CAN ACTIVE RIVAL INTERACTIONS EXPOSE ADVERSARIAL VULNERABILITIES IN DEEP NETWORKS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Adversarial attacks exploit the decision geometry of deep neural networks to expose vulnerabilities under small input perturbations. However, these attacks primarily optimize top-1 misclassification losses or manipulate ranked outputs, leaving dynamic pairwise interactions among non-source rival classes closest to overtaking the source class underexplored, which may, in turn, limit attack effectiveness. In particular, probing a direction associated with one rival class can reveal changes in another rival class's normalized margin gradient. This reveals a potential source of attack potency: pairwise margin-field responses provide geometric information beyond individual rival gradients at the current iterate and can expose adversarial directions missed by independent classwise optimization or single-margin ascent. We therefore introduce **Rival Interaction Optimization for Holistic Attack** (`RoHoAttack`), a geometry-aware white-box attack that exploits active top-K non-source rival-class selection, reciprocal probing, residual refinement, and projected lookahead, thereby retaining the highest-margin evaluated ℓ_p-feasible candidate while refreshing the active rival classes after each update. Across natural-vision and medical-imaging benchmarks, `RoHoAttack` achieves higher attack success rates and reduces robust accuracy by up to ≈ 12.5 points over leading baselines, thereby demonstrating its attack effectiveness. Code

1 INTRODUCTION

Deep neural architectures have witnessed tremendous progress across computer vision, language understanding, scientific discovery, and foundation-model `AI` systems (Dhar et al., 2026b; Paniagua et al., 2025). Despite these advances, adversarial vulnerability remains a fundamental obstacle to trustworthy real-world use (Goodfellow et al., 2015; Dhar et al., 2026b). Adversarial attacks therefore play a central role in robustness research: white-box adversaries can access model parameters, logits, and gradients to search for high-potency perturbations (Madry et al., 2018). This, in turn, provides an upper-bound-style setting for testing architectures and defenses against adversarial perturbations (Carlini & Wagner, 2017; Madry et al., 2018).

Classical adversarial attacks (Madry et al., 2018; Carlini & Wagner, 2017; Croce & Hein, 2020b) primarily induce top-1 misclassification by crossing decision boundaries (Paniagua et al., 2025) (Fig. 1(A)). However, recent studies broaden this view by exploring classifiers' richer decision structure through logits, margins, confidence scores, top-K prediction sets, and class rankings (Paniagua et al., 2023; Reza et al., 2025). From this perspective, query-based (Chen et al., 2020) and geometric attacks (Rahmati et al., 2020; Reza et al., 2023; 2025) exploit boundary information under restricted model access. In contrast, ordered top-K attacks (Zhang & Wu, 2020) (Fig. 1(B)) impose ranking constraints through feature-space optimization (Paniagua et al., 2023) or linearize logit-ranking constraints for image-space search within adversarial-Jacobian subspaces (Paniagua et al., 2025).

Despite their progress, these attacks leave dynamic pairwise interactions among the strongest non-source rival classes underexplored, thereby potentially limiting attack effectiveness. In particular, near a decision boundary, a probe associated with one rival class can reveal changes in another's margin gradient, thereby suggesting candidate directions beyond isolated margin ascent. This observation motivates our central question: ***Can active rival interactions guide white-box adversarial search to improve its effectiveness within the same ℓ_p budget?***

We answer this question by introducing **Rival Interaction Optimization for Holistic Attack** (`RoHoAttack`), a geometry-aware white-box attack that draws inspiration from Lie brackets (Murray & Sastry, 1993) to explore rival-class interactions under fixed ℓ_p perturbation constraints. At each iteration, it probes reciprocal margin-gradient responses among active top-K non-source rival

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.

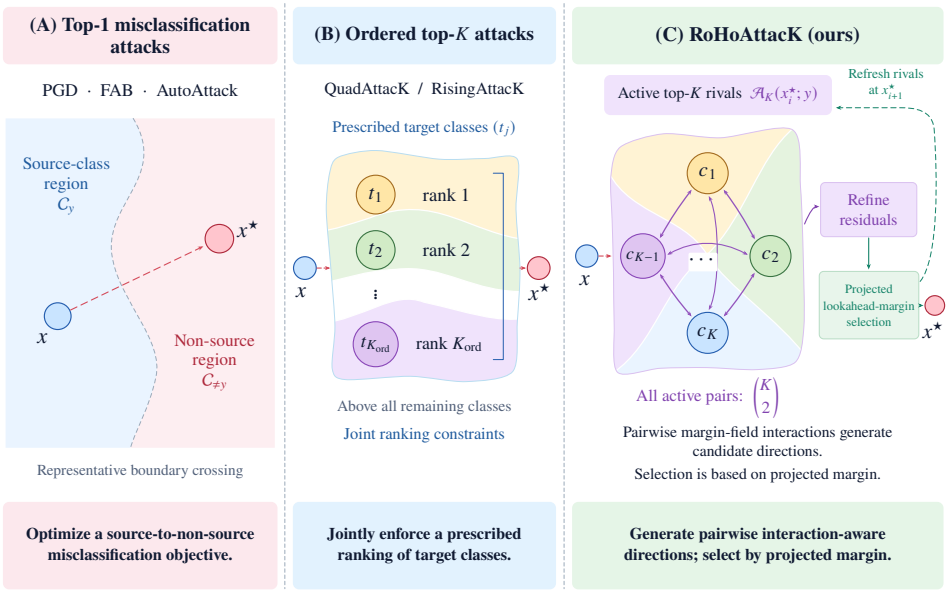

Figure 1: **High-level conceptual comparison of adversarial attacks.** **(A)** Attacks such as PGD (Madry et al., 2018), FAB (Croce & Hein, 2020a), and AutoAttack (Croce & Hein, 2020b) induce top-1 misclassification through source-class boundary crossing. **(B)** Ordered top-K attacks enforce prescribed target rankings through feature-space optimization in QuadAttacK (Paniagua et al., 2023) or image-space search within adversarial-Jacobian subspaces in RisingAttacK (Paniagua et al., 2025). These approaches leave dynamic pairwise margin-field responses among competitive non-source rivals underexplored, potentially overlooking interaction-induced adversarial directions. **(C)** RoHoAttacK refines pairwise margin-field residuals among active top-K non-source rivals into candidate directions, selects the highest-margin evaluated candidate after projection into the ℓ_p-feasible region, and refreshes rivals after each update. This interaction-aware search achieves higher attack success rates and lower robust accuracy than the evaluated baselines (Table 1; Fig. 3).

classes, refines the resulting cross-response residuals into candidate directions, and uses projected lookahead to retain the highest-margin evaluated feasible candidate (Fig. 1(C)). Refreshing the active rival classes after each update adapts this constraint-aligned search to evolving class competition, thereby enabling `RoHoAttack` to improve attack effectiveness by exposing vulnerabilities that prior attacks may miss. Our main contributions are summarized as follows:

1. **Geometry-aware adversarial attack.** We introduce `RoHoAttack`, a new state-of-the-art white-box ℓ_p-bounded adversarial attack for exposing vulnerabilities in standard classifiers (**Table 1**) and adversarially robust architectures (**Fig. 3**).
2. **Interaction-aware adversarial optimization.** We propose a projection-aligned optimization strategy that jointly exploits active top-K rival-class selection, Lie-bracket-inspired pairwise direction generation, and residual refinement, while projected lookahead selects the evaluated candidate with the largest realized adversarial margin increase to improve attack effectiveness.
3. **Comprehensive adversarial evaluation.** We evaluate `RoHoAttack` against standard and recent attacks across convolutional and transformer-based classifiers on diverse natural-vision and medical-imaging benchmarks (**Tables 1–2 and Fig. 3**).

