

420 **Part I**

421 **Appendix**

422 **Table of Contents**

---

|     |  |           |
|-----|--|-----------|
| 424 | <b>A Proof of Theorem 6.2</b>  | <b>14</b> |
| 425 | <b>B The Parametrization Gap</b>   | <b>15</b> |
| 426 | <b>C Additional Experimental Details</b>                                       | <b>17</b> |
| 427 | <b>D Constraint Satisfaction in LORE</b>                                       | <b>18</b> |
| 428 | <b>E Generalization Gap in Adversarial Training</b>                            | <b>18</b> |
| 429 | <b>F Deviation Between Cosine Similarities</b>                                 | <b>19</b> |
| 430 | <b>G Impact of Dual Network on Model Performance</b>                           | <b>20</b> |
| 431 | G.1 Comparison of Alternative Architectures for the Dual Function . . . . .    | 20        |
| 432 | G.2 Effect of the Dual Network $\lambda_\omega(x)$ on Clean Accuracy . . . . . | 21        |
| 433 | <b>H Revisiting Embedding Models</b>   | <b>21</b> |
| 434 | <b>I Impact of <math>K</math> on Model Performance</b>                         | <b>23</b> |
| 435 | <b>J Additional Experimental Results</b>                                       | <b>23</b> |
| 436 | J.1 Square Attack Evaluation . . . . .   | 24        |
| 437 | J.2 Evaluation Under Gaussian Noise Corruption . . . . .                       | 24        |
| 438 | J.3 In-domain Image Classification . . . . .                                   | 25        |
| 439 | J.4 Zero-shot Image Classification . . . . .                                   | 25        |
| 440 | J.5 Out-of-Distribution Robustness . . . . .                                   | 25        |
| 441 | <b>K Computation and Efficiency Analysis</b>                                   | <b>25</b> |
| 442 | K.1 Convergence Efficiency. . . . .  | 26        |
| 443 | K.2 Impact of $\lambda_\omega$ architecture on Training Time. . . . .          | 26        |
| 444 | K.3 Impact of $K$ on Training Time. . . . .                                    | 26        |

---

## 448 A Proof of Theorem 6.2

449 We define the following quantities:

$$R = \min_{\phi \in \mathcal{H}} \ell_{\text{adv}}(\phi; \phi_0), \quad R_\rho = \min_{\phi \in \mathcal{H}_\rho} \ell_{\text{adv}}(\phi; \phi_0).$$

450 And

$$\begin{aligned} \phi_\rho^* &= \operatorname{argmin}_{\phi \in \mathcal{H}_\rho} \mathbb{E}_{x \sim \mathcal{D}} [\ell_{\text{adv}}(\phi, x; \phi_{\theta_0})] = \operatorname{argmin}_{\phi \in \mathcal{H}_\rho} \ell_{\text{adv}}(\phi; \phi_{\theta_0}), \\ \text{where } \ell_{\text{adv}}(\phi, x; \phi_{\theta_0}) &\triangleq \max_{\delta \in \Delta} d(\phi(x + \delta), \phi_{\theta_0}(x)). \end{aligned} \quad (9)$$

Since  $\mathcal{H}_\rho \subset \mathcal{H}$ , we have  $R_\rho \geq R$ . The sub-optimality gap is bounded by:

$$0 \leq R_\rho - R \leq \sqrt{k} \cdot \text{Lipschitz}(\phi_\rho^* - \phi^*)\varepsilon + \|\phi_\rho^* - \phi^*\| \leq \sqrt{k}(L_\rho^* + L')\varepsilon + \|\phi_\rho^* - \phi^*\|.$$

451 *Proof.* Let  $\delta_1^* = \arg \max_{\delta \in \Delta} d(\phi_1(x + \delta), \phi_0(x))$  denote the perturbation that maximizes the  
452 adversarial loss for model  $\phi_1$  relative to the reference model  $\phi_0$ . For any two models  $\phi_1, \phi_2$ , we aim  
453 to bound the difference in their adversarial losses, considering assumption 6.1:

$$|\ell_{\text{adv}}(\phi_1, x; \phi_0) - \ell_{\text{adv}}(\phi_2, x; \phi_0)| \quad (10)$$

$$= \left| \max_{\delta \in \Delta} d(\phi_1(x + \delta), \phi_0(x)) - \max_{\delta \in \Delta} d(\phi_2(x + \delta), \phi_0(x)) \right| \quad (11)$$

$$\leq |d(\phi_1(x + \delta_1^*), \phi_0(x)) - d(\phi_2(x + \delta_1^*), \phi_0(x))| \quad (12)$$

$$\leq \|\phi_1(x + \delta_1^*) - \phi_2(x + \delta_1^*)\|. \quad (13)$$

454 Alternatively, we set the Lipschitz constant to 1 for brevity; however, it is worth noting that in general,  
455 this constant appears explicitly in the bound and should be carried through the analysis. Note that  
456 without loss of generality, we have assumed that  $\max_{\delta \in \Delta} d(\phi_1(x + \delta), \phi_0(x)) - \max_{\delta \in \Delta} d(\phi_2(x + \delta), \phi_0(x)) \geq 0$ . If that is not the case, then we have to replace  $\delta_1^*$  with  $\delta_2^* = \arg \max_{\delta \in \Delta} d(\phi_2(x + \delta), \phi_0(x))$  throughout the proof. We now decompose this difference using the triangle inequality:

$$\|\phi_1(x + \delta_1^*) - \phi_2(x + \delta_1^*)\| \quad (14)$$

$$= \|(\phi_1(x + \delta_1^*) - \phi_1(x)) - (\phi_2(x + \delta_1^*) - \phi_2(x)) + (\phi_1(x) - \phi_2(x))\| \quad (15)$$

$$\leq \|(\phi_1(x + \delta_1^*) - \phi_2(x + \delta_1^*)) - (\phi_1(x) - \phi_2(x))\| + \|\phi_1(x) - \phi_2(x)\|. \quad (16)$$

459 Assuming that the function  $\phi_1 - \phi_2$  is Lipschitz continuous with constant  $L$ , we obtain:

$$\|(\phi_1(x + \delta_1^*) - \phi_2(x + \delta_1^*)) - (\phi_1(x) - \phi_2(x))\| \leq L\|\delta_1^*\|. \quad (17)$$

460 Assuming  $\delta_1^* \in \mathbb{R}^k$  with  $\|\delta_1^*\|_2 \leq \varepsilon\sqrt{k}$ —which corresponds to assuming that  $d$  is Lipschitz with  
461 respect to the Euclidean norm, and that  $\varepsilon$  is bounded in the  $\ell_\infty$  norm—we have:

$$|\ell_{\text{adv}}(\phi_1, x; \phi_0) - \ell_{\text{adv}}(\phi_2, x; \phi_0)| \leq L\sqrt{k}\varepsilon + \|\phi_1(x) - \phi_2(x)\|. \quad (18)$$

462 We now extend this pointwise bound to the expected adversarial loss:

$$|\ell_{\text{adv}}(\phi_1; \phi_0) - \ell_{\text{adv}}(\phi_2; \phi_0)| = |\mathbb{E}_x [\ell_{\text{adv}}(\phi_1, x; \phi_0) - \ell_{\text{adv}}(\phi_2, x; \phi_0)]| \quad (19)$$

$$\leq \mathbb{E}_x [|\ell_{\text{adv}}(\phi_1, x; \phi_0) - \ell_{\text{adv}}(\phi_2, x; \phi_0)|] \quad (20)$$

$$\leq \mathbb{E}_x [L\sqrt{k}\varepsilon + \|\phi_1(x) - \phi_2(x)\|] \quad (21)$$

$$= L\sqrt{k}\varepsilon + \|\phi_1 - \phi_2\|, \quad (22)$$

463 where  $\|\phi_1 - \phi_2\|$  denotes the expected difference in their outputs over the input distribution.

464 Similarly, setting  $\varepsilon = 0$ , the adversarial loss reduces to the clean loss, yielding a corresponding  
465 bound:

$$|\ell_{\text{clean}}(\phi_1, \phi_0) - \ell_{\text{clean}}(\phi_2, \phi_0)| \leq \|\phi_1 - \phi_2\|.$$

466 Finally, applying this to the case  $\phi_1 = \phi_\rho^*$  and  $\phi_2 = \phi^*$ , we obtain the desired bound on the deviation  
467 in adversarial loss due to constraining the hypothesis space.  $\square$

## B The Parametrization Gap

In this paper, we solved our constrained optimization problem, by optimizing its dual problem, and using neural networks for the lagrangian multipliers. particularly solving:

$$d^* = \max_{\psi \in \Psi} \min_{\theta \in \Theta} \mathbb{E}[\max_{\delta \in \Delta} d(\phi_\theta(x + \delta), \phi_0(x)) + \lambda_\psi(x)(d(\phi_\theta(x), \phi_0(x)) - \rho m(x))]$$

Instead of the original constrained optimization problem. in this section we are interested in deriving some bounds on the duality gap, following the proof from Robey et al. [2021].

*Proof.*

**Assumption B.1.** For all  $g \in \text{conv}(\mathcal{H})$ , there exists  $\tilde{\theta} \in \Theta$  such that

$$\|\phi_{\tilde{\theta}} - g^*\| \leq \eta,$$

where  $\eta > 0$  is a sufficiently small constant.

First, ignoring the parametrization of  $\lambda$ , and assuming  $\lambda$  can be any function from  $\Lambda = \{\lambda : \mathcal{X} \rightarrow \mathbb{R}_+\}$ , we consider the Lagrangian:

$$L(\phi, \lambda) = \mathbb{E} \left[ \max_{\delta \in \Delta} d(\phi(x + \delta), \phi_0(x)) + \lambda(x)(d(\phi(x), \phi_0(x)) - \rho m(x)) \right]$$

$$d^* = \sup_{\lambda \in \Lambda} \inf_{\theta \in \Theta} L(\phi_\theta, \lambda)$$

Now consider the original problem:

$$p^* = \inf_{\theta \in \Theta} \mathbb{E}_{x \sim \mathcal{D}} \left[ \max_{\delta \in \Delta} d(\phi_\theta(x + \delta), \phi_0(x)) \right],$$

$$\text{s.t. } d(\phi_\theta(x), \phi_{\theta_0}(x)) \leq \rho m(x), \quad \text{for almost every } x \in \mathcal{X}$$

If the function class  $\mathcal{H}$  parametrized by  $\theta$  were convex, this would be a convex program. Since by definition,  $\phi_0 \in \mathcal{H} = \{\phi_\theta : \theta \in \Theta\}$ , there exists  $\theta \in \Theta$  such that  $d(\phi_\theta(x), \phi_0(x)) = 0 < \rho m(x)$  for all  $x \in \mathcal{X}$ . Thus, Slater's condition is satisfied.

