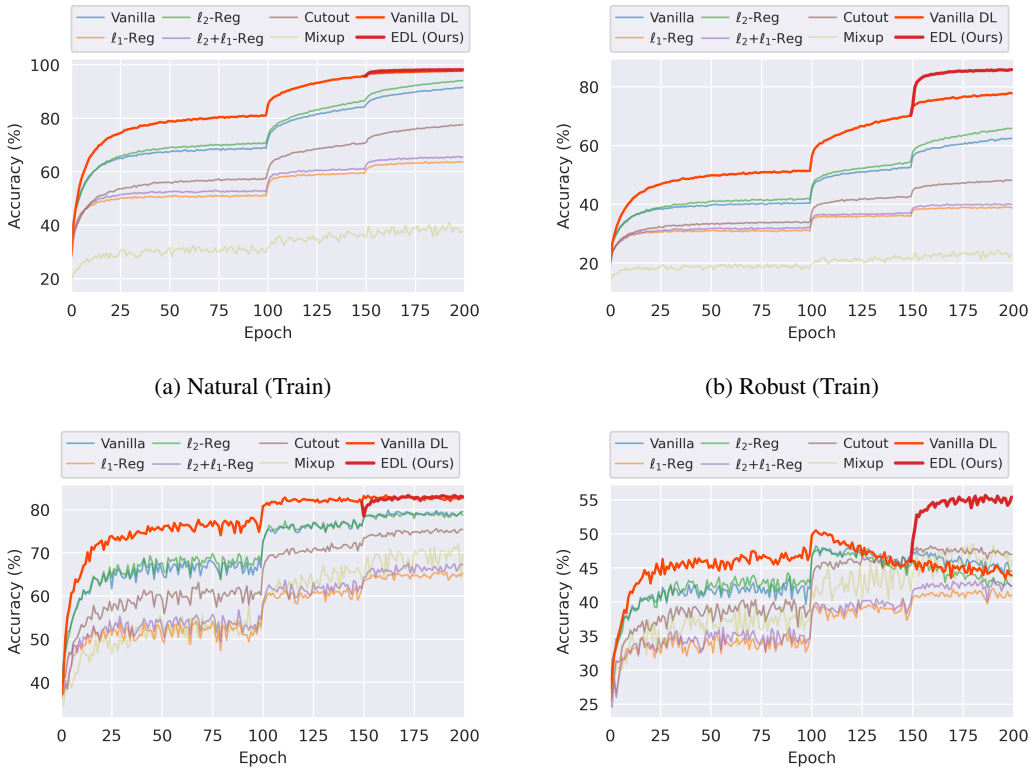
(e) Cutout

(f) Mixup

Figure 13: Training curves of baselines.

E.2.2 Comparison of All Methods

To compare all methods, we plot the natural and robust performance on the training and testing sets in Figure 14. The figures show the consistent advantage of our Elastic DL over other methods.



(a) Natural (Train)

(b) Robust (Train)

(c) Natural (Test)

(d) Robust (Test)

Figure 14: Comparison of training curves across all methods.

E.3 Ablation Studies

E.3.1 Universality

Universality across backbones, datasets, and attacks. We conduct ablation studies across different backbones, datasets, and attacks in Table 7, Table 8, and Table 9. Our proposed method shows consistent effectiveness across all settings.

Table 7: Adversarial robustness on CIFAR10 with different backbones.

Method	Natural	PGD	FGSM	C&W	AA
Vanilla DL + ResNet10	81.55	45.48	52.53	45.85	41.60
Elastic DL + ResNet10	82.69	49.54	64.52	57.37	46.30
Vanilla DL + ResNet18	83.28	45.64	53.88	41.22	43.70
Elastic DL + ResNet18	83.57	53.22	69.35	60.80	52.90
Vanilla DL + ResNet34	82.45	45.37	54.32	42.12	44.40
Elastic DL + ResNet34	82.95	55.88	70.19	61.74	53.80
Vanilla DL + ResNet50	81.22	46.83	53.75	43.64	45.10
Elastic DL + ResNet50	81.07	58.33	69.38	64.87	56.70

Table 8: Adversarial robustness on CIFAR100 with different backbones.