Related work. We include an extensive discussion of related work in App. A.

2 Method

2.1 Problem Definition

We first specify the model and the adversary's access, then define the source-displacement objective under ℓ_p constraints and the active top-K rival-margin loss used for adversarial optimization.

Model under attack. Let $(x, y) \in [l, u]^D \times \mathcal{Y}$ denote a clean image and its ground-truth label, where $D = \mathrm{Ch} \times H \times W$ is the input dimension, $\mathcal{Y} = \{1, \ldots, C\}$ is the C-class label space, and $[l, u]$ is the valid pixel range. Here, Ch, H, and W denote the channel count, height, and width, respectively. We consider a trained classifier $\ell(\cdot; \Theta) : [l, u]^D \to \mathbb{R}^C$, with parameters Θ, logit

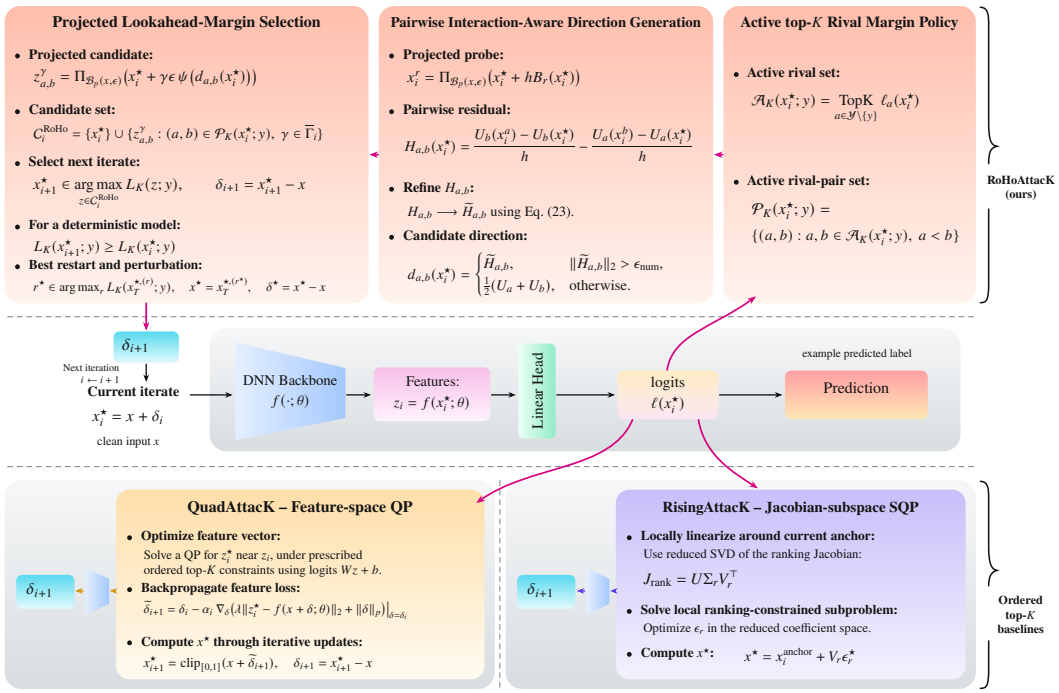

Figure 2: Workflow comparison between our proposed **RoHoAttacK** (top) and the prior art (bottom), QuadAttacK (Paniagua et al., 2023) and RisingAttacK (Paniagua et al., 2025). For a clean image $x \in [l,u]^D$, RoHoAttacK forms active top-K rival pairs, generates interaction-aware directions from reciprocal margin-gradient responses, and selects the projected feasible update with the largest realized adversarial margin progress.

vector $\ell(x) \equiv \ell(x;\Theta)$, and class logits $\ell_c(x)$ for $c \in \mathcal{Y}$. In the white-box setting (Carlini & Wagner, 2017; Madry et al., 2018; Chen & Liu, 2023), the adversary accesses model parameters, logits, and input gradients without modifying Θ. The objective is to find a small input-space perturbation that displaces the source class from the leading decision position.

Source-displacement objective. We evaluate clean samples correctly classified as y and construct adversarial candidates as $x^\star = x + \delta$, where $\delta \in \mathbb{R}^D$ is the input perturbation. At iteration i, $x_i^\star = x + \delta_i$, where δ_i is the total perturbation relative to x; the next selected candidate gives $\delta_{i+1} = x_{i+1}^\star - x$ (Fig. 2). For an ℓ_p norm with $p \in \{1, 2, \infty\}$ and perturbation radius $\epsilon > 0$, the feasible candidate region is

$$\mathcal{B}_p(x,\epsilon) = \left\{ x^\star \in [l,u]^D : \|x^\star - x\|_p \le \epsilon \right\}. \tag{1}$$

A feasible candidate succeeds when a non-source class strictly overtakes the source class, thereby defining the source-displacement region

$$\mathcal{S}_p(x,y) = \{ x^\star \in \mathcal{B}_p(x,\epsilon) : \exists a \in \mathcal{Y} \setminus \{y\},\ \ell_a(x^\star) > \ell_y(x^\star) \}. \tag{2}$$

Active top-K rival-margin loss. Let $K \in \{1, \dots, C-1\}$ denote the number of active non-source rival classes. At a candidate $x^\star$, we select

$$\mathcal{A}_K(x^\star; y) = \mathrm{TopK}_{a \in \mathcal{Y} \setminus \{y\}}\, \ell_a(x^\star), \tag{3}$$

where TopK returns the indices of the K largest non-source logits. We recompute this set at $x_i^\star$ each iteration to track evolving class competition. Each non-source class a has source-relative margin

$$m_a(x^\star; y) = \ell_a(x^\star) - \ell_y(x^\star), \qquad a \in \mathcal{Y} \setminus \{y\}. \tag{4}$$

Since the active set contains a highest-scoring non-source class, the active rival loss satisfies

$$\mathcal{L}_K(x^\star; y) = \max_{a \in \mathcal{A}_K(x^\star; y)} m_a(x^\star; y) = \max_{a \in \mathcal{Y} \setminus \{y\}} \left[\ell_a(x^\star) - \ell_y(x^\star) \right]. \tag{5}$$

For feasible $x^\star$, $\mathcal{L}_K(x^\star; y) > 0$ is therefore equivalent to $x^\star \in \mathcal{S}_p(x,y)$ (Proposition 3; Appendix B.1). Thus, K controls candidate-direction generation without changing the source-displacement loss. The corresponding feasible perturbation set is

$$\Delta_p(x,\epsilon) = \left\{\delta \in \mathbb{R}^D : x+\delta \in [l,u]^D,\ \|\delta\|_p \le \epsilon\right\}. \tag{6}$$

The attack therefore seeks

$$\max_{\delta \in \Delta_p(x,\epsilon)} \mathcal{L}_K(x+\delta; y). \tag{7}$$

We select the final iterate with the highest active rival loss across restarts as $x^\star$, thereby obtaining the adversarial perturbation $\delta^\star = x^\star - x$. The ℓ_p-norm definition and minimum-perturbation formulation are provided in Appendix B.2. Eq. 7 specifies what should be optimized: maximizing the active rival loss within the feasible ℓ_p region used for evaluation. What remains is to determine how candidate directions should be generated from the active top-K rival structure and how a projected candidate should be selected to expose adversarial vulnerabilities. We therefore aim to develop a robust and potent attack through a direction-selection mechanism that generates interaction-aware directions and selects, among evaluated projected candidates, the update with the largest realized active-loss increase to improve attack effectiveness.