Therefore, if  $\mathcal{H}$  were convex, we would have strong duality, i.e.,  $p^* = d^*$ . However, for most typical neural networks,  $\mathcal{H}$  is non-convex. By weak duality, we always have:  $d^* \leq p^*$ . To derive a lower bound, consider the following problem for some positive constant  $\eta > 0$ :

$$\tilde{p}^* = \inf_{g \in \text{conv}(\mathcal{H})} \mathbb{E}_{x \sim \mathcal{D}} \left[ \max_{\delta \in \Delta} d(g(x + \delta), \phi_0(x)) \right],$$

$$\text{s.t. } d(g(x), \phi_{\theta_0}(x)) \leq \rho m(x) - \eta, \quad \text{for almost every } x \in \mathcal{X}$$

This is now a convex program. Since  $\phi_0$  itself satisfies  $d(\phi_0(x), \phi_0(x)) = 0 < \rho m(x) - \eta$ , Slater's condition is again satisfied. Hence, strong duality holds, and the Lagrangian becomes:

$$\tilde{L}(g, \lambda) = \mathbb{E} \left[ \max_{\delta \in \Delta} d(g(x + \delta), \phi_0(x)) + \lambda(x)(d(g(x), \phi_0(x)) - \rho m(x) + \eta) \right] = L(g, \lambda) + \eta \mathbb{E}[\lambda(x)]$$

Thus,

$$\tilde{p}^* = \sup_{\lambda \in \Lambda} \inf_{g \in \text{conv}(\mathcal{H})} \tilde{L}(g, \lambda)$$

Assuming the infimum and supremum are attained at  $g^*$  and  $\tilde{\lambda}^*$ , we have:

$$d^* - \sup_{\lambda \in \Lambda} \inf_{\theta \in \Theta} L(\phi_\theta, \lambda) \geq \inf_{\theta \in \Theta} L(\phi_\theta, \tilde{\lambda}^*) = \inf_{\phi \in \mathcal{H}} L(\phi, \tilde{\lambda}^*) \geq \inf_{g \in \text{conv}(\mathcal{H})} L(g, \tilde{\lambda}^*)$$

Using the relation between  $\tilde{L}$  and  $L$ , we obtain:

$$d^* \geq \inf_{g \in \text{conv}(\mathcal{H})} L(g, \tilde{\lambda}^*) = \inf_g \left[ \tilde{L}(g, \tilde{\lambda}^*) - \eta \mathbb{E}[\tilde{\lambda}(x)] \right] = \tilde{L}(g^*, \tilde{\lambda}^*) - \eta \mathbb{E}[\tilde{\lambda}(x)] = \tilde{p}^* - \eta \mathbb{E}[\tilde{\lambda}(x)]$$

491 Since  $g^*$  is strictly feasible for the relaxed constraint, the complementary slackness condition implies  
 492  $\tilde{\lambda}^*(x) = 0$  for almost every  $x$ . Therefore:

$$d^* \geq \tilde{p}^* = \mathbb{E} \left[ \max_{\delta \in \Delta} d(g^*(x + \delta), \phi_0(x)) \right] = \ell_{\text{adv}}(g^*, \phi_0)$$

493 Using the suboptimality bound from Lemma A, we have:

$$|\ell_{\text{adv}}(g^*, \phi_0) - \ell_{\text{adv}}(\phi_{\tilde{\theta}}, \phi_0)| \leq L\sqrt{k}\varepsilon + \|g^* - \phi_{\tilde{\theta}}\|, \quad |\ell_{\text{clean}}(g^*, \phi_0) - \ell_{\text{clean}}(\phi_{\tilde{\theta}}, \phi_0)| \leq \|g^* - \phi_{\tilde{\theta}}\|$$

494 Since  $g^*$  is feasible in the relaxed problem with the stricter constraint  $d(g^*(x), \phi_0(x)) \leq \rho m(x) - \eta$ ,  
 495 for some  $\tilde{\theta}^* \in \Theta$  approximating  $g^*$  such that  $\|g^* - \phi_{\tilde{\theta}^*}\| < \eta$ , the function  $\phi_{\tilde{\theta}^*}$  is feasible in the  
 496 original problem, because for almost every  $x \in \mathcal{X}$

$$\ell_{\text{clean}}(\phi_{\tilde{\theta}^*}(x), \phi_0(x)) \leq \ell_{\text{clean}}(g^*(x), \phi_0(x)) + \|g^* - \phi_{\tilde{\theta}^*}\| \leq \rho m(x) - \eta + \|g^* - \phi_{\tilde{\theta}^*}\| < \rho m(x).$$

497 Since  $p^*$  is the minimum over all feasible  $\theta \in \Theta$ , it follows that  $p^* \leq \ell_{\text{adv}}(\phi_{\tilde{\theta}^*}, \phi_0)$ .

$$d^* \geq \ell_{\text{adv}}(g^*, \phi_0) \geq \ell_{\text{adv}}(\phi_{\tilde{\theta}^*}, \phi_0) - L\sqrt{k}\varepsilon - \|g^* - \phi_{\tilde{\theta}^*}\| \geq \ell_{\text{adv}}(\phi_{\tilde{\theta}^*}, \phi_0) - L\sqrt{k}\varepsilon - \eta$$

498 Therefore, since  $\theta^*$  is the optimal solution to the original problem:

$$d^* \geq p^* - L\sqrt{k}\varepsilon - \eta$$

499 Finally, we note that although we have treated the dual space as the full infinite-dimensional set  $\Lambda =$   
 500  $\{\lambda : \mathcal{X} \rightarrow [0, \infty)\}$ , in this work we have restricted  $\lambda$  to a finite-dimensional family  $\{\lambda_\omega\}_{\omega \in \Omega} \subset \Lambda$   
 501 that uniformly approximates its elements. Concretely, if for every  $\lambda \in \Lambda$  there exists  $\omega \in \Omega$  with  
 502  $\|\lambda - \lambda_\omega\|_{L^1(\mathcal{D})} \leq \xi$ , then replacing

$$\sup_{\lambda \in \Lambda} \inf_{\phi \in \mathcal{H}} L(\phi, \lambda) \quad \text{by} \quad \sup_{\omega \in \Omega} \inf_{\phi \in \mathcal{H}} L(\phi, \lambda_\omega)$$

503 only incurs an arbitrarily small error  $O(\xi)$ . All weak-duality arguments carry over immediately, and  
 504 under Slater's condition the resulting strong-duality statement remains valid up to this negligible  
 505 approximation.

506 One viewpoint is that limiting the expressivity of  $\lambda_\omega$  through parametrization effectively relaxes the  
 507 constraints, as the network cannot fully ensure the constraints are met. As an extreme case, consider  
 508 when  $\lambda_\omega(x) = \omega$  for all  $x \in \mathcal{X}$ . In this case:

$$\lambda_\omega(x) \equiv \omega, \quad \omega \geq 0.$$

509 Then the (relaxed) Lagrangian becomes

$$L(\phi, \omega) = \mathbb{E} \left[ \max_{\delta \in \Delta} d(\phi(x + \delta), \phi_0(x)) \right] + \omega \mathbb{E} [d(\phi(x), \phi_0(x)) - \rho m(x)].$$

510 Optimizing first over  $\phi$  (so that  $\mathbb{E}[\max_{\delta} d]$  is fixed) and then taking the supremum over  $\omega \geq 0$  forces

$$\mathbb{E} [d(\phi(x), \phi_0(x))] - \rho \mathbb{E} [m(x)] \leq 0,$$

511 otherwise  $L(\phi, \omega) \rightarrow +\infty$  as  $\omega \rightarrow +\infty$ . In other words, the constant- $\lambda$  relaxation exactly enforces

$$\mathbb{E} [d(\phi(x), \phi_0(x))] \leq \rho \mathbb{E} [m(x)].$$

512 Thus, by limiting the expressivity of  $\lambda$ , we move from the original per-sample constraint

$$d(\phi(x), \phi_0(x)) \leq \rho m(x) \quad \forall x,$$

513 to the weaker but still meaningful average constraint

$$\mathbb{E} [d(\phi(x), \phi_0(x))] \leq \rho \mathbb{E} [m(x)].$$

514

□

## C Additional Experimental Details

In this appendix, we provide further experimental details beyond those given in the main text. Experiments were conducted using 8 NVIDIA HGX H100 80GB GPUs.

**Training hyperparameters.** We report below the training settings used across all experiments. Unless otherwise noted, all models were trained using AdamW with a weight decay of  $1 \times 10^{-4}$ , a cosine learning rate scheduler, and adversarial training with PGD (10 iterations, step size  $\varepsilon/4$ ) under an  $\ell_\infty$  constraint. Each  $\lambda$  network used the 2-layer `linear_mlp` architecture, with a hidden dimension of 512, and was optimized via  $K = 5$  inner primal updates with learning rate  $5 \times 10^{-4}$ . More experimental details are provided in Table 4. Additional information about all figures and tables in the paper is summarized in Table 5.

Table 4: Training hyperparameters for all models trained with LORE. All models trained with FARE use the same number of epochs and learning rate as the corresponding LORE setting.