Method	Natural	PGD	FGSM	C&W	AA
Vanilla DL + ResNet10	55.94	22.45	26.57	18.90	21.00
Elastic DL + ResNet10	55.20	26.30	35.34	26.45	22.60
Vanilla DL + ResNet18	57.24	22.17	26.81	17.43	21.60
Elastic DL + ResNet18	57.70	27.27	37.62	28.87	26.30
Vanilla DL + ResNet34	56.18	21.77	26.14	16.38	20.80
Elastic DL + ResNet34	56.38	32.67	43.39	39.34	29.20
Vanilla DL + ResNet50	54.01	22.39	26.40	18.40	20.90
Elastic DL + ResNet50	54.64	30.29	41.48	35.24	28.10

Table 9: Adversarial robustness on Tiny-ImageNet with different backbones.

Method	Natural	PGD	FGSM	C&W	AA
Vanilla DL + ResNet10	49.60	27.17	32.46	37.91	20.20
Elastic DL + ResNet10	50.12	32.93	39.64	40.10	24.90
Vanilla DL + ResNet18	50.22	31.45	36.46	39.02	30.90
Elastic DL + ResNet18	50.52	37.60	43.10	46.64	36.30
Vanilla DL + ResNet34	50.03	33.54	37.24	37.19	29.30
Elastic DL + ResNet34	50.40	34.80	41.75	44.72	34.60
Vanilla DL + ResNet50	50.35	34.42	37.63	38.86	31.20
Elastic DL + ResNet50	50.39	37.38	42.06	41.09	35.40

E.3.2 Orthogonality to Adversarial Training

Our proposed Elastic DL framework incorporates structural priors into neural networks, complementing existing adversarial training techniques. As shown in Table 3 and Figure 15, Elastic DL can be integrated with various adversarial training methods (PGD-AT, TRADES-2.0/0.2, HAT) to consistently enhance performance.

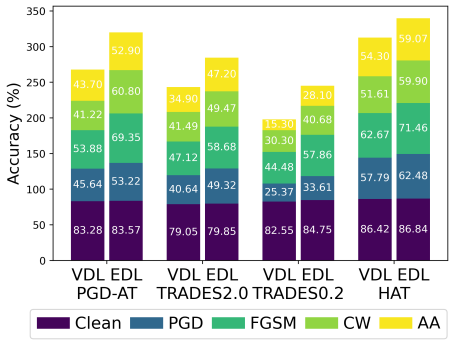


Figure 15: Different adversarial training methods. Our Elastic DL is orthogonal to existing adversarial training methods and can be combined with them to further improve performance.

E.3.3 Different Budget Measurements

In addition to the ℓ_∞-norm attack (PGD-ℓ_∞), we also validate the consistent effectiveness of our Elastic DL against ℓ_2-norm (PGD-ℓ_2) and ℓ_1-norm (SparseFool) attacks in Figure 16.

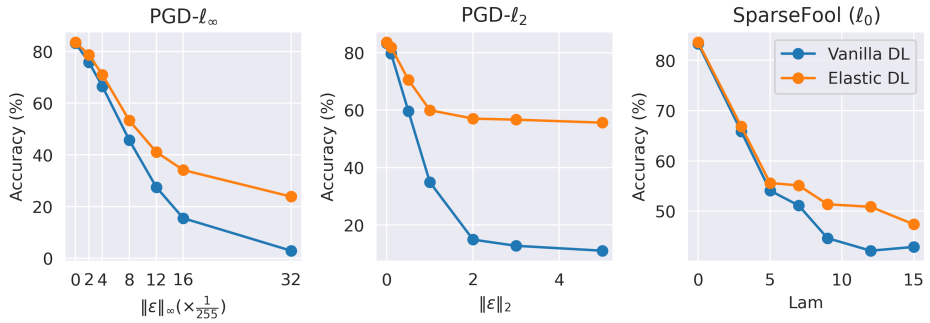


Figure 16: Different attack measurements. Our Elastic DL consistently outperforms Vanilla DL across attacks (PGD-ℓ_∞, PGD-ℓ_2, SparseFool) evaluated under various metrics ($\ell_\infty, \ell_2, \ell_0$ norms).

E.3.4 Zero-Order Gradient Analysis

To further validate that our method does not introduce obfuscated gradients, we estimate the gradient with a zero-order method ($\frac{\partial f}{\partial x} \approx \frac{f(x+\epsilon)-f(x)}{\epsilon}$) and compare it against the gradient computed by autograd. The results in Table 10 show that the relative difference between the autograd and zero-order gradients ($\frac{|\text{Grad}_{\text{zero}} - \text{Grad}_{\text{auto}}|}{|\text{Grad}_{\text{auto}}|}$) is negligible. Moreover, the error does not accumulate or grow with the number of model layers, confirming that our method does not introduce gradient-related issues.

E.3.5 Attack with Backward Pass Differentiable Approximation (BPDA)

To address concerns about gradient masking from the soft-thresholding operation, we explicitly evaluate our method under an adaptive attack based on Backward Pass Differentiable Approximation (BPDA). Concretely, during the forward pass we retain the exact proximal operator $y = \mathcal{T}_{\lambda t}(x)$, while in the backward pass we replace the gradient of the soft-thresholding operation with an identity

NUM. OF LAYERS	3	5	8	10	15	20	25	30	40	50
RELATIVE ERROR	0.00066	0.00095	0.00152	0.00092	0.00118	0.00098	0.00112	0.00082	0.00060	0.00093

Table 10: Zero-order gradient analysis: relative difference between autograd and zero-order gradients across network depths.

mapping, $\frac{\partial y}{\partial x} \approx I$. Under this stronger adaptive attack setting, robustness remains stable, indicating that the observed robustness is not solely due to gradient obfuscation.

BUDGET	0	8/255	16/255	32/255	64/255
ELASTIC DL	83.57	53.29	34.13	23.84	12.45
ELASTIC DL + BPDA	83.57	53.20	34.08	23.51	12.44

Table 11: Adversarial robustness under standard adaptive attack versus BPDA attack, showing comparable performance and ruling out gradient obfuscation.

E.3.6 HIDDEN EMBEDDING VISUALIZATION

We conduct visualization analyses on the hidden embeddings to gain further insight into the effectiveness of our proposed Elastic DL. We begin by quantifying the relative difference between clean embeddings ($\boldsymbol{x}_i$) and their adversarially attacked counterparts ($\boldsymbol{x}'_i$) across all layers of the network in Figure 17. As shown in Figure 18 and Figure 19, adversarial perturbations induce a substantially larger relative difference in the hidden embeddings of Vanilla DL than in those of Elastic DL, with this gap becoming increasingly pronounced in deeper layers, indicating that Elastic DL is markedly more effective at containing the propagation of adversarial perturbations through the network rather than merely suppressing them at the input. Beyond this layer-wise quantification, we further visualize representative embedding instances to illustrate the effect qualitatively: under adversarial perturbation, the disrupted hidden embedding patterns cause Vanilla DL to shift its prediction away from the groundtruth label, whereas Elastic DL preserves embedding patterns close to those of the clean input and thus maintains the correct prediction.

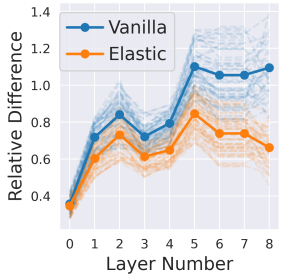
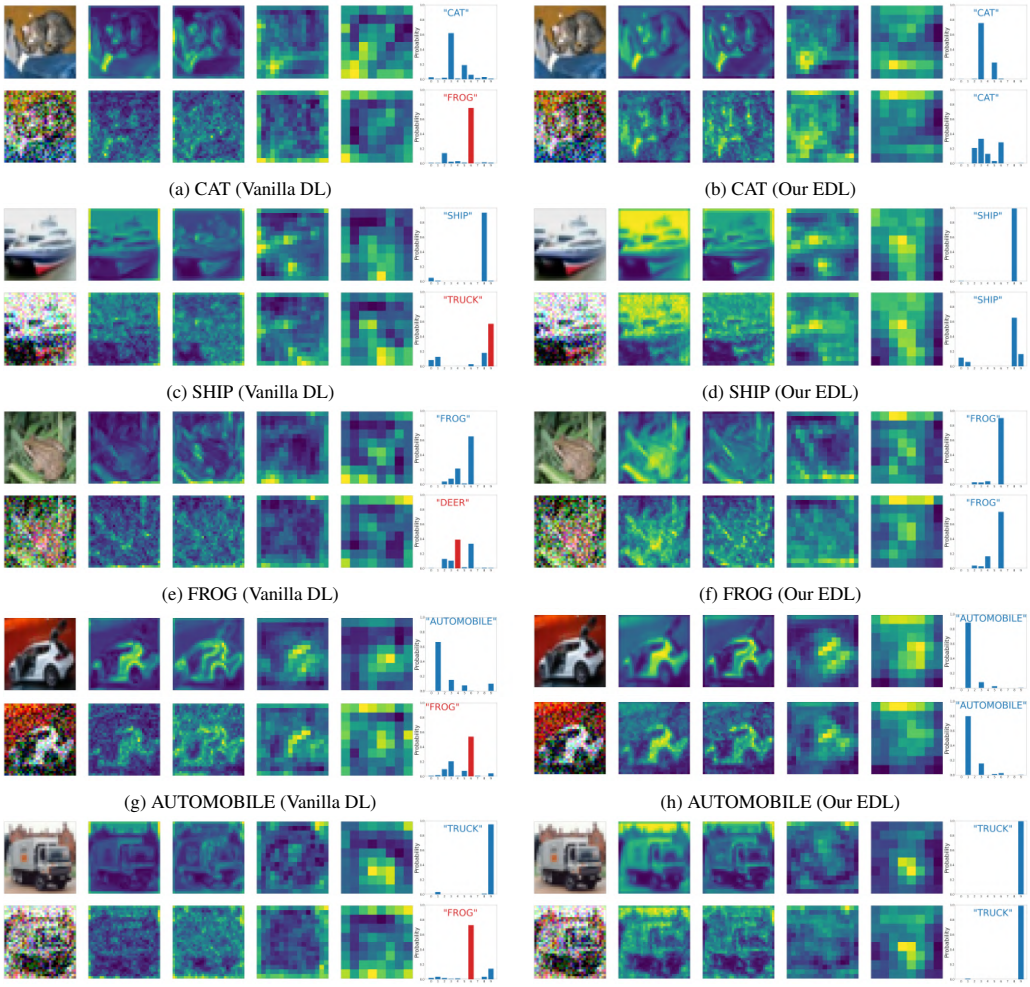


Figure 17: Embedding difference across layers.

Here are instances of CAT, SHIP, FROG, AUTOMOBILE, and TRUCK:



(a) CAT (Vanilla DL) (b) CAT (Our EDL)

(c) SHIP (Vanilla DL) (d) SHIP (Our EDL)

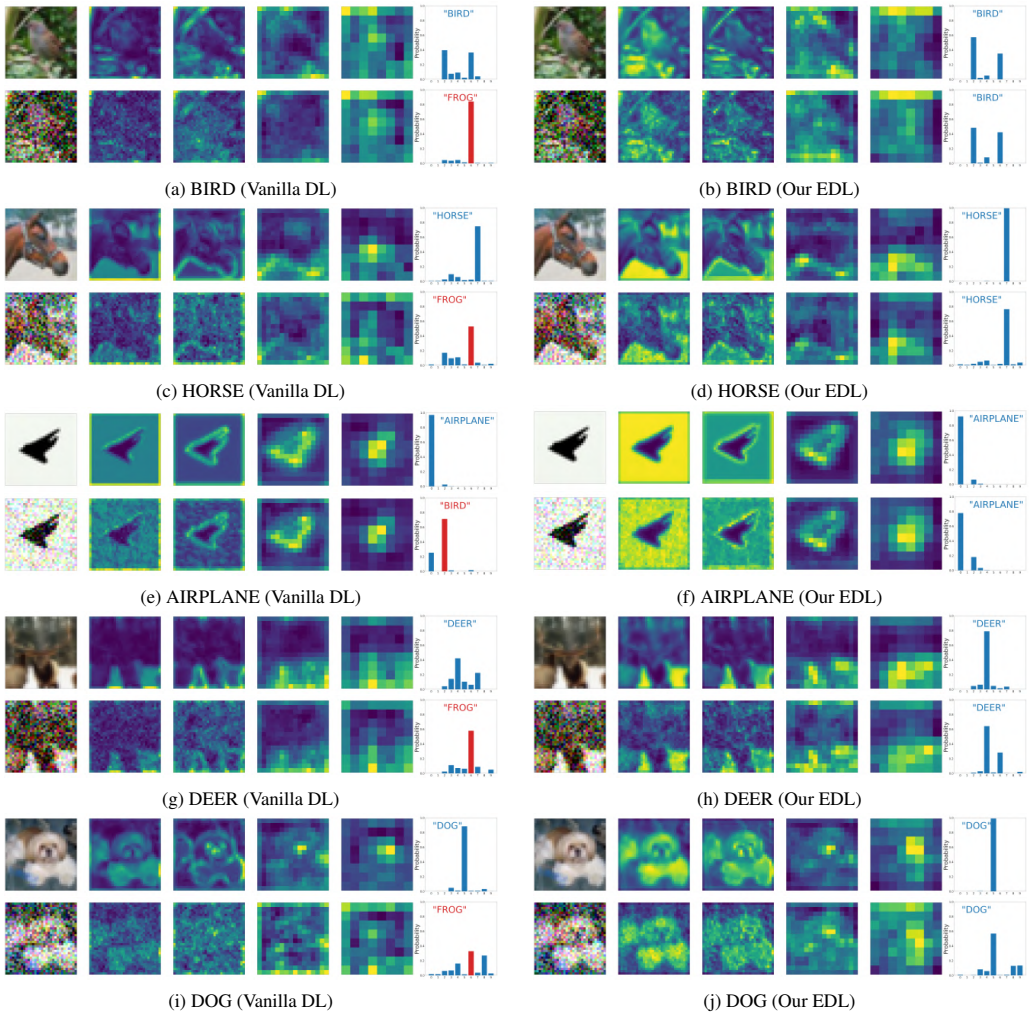
(e) FROG (Vanilla DL) (f) FROG (Our EDL)

(g) AUTOMOBILE (Vanilla DL) (h) AUTOMOBILE (Our EDL)

(i) TRUCK (Vanilla DL) (j) TRUCK (Our EDL)

Figure 18: Hidden embedding visualization (Part 1).

Here are instances of BIRD, HORSE, AIRPLANE, DEER, and DOG:



(a) BIRD (Vanilla DL) (b) BIRD (Our EDL)

(c) HORSE (Vanilla DL) (d) HORSE (Our EDL)

(e) AIRPLANE (Vanilla DL) (f) AIRPLANE (Our EDL)

(g) DEER (Vanilla DL) (h) DEER (Our EDL)

(i) DOG (Vanilla DL) (j) DOG (Our EDL)

Figure 19: Hidden embedding visualization (Part 2).

E.3.7 Reconstruction Process

Image & noise reconstruction. In conventional feedforward neural networks, adding a perturbation ϵ to the input can lead the model to make incorrect predictions. As illustrated in Figure 20, our approach instead reconstructs both the clean image x and the perturbation ϵ through a dictionary learning process. To evaluate the effectiveness of our method, we quantify the reconstruction error between the recovered noise $\hat{\epsilon}$ under our Elastic DL framework and noise generated by various methods (random noise, transfer noise from ResNet/Vanilla DL, and adaptive noise from Elastic DL). As shown in Table 12, the recovered noise from our approach exhibits the smallest difference relative to the adaptive noise in Elastic DL, demonstrating that our proposed framework more effectively reconstructs and mitigates the impact of adversarial noise on predictions.

Table 12: Reconstruction error between the recovered noise $\hat{\epsilon}$ and various input noises, including random noise (ϵ_{random}), transfer noise from ResNet (ϵ_{resnet}) and Vanilla DL ($\epsilon_{\text{vanilla}}$), and adaptive noise from our Elastic DL ($\epsilon_{\text{elastic}}$). Our Elastic DL achieves the smallest reconstruction error, indicating that our approach can adaptively recover and neutralize the input perturbation, thereby mitigating its impact.

ERROR	$\|\cdot\|_1$	$\|\cdot\|_2$	$\|\cdot\|_\infty$
$\epsilon_{\text{RANDOM}} - \hat{\epsilon}$	1294.75 ± 406.78	26.09 ± 7.04	0.901 ± 0.10
$\epsilon_{\text{RESNET}} - \hat{\epsilon}$	131.51 ± 10.53	2.93 ± 0.22	0.163 ± 0.01
$\epsilon_{\text{VANILLA}} - \hat{\epsilon}$	129.07 ± 13.22	2.85 ± 0.26	0.157 ± 0.01
$\epsilon_{\text{ELASTIC}} - \hat{\epsilon}$	$\mathbf{122.62 \pm 9.92}$	$\mathbf{2.69 \pm 0.22}$	$\mathbf{0.149 \pm 0.01}$

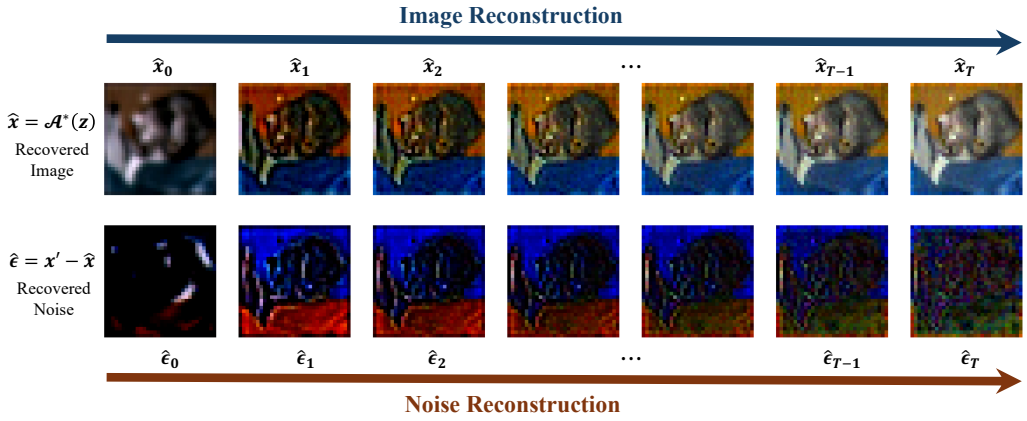


Figure 20: Reconstruction process.

Here are instances of the reconstruction process on ImageNet:

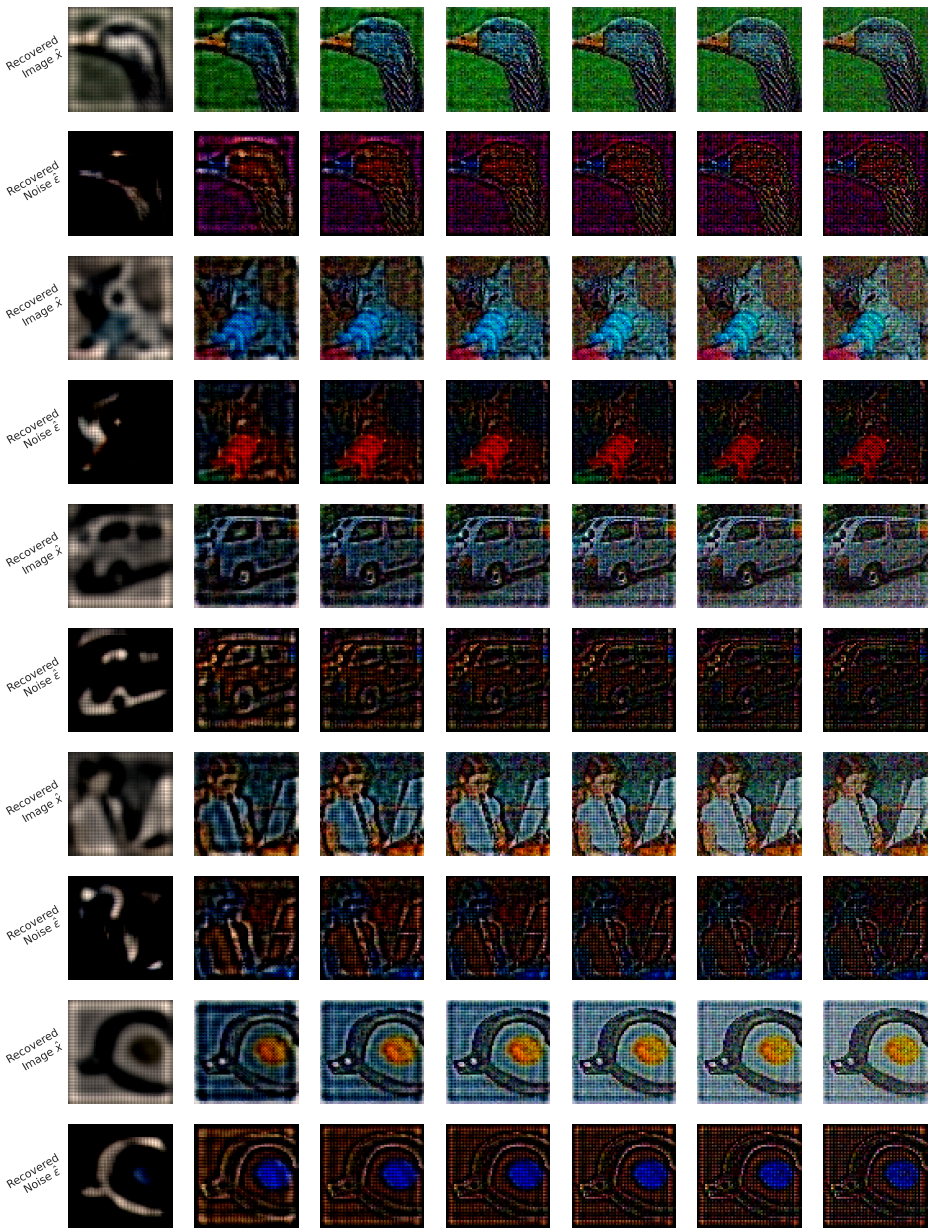


Figure 21: Reconstruction process (ImageNet).

Here are instances of the reconstruction process on CIFAR10 (Part 1):

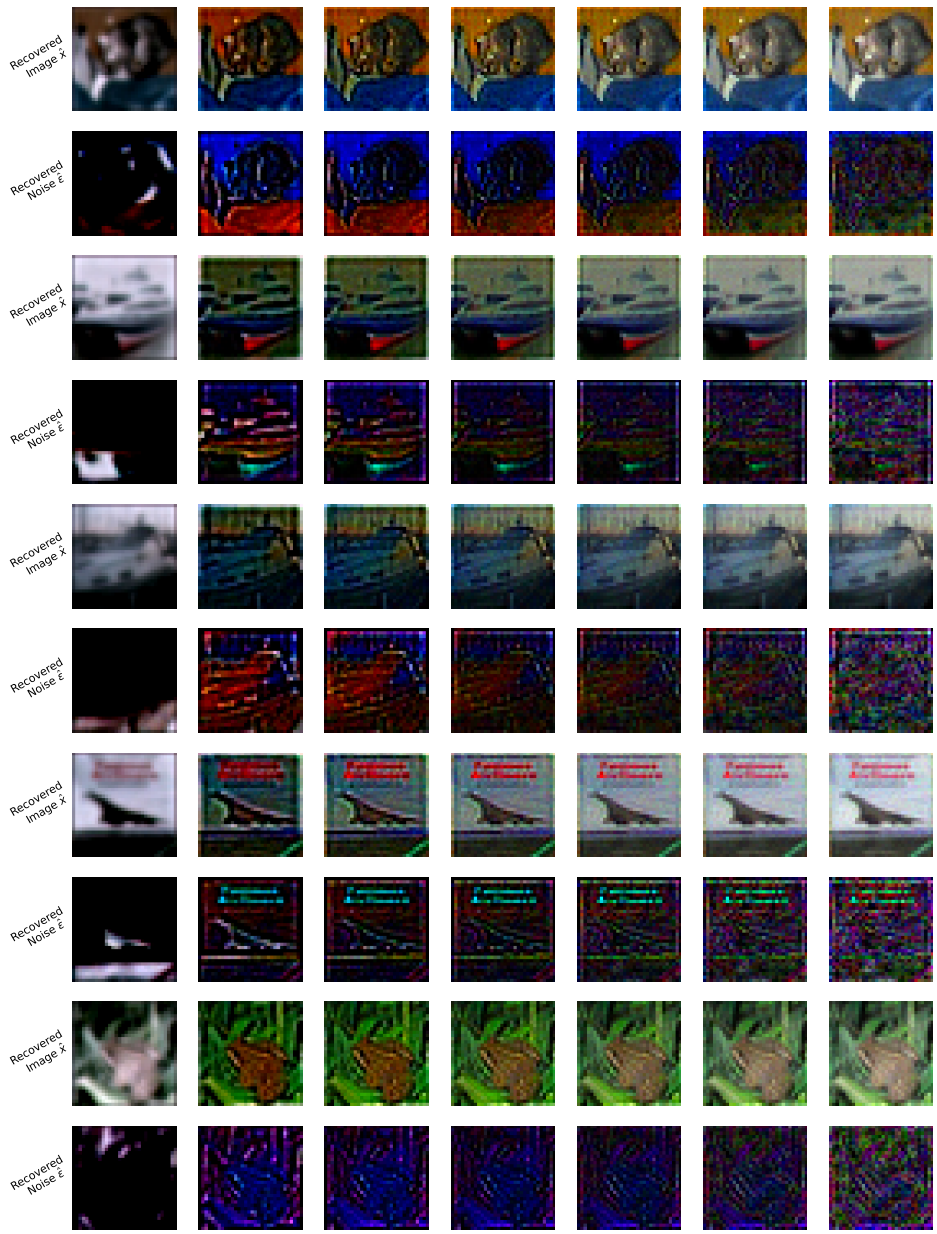


Figure 22: Reconstruction process (CIFAR10, Part 1).

Here are instances of the reconstruction process on CIFAR10 (Part 2):

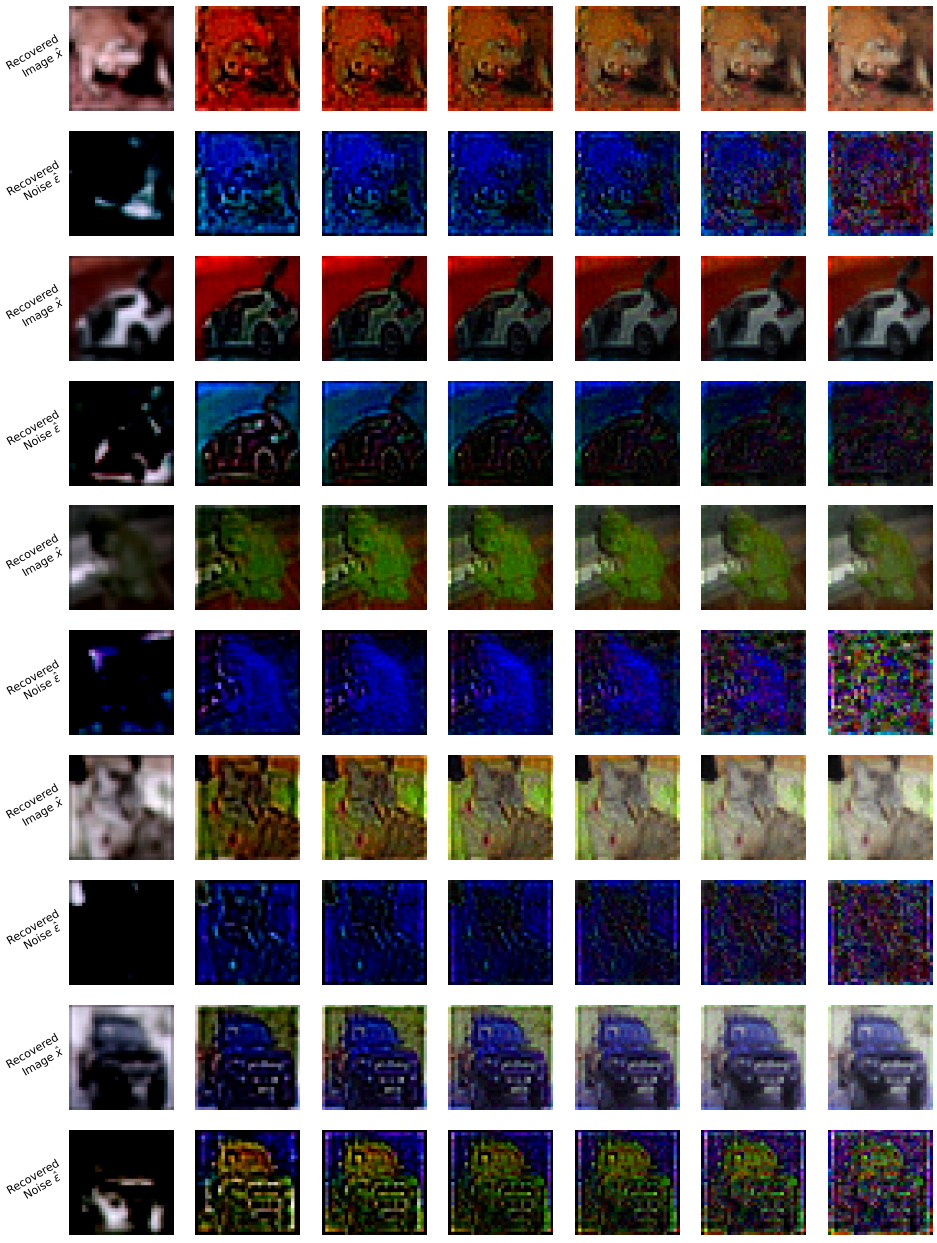


Figure 23: Reconstruction process (CIFAR10, Part 2).