2.2 Rival Interaction Optimization for Holistic Attack

Motivation. The objective in Eq. 7 specifies the loss to maximize, but not how effective candidate directions should be generated and selected. This distinction is important: several active non-source rival classes can compete simultaneously, yet an isolated margin gradient captures only one rival's local response. Consequently, prior attacks relying on isolated margin ascent may have limited effectiveness in discovering directions revealed by reciprocal rival-gradient responses. This motivates an interaction-aware search that exploits local class competition while assessing candidate quality after enforcing feasibility.

To address this requirement, we introduce **Rival Interaction Optimization for Holistic Attack** (`RoHoAttack`), a geometry-aware white-box ℓ_p-bounded attack that unifies interaction-aware exploration and projection-aligned selection to improve attack effectiveness. Drawing inspiration from Lie brackets (Murray & Sastry, 1993), `RoHoAttack` forms active top-K rival pairs and refines their reciprocal cross-response residuals into candidate directions. Projected lookahead-margin selection then chooses, among evaluated projected candidates, the update with the largest realized active-loss increase. Thus, rival interactions guide exploration, while realized adversarial progress determines the accepted update. At iteration i, we maintain a feasible iterate $x_i^\star \in \mathcal{B}_p(x,\epsilon)$, with total perturbation δ_i relative to the clean image x:

$$x_i^\star = \Pi_{\mathcal{B}_p(x,\epsilon)}(x+\delta_i) = x+\delta_i, \qquad \delta_i = x_i^\star - x. \tag{8}$$

Around this iterate, we construct the candidate direction family

$$\mathcal{D}_K^{\mathrm{H}}(x_i^\star; y) = \{d_{a,b}(x_i^\star) : (a,b) \in \mathcal{P}_K(x_i^\star; y)\}, \tag{9}$$

where $\mathcal{P}_K$ is the active rival-pair set and $d_{a,b}$ is the direction generated by pair (a,b) such that a,b are distinct non-source class indices. Rather than accepting these directions directly, we evaluate their feasible realizations. Given a finite signed step set $\bar{\Gamma}_i$ containing zero, we form

$$\mathcal{C}_i^{\mathrm{RoHo}} = \{x_i^\star\} \cup \left\{\Pi_{\mathcal{B}_p(x,\epsilon)}(x_i^\star + \gamma\epsilon\psi(d)) : d \in \mathcal{D}_K^{\mathrm{H}}(x_i^\star; y),\ \gamma \in \bar{\Gamma}_i\right\}, \tag{10}$$

where ψ is the bounded direction map and $\Pi_{\mathcal{B}_p(x,\epsilon)}$ enforces the perturbation and image-domain constraints. We then select the next iterate and compute the corresponding perturbation:

$$x_{i+1}^\star \in \operatorname*{arg\,max}_{z \in \mathcal{C}_i^{\mathrm{RoHo}}} \mathcal{L}_K(z; y), \qquad \delta_{i+1} = x_{i+1}^\star - x. \tag{11}$$

We subsequently refresh the active rival classes at $x+\delta_{i+1}$, thereby adapting the search to evolving class competition. Fig. 2 illustrates the workflow, and Algorithms 1–3 provide its implementation (see App. E). The following subsections detail the three components and their formulations.

2.2.1 Active Top-K Rival-Margin Policy

Motivation. The active set in Eq. 3 identifies competitive non-source rival classes, while Eq. 4 measures their source-relative margins. However, knowing which rivals are competitive does not determine which reciprocal responses should guide direction generation. This distinction is important: considering rival margins independently can overlook directions revealed by their interactions.

Table 1: **Attack effectiveness comparison against non-robust classifiers on ImageNet-1K.** ASR, active rival margin M_K, perturbation norms, and average computation time (s/image) across CNN and transformer architectures at $\epsilon = 8/255$. **Green** denotes best performance.

(a) ResNet-50

Attack	ASR↑	M_K ↑	ℓ_1 ↓	ℓ_2 ↓	ℓ_∞ ↓	Time↓
PGD	0.9254	3.7028	4.9134	2140.3821	0.0314	**0.4217**
APGD-DLR	0.9473	4.3274	4.3815	1960.7428	0.0314	0.8239
C&W	0.9186	3.9841	3.7638	1586.5172	0.0264	4.7046
DeepFool	0.8865	2.4572	2.7042	1110.6385	0.0212	1.0834
FAB	0.9148	3.3259	2.8037	1188.9216	0.0230	1.4827
APAA	0.9526	4.6243	4.1249	1875.6394	0.0304	0.7241
PMA	0.9551	4.7428	3.6824	1568.7421	0.0261	1.3259
MALT	0.9584	4.8365	3.5148	1492.3184	0.0254	1.1184
Prompt2Perturb	0.9372	4.4213	3.2467	1378.5062	0.0243	5.9185
RoHoAttacK	**0.9976**	**6.2847**	**2.5128**	**1035.4273**	**0.0196**	8.4264

(b) DenseNet-121

Attack	ASR↑	M_K ↑	ℓ_1 ↓	ℓ_2 ↓	ℓ_∞ ↓	Time↓
PGD	0.9180	3.6119	5.0518	2215.2743	0.0314	**0.4528**
APGD-DLR	0.9435	4.1867	4.6531	2055.7189	0.0314	0.9136
C&W	0.9123	3.8214	3.9515	1664.4925	0.0272	5.1068
DeepFool	0.8762	2.2986	2.7836	1152.6842	0.0218	1.1947
FAB	0.9095	3.2148	2.9145	1235.5371	0.0239	1.6384
APAA	0.9487	4.4826	4.4218	1986.3289	0.0309	0.8125
PMA	0.9518	4.5962	3.8246	1658.4927	0.0268	1.3842
MALT	0.9551	4.7048	3.6519	1572.8364	0.0259	1.1765
Prompt2Perturb	0.9324	4.2865	3.3825	1426.7148	0.0247	6.2048
RoHoAttacK	**0.9964**	**6.0186**	**2.5841**	**1073.5127**	**0.0201**	8.8735

(c) ViT-B

Attack	ASR↑	M_K ↑	ℓ_1 ↓	ℓ_2 ↓	ℓ_∞ ↓	Time↓
PGD	0.9015	3.1647	5.1827	2248.5193	0.0314	**0.4842**
APGD-DLR	0.9324	4.0385	4.7739	2094.8625	0.0314	0.9328
C&W	0.9062	3.5428	4.0841	1715.4368	0.0278	5.3487
DeepFool	0.8649	2.1026	2.9138	1196.4726	0.0225	1.2549
FAB	0.8955	2.9537	3.0187	1270.6941	0.0245	1.7286
APAA	0.9418	4.3162	4.5634	2015.8172	0.0310	0.8423
PMA	0.9449	4.4327	3.9687	1712.5843	0.0272	1.4639
MALT	0.9476	4.5581	3.7469	1614.9085	0.0264	1.2315
Prompt2Perturb	0.9243	4.1164	3.4826	1488.2731	0.0251	6.6874
RoHoAttacK	**0.9937**	**5.7824**	**2.7015**	**1118.4362**	**0.0206**	9.1847

(d) DEiT-B

Attack	ASR↑	M_K ↑	ℓ_1 ↓	ℓ_2 ↓	ℓ_∞ ↓	Time↓
PGD	0.8987	3.0916	4.9185	2108.7534	0.0314	**0.4631**
APGD-DLR	0.9296	3.9624	4.5526	1956.4982	0.0314	0.8916
C&W	0.9025	3.4182	3.8849	1648.2769	0.0269	5.0248
DeepFool	0.8674	2.1873	2.7517	1132.5947	0.0217	1.1375
FAB	0.8943	3.0145	2.8435	1171.4286	0.0233	1.5832
APAA	0.9396	4.2149	4.1378	1915.7294	0.0308	0.7939
PMA	0.9418	4.3265	3.8124	1635.9276	0.0267	1.4172
MALT	0.9442	4.4517	3.5936	1548.3819	0.0258	1.1834
Prompt2Perturb	0.9210	4.0386	3.3175	1396.4528	0.0245	6.3518
RoHoAttacK	**0.9928**	**5.5119**	**2.5426**	**1054.8937**	**0.0199**	8.7362

We therefore design `RoHoAttack` with an active rival-pair policy that extends exploration beyond isolated margin ascent while restricting pairwise search to active rival classes.

For $2 \le K \le C-1$, let $\mathcal{P}_K(x_i^\star; y)$ denote the set of unordered pairs drawn from the active rival classes at iterate $x_i^\star = x + \delta_i$:

$$\mathcal{P}_K(x_i^\star; y) = \{(a, b) : a, b \in \mathcal{A}_K(x_i^\star; y),\ a < b\}\,. \tag{12}$$

Here, $a < b$ represents each unordered pair once. Whereas $\mathcal{A}_K$ contains individual rival classes, $\mathcal{P}_K$ determines which rival-gradient responses are jointly probed. The following proposition characterizes the resulting search and its consistency with the source-displacement objective.

Proposition 1 (Focused active-pair interaction search)**.** *For $2 \le K \le C-1$, the active pair set has cardinality $|\mathcal{P}_K(x_i^\star; y)| = \binom{K}{2}$, compared with $\binom{C-1}{2}$ over all non-source classes. It therefore retains the fraction $\binom{K}{2}/\binom{C-1}{2}$ of these interactions. For every feasible candidate z, evaluating $\mathcal{L}_K(z; y)$ with its active set $\mathcal{A}_K(z; y)$ preserves the exact source-displacement loss in Proposition 3.*

Proof. See Appendix C.1.

2.2.2 Pairwise Interaction-Aware Direction Generation

Motivation. Pairwise interaction-aware direction generation converts the active rival-pair structure into candidate perturbation directions in Eq. 9. A single source-relative margin gradient captures the first-order variation of one rival's margin, as in margin- and boundary-based attacks (Moosavi-Dezfooli et al., 2016; Carlini & Wagner, 2017; Croce & Hein, 2020a), whereas top-K attacks explore richer ranked prediction structure (Tursynbek et al., 2022; Paniagua et al., 2023; 2025). Accordingly, `RoHoAttack` probes reciprocal margin-gradient responses between two active rival classes and refines their residual response to generate interaction-aware directions beyond isolated single-margin ascent, drawing inspiration from Lie brackets (Murray & Sastry, 1993). The resulting directions form $\mathcal{D}_K^{\mathrm{H}}(x_i^\star; y)$ and are evaluated by the projected selection rule in Eq. 11.

To construct this family, `RoHoAttack` makes rival gradients comparable, uses them to design feasible probes, and measures how each probe changes the other rival's gradient. We refine the difference between reciprocal changes into a candidate direction. For each active rival $r \in \mathcal{A}_K(x_i^\star; y)$, we differentiate its source-relative margin at an evaluation input s and normalize the resulting gradient:

$$g_r(s) = \nabla_s m_r(s; y), \qquad U_r(s) = \frac{g_r(s)}{\sqrt{D^{-1}\sum_{j=1}^{D} g_{r,j}(s)^2 + \rho}}. \tag{13}$$

Here, $g_{r,j}$ is gradient coordinate j and $\rho > 0$ stabilizes the denominator. The input s is either the current iterate or a probe input. Normalization reduces the influence of unequal raw gradient scales when comparing reciprocal responses. Rather than applying $U_r(x_i^\star)$ directly as an attack update, we use it to determine where to measure a neighboring gradient response. With probe scale $h = \eta\epsilon$, where $\eta > 0$ controls the probing factor, we define $B_r(x_i^\star) = \tanh(\mathrm{RMSNorm}(U_r(x_i^\star)))$, where $\mathrm{RMSNorm}$ applies root-mean-square normalization and coordinatewise $\tanh$ bounds the probe direction. For an active pair $(a, b) \in \mathcal{P}_K(x_i^\star; y)$, we formalize

$$x_i^r = \Pi_{\mathcal{B}_p(x,\epsilon)}(x_i^\star + hB_r(x_i^\star)), \qquad r \in \{a, b\}. \tag{14}$$

The projection enforces both the image-domain and perturbation constraints. We then evaluate U_b at x_i^a and U_a at x_i^b, retaining the selected class indices during probing. Thus, each probe tests how a movement associated with one rival changes the other rival's gradient, while both responses are measured relative to $x_i^\star$. To isolate these changes from the gradients at the current iterate, we subtract their respective baseline responses and form

$$H_{a,b}(x_i^\star) = \frac{U_b(x_i^a) - U_b(x_i^\star)}{h} - \frac{U_a(x_i^b) - U_a(x_i^\star)}{h}. \tag{15}$$

Dividing by h expresses each change relative to the probe scale; subtracting the reciprocal changes captures their difference rather than averaging the original rival gradients. The resulting cross-response residual $H_{a,b}(x_i^\star) \in \mathbb{R}^D$ therefore uses information about how the gradients vary around the current iterate. The following proposition relates this information to Lie brackets.

Proposition 2 (Pairwise residual estimates local margin-gradient interaction)**.** *Assume U_a, U_b are continuously differentiable near $x_i^\star$ with locally Lipschitz Jacobians, and both probe projections in Eq. 14 are locally inactive. Holding $x_i^\star$ fixed, as $h \to 0$,*

$$H_{a,b}(x_i^\star) = J_{U_b}(x_i^\star)B_a(x_i^\star) - J_{U_a}(x_i^\star)B_b(x_i^\star) + \mathcal{O}(h), \tag{16}$$

where J_{U_r} is the input Jacobian of U_r; for $B_r = U_r$ with $r \in \{a, b\}$, the leading term is the Lie bracket:

$$H_{a,b}(x_i^\star) = [U_a, U_b](x_i^\star) + \mathcal{O}(h) \quad \textit{and} \quad [U_a, U_b] = J_{U_b}U_a - J_{U_a}U_b. \tag{17}$$

Proof. See Appendix C.2.

The Jacobian–direction products quantify how one rival's normalized gradient changes along the other's probe direction. This explains what reciprocal probing estimates; the Lie-bracket identity applies to the $B_r = U_r$ specialization, while the implemented probes remain bounded and projected.

Residual refinement. A cross-response residual can contain components along the original rival gradients, so using it directly may repeat classwise movements. We refine it to emphasize the response outside those directions. Let $U_r = U_r(x_i^\star)$ and $H_{a,b} = H_{a,b}(x_i^\star)$, we subtract the regularized component of U_b along U_a to obtain $U_b^\perp$, then subtract the components of $H_{a,b}$ along U_a and $U_b^\perp$:

$$\begin{aligned} \mathrm{Proj}_u^\rho(v) &= \frac{\langle v, u\rangle}{\|u\|_2^2 + \rho}u \quad \text{and} \quad U_b^\perp = U_b - \mathrm{Proj}_{U_a}^\rho(U_b), \\ \widetilde{H}_{a,b} &= H_{a,b} - \mathrm{Proj}_{U_a}^\rho(H_{a,b}) - \mathrm{Proj}_{U_b^\perp}^\rho(H_{a,b}). \end{aligned} \tag{18}$$

Here, $\mathrm{Proj}_u^\rho(v)$ denotes the regularized projection of v along u, $U_b^\perp$ the rival-gradient residual, and $\widetilde{H}_{a,b}$ the refined cross-response residual. The lemma 1 establishes exact separation when $\rho = 0$.

Lemma 1 (Residual refinement removes first-order classwise components)**.** *When $\rho = 0$ and $U_a, U_b^\perp$ are nonzero, the refined residual satisfies*

$$\langle \widetilde{H}_{a,b}, U_a\rangle = 0, \qquad \langle \widetilde{H}_{a,b}, U_b^\perp\rangle = 0. \tag{19}$$

For $\rho > 0$, Eq. 18 provides numerically stabilized, regularized approximate orthogonal separation.

Proof. See Appendix C.3.

The lemma identifies the classwise components removed by refinement; projected lookahead subsequently evaluates the remaining direction's effect on the attack loss. When the refined residual is small, we retain an averaged-gradient candidate rather than rely on a weak interaction response:

$$d_{a,b}(x_i^\star) = \begin{cases} \widetilde{H}_{a,b}, & \|\widetilde{H}_{a,b}\|_2 > \varepsilon_{\mathrm{num}}, \\ \frac{1}{2}(U_a + U_b), & \text{otherwise}, \end{cases} \tag{20}$$

where $\varepsilon_{\text{num}} > 0$ is the residual-norm threshold. Applying this rule to every active pair produces $\mathcal{D}_K^{\text{H}}(x_i^\star; y)$. The next component converts these directions into feasible candidates and compares their achieved active rival losses via Eq. 11, thereby determining the next iterate and its perturbation.

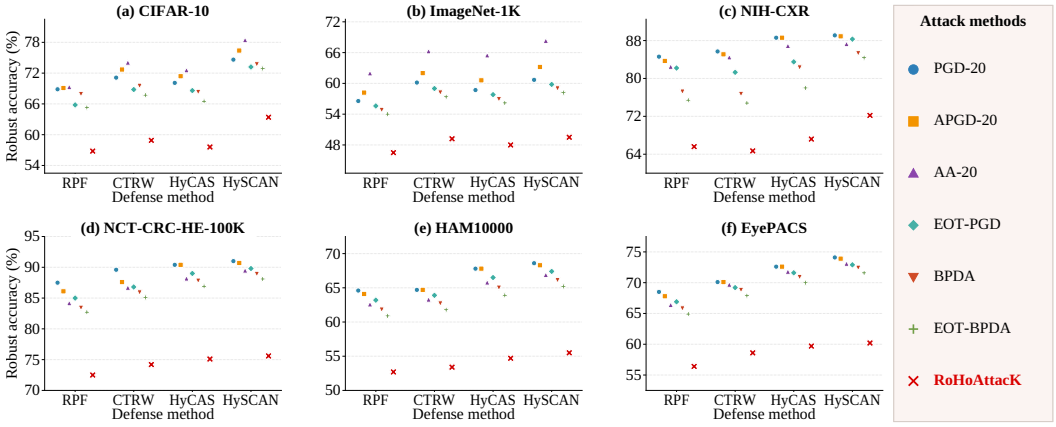

Figure 3: **Attack effectiveness comparison exposing vulnerabilities in robust defenses.** Robust accuracy (%) of RPF, CTRW, HyCAS, and HySCAN on CIFAR-10, ImageNet-1K, NIH-CXR, NCT-CRC-HE-100K, HAM10000, and EyePACS at $\epsilon = 8/255$.

2.2.3 Projected Lookahead Loss Optimization

Motivation. Once the interaction-aware direction family $\mathcal{D}_K^{\text{H}}(x_i^\star; y)$ is generated, the central question becomes which candidate should be trusted. A direction may arise from a large cross-response residual, but this does not guarantee loss improvement after bounded mapping, bidirectional search, and projection into $\mathcal{B}_p(x, \epsilon)$. Therefore, `RoHoAttack` selects candidates by achieved loss after projection, thereby aligning direction selection with the constrained maximization objective in Eq. 7 (Madry et al., 2018; Croce & Hein, 2020b;a). To implement this selection, we evaluate both orientations of each direction at multiple scales, project the resulting candidates, and compare their active rival losses before accepting an update. Let $T > 1$ denote the number of iterations, $i \in \{0, \ldots, T-1\}$ the iteration index, and $\alpha > 0$ the base search scale. We define the search scale and bidirectional lookahead schedule:

$$\alpha_i = \alpha\left[0.55 + 0.45\cos\left(\frac{\pi i}{T-1}\right)\right] \;\Rightarrow\; \Gamma_i = \alpha_i\{\pm 1, \pm 0.5, \pm 0.25\}, \quad \bar{\Gamma}_i = \Gamma_i \cup \{0\}. \tag{21}$$

Here, the search scale α_i decreases from α to 0.1α across iterations, allowing larger exploratory moves early and smaller updates later. Γ_i contains nonzero factors for both orientations at relative magnitudes $\{1, 0.5, 0.25\}$, while $\bar{\Gamma}_i$ adds $\gamma = 0$ to retain the current iterate for comparison. For each active pair $(a, b) \in \mathcal{P}_K(x_i^\star; y)$ and $\gamma \in \bar{\Gamma}_i$, we apply the bounded map ψ to $d_{a,b}(x_i^\star)$, scale by $\gamma\epsilon$, and add to $x + \delta_i$ before projection, yielding the projected candidate:

$$z_{a,b}^{\gamma} = \Pi_{\mathcal{B}_p(x,\epsilon)}(x + \delta_i + \gamma\epsilon\psi(d_{a,b}(x_i^\star)))\,. \tag{22}$$

Here, $z_{a,b}^{\gamma}$ is the feasible image for pair (a, b) and scale γ. All candidates originate from $x_i^\star = x + \delta_i$, with projection constraining their total perturbation relative to the clean image x. Before updating, we compare their source-displacement losses using each candidate's active rival set, rather than residual magnitudes. The lookahead-margin (LM) score $q_{\text{LM}}(a, b \mid x_i^\star)$ denotes the pair's best projected loss over $\bar{\Gamma}_i$. The pair score, selected pair and scale factor, and perturbation update are

$$q_{\text{LM}}(a, b \mid x_i^\star) = \max_{\gamma \in \bar{\Gamma}_i} \mathcal{L}_K(z_{a,b}^{\gamma}; y), \quad \text{and} \quad (a_i^\star, b_i^\star, \gamma_i^\star) \in \underset{\substack{(a,b)\in\mathcal{P}_K(x_i^\star;y)\\ \gamma\in\bar{\Gamma}_i}}{\arg\max} \mathcal{L}_K(z_{a,b}^{\gamma}; y),$$
$$x_{i+1}^\star = z_{a_i^\star, b_i^\star}^{\gamma_i^\star}, \qquad \text{then} \qquad \delta_{i+1} = x_{i+1}^\star - x. \tag{23}$$

Since the current loss is common to every comparison, this rule selects, among evaluated projected candidates, the update with the largest realized active-loss increase. We then refresh the active rival classes at $x + \delta_{i+1}$, thereby adapting subsequent direction generation to evolving class competition. The following theorem characterizes the resulting per-iteration guarantees.

Table 2: **Ordered top-K attack effectiveness comparison on ImageNet-1K with 30 optimization steps.** Mean ASR, perturbation norms, and average computation time (s/image) for `QuadAttacK`, `RisingAttacK`, and `RoHoAttack-O30` across four classifiers and $K_{\text{ord}} \in \{1, 5, 10, 15, 20\}$. **Green** denotes best.

(a) ResNet-50

Top-K	Method	Mean ASR↑	Mean ℓ_1 ↓	Mean ℓ_2 ↓	Mean ℓ_∞ ↓	Time (s/img)↓
Top-20	QuadAttacK$_{30}$	0.0978	9.0948	2850.0433	0.0858	**1.7264**
	RisingAttacK$_{30}$	0.0666	3.4854	1022.5585	0.0269	3.7216
	RoHoAttacK-O$_{30}$	**0.2778**	**3.0672**	**899.8515**	**0.0242**	5.9546
Top-15	QuadAttacK$_{30}$	0.4922	7.8296	2451.8036	0.0717	**1.7382**
	RisingAttacK$_{30}$	0.5856	2.9944	873.3877	0.0234	2.8794
	RoHoAttacK-O$_{30}$	**0.7110**	**2.6351**	**768.5812**	**0.0207**	4.6070
Top-10	QuadAttacK$_{30}$	0.8460	6.3547	1994.8023	0.0544	**1.7593**
	RisingAttacK$_{30}$	0.8064	2.1748	630.0922	0.0175	1.7965
	RoHoAttacK-O$_{30}$	**0.9060**	**1.9138**	**554.4811**	**0.0158**	2.8744
Top-5	QuadAttacK$_{30}$	0.9590	3.9539	1246.4929	0.0300	2.1458
	RisingAttacK$_{30}$	0.9504	1.4693	420.0254	0.0124	**0.9517**
	RoHoAttacK-O$_{30}$	**0.9840**	**1.2325**	**351.8154**	**0.0105**	3.4333
Top-1	QuadAttacK$_{30}$	0.9772	1.4244	461.1199	0.0080	2.6411
	RisingAttacK$_{30}$	0.9986	0.9155	251.6174	0.0088	**0.3201**
	RoHoAttacK-O$_{30}$	**1.0000**	**0.7626**	**209.3175**	**0.0072**	4.2258

(b) DenseNet-121

Top-K	Method	Mean ASR↑	Mean ℓ_1 ↓	Mean ℓ_2 ↓	Mean ℓ_∞ ↓	Time (s/img)↓
Top-20	QuadAttacK$_{30}$	0.0330	9.8564	3072.6790	0.0923	**2.0206**
	RisingAttacK$_{30}$	0.4500	6.1377	1786.5613	0.0485	4.6070
	RoHoAttacK-O$_{30}$	**0.6100**	**5.4012**	**1572.1739**	**0.0437**	7.3712
Top-15	QuadAttacK$_{30}$	0.5088	8.6281	2697.9823	0.0771	**1.8919**
	RisingAttacK$_{30}$	0.9362	4.7380	1362.7350	0.0387	3.5501
	RoHoAttacK-O$_{30}$	**0.9798**	**4.1694**	**1199.2068**	**0.0336**	5.6802
Top-10	QuadAttacK$_{30}$	0.9392	6.6701	2098.0095	0.0531	**1.8918**
	RisingAttacK$_{30}$	0.9880	2.9210	827.1606	0.0253	2.2937
	RoHoAttacK-O$_{30}$	**1.0000**	**2.5705**	**727.9013**	**0.0225**	3.6699
Top-5	QuadAttacK$_{30}$	0.9924	3.9526	1253.5745	0.0262	1.8502
	RisingAttacK$_{30}$	0.9982	1.6603	457.9204	0.0156	**1.2082**
	RoHoAttacK-O$_{30}$	**1.0000**	**1.4216**	**392.0683**	**0.0137**	2.9603
Top-1	QuadAttacK$_{30}$	0.9960	1.5144	501.4779	0.0070	1.5519
	RisingAttacK$_{30}$	**1.0000**	1.0708	280.0300	0.0116	**0.4255**
	RoHoAttacK-O$_{30}$	**1.0000**	**0.9214**	**239.2618**	**0.0063**	2.4830

(c) ViT-B

Top-K	Method	Mean ASR↑	Mean ℓ_1 ↓	Mean ℓ_2 ↓	Mean ℓ_∞ ↓	Time (s/img)↓
Top-20	QuadAttacK$_{30}$	0.0004	6.3770	1992.7502	0.0533	**2.6210**
	RisingAttacK$_{30}$	0.4956	4.9482	1343.2135	0.0473	7.9615
	RoHoAttacK-O$_{30}$	**0.6620**	**4.2531**	**1149.2845**	**0.0410**	12.7384
Top-15	QuadAttacK$_{30}$	0.0056	4.7982	1495.8188	0.0385	**2.4245**
	RisingAttacK$_{30}$	0.7510	3.8944	1038.7394	0.0379	6.0305
	RoHoAttacK-O$_{30}$	**0.8510**	**3.1817**	**847.2894**	**0.0325**	9.6488
Top-10	QuadAttacK$_{30}$	0.0252	3.4999	1094.6987	0.0261	**2.3034**
	RisingAttacK$_{30}$	0.7112	2.6247	684.1576	0.0267	4.1602
	RoHoAttacK-O$_{30}$	**0.8112**	**2.1835**	**568.1873**	**0.0230**	6.6563
Top-5	QuadAttacK$_{30}$	0.5024	3.2930	1029.8490	0.0242	**2.1108**
	RisingAttacK$_{30}$	0.5980	1.6101	406.4644	0.0174	2.2197
	RoHoAttacK-O$_{30}$	**0.6980**	**1.3587**	**343.0915**	**0.0151**	3.5515
Top-1	QuadAttacK$_{30}$	0.9958	1.5681	508.0591	0.0081	1.3040
	RisingAttacK$_{30}$	0.9362	0.6578	149.1661	0.0086	**0.6417**
	RoHoAttacK-O$_{30}$	**0.9992**	**0.5789**	**130.3188**	**0.0073**	2.0864

(d) DEiT-B

Top-K	Method	Mean ASR↑	Mean ℓ_1 ↓	Mean ℓ_2 ↓	Mean ℓ_∞ ↓	Time (s/img)↓
Top-20	QuadAttacK$_{30}$	0.0032	6.2491	1950.0525	0.0524	**2.1503**
	RisingAttacK$_{30}$	0.6348	3.8373	1045.3953	0.0366	7.9624
	RoHoAttacK-O$_{30}$	**0.7348**	**3.3314**	**904.1626**	**0.0322**	12.7398
Top-15	QuadAttacK$_{30}$	0.0338	4.9874	1558.1460	0.0386	**2.0234**
	RisingAttacK$_{30}$	0.9278	3.1490	838.2295	0.0310	6.0263
	RoHoAttacK-O$_{30}$	**0.9732**	**2.7219**	**725.3367**	**0.0266**	9.6421
Top-10	QuadAttacK$_{30}$	0.1298	3.5782	1123.3760	0.0256	**1.9552**
	RisingAttacK$_{30}$	0.9200	2.1465	556.2741	0.0222	4.1613
	RoHoAttacK-O$_{30}$	**0.9648**	**1.8423**	**477.2476**	**0.0197**	6.6581
Top-5	QuadAttacK$_{30}$	0.7794	3.2526	1024.0607	0.0225	**1.7718**
	RisingAttacK$_{30}$	0.8800	1.3450	334.9398	0.0149	2.2165
	RoHoAttacK-O$_{30}$	**0.9400**	**1.1527**	**286.3053**	**0.0129**	3.5464
Top-1	QuadAttacK$_{30}$	0.9994	1.3899	459.3060	0.0063	1.4404
	RisingAttacK$_{30}$	0.9772	0.5249	114.2980	0.0073	**0.6426**
	RoHoAttacK-O$_{30}$	**1.0000**	**0.4619**	**100.5822**	**0.0057**	2.3046

Theorem 1 (Projection-aligned active-loss maximization). *For a deterministic classifier and feasible iterate $x_i^\star$, let $\mathcal{C}_i^{\mathrm{RoHo}}$ be the finite candidate family in Eq. 10, with $0 \in \bar{\Gamma}_i$. Then every $z \in \mathcal{C}_i^{\mathrm{RoHo}}$ satisfies $z \in \mathcal{B}_p(x, \epsilon)$, and $x_{i+1}^\star \in \arg\max_{z \in \mathcal{C}_i^{\mathrm{RoHo}}} \mathcal{L}_K(z; y)$. Consequently,*

$$\mathcal{L}_K(x_{i+1}^\star; y) \geq \mathcal{L}_K(x_i^\star; y), \qquad \mathcal{L}_K(x_{i+1}^\star; y) \geq \max_{z \in \mathcal{C}_i^{\mathrm{sub}}} \mathcal{L}_K(z; y) \tag{24}$$

for any nonempty evaluated subset $\mathcal{C}_i^{\mathrm{sub}} \subseteq \mathcal{C}_i^{\mathrm{RoHo}}$.

Proof. See Appendix C.4. Since a positive active loss exactly encodes source displacement (Prop. 3), the robustness implication and source-preservation analysis are provided in Cor. 1 (App. C.5).

Restart selection. Different feasible initializations can expose different rival interactions during the search. We therefore repeat the procedure across $R \geq 1$ restarts and compare their endpoint losses. Here, $r \in \{1, \ldots, R\}$ indexes restarts, $\xi^{(r)} \in \mathbb{R}^D$ is an initialization offset, and $x_T^{\star,(r)}$ is the endpoint after T iterations of restart r. Following Algorithm 1, we set $\xi^{(1)} = 0$ for the clean initialization and sample the remaining offsets before projection:

$$\begin{aligned} x_0^{\star,(r)} &= \Pi_{\mathcal{B}_p(x,\epsilon)}(x + \xi^{(r)}) \quad \text{and} \quad r^\star \in \arg\max_{r \in \{1,\ldots,R\}} \mathcal{L}_K(x_T^{\star,(r)}; y), \\ x^\star &= x_T^{\star,(r^\star)}, \qquad \text{then} \qquad \delta^\star = x^\star - x. \end{aligned} \tag{25}$$

We retain the endpoint of the selected restart $r^\star$ as the final adversarial image $x^\star$, yielding the perturbation $\delta^\star = x^\star - x$. The active-loss criterion governs both iterative candidate and final restart selection. Search-cost analysis appears in App. C.6.

3 Results

Datasets. We evaluate on `CIFAR-10` (Krizhevsky, 2009), `ImageNet-1K` (Deng et al., 2009), `NIH-CXR` (Wang et al., 2017), `HAM10000` (Tschandl et al., 2018), `EyePACS` (Cuadros & Bresnick,

2009), and NCT-CRC-HE-100K (Kather et al., 2018). Following the sampling protocol of ordered baselines QuadAttacK (Paniagua et al., 2023) and RisingAttacK (Paniagua et al., 2025), we select 1,000 ImageNet validation images, one per class, correctly classified by all four models.

Models, defenses, and attack baselines. Classifiers include ResNet-50 (He et al., 2016), DenseNet-121 (Huang et al., 2017), ViT-B (Dosovitskiy, 2020), and DeiT-B (Touvron et al., 2021); defenses include RPF (Dong & Xu, 2023), CTRW (Ma et al., 2023), HyCAS (Dhar et al., 2026b), and HySCAN (Dhar et al., 2026a). We compare against diverse attacks, including PGD (Madry et al., 2018), APGD (including its DLR variant) (Croce & Hein, 2020b), C&W (Carlini & Wagner, 2017), DeepFool (Moosavi-Dezfooli et al., 2016), FAB (Croce & Hein, 2020a), APAA (Yuan et al., 2024), PMA (Xie et al., 2025), MALT (Melamed et al., 2024), Prompt2Perturb (Medghalchi et al., 2025), AutoAttack (AA) (Croce & Hein, 2020b), and adaptive attacks: EOT-PGD (Athalye et al., 2018b), BPDA and EOT-BPDA (Athalye et al., 2018b;a).

Evaluation details. Source-displacement and defense evaluations use an ℓ_∞ budget of $\epsilon = 8/255$, with non-robust evaluations at $16/255$. Consequently, PGD, APGD, and AA use 20 steps; EOT-PGD and EOT-BPDA average gradients over 10 stochastic realizations per step. RoHoAttack uses $K = 5$ active rivals. Following QuadAttacK and RisingAttacK, ordered comparisons use identical target lists and matched iteration counts of 30 or 60 for $K_{\text{ord}} \in \{1, 5, 10, 15, 20\}$. We report attack success rate (ASR), active rival margin M_K, robust accuracy, runtime, and ℓ_1, ℓ_2, and ℓ_∞ perturbation norms over successful attacks. Further details appear in App. D.

3.1 Performance comparisons

Adversarial attacks against non-robust models and robust defenses. Across non-robust CNN and transformer classifiers on the ImageNet-1K benchmark, RoHoAttack improves ASR by 3.9–4.9 points and active rival margins by 23.8%–29.9% over MALT, the leading baseline at $\epsilon = 8/255$ (Table 1). The advantage persists at $\epsilon = 16/255$, with ASR improvements of 3.2–4.6 points and rival-margin improvements of up to $\approx 30.8\%$ (Table 5; App. F). RoHoAttack exposes vulnerabilities in diverse defenses, including RPF, CTRW, HyCAS, and HySCAN, outperforming standard and adaptive attacks such as AutoAttack and EOT-BPDA across CIFAR-10, ImageNet-1K, NIH-CXR, NCT-CRC-HE-100K, HAM10000, and EyePACS. At $\epsilon = 8/255$, it reduces robust accuracy by up to ≈ 12.5 percentage points relative to the leading attack (see Fig. 3).

These improvements support jointly exploiting reciprocal rival-field responses and projected lookahead selection. Their coordination identifies useful interaction directions and evaluates their adversarial effect after enforcing the perturbation constraint, strengthening attacks over the individual component variants on ImageNet-1K (Tables 10 and 12; App. G).

Ordered top-K attack effectiveness. RoHoAttack variants extend these improvements to ordered top-K attacks on ImageNet-1K, denoted RoHoAttack-O, with subscripts indicating step budgets. Compared with the leading baseline in each setting among QuadAttacK and RisingAttacK, RoHoAttack-O_{30} and RoHoAttack-O_{60} improve mean ASR by up to ≈ 18.0 and ≈ 7.9 points, respectively, with pronounced 30-step improvements at Top-20 and Top-15. Higher success accompanies smaller perturbations: maximum reductions in mean ℓ_1, ℓ_2, and ℓ_∞ norms are $\approx 18.3\%$, $\approx 18.4\%$, and $\approx 15.3\%$ at 30 steps and $\approx 13.3\%$, $\approx 13.5\%$, and $\approx 12.3\%$ at 60 steps, relative to the lowest corresponding baseline values (Tables 2 and 4; App. F).

These results support exploiting rival-field interactions when attack success requires coordinating multiple ranked outputs. With target ordering fixed, interaction-aware directions guide exploration to coordinate ranked outputs, while projected lookahead evaluates the achieved ordered objective; ablations support their complementary roles (Table 14; App. G). Best-, mean-, and worst-case comparisons across sampled target lists extend evaluation beyond average ASR (Tables 6–9; App. F). *More results, ablations, and discussion are provided in Apps. F, G, and H, respectively.*

4 Conclusion

We introduced RoHoAttack, a new white-box attack combining active rival selection, reciprocal probing, residual refinement, and projected lookahead under ℓ_p constraints. Across diverse benchmarks, it strengthens attacks on non-robust models and reduces robust accuracy by up to ≈ 12.5 points. Ordered top-K variants improve mean ASR by up to ≈ 18.0 points with smaller perturbations. Future work will explore adaptive pair pruning and candidate batching for faster evaluation.

AI USE STATEMENT

We selected few sentences at a time and asked GPT to check the grammar.

REFERENCES

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 274–283. PMLR, 2018a. URL https://proceedings.mlr.press/v80/athalye18a.html.

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 284–293. PMLR, 2018b. URL https://proceedings.mlr.press/v80/athalye18b.html.

Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, 2017. doi: 10.1109/SP.2017.49.

Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. HopSkipJumpAttack: A query-efficient decision-based attack. In *2020 IEEE Symposium on Security and Privacy (SP)*, pp. 1277–1294. IEEE, 2020.

Pin-Yu Chen and Sijia Liu. Holistic adversarial robustness of deep learning models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 15411–15420, 2023.

Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2196–2205. PMLR, 2020a. URL https://proceedings.mlr.press/v119/croce20a.html.

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, 2020b.

Jorge Cuadros and George Bresnick. EyePACS: An adaptable telemedicine system for diabetic retinopathy screening. *Journal of Diabetes Science and Technology*, 3(3):509–516, 2009.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Joy Dhar, Manish Kumar Pandey, Behzad Bozorgtabar, Nayyar Zaidi, Wenyu Zhang, Wei-Hong Li, Tingting Mu, Dwarikanath Mahapatra, Mahsa Baktashmotlagh, Trung Le, et al. Does a hybrid space-aware randomized defense improve empirical and certified adversarial robustness? In *Forty-third International Conference on Machine Learning*, 2026a.

Joy Dhar, Song Xia, Manish Kumar Pandey, Maryam Haghighat, Azadeh Alavi, Ferdous Sohel, Wenyu Zhang, and Nayyar Zaidi. Certified vs. empirical adversarial robustness via hybrid convolutions with attention stochasticity. In *International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=sYk9GaFHEf.

Minjing Dong and Chang Xu. Adversarial robustness via random projection filters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4077–4086, 2023.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6572.

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.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 2261–2269. Ieee, 2017.

Kaggle and EyePACS. Diabetic retinopathy detection. Kaggle competition dataset, 2015. URL https://www.kaggle.com/c/diabetic-retinopathy-detection.

Jakob Nikolas Kather, Niels Halama, and Alexander Marx. 100,000 histological images of human colorectal cancer and healthy tissue, April 2018. URL https://doi.org/10.5281/zenodo.1214456.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Yanxiang Ma, Minjing Dong, and Chang Xu. Adversarial robustness through random weight sampling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Yasamin Medghalchi, Moein Heidari, Clayton Allard, Leonid Sigal, and Ilker Hacihaliloglu. Prompt2perturb (p2p): Text-guided diffusion-based adversarial attacks on breast ultrasound images. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28564–28574. IEEE, 2025.

Odelia Melamed, Gilad Yehudai, and Adi Shamir. Malt powers up adversarial attacks. *Advances in Neural Information Processing Systems*, 37:44366–44392, 2024.

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. doi: 10.1109/CVPR.2016.282.

Richard M. Murray and S. Shankar Sastry. Nonholonomic motion planning: Steering using sinusoids. *IEEE Transactions on Automatic Control*, 38(5):700–716, May 1993. doi: 10.1109/9.277235.

Thomas Paniagua, Ryan Grainger, and Tianfu Wu. Quadattack: A quadratic programming approach to learning ordered top-k adversarial attacks. In *Advances in Neural Information Processing Systems*, volume 36, pp. 64112–64143, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/ca06d48a3ccce45e97fbcf4467026c30-Abstract-Conference.html.

Thomas Paniagua, Chinmay Savadikar, and Tianfu Wu. Adversarial perturbations are formed by iteratively learning linear combinations of the right singular vectors of the adversarial jacobian. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*. PMLR, 2025. URL https://proceedings.mlr.press/v267/paniagua25a.html.

Ali Rahmati, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, and Huaiyu Dai. GeoDA: A geometric framework for black-box adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8446–8455, 2020.

Md Farhamdur Reza, Ali Rahmati, Tianfu Wu, and Huaiyu Dai. CGBA: Curvature-aware geometric black-box attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 124–133, 2023.

Md Farhamdur Reza, Richeng Jin, Tianfu Wu, and Huaiyu Dai. GSBAK: Top-k geometric score-based black-box attack. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=3vELOeAhqP.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115 (3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357. PMLR, 2021.

Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The HAM10000 dataset: A large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 5: 180161, 2018. doi: 10.1038/sdata.2018.161.

Nurislam Tursynbek, Aleksandr Petiushko, and Ivan Oseledets. Geometry-inspired top-k adversarial perturbations. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3398–3407, 2022. URL https://openaccess.thecvf.com/content/WACV2022/html/Tursynbek_Geometry-Inspired_Top-K_Adversarial_Perturbations_WACV_2022_paper.html.

Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. ChestX-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3462–3471, 2017. doi: 10.1109/CVPR.2017.369. Dataset: NIH ChestX-ray (ChestX-ray8/14).

Yong Xie, Weijie Zheng, Hanxun Huang, Guangnan Ye, and Xingjun Ma. Towards million-scale adversarial robustness evaluation with stronger individual attacks. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30702–30711. IEEE, 2025.

Zheng Yuan, Jie Zhang, Zhaoyan Jiang, Liangliang Li, and Shiguang Shan. Adaptive perturbation for adversarial attack. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8): 5663–5676, 2024. doi: 10.1109/TPAMI.2024.3367773.

Zekun Zhang and Tianfu Wu. Learning ordered top-k adversarial attacks via adversarial distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 776–777, 2020.

APPENDIX

A RELATED WORK

A.1 WHITE-BOX ADVERSARIAL OPTIMIZATION

White-box attacks exploit model gradients to manipulate predictions. `PGD` (Madry et al., 2018) performs projected loss ascent, `C&W` (Carlini & Wagner, 2017) balances classification margins and perturbation size, and `DeepFool` (Moosavi-Dezfooli et al., 2016) and `FAB` (Croce & Hein, 2020a) exploit local boundary geometry. `APGD` adapts optimization steps, while `AutoAttack` combines complementary attacks (Croce & Hein, 2020b). Recent approaches improve perturbation scaling (`APAA` (Yuan et al., 2024)), probability-margin objectