Fast Adaptive Test-Time Defense with Robust Features

Anurag Singh*[,1], **Mahalakshmi Sabanayagam*[,2]**, Krikamol Muandet[1], Debarghya Ghoshdastidar[2]

[1] CISPA–Helmholtz Center for Information Security, [2] TU Munich

Objective

Given a trained model, how to improve robustness to adversarial attacks at ***test-time***?

Can we do it ***efficiently*** without additional computation costs?

Analysis: Generalized Additive Model

Let $x \in \mathbb{R}^d$, $y \in \{0,1\}^C$ and $y = h(x) + \epsilon$ where $\epsilon \in \mathbb{R}^C, \mathbb{E}[\epsilon_c] = 0; \mathbb{E}[\epsilon_c^2] \le \sigma^2$.

Trained model: Generalized Additive Model

$$h(x) = \beta^T \phi(x)$$

Captures Neural Networks!

Let β_c, y_c be the c-th column of β, y and $\Sigma = \mathbb{E}_x[\phi(x)\phi(x)^T] = U\Lambda U^T$.

Feature Robustness

$$s_{\mathscr{D},\beta,c}(f) = \mathbb{E}_{(x,y)\sim\mathscr{D}}\left[\inf_{\|\tilde{x}-x\|_2 \le \Delta} y_c \beta_c^\top f(\tilde{x})\right]$$

Feature Information

$$\rho_{\mathscr{D},\beta,c}(f) = \mathbb{E}_{(x,y)\sim\mathscr{D}}\left[y_c \beta_c^\top f(x)\right]$$

Check out the paper here!

Theoretical Results

Goal: $h(x) = h_{robust}(x) + h_{nonrobust}(x)$

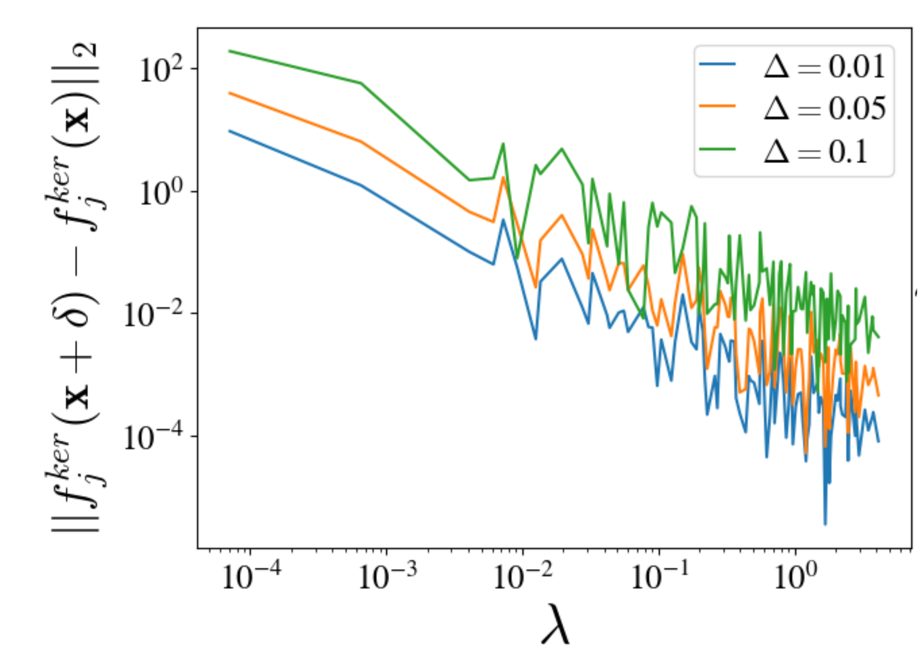
Lower Bound on $s_{\mathscr{D},\beta,c}(f)$

Let ϕ be L-Lipschitz and for any $f(x) = M^T\phi(x)$ i.e. linear transformation of ϕ

$$s_{\mathscr{D},\beta,c}(f) \ \ge \ \beta_c^\top \Sigma M \beta_c - L\Delta \|M\|_{op} \|\beta_c\|_{\mathscr{H}} \sqrt{\sigma^2 + \beta_c^\top \Sigma \beta_c}$$

Features with large $s_c(u_i)$ are Robust

For $M = PP^T$ where P is orthonormal basis, the lower bound on $s_{\mathscr{D},\beta,c}(f)$ is maximized when $f(x) = \tilde{U}\tilde{U}^\top\phi(x)$ where $\tilde{U}$ is the K eigenvectors with $s_c(u_i) = \lambda_i(\beta_c^\top u_i)^2$ largest

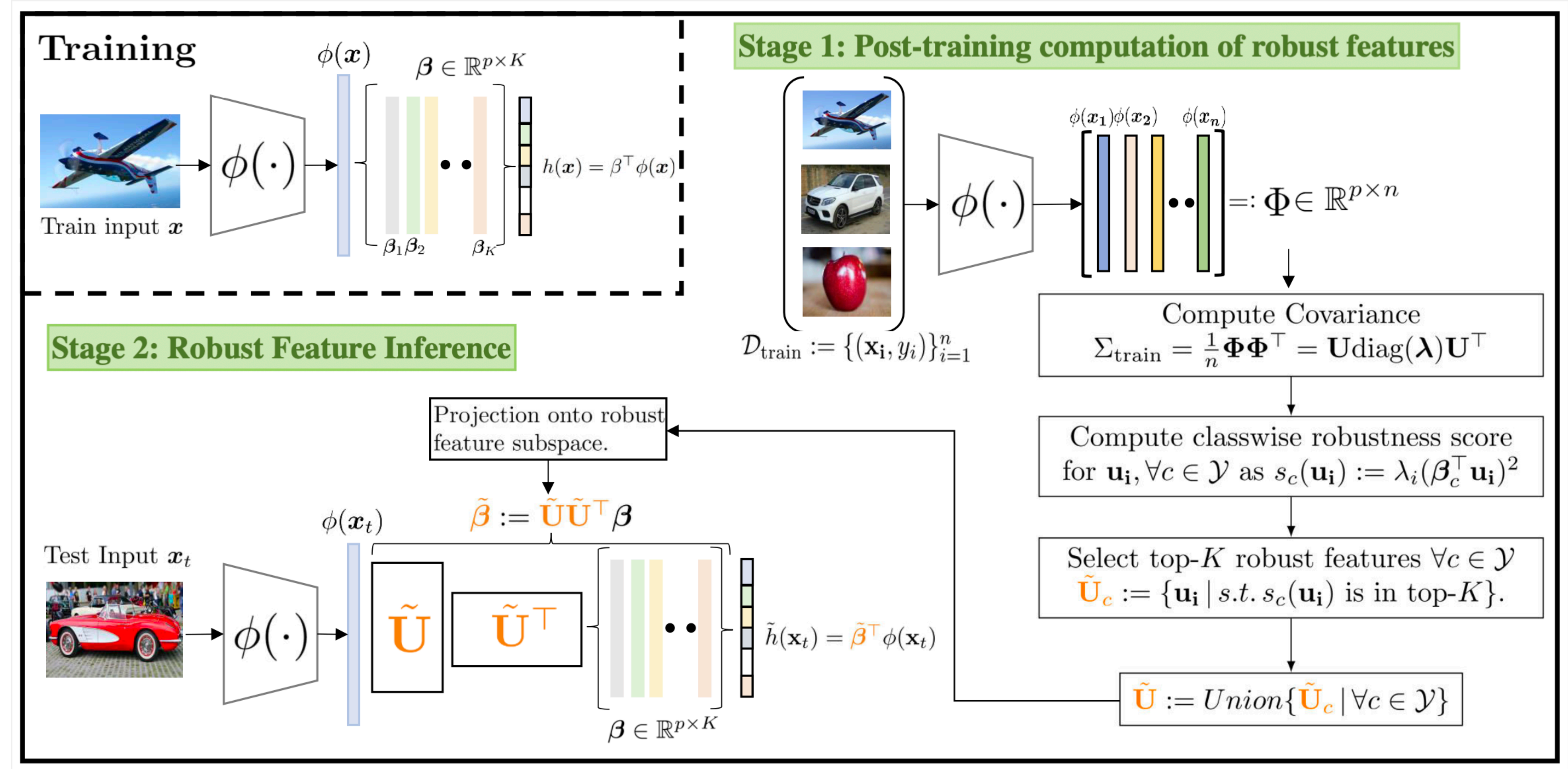


Robust Features are Informative

For $f(x) = \tilde{U}\tilde{U}^\top\phi(x)$, feature information

$$\rho_{\mathscr{D},\beta,c}(f) = \sum_{i=1}^{K} s_c(u_i)$$

Robust Feature Inference (RFI)



CIFAR-10 Training	Clean		$\ell_\infty(\epsilon = \frac{8}{255})$		$\ell_2(\epsilon = 0.5)$	
	Method	+RFI	Method	+RFI	Method	+RFI
PGD	**83.53**	83.22	42.20	**43.29**	54.61	**55.03**
IAT	**91.86**	91.26	44.76	**46.95**	62.53	**64.31**
C&W attack	**85.11**	84.97	40.01	**42.56**	55.02	**56.79**

Base Method	Adaptive Defense	Clean	APGD-CE	APGD-DLR	RobustBench
WideResNet-34-10	None	85.34	50.12	56.80	53.42
	Anti-adv [8×]	**85.40**	50.10	57.50	50.98
	SODEF [2×]	85.10	50.60	56.50	50.09
	RFI [1×]	85.30	**51.62**	**58.97**	**54.86**
WideResNet-28-10 Current SOTA	None	**92.44**	70.23	67.82	67.31
	Anti-adv [8×]	**92.44**	68.90	65.91	66.52
	SODEF [2×]	92.01	67.53	65.08	64.20
	RFI [1×]	92.34	**70.36**	**67.90**	**67.50**

References

[1] Ilyas, Andrew, et al. Adversarial examples are not bugs, they are features. NeurIPS 2019
[2] Rice, Leslie, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. ICML 2020
[3] Wang, Zekai, et al. Better diffusion models further improve adversarial training. ICML 2023
[4] Alfarra, Motasem, et al. Combating adversaries with anti-adversaries. AAAI 2022
[5] Kang, Qiyu, et al. Stable neural ode with lyapunov-stable equilibrium points for defending against adversarial attacks. NeurIPS 2021