| Model                    | Training Dataset | Epochs | Batch size (per device) | LR   | $\rho$ | $K$ -iter | $\lambda$ LR |
|--------------------------|------------------|--------|-------------------------|------|--------|-----------|--------------|
| <i>LORE<sup>1</sup></i>  |                  |        |                         |      |        |           |              |
| CLIP ViT-B/32            | ImageNet         | 2      | 448                     | 1e-5 | 0.1    | 5         | 5e-4         |
| CLIP ViT-B/32            | ImageNet-100     | 5      | 448                     | 1e-5 | 0.1    | 5         | 5e-4         |
| CLIP ViT-B/32            | CIFAR10          | 5      | 448                     | 1e-5 | 0.01   | 5         | 5e-4         |
| <i>LORE<sup>2</sup></i>  |                  |        |                         |      |        |           |              |
| CLIP ViT-B/16            | ImageNet-100     | 5      | 128                     | 1e-5 | 0.1    | 5         | 5e-4         |
| CLIP ViT-B/32-LAION      | ImageNet-100     | 5      | 448                     | 1e-5 | 0.15   | 5         | 5e-4         |
| CLIP ConvNeXt-B          | ImageNet-100     | 5      | 64                      | 1e-5 | 0.15   | 5         | 5e-4         |
| CLIP ViT-B/32            | ImageNet         | 2      | 448                     | 1e-5 | 0.1    | 5         | 5e-4         |
| CLIP ViT-B/32            | ImageNet-100     | 5      | 448                     | 1e-5 | 0.1    | 5         | 5e-4         |
| CLIP ViT-B/32            | CIFAR10          | 5      | 448                     | 1e-5 | 0.01   | 5         | 5e-4         |
| <i>LORE<sup>4</sup></i>  |                  |        |                         |      |        |           |              |
| CLIP ViT-B/16            | ImageNet-100     | 5      | 128                     | 1e-5 | 0.2    | 5         | 5e-4         |
| CLIP ViT-B/32-LAION      | ImageNet-100     | 5      | 448                     | 1e-5 | 0.15   | 5         | 5e-4         |
| CLIP ConvNeXt-B          | ImageNet-100     | 5      | 64                      | 1e-5 | 0.15   | 5         | 5e-4         |
| DINOv2 ViT-S/14          | ImageNet         | 2      | 128                     | 1e-5 | 0.05   | 5         | 5e-4         |
| DINOv2 ViT-B/14          | ImageNet         | 1      | 64                      | 1e-5 | 0.1    | 5         | 5e-4         |
| CLIP ViT-B/32            | ImageNet         | 3      | 448                     | 1e-5 | 0.1    | 5         | 5e-4         |
| CLIP ViT-B/32            | ImageNet-100     | 5      | 448                     | 1e-5 | 0.1    | 5         | 5e-4         |
| CLIP ViT-B/32            | CIFAR10          | 5      | 448                     | 1e-5 | 0.01   | 5         | 5e-4         |
| <i>LORE<sup>6</sup></i>  |                  |        |                         |      |        |           |              |
| CLIP ViT-B/32            | ImageNet         | 3      | 448                     | 1e-5 | 0.2    | 5         | 5e-4         |
| CLIP ViT-B/32            | ImageNet-100     | 5      | 448                     | 1e-5 | 0.1    | 5         | 5e-4         |
| <i>LORE<sup>8</sup></i>  |                  |        |                         |      |        |           |              |
| CLIP ViT-B/16            | ImageNet-100     | 5      | 128                     | 1e-5 | 0.2    | 5         | 5e-4         |
| CLIP ViT-B/32-LAION      | ImageNet-100     | 5      | 448                     | 1e-5 | 0.15   | 5         | 5e-4         |
| CLIP ConvNeXt-B          | ImageNet-100     | 5      | 64                      | 1e-5 | 0.15   | 5         | 5e-4         |
| DINOv2 ViT-S/14          | ImageNet         | 2      | 128                     | 1e-5 | 0.05   | 5         | 5e-4         |
| DINOv2 ViT-B/14          | ImageNet         | 1      | 64                      | 1e-5 | 0.1    | 5         | 5e-4         |
| CLIP ViT-B/32            | ImageNet         | 3      | 448                     | 1e-5 | 0.2    | 5         | 5e-4         |
| CLIP ViT-B/32            | ImageNet-100     | 5      | 448                     | 1e-5 | 0.1    | 5         | 5e-4         |
| <i>LORE<sup>10</sup></i> |                  |        |                         |      |        |           |              |
| CLIP ViT-B/32            | ImageNet         | 3      | 448                     | 1e-5 | 0.2    | 5         | 5e-4         |
| CLIP ViT-B/32            | ImageNet-100     | 5      | 448                     | 1e-5 | 0.1    | 5         | 5e-4         |
| <i>LORE<sup>16</sup></i> |                  |        |                         |      |        |           |              |
| DINOv2 ViT-S/14          | ImageNet         | 2      | 128                     | 1e-5 | 0.05   | 5         | 5e-4         |
| DINOv2 ViT-B/14          | ImageNet         | 1      | 64                      | 1e-5 | 0.1    | 5         | 5e-4         |

Table 5: Details of each Figure and Table used in the paper.

| Figure/Table | Model   | Training Dataset | Additional Notes                                |
|--------------|---|------------------|---|
| Figure 1-a   | FARE <sup>2,4,6,8,10</sup> , LORE <sup>10</sup>             | ImageNet-100     | Initial performance drop                        |
| Figure 1-b   | FARE <sup>2</sup> , LORE <sup>2</sup>                       | ImageNet-100     | Robustness–accuracy Pareto front                |
| Figure 3     | LORE <sup>2</sup>   | ImageNet-100     | Controllability of LORE                         |
| Figure 4-a   | FARE <sup>2</sup> , LORE <sup>2</sup>                       | ImageNet         | OOD robustness                                  |
| Figure 4-b   | FARE <sup>1,2,4,6,8,10</sup> , LORE <sup>1,2,4,6,8,10</sup> | ImageNet         | Effect on image embedding interpretability      |
| Figure 5     | FARE <sup>1,2,4,6,8,10</sup> , LORE <sup>1,2,4,6,8,10</sup> | ImageNet         | Accuracy & robust accuracy across $\varepsilon$ |
| Figure 6     | LORE <sup>4</sup>   | ImageNet         | Fidelity analysis of LORE                       |
| Table 1      | FARE <sup>2,4</sup> , LORE <sup>2,4</sup>                   | ImageNet-100     | In-domain image classification                  |
| Table 2      | FARE <sup>4,8</sup> , LORE <sup>4,8</sup>                   | ImageNet-100     | In-domain image classification (DINOv2)         |
| Table 3      | FARE <sup>1,2,4</sup> , LORE <sup>1,2,4</sup>               | ImageNet         | Zero-shot image classification                  |

## D Constraint Satisfaction in LORE

To further understand the behavior of constraint enforcement under varying adversarial budgets, we visualize the distribution of distances between clean embeddings and their corresponding pre-trained references throughout training for  $\varepsilon = 1, 2$ , and 4 in Figure 7. As we can observe, larger perturbation strengths lead to greater deviation from the pre-trained reference in the early stages of training. This early-stage divergence results in a more pronounced initial drop in the model’s nominal performance. However, LORE is able to effectively regulate this deviation through its constraint-aware mechanism, gradually aligning the clean embeddings back within the  $\rho$  threshold. This demonstrates the robustness and practicality of LORE in preserving clean performance even under severe adversarial training regimes.

In contrast, due to the lack of such constraint regulation in FARE, the distance between clean embeddings and pre-trained references cannot be reliably controlled. As a result, FARE experiences a catastrophic initial drop in nominal accuracy, particularly under larger perturbation budgets. This failure to maintain embedding fidelity further underscores the importance of the dual network in LORE for stabilizing the training process and preserving clean accuracy.

To better illustrate this behavior, all subfigures in Figure 7 show the distance distributions beginning from the 20th training iteration onward. These comparisons clearly highlight the contrast between LORE’s effective enforcement of the proximity constraint and FARE’s limited capability to manage deviation across increasing adversarial strengths.

## E Generalization Gap in Adversarial Training

**Theorem E.1** (Generalization Gap in Adversarial Training). *It is well known that the generalization gap for a given loss function is upper bounded by complexity measures, giving rise to theoretical justifications of the bias-variance trade-off. Assuming bounded norm embeddings, i.e.,  $\|\phi(x)\|_2 \leq K$  for all  $x, \phi$ , we can see that the uniform loss bound  $B$  satisfies:*

$$B := \sup_{\phi, x} |\ell(\phi, x)| = \sup_{\phi, x} \max_{\delta} \|\phi(x + \delta) - \phi_0(x)\|_2 \leq \sup_{\phi, x} \max_{\delta} [\|\phi(x + \delta)\|_2 + \|\phi_0(x)\|_2] \leq 2K.$$

Therefore, with probability at least  $1 - 2\delta$ ,

$$\left| \mathbb{E}_x[\ell(\phi, x)] - \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \ell(\phi, x_i) \right| \leq 2\mathfrak{R}_n(\mathcal{L}_{\mathcal{H}}) + B \sqrt{\frac{\log(1/\delta)}{2|\mathcal{D}|}} \leq 2\mathfrak{R}_n(\mathcal{L}_{\mathcal{H}}) + K \sqrt{\frac{2\log(1/\delta)}{|\mathcal{D}|}},$$

where  $\mathcal{L}_{\mathcal{H}} = \{\ell(\phi, \cdot) \mid \phi \in \mathcal{H}\}$  is the loss class induced by hypothesis class  $\mathcal{H}$ ,  $\mathfrak{R}_n(\mathcal{L}_{\mathcal{H}})$  is the empirical Rademacher complexity of  $\mathcal{L}_{\mathcal{H}}$ , and  $\sup_{\phi, x} |\ell(\phi, x)| \leq B$ .

In adversarial training, the loss class  $\mathcal{L}_{\mathcal{H}}$  becomes extremely complex due to the inner  $\max_{\delta}$  operation, leading to large Rademacher complexity  $\mathfrak{R}_n(\mathcal{L}_{\mathcal{H}})$ . This explains why adversarial training requires significantly more samples for generalization compared to standard training.

When we restrict to simpler hypothesis classes  $\mathcal{H}_{\rho} \subset \mathcal{H}_{\text{org}}$  (through techniques like Lipschitz constraints or norm bounds), the Rademacher complexity decreases, potentially improving generalization.

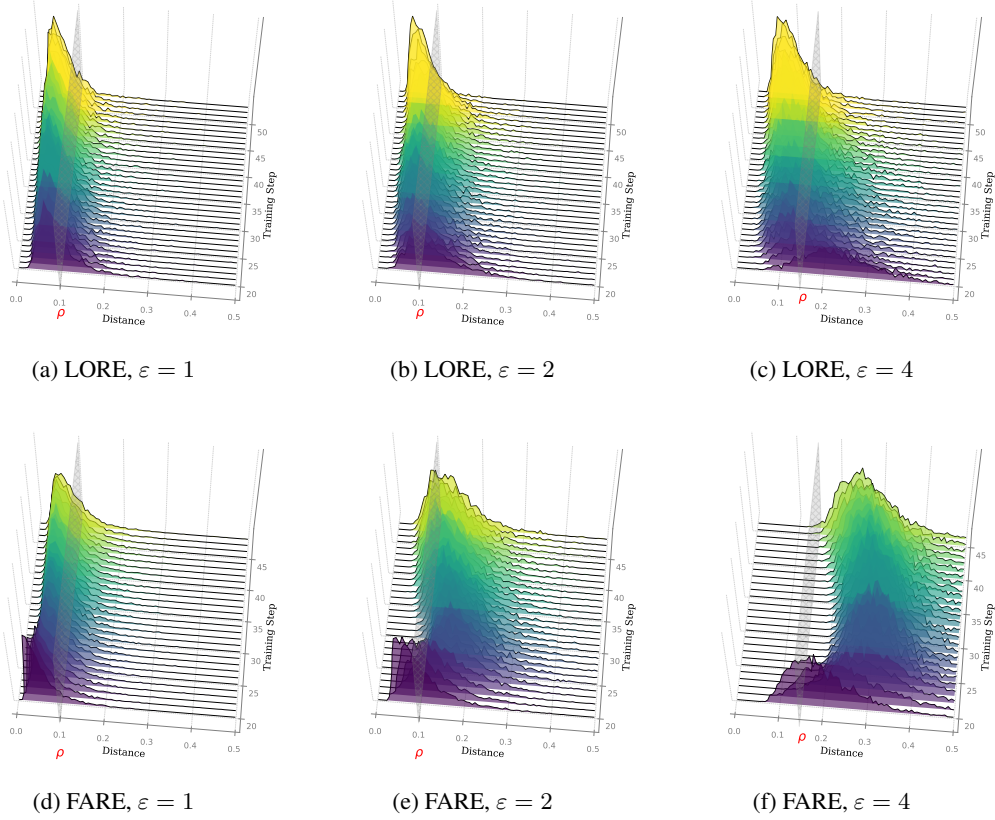


Figure 7: Constraint satisfaction comparison between LORE and FARE across adversarial training budgets  $\varepsilon = 1, 2, 4$ . **(Top):** LORE maintains strong proximity between clean embeddings and pre-trained references throughout training, with distances concentrating below the  $\rho$  threshold. **(Bottom):** FARE exhibits weaker fidelity preservation and fails to effectively regulate distance under increasing adversarial strength.

557 However, the bounds we derived are notoriously crude; they fail to capture important phenomena like  
558 double descent and often dramatically overestimate the actual generalization gap in practice.

## 559 F Deviation Between Cosine Similarities

**Assumption F.1.** For each input  $x$ , let

$$u = \phi_{\text{org}}(x), \quad \hat{u} = \phi_{\theta}(x),$$

and suppose

$$\|\hat{u} - u\|_2^2 \leq \rho \|u\|_2^2 \quad \text{for some } \rho \in [0, 1).$$

**Proposition F.2.** Under the above assumption, for any nonzero  $v \in \mathbb{R}^n$ ,

$$|S_C(u, v) - S_C(\hat{u}, v)| \leq 2\sqrt{\rho}.$$

We show that enforcing

$$\|\phi_{\theta}(x) - \phi_{\text{org}}(x)\|_2^2 \leq \rho \|\phi_{\text{org}}(x)\|_2^2$$

560 implies a uniform bound on the change in cosine similarity to any fixed vector  $v \in \mathbb{R}^n$ .

*Proof.* Write

$$S_C(u, v) = \frac{v^T u}{\|v\| \|u\|}, \quad S_C(\hat{u}, v) = \frac{v^T \hat{u}}{\|v\| \|\hat{u}\|},$$

so

$$|S_C(u, v) - S_C(\hat{u}, v)| = \left| \frac{v^T}{\|v\|} \left( \frac{u}{\|u\|} - \frac{\hat{u}}{\|\hat{u}\|} \right) \right| \leq \left\| \frac{u}{\|u\|} - \frac{\hat{u}}{\|\hat{u}\|} \right\|.$$

Now decompose

$$\frac{u}{\|u\|} - \frac{\hat{u}}{\|\hat{u}\|} = \left( \frac{u}{\|u\|} - \frac{\hat{u}}{\|u\|} \right) + \left( \frac{\hat{u}}{\|u\|} - \frac{\hat{u}}{\|\hat{u}\|} \right),$$

so by the triangle inequality,

$$\left\| \frac{u}{\|u\|} - \frac{\hat{u}}{\|\hat{u}\|} \right\| \leq \frac{\|u - \hat{u}\|}{\|u\|} + \|\hat{u}\| \left| \frac{1}{\|u\|} - \frac{1}{\|\hat{u}\|} \right|.$$

Since  $\|\hat{u}\| - \|u\| \leq \|u - \hat{u}\|$  and  $\|\hat{u}\| \leq \|u\| + \|u - \hat{u}\|$ , one shows easily

$$\|\hat{u}\| \left| \frac{1}{\|u\|} - \frac{1}{\|\hat{u}\|} \right| \leq \frac{\|u - \hat{u}\|}{\|u\|}.$$

Hence

$$|S_C(u, v) - S_C(\hat{u}, v)| \leq 2 \frac{\|u - \hat{u}\|}{\|u\|} \leq 2\sqrt{\rho},$$

as claimed.  $\square$

**Remark.** In vision-language models one may take  $v = \psi(t)$ , the text embedding of prompt  $t$ , so the same bound guarantees  $|S_C(\phi_{\text{org}}(x), \psi(t)) - S_C(\phi_{\theta}(x), \psi(t))| \leq 2\sqrt{\rho}$ .

## G Impact of Dual Network on Model Performance

### G.1 Comparison of Alternative Architectures for the Dual Function

In Figure 8, we compare the clean and robust accuracy achieved by two different architectures used for the dual function: a simple scalar value and a network-based (sample-dependent) function, as adopted in the current LORE setting. While both configurations perform comparably in terms of clean accuracy, the network-based dual function generalizes substantially better on adversarial examples, leading to consistently higher robust accuracy throughout training. This highlights the importance of a flexible, sample-adaptive mechanism in effectively enforcing robustness constraints during adversarial fine-tuning.

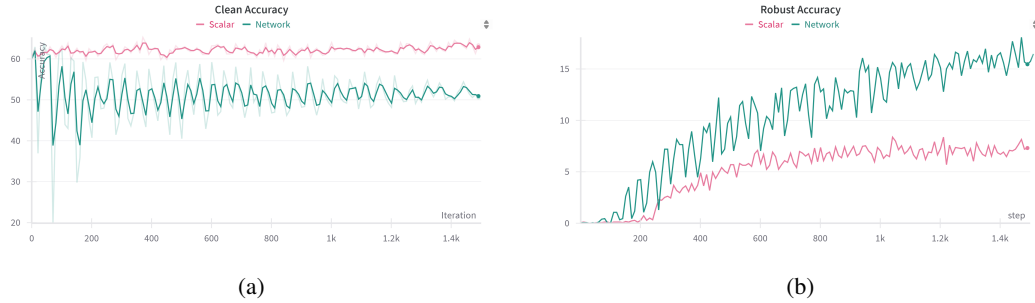


Figure 8: Comparison of clean and robust accuracy when using different architectures for the dual function in LORE. **(a):** Clean accuracy over training steps. **(b):** Robust accuracy over training steps. The *Network*-based dual function (sample-based) used in the current LORE setting leads to significantly higher robust accuracy compared to the *Scalar* baseline, while maintaining competitive clean accuracy.



## 573 G.2 Effect of the Dual Network $\lambda_\omega(x)$ on Clean Accuracy

574 Figure 9 illustrates the impact of the dual network  $\lambda_\omega(x)$  on model performance. As shown, at the  
 575 initial steps, higher values of  $\lambda_\omega(x)$  help maintain the model’s nominal performance, ensuring it  
 576 performs well on clean data. In contrast, for FARE, due to the absence of such a proximity constraint  
 577 during the early iterations, the model, lacking robustness, passes through suboptimal states, leading  
 578 to a significant drop in nominal performance.

579 To further support this observation, we present comprehensive results in Figure 10 and Figure 11,  
 580 showcasing the behavior of the dual network and its impact across different architectures. In Figure 10,  
 581 experiments on DINOv2 models (base and small) demonstrate that LORE consistently achieves  
 582 higher clean accuracy compared to FARE, especially in the early stages of training, while adaptively  
 583 modulating  $\lambda_\omega(x)$  to control constraint satisfaction. Similarly, Figure 11 reports the performance  
 584 of ViT-B/16 and ConvNeXt-B models, confirming the effectiveness and generality of the dual  
 585 network across various architectures and perturbation strengths. These results highlight that LORE’s  
 586 constraint-aware mechanism is stable, avoiding the sharp degradation commonly observed in FARE  
 587 adversarial fine-tuning.

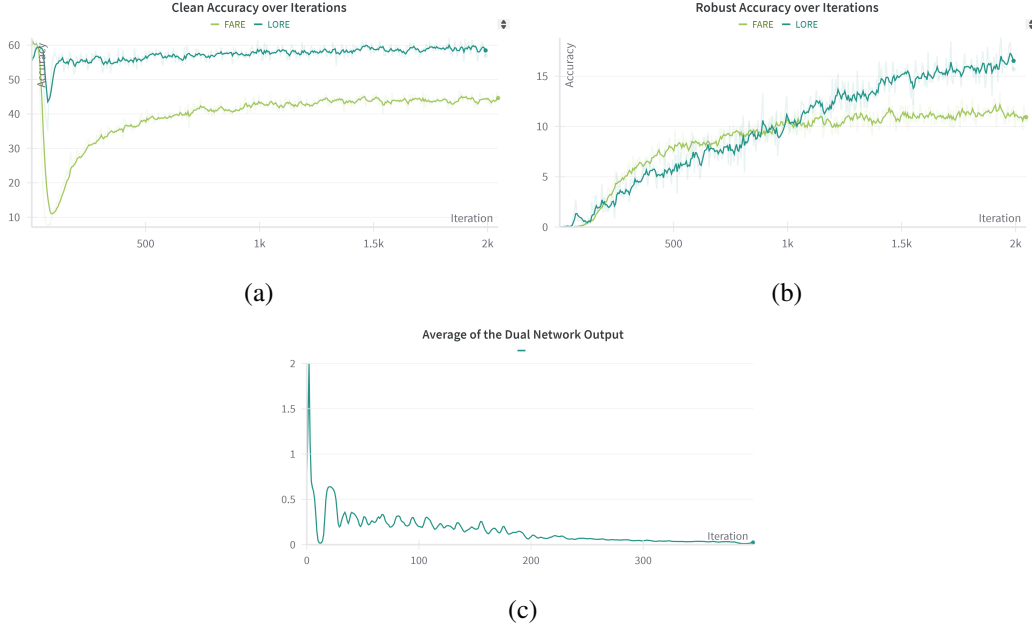


Figure 9: Comparison of the performance of two methods and the output of the dual network. (a) Clean accuracies over iterations, (b) Robust accuracies over iterations, (c) Average output of the dual network  $\lambda_\omega(x)$ .

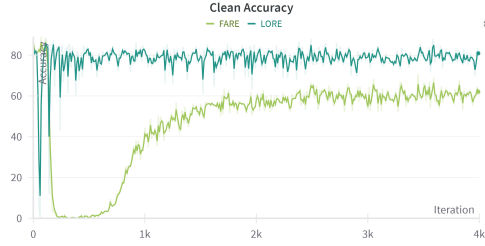
## 588 H Revisiting Embedding Models

589 **CLIP** [Radford et al., 2021]. A major part of our experiments builds upon CLIP, which consists  
 590 of an image encoder  $\phi$  and a text encoder  $\psi$  that map images and text descriptions into a shared  
 591 embedding space. For zero-shot classification, textual descriptions are typically formatted as "This  
 592 is a photo of a [CLS]", where [CLS] represents class labels. The probability of assigning an  
 593 image  $x$  to a class  $\hat{y}$  is computed via a softmax over cosine similarities:

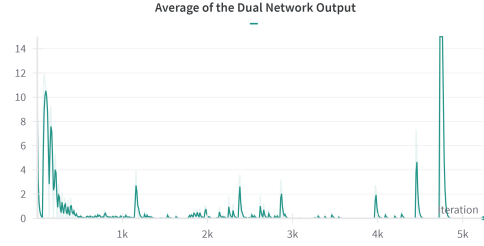
$$p(\hat{y} | x) = \frac{\exp(\cos(\psi(t_{\hat{y}}), \phi(x))/\tau)}{\sum_{j=1}^K \exp(\cos(\psi(t_j), \phi(x))/\tau)}. \quad (23)$$

594 where  $\tau$  is a temperature parameter, and  $K$  denotes the number of classes.

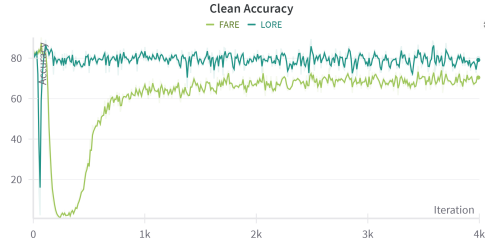
595 **DINOv2** [Oquab et al., 2024]. In addition to CLIP, we incorporate DINOv2, a powerful self-  
 596 supervised visual transformer-based encoder, into our exploration of embedding models. While



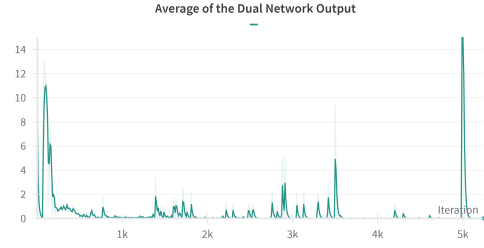
(a) DINOv2-base, Clean accuar cay ( $\varepsilon = 16$ )



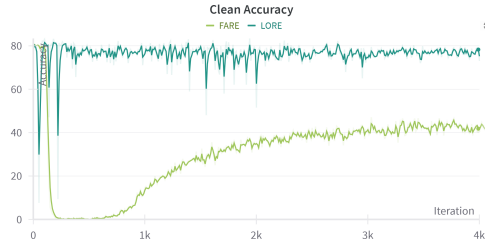
(b) DINOv2-base, Dual Network Output ( $\varepsilon = 16$ )



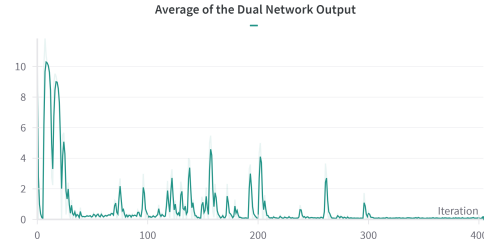
(c) DINOv2-base, Clean accuar cay ( $\varepsilon = 8$ )



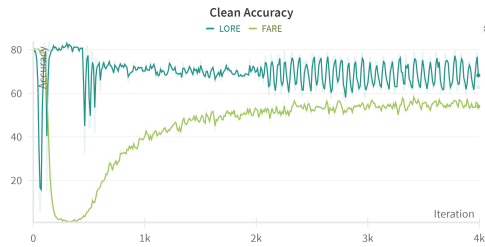
(d) DINOv2-base, Dual Network Output ( $\varepsilon = 8$ )



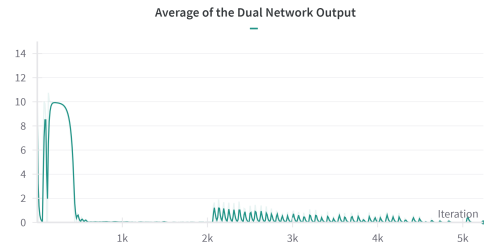
(e) DINOv2-small, Clean accuar cay ( $\varepsilon = 16$ )



(f) DINOv2-small, Dual Network Output ( $\varepsilon = 16$ )



(g) DINOv2-small, Clean accuar cay ( $\varepsilon = 8$ )



(h) DINOv2-small, Dual Network Output ( $\varepsilon = 8$ )

Figure 10: Comparison of LORE and FARE on DINOv2-base and DINOv2-small models under different adversarial budgets  $\varepsilon \in \{8, 16\}$ . **Left:** Clean accuracy over training iterations, illustrating LORE’s superior stability and performance. **Right:** Average output of the dual network  $\lambda_\omega(x)$  across iterations, highlighting how LORE dynamically adjusts its constraint enforcement.

CLIP provides a joint image-text embedding space, DINOv2 focuses solely on visual representation learning. This complementary perspective allows us to compare and leverage both multimodal and unimodal embedding paradigms.

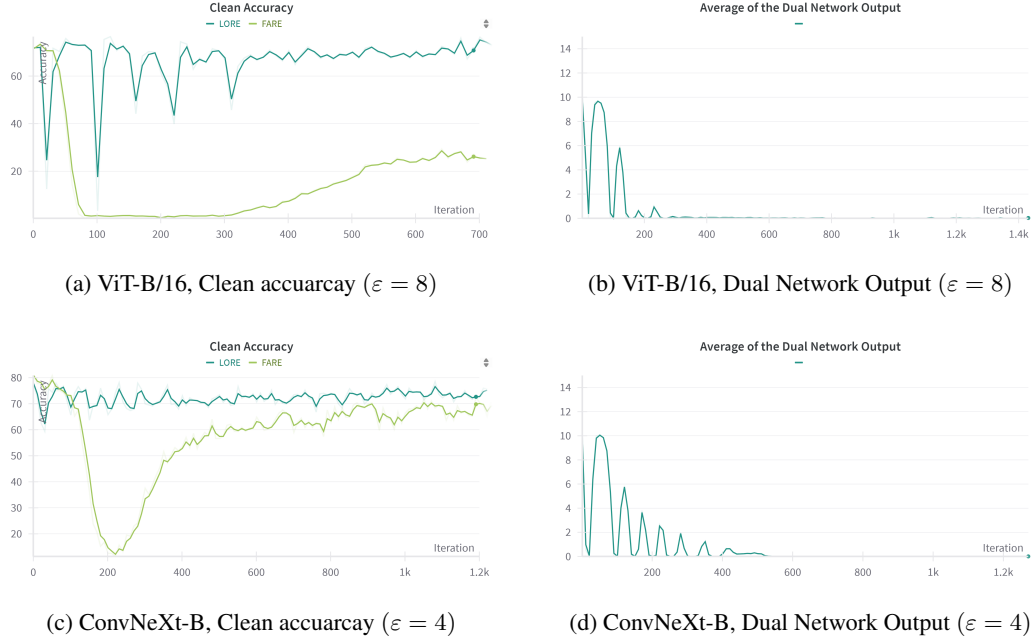


Figure 11: Comparison of LORE and FARE on ViT-B/16 and ConvNeXt-B models under adversarial fine-tuning. **Left:** Clean accuracy over training iterations ( $\varepsilon = 8$  for ViT-B/16 and  $\varepsilon = 4$  for ConvNeXt-B), showing LORE’s improved stability and performance. **Right:** Average output of the dual network  $\lambda_\omega(x)$ , indicating LORE’s dynamic constraint adjustment during training.

DINOv2 learns visual features by minimizing a cross-view prediction loss between student and teacher networks. Given  $N$  image views, the loss is computed as:

$$\mathcal{L}_{\text{DINO}} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C q_{ic} \log p_{ic}, \quad (24)$$

where  $q_{ic}$  and  $p_{ic}$  are the teacher and student probabilities for class  $c$  and view  $i$ , respectively, and  $C$  is the number of output dimensions.

Our broader work centers around embedding models, with a primary emphasis on CLIP, while also investigating the capabilities and representations of models like DINOv2. In general, we found that the DINOv2 model has much richer image embeddings than CLIP and can achieve much higher robustness over the same perturbation and dataset.

## I Impact of $K$ on Model Performance

LORE alternates between  $K$  steps of updating the primal encoder and one step of updating the dual network. In this section, we study the effect of the hyperparameter  $K$  on final performance. As shown in Table 6, increasing  $K$  leads to improved clean accuracy (Acc) and robust accuracy (RAcc), particularly when moving from  $K = 1$  to  $K = 3$  or 5. This demonstrates that more frequent primal updates between dual updates help stabilize training and improve performance. Based on this observation, we choose  $K = 5$  as the default in our final LORE implementation. For further discussion on how  $K$  impacts computational cost and training time, see Appendix K.

## J Additional Experimental Results

In this section, we present additional experiments to further validate the robustness and generalization capabilities of our proposed method. These evaluations span multiple settings, including black-box adversarial attacks (e.g., Square Attack), Gaussian noise corruption, in-domain and zero-shot classification, and out-of-distribution (OOD) robustness. By comparing against the FARE baseline

Table 6: Effect of the  $K$  hyperparameter on model performance. Clean accuracy (Acc) and robust accuracy (RAcc) (%) are reported for various values of  $K$ . Results are based on ViT-B/32 trained with LORE<sup>2</sup> evaluated on ImageNet-100 under  $\varepsilon = 2/255$  APGD attack.

| <b>K</b>        | 1     | 2     | 3     | 5     | 7     | 10    |
|-----------------|-------|-------|-------|-------|-------|-------|
| <b>Acc (%)</b>  | 64.11 | 66.43 | 64.58 | 60.19 | 59.96 | 56.46 |
| <b>RAcc (%)</b> | 9.54  | 13.78 | 19.48 | 27.01 | 31.81 | 39.71 |

Table 7: Evaluation on Square Attack, a Black-Box attacks, averaged over the previous mentioned 13 zero-shot datasets

| Method            | Backbone | Clean       | $\varepsilon = 1$ | $\varepsilon = 2$ | $\varepsilon = 4$ | $\varepsilon = 6$ |
|-------------------|----------|-------------|-------------------|-------------------|-------------------|-------------------|
| FARE <sup>4</sup> | ViT-B/32 | 42.6        | 40.0              | 36.4              | 30.0              | 23.6              |
| LORE <sup>4</sup> | ViT-B/32 | <b>50.1</b> | <b>43.9</b>       | <b>40.3</b>       | <b>33.5</b>       | <b>27.1</b>       |

across diverse conditions and datasets, we demonstrate that LORE consistently achieves superior performance, particularly under challenging threat models and distributional shifts.

### J.1 Square Attack Evaluation

In this section, we evaluate the robustness of our fine-tuning approach against a black-box adversarial attack known as the Square Attack [Andriushchenko et al., 2020]. Unlike gradient-based methods, Square Attack operates without access to model gradients and perturbs the input using a query-efficient, score-based strategy. This makes it a strong candidate for evaluating real-world robustness where white-box access is not feasible. We conduct experiments on LORE<sup>4</sup>, which is adversarially fine-tuned on ImageNet, to assess how well the model generalizes to such black-box settings. The results, summarized in Table 7, show that our method consistently outperforms the baseline under this challenging threat model.

### J.2 Evaluation Under Gaussian Noise Corruption

To further assess the robustness of our method, we evaluate the performance of LORE<sup>4</sup> and FARE<sup>4</sup> under varying levels of Gaussian noise corruption. We use the ViT-B/32 CLIP model, with both methods fine-tuned on ImageNet. As shown in Figure 13, LORE<sup>4</sup> consistently maintains higher accuracy than FARE<sup>4</sup> across a wide range of noise strengths ( $\sigma$ ), especially in low to moderate noise settings. The right subplot illustrates the accuracy gap, highlighting LORE<sup>4</sup>'s robustness advantage up to  $\sigma = 40$ , beyond which the performance of both models converges as the corruption becomes extreme. This evaluation further supports the generalization capabilities of our method in the presence of unseen corruptions. Figure 12 provides a visual illustration of how a single image degrades under increasing levels of Gaussian noise.



Figure 12: Visualization of a single image under increasing levels of Gaussian noise ( $\sigma = 0, 32, 64, 128$ ). This figure helps set reasonable expectations for model performance by illustrating how perceptual degradation progresses with noise intensity.

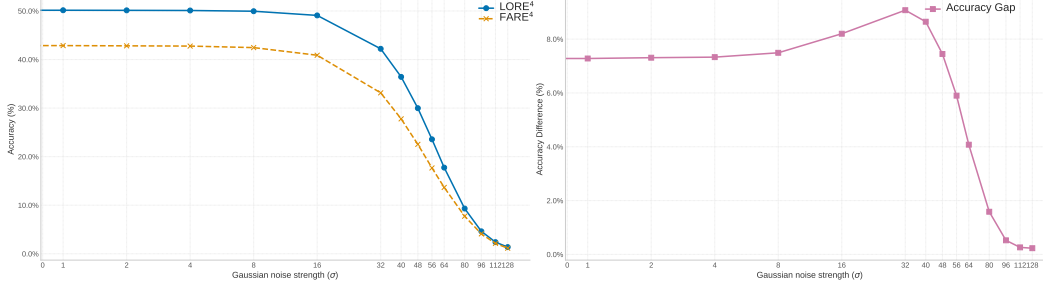


Figure 13: Robustness evaluation under Gaussian noise corruption. **Left:** Classification accuracy of LORE<sup>4</sup> and FARE<sup>4</sup> under increasing Gaussian noise strength ( $\sigma$ ). LORE<sup>4</sup> consistently outperforms FARE<sup>4</sup>, particularly in moderate noise regimes. **Right:** Accuracy gap between LORE<sup>4</sup> and FARE<sup>4</sup>, showing a stable and significant margin up to  $\sigma = 40$ , after which the gap decreases as both models degrade under extreme noise conditions.

Table 8: Clean and adversarial accuracy for in-domain image classification on ImageNet-100 across different CLIP vision encoders, evaluated using the APGD attack.

| Method            | Backbone       | Clean       | $\varepsilon = 1$ | $\varepsilon = 2$ | $\varepsilon = 4$ | $\varepsilon = 8$ |
|-------------------|----------------|-------------|-------------------|-------------------|-------------------|-------------------|
| FARE <sup>8</sup> | ViT-B/16       | 26.5        | 20.4              | 17.0              | 10.3              | 2.3               |
| LORE <sup>8</sup> | ViT-B/16       | <b>70.5</b> | <b>53.6</b>       | <b>48.5</b>       | <b>37.8</b>       | <b>17.8</b>       |
| FARE <sup>8</sup> | ViT-B/32 LAION | 17.0        | 11.3              | 7.3               | 3.16              | 0.40              |
| LORE <sup>8</sup> | ViT-B/32 LAION | <b>28.2</b> | <b>12.1</b>       | <b>10.0</b>       | <b>6.54</b>       | <b>3.51</b>       |
| FARE <sup>8</sup> | ConvNeXt-B     | 61.6        | 55.3              | 48.5              | 35.7              | 43.4              |
| LORE <sup>8</sup> | ConvNeXt-B     | <b>72.2</b> | <b>56.2</b>       | <b>49.1</b>       | <b>38.3</b>       | <b>47.2</b>       |

### J.3 In-domain Image Classification

Table 8 presents a comparison of clean and adversarial accuracy across various CLIP-based vision backbones, all trained with  $\varepsilon = 8$ , on the ImageNet-100 dataset under the APGD attack.

### J.4 Zero-shot Image Classification

Table 9 presents the results of different settings for zero-shot image classification using the ViT-B/32 CLIP model, highlighting the superiority of our method over the FARE baseline. Additionally, for a more challenging scenario, Table 10 reports model performance under high-intensity adversarial attacks, further demonstrating the resilience of our approach. These tables serve as the complete version of the results summarized in the main paper.

### J.5 Out-of-Distribution Robustness

As shown in Figure 14, increasing the training perturbation strength leads to greater degradation in out-of-distribution (OOD) robustness across common corruptions in ImageNet-C. Despite this trend, models trained with LORE consistently exhibit better robustness compared to those trained with the FARE method, highlighting LORE’s superior generalization under distributional shifts.

## K Computation and Efficiency Analysis

In this section, we analyze the computational aspects of LORE in terms of training time, efficiency, and convergence behavior. While LORE introduces an additional dual network and constraint enforcement mechanism, we find that its cost remains practical and comparable to FARE baselines.

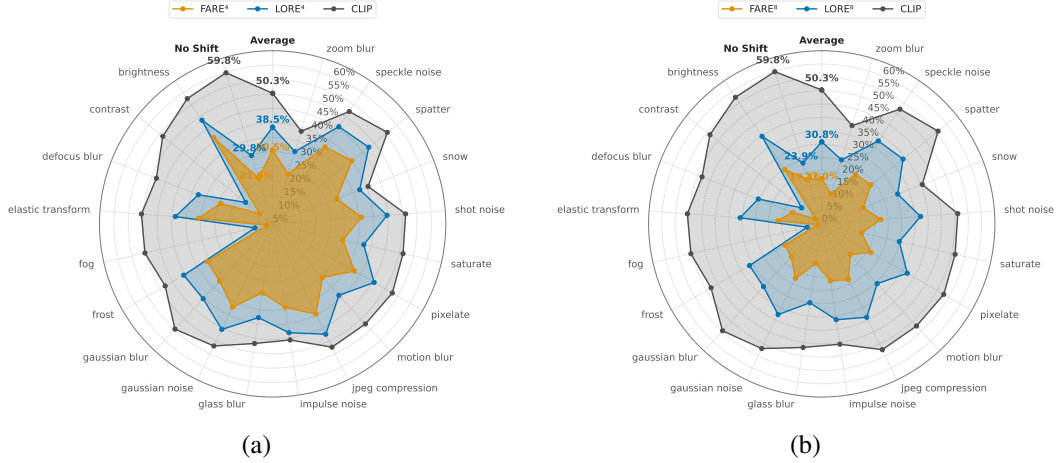


Figure 14: Robustness to common corruptions on ImageNet-C as an OOD evaluation for models trained with (a)  $\varepsilon = 4$ , and (b)  $\varepsilon = 8$ . As we can observe, as the training perturbation strength increases, the degradation in OOD robustness also increases. Nevertheless, LORE consistently shows lower degradation compared to models trained with the FARE method.

### 660 K.1 Convergence Efficiency.

661 To compare the training efficiency of FARE and LORE, we measure the total training time (in  
662 minutes) required to reach specific robust accuracy (RAcc) thresholds on the validation set. Table 11  
663 reports this comparison across various model backbones, including ViT-B/16, ViT-B/32, ConvNeXt-  
664 B, ViT-S/14, and ViT-B/14. LORE often reaches target RAcc levels in fewer training minutes than  
665 FARE, highlighting its superior optimization efficiency and stability.

### 666 K.2 Impact of $\lambda_\omega$ architecture on Training Time.

667 The results in Table 12 show that LORE’s current dual network design is not only significantly more  
668 efficient than using a separate pretrained CLIP model as a dual network, but also achieves comparable  
669 runtime to simpler parameterizations (scalar or linear forms). This demonstrates that LORE achieves  
670 computational efficiency without sacrificing expressive capacity.

### 671 K.3 Impact of $K$ on Training Time.

672 As described in Section 5, LORE alternates between  $K$  primal updates and one dual update per batch.  
673 While Section I analyzes the impact of  $K$  on final performance, here in Fig. 15, we empirically  
674 examine how varying  $K$  affects training time.

675

Table 9: A comprehensive evaluation of clean and adversarial performance is conducted across various image classification datasets using the ViT-B/32 CLIP model. All models are trained on ImageNet and evaluated in a zero-shot setting across diverse benchmarks. Table demonstrates the increase ( $\uparrow$ ) in performance of our method relative to the corresponding FARE models.

| Eval.              | Vision encoder    | Zero-shot datasets |         |      |         |          |      |         |      |         |            |            |      |            |                      | Average Zero-shot    |
|--------------------|-------------------|--------------------|---------|------|---------|----------|------|---------|------|---------|------------|------------|------|------------|----------------------|----------------------|
|                    |                   | ImageNet           | CalTech | Cars | CIFAR10 | CIFAR100 | DTD  | EuroSAT | FGVC | Flowers | ImageNet-R | ImageNet-S | PCAM | OxfordPets | STL-10               |                      |
| clean              | CLIP              | 59.8               | 84.1    | 59.6 | 89.7    | 63.3     | 44.4 | 46.1    | 19.6 | 66.3    | 69.3       | 42.3       | 62.3 | 87.5       | 97.2                 | 64.0                 |
|                    | FARE <sup>1</sup> | 56.6               | 84.0    | 56.3 | 86.4    | 61.1     | 40.5 | 27.2    | 18.1 | 62.0    | 66.4       | 40.5       | 55.5 | 86.1       | 95.8                 | 60.0                 |
|                    | LORE <sup>1</sup> | 57.4               | 84.4    | 55.9 | 88.5    | 64.5     | 40.1 | 29.9    | 16.7 | 61.3    | 67.2       | 41.5       | 53.8 | 86.9       | 96.3                 | 60.5 $\uparrow 0.5$  |
|                    | FARE <sup>2</sup> | 52.9               | 82.2    | 49.7 | 76.3    | 51.1     | 36.4 | 18.4    | 15.7 | 53.3    | 60.4       | 35.9       | 48.2 | 82.7       | 93.0                 | 54.1                 |
|                    | LORE <sup>2</sup> | 55.7               | 83.0    | 51.0 | 83.4    | 59.7     | 37.2 | 23.0    | 15.9 | 54.5    | 63.4       | 39.3       | 51.2 | 84.3       | 94.5                 | 57.0 $\uparrow 2.9$  |
|                    | FARE <sup>4</sup> | 42.6               | 78.1    | 36.5 | 55.9    | 35.8     | 28.8 | 15.7    | 10.6 | 36.1    | 49.3       | 27.1       | 50.0 | 71.8       | 85.6                 | 44.7                 |
|                    | LORE <sup>4</sup> | 50.1               | 80.3    | 40.1 | 72.4    | 49.6     | 32.4 | 17.7    | 11.4 | 39.7    | 55.1       | 33.6       | 50.0 | 79.3       | 90.4                 | 50.2 $\uparrow 5.5$  |
|                    | FARE <sup>6</sup> | 33.0               | 73.0    | 24.7 | 40.0    | 24.9     | 23.7 | 15.2    | 6.24 | 22.7    | 39.5       | 20.2       | 50.0 | 52.4       | 75.0                 | 36.0                 |
|                    | LORE <sup>6</sup> | 42.6               | 75.3    | 28.7 | 61.5    | 35.8     | 26.1 | 16.4    | 8.25 | 25.8    | 45.2       | 26.2       | 50.0 | 69.3       | 84.2                 | 42.5 $\uparrow 6.5$  |
|                    | FARE <sup>8</sup> | 27.6               | 69.1    | 17.0 | 34.2    | 20.2     | 20.6 | 15.0    | 4.68 | 16.5    | 34.1       | 16.8       | 50.0 | 37.6       | 67.3                 | 31.0                 |
|                    | LORE <sup>8</sup> | 41.3               | 74.6    | 24.9 | 61.5    | 35.9     | 24.5 | 16.0    | 7.14 | 22.8    | 43.0       | 24.4       | 50.0 | 67.4       | 83.5                 | 41.2 $\uparrow 10.2$ |
| FARE <sup>10</sup> | 23.2              | 66.0               | 12.8    | 31.3 | 17.5    | 18.8     | 15.0 | 4.11    | 13.7 | 30.2    | 14.4       | 50.0       | 27.6 | 61.9       | 28.0                 |                      |
| LORE <sup>10</sup> | 40.5              | 74.3               | 23.8    | 64.8 | 38.8    | 24.3     | 16.2 | 6.75    | 22.0 | 41.9    | 22.9       | 50.0       | 66.4 | 84.2       | 41.2 $\uparrow 13.2$ |                      |
| $\epsilon = 1.0$   | CLIP              | 0.0                | 0.0     | 0.0  | 0.0     | 0.0      | 0.0  | 0.0     | 0.0  | 0.0     | 0.0        | 0.1        | 0.0  | 0.0        | 0.0                  | 0.0                  |
|                    | FARE <sup>1</sup> | 27.8               | 68.6    | 16.1 | 61.0    | 35.6     | 22.5 | 6.1     | 2.9  | 30.6    | 34.4       | 22.5       | 24.7 | 55.8       | 82.2                 | 35.6                 |
|                    | LORE <sup>1</sup> | 32.9               | 71.0    | 18.7 | 67.1    | 40.0     | 23.7 | 9.4     | 4.2  | 33.5    | 37.6       | 24.8       | 28.3 | 60.5       | 84.1                 | 38.7 $\uparrow 3.1$  |
|                    | FARE <sup>2</sup> | 34.3               | 75.2    | 22.6 | 60.1    | 35.4     | 24.7 | 12.6    | 5.3  | 33.9    | 39.7       | 24.1       | 30.4 | 64.8       | 83.3                 | 39.4                 |
|                    | LORE <sup>2</sup> | 39.3               | 76.3    | 23.3 | 67.0    | 43.2     | 26.4 | 12.3    | 6.5  | 35.8    | 42.4       | 26.4       | 39.0 | 68.5       | 85.6                 | 42.5 $\uparrow 3.1$  |
|                    | FARE <sup>4</sup> | 33.2               | 74.8    | 21.4 | 44.9    | 28.0     | 22.4 | 14.0    | 5.8  | 27.3    | 37.1       | 21.3       | 50.2 | 59.3       | 77.7                 | 37.2                 |
|                    | LORE <sup>4</sup> | 41.8               | 77.2    | 24.1 | 61.2    | 39.9     | 24.5 | 14.3    | 7.8  | 30.2    | 41.6       | 25.5       | 50.2 | 68.8       | 83.2                 | 42.2 $\uparrow 5.0$  |
|                    | FARE <sup>6</sup> | 26.3               | 70.7    | 15.2 | 32.8    | 20.0     | 19.5 | 14.1    | 3.6  | 19.3    | 30.0       | 15.1       | 50.2 | 43.5       | 70.2                 | 31.1                 |
|                    | LORE <sup>6</sup> | 36.2               | 74.4    | 18.9 | 52.2    | 30.6     | 20.8 | 15.4    | 6.3  | 22.3    | 34.2       | 22.1       | 50.2 | 60.5       | 78.1                 | 37.4 $\uparrow 6.3$  |
|                    | FARE <sup>8</sup> | 23.1               | 66.9    | 10.8 | 28.2    | 16.7     | 17.7 | 14.6    | 3.1  | 15.5    | 25.3       | 12.5       | 50.2 | 30.9       | 62.7                 | 27.3                 |
|                    | LORE <sup>8</sup> | 35.5               | 73.6    | 15.9 | 51.1    | 31.2     | 20.1 | 14.2    | 5.8  | 20.0    | 33.3       | 20.7       | 50.2 | 58.6       | 77.4                 | 36.3 $\uparrow 9.0$  |
| FARE <sup>10</sup> | 19.0              | 65.4               | 8.20    | 26.3 | 14.3    | 16.7     | 14.8 | 3.0     | 13.1 | 22.0    | 11.0       | 50.2       | 23.7 | 57.2       | 25.1                 |                      |
| LORE <sup>10</sup> | 33.8              | 71.2               | 14.3    | 51.7 | 30.1    | 19.5     | 12.9 | 4.2     | 18.9 | 31.5    | 19.4       | 50.2       | 53.9 | 76.7       | 35.0 $\uparrow 9.9$  |                      |
| $\epsilon = 2.0$   | CLIP              | 0.0                | 0.0     | 0.0  | 0.0     | 0.0      | 0.0  | 0.0     | 0.0  | 0.0     | 0.0        | 0.1        | 0.0  | 0.0        | 0.0                  | 0.0                  |
|                    | FARE <sup>1</sup> | 8.0                | 43.5    | 1.9  | 31.0    | 14.7     | 12.9 | 0.6     | 0.2  | 6.8     | 13.4       | 11.7       | 14.1 | 15.9       | 54.9                 | 17.0                 |
|                    | LORE <sup>1</sup> | 13.1               | 49.0    | 3.3  | 37.9    | 19.0     | 14.2 | 2.5     | 0.5  | 10.1    | 17.6       | 13.1       | 19.1 | 23.1       | 61.2                 | 20.8 $\uparrow 3.8$  |
|                    | FARE <sup>2</sup> | 19.3               | 59.9    | 7.7  | 41.2    | 22.8     | 17.8 | 9.6     | 1.5  | 16.4    | 24.2       | 15.9       | 23.4 | 38.6       | 68.6                 | 26.7                 |
|                    | LORE <sup>2</sup> | 24.0               | 63.3    | 8.6  | 47.2    | 27.2     | 18.2 | 10.6    | 1.7  | 18.5    | 26.0       | 18.4       | 28.0 | 44.4       | 73.1                 | 29.6 $\uparrow 2.9$  |
|                    | FARE <sup>4</sup> | 24.1               | 65.5    | 10.4 | 36.0    | 21.6     | 18.8 | 12.3    | 2.7  | 17.9    | 27.7       | 15.8       | 50.0 | 44.4       | 68.8                 | 30.1                 |
|                    | LORE <sup>4</sup> | 32.6               | 69.5    | 12.4 | 50.8    | 29.6     | 20.9 | 13.0    | 3.3  | 21.6    | 32.3       | 20.0       | 50.1 | 55.9       | 76.1                 | 35.0 $\uparrow 4.9$  |
|                    | FARE <sup>6</sup> | 20.2               | 64.6    | 8.4  | 27.4    | 16.7     | 17.1 | 13.0    | 1.8  | 14.3    | 23.6       | 11.8       | 50.2 | 33.3       | 62.1                 | 26.5                 |
|                    | LORE <sup>6</sup> | 30.1               | 68.3    | 10.7 | 44.0    | 25.6     | 18.5 | 13.8    | 3.7  | 16.8    | 27.6       | 17.0       | 50.2 | 49.7       | 71.2                 | 32.1 $\uparrow 5.6$  |
|                    | FARE <sup>8</sup> | 17.4               | 62.7    | 6.4  | 24.5    | 13.7     | 15.4 | 13.1    | 1.5  | 11.2    | 19.7       | 10.7       | 50.2 | 24.0       | 56.2                 | 23.8                 |
|                    | LORE <sup>8</sup> | 30.9               | 68.8    | 10.5 | 43.3    | 25.7     | 18.5 | 13.5    | 3.2  | 16.6    | 27.8       | 17.0       | 50.2 | 49.2       | 71.6                 | 31.9 $\uparrow 8.1$  |
| FARE <sup>10</sup> | 15.1              | 60.0               | 5.0     | 23.5 | 11.8    | 14.3     | 13.5 | 1.7     | 10.6 | 18.5    | 9.2        | 50.2       | 18.4 | 52.5       | 22.2                 |                      |
| LORE <sup>10</sup> | 29.7              | 66.8               | 8.8     | 38.8 | 24.1    | 17.9     | 12.4 | 2.5     | 14.9 | 27.0    | 16.2       | 50.2       | 45.5 | 69.1       | 30.3 $\uparrow 8.1$  |                      |
| $\epsilon = 4.0$   | CLIP              | 0.0                | 0.0     | 0.0  | 0.0     | 0.0      | 0.0  | 0.0     | 0.0  | 0.0     | 0.0        | 0.1        | 0.0  | 0.0        | 0.0                  | 0.0                  |
|                    | FARE <sup>1</sup> | 0.3                | 6.3     | 0.0  | 1.7     | 2.0      | 2.3  | 0.0     | 0.0  | 0.1     | 2.6        | 2.4        | 0.9  | 0.0        | 5.3                  | 1.8                  |
|                    | LORE <sup>1</sup> | 0.7                | 9.7     | 0.0  | 3.5     | 3.1      | 4.0  | 0.0     | 0.0  | 0.2     | 3.8        | 2.8        | 2.7  | 0.0        | 9.4                  | 3.0 $\uparrow 1.2$   |
|                    | FARE <sup>2</sup> | 3.2                | 27.5    | 0.5  | 12.3    | 7.0      | 7.7  | 4.3     | 0.0  | 2.4     | 6.8        | 5.1        | 15.8 | 3.0        | 30.1                 | 9.4                  |
|                    | LORE <sup>2</sup> | 5.7                | 31.1    | 0.7  | 13.0    | 8.2      | 9.7  | 0.8     | 0.0  | 3.1     | 8.3        | 6.5        | 18.2 | 7.2        | 33.5                 | 10.8 $\uparrow 1.4$  |
|                    | FARE <sup>4</sup> | 10.7               | 46.3    | 1.5  | 19.7    | 11.8     | 11.9 | 10.2    | 0.6  | 6.4     | 11.4       | 8.7        | 45.2 | 16.2       | 46.1                 | 18.2                 |
|                    | LORE <sup>4</sup> | 17.8               | 54.2    | 2.8  | 27.4    | 16.8     | 14.4 | 10.0    | 0.6  | 8.0     | 16.4       | 11.7       | 48.4 | 25.5       | 56.1                 | 22.5 $\uparrow 4.3$  |
|                    | FARE <sup>6</sup> | 11.6               | 50.5    | 1.6  | 19.2    | 9.8      | 12.1 | 11.1    | 0.6  | 6.3     | 12.7       | 7.4        | 50.2 | 15.8       | 46.0                 | 18.7                 |
|                    | LORE <sup>6</sup> | 19.2               | 57.0    | 3.5  | 26.2    | 16.4     | 13.9 | 12.7    | 1.0  | 8.9     | 16.9       | 10.5       | 50.2 | 26.9       | 57.0                 | 23.2 $\uparrow 4.5$  |
|                    | FARE <sup>8</sup> | 10.9               | 50.0    | 1.5  | 18.3    | 9.2      | 11.4 | 11.8    | 0.7  | 6.3     | 11.9       | 6.5        | 50.2 | 12.4       | 44.3                 | 18.0                 |
|                    | LORE <sup>8</sup> | 21.7               | 58.8    | 4.1  | 28.0    | 17.2     | 13.8 | 12.8    | 1.1  | 9.5     | 17.7       | 10.9       | 50.2 | 31.5       | 59.0                 | 24.2 $\uparrow 6.2$  |
| FARE <sup>10</sup> | 9.03              | 48.3               | 1.1     | 17.7 | 8.3     | 10.4     | 11.5 | 0.4     | 5.5  | 11.1    | 5.4        | 50.2       | 10.6 | 41.9       | 17.1                 |                      |
| LORE <sup>10</sup> | 21.1              | 56.8               | 3.5     | 22.1 | 14.8    | 13.7     | 11.8 | 0.9     | 9.3  | 17.4    | 10.6       | 50.2       | 28.8 | 52.9       | 22.5 $\uparrow 5.4$  |                      |



Table 10: **Evaluation under high-intensity adversarial attacks.** A comprehensive assessment of clean and adversarial performance is conducted across various image classification datasets using the ViT-B/32 CLIP model. All models are trained on ImageNet and evaluated in a zero-shot setting across diverse benchmarks.

| Eval.             | Vision encoder     | ImageNet | Zero-shot datasets |      |         |          |      |         |      |         |            |            |      |            |        | Average Zero-shot      |
|-------------------|--------------------|----------|--------------------|------|---------|----------|------|---------|------|---------|------------|------------|------|------------|--------|------------------------|
|                   |                    |          | CalTech            | Cars | CIFAR10 | CIFAR100 | DTD  | EuroSAT | FGVC | Flowers | ImageNet-R | ImageNet-S | PCAM | OxfordPets | STL-10 |                        |
| $\epsilon = 6.0$  | FARE <sup>6</sup>  | 5.5      | 28.5               | 0.1  | 11.4    | 6.2      | 7.7  | 10.4    | 0.0  | 2.7     | 5.8        | 3.5        | 50.2 | 3.6        | 30.4   | 12.3                   |
|                   | LORE <sup>6</sup>  | 10.4     | 40.1               | 0.6  | 13.7    | 9.6      | 9.9  | 11.9    | 0.1  | 4.5     | 9.4        | 6.6        | 50.2 | 11.4       | 40.1   | 16.0 <span>↑3.7</span> |
|                   | FARE <sup>8</sup>  | 5.5      | 30.6               | 0.0  | 12.5    | 6.1      | 7.8  | 11.4    | 0.0  | 3.1     | 6.2        | 3.7        | 50.2 | 3.9        | 29.6   | 12.7                   |
|                   | LORE <sup>8</sup>  | 12.9     | 44.6               | 1.4  | 15.0    | 10.9     | 10.4 | 12.2    | 0.6  | 5.3     | 11.0       | 7.4        | 50.2 | 15.3       | 43.2   | 17.5 <span>↑4.8</span> |
|                   | FARE <sup>10</sup> | 5.3      | 31.4               | 0.0  | 13.7    | 5.4      | 7.3  | 11.8    | 0.0  | 3.2     | 6.1        | 3.6        | 50.2 | 4.2        | 28.5   | 12.7                   |
|                   | LORE <sup>10</sup> | 13.6     | 45.0               | 1.5  | 10.7    | 9.6      | 10.1 | 11.3    | 0.6  | 5.4     | 11.3       | 7.1        | 50.2 | 14.2       | 35.3   | 16.3 <span>↑3.6</span> |
| $\epsilon = 8.0$  | FARE <sup>6</sup>  | 1.9      | 16.8               | 0.0  | 5.9     | 3.5      | 5.7  | 9.2     | 0.0  | 0.7     | 3.2        | 2.2        | 50.2 | 0.7        | 13.8   | 8.6                    |
|                   | LORE <sup>6</sup>  | 4.8      | 24.3               | 0.2  | 6.7     | 5.0      | 7.4  | 8.1     | 0.0  | 2.0     | 5.5        | 4.0        | 50.2 | 3.6        | 22.3   | 10.7 <span>↑2.1</span> |
|                   | FARE <sup>8</sup>  | 2.2      | 19.1               | 0.0  | 8.4     | 3.8      | 5.4  | 10.0    | 0.0  | 1.4     | 3.2        | 2.2        | 50.2 | 0.9        | 16.2   | 9.3                    |
|                   | LORE <sup>8</sup>  | 7.5      | 30.4               | 0.4  | 7.6     | 6.2      | 8.0  | 10.2    | 0.0  | 3.8     | 6.5        | 4.8        | 50.2 | 6.6        | 25.6   | 12.3 <span>↑3.0</span> |
|                   | FARE <sup>10</sup> | 2.6      | 19.5               | 0.0  | 9.1     | 4.0      | 5.2  | 10.1    | 0.0  | 1.6     | 3.3        | 2.1        | 50.2 | 1.2        | 16.7   | 9.4                    |
|                   | LORE <sup>10</sup> | 8.0      | 31.9               | 0.4  | 5.0     | 5.3      | 8.0  | 10.6    | 0.0  | 4.0     | 6.8        | 5.0        | 50.2 | 6.8        | 20.0   | 11.8 <span>↑2.4</span> |
| $\epsilon = 10.0$ | FARE <sup>6</sup>  | 0.7      | 9.2                | 0.0  | 2.8     | 1.9      | 3.3  | 6.8     | 0.0  | 0.1     | 1.5        | 1.2        | 50.0 | 0.2        | 4.4    | 6.3                    |
|                   | LORE <sup>6</sup>  | 1.7      | 13.9               | 0.0  | 2.8     | 2.7      | 4.8  | 0.1     | 0.0  | 0.7     | 3.2        | 2.2        | 50.1 | 0.4        | 9.5    | 6.9 <span>↑0.6</span>  |
|                   | FARE <sup>8</sup>  | 0.8      | 10.5               | 0.0  | 3.9     | 2.2      | 3.6  | 8.0     | 0.0  | 0.5     | 1.5        | 1.1        | 50.1 | 0.5        | 6.5    | 6.8                    |
|                   | LORE <sup>8</sup>  | 3.2      | 18.9               | 0.1  | 3.7     | 3.3      | 5.6  | 2.9     | 0.0  | 1.6     | 4.2        | 3.0        | 50.1 | 2.1        | 13.8   | 8.4 <span>↑1.6</span>  |
|                   | FARE <sup>10</sup> | 1.0      | 11.5               | 0.0  | 5.1     | 2.3      | 3.4  | 8.5     | 0.0  | 0.6     | 1.8        | 1.2        | 50.1 | 0.6        | 7.6    | 7.1                    |
|                   | LORE <sup>10</sup> | 4.0      | 20.2               | 0.1  | 2.1     | 3.2      | 6.0  | 7.1     | 0.0  | 1.7     | 4.9        | 3.3        | 50.2 | 2.8        | 11.2   | 8.7 <span>↑1.6</span>  |

Table 11: Training time (in minutes) required to reach different robust accuracy (RAcc) thresholds under PGD attack ( $\epsilon = 2/255$ ). Lower is better. All models were trained using 4 NVIDIA H100 80GB GPUs.

| Model      | Method | 10% | 20% | 30% | 35% | Dataset      |
|------------|--------|-----|-----|-----|-----|--------------|
| ViT-B/16   | FARE   | 23  | 38  | 64  | 98  | ImageNet-100 |
|            | LORE   | 11  | 17  | 23  | 31  |              |
| ViT-B/32   | FARE   | 30  | 86  | —   | —   | ImageNet     |
|            | LORE   | 25  | 68  | 185 | —   |              |
| ViT-B/32   | FARE   | 62  | 115 | 165 | 285 | ImageNet-100 |
|            | LORE   | 47  | 84  | 148 | 268 |              |
| ConvNeXt-B | FARE   | 19  | 26  | 34  | 40  | ImageNet-100 |
|            | LORE   | 52  | 57  | 60  | 63  |              |
| ViT-S/14   | FARE   | 25  | 43  | 100 | 181 | ImageNet     |
|            | LORE   | 39  | 71  | 124 | 126 |              |
| ViT-B/14   | FARE   | 22  | 28  | 43  | 56  | ImageNet     |
|            | LORE   | 29  | 32  | 42  | 53  |              |

Table 12: Training time (in seconds) for 50 iterations of LORE using 4xH100 GPUs across different architectures for  $\lambda_\omega$ . The *Linear* model uses a single-layer projection: `nn.Linear(self.embedding_size, 1)`. Lower is better.

| Architecture                         | Training Time (s) |
|--------------------------------------|-------------------|
| LORE (Current Dual Network)          | 551               |
| Scalar $\lambda$ (input-independent) | 533               |
| Linear $\lambda$ (input-independent) | 540               |
| Pretrained CLIP                      | 692               |



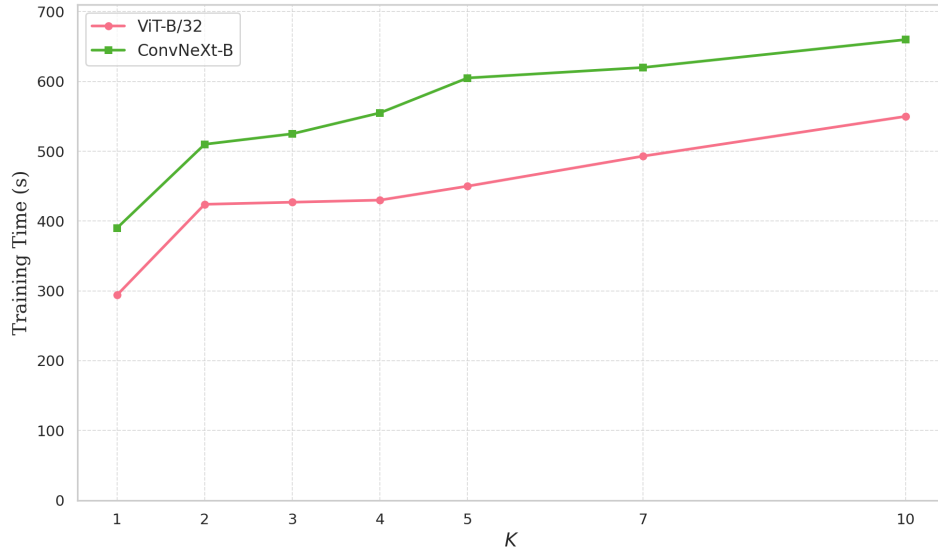


Figure 15: Figure: Training time (in seconds) for completing 30 training iterations of LORE under different values of  $K$ , using 8×H100 GPUs. Results are reported for two architectures: ViT-B/32 and ConvNeXt-B. Increasing  $K$  slightly raises the training time due to more frequent primal updates, with consistent trends across both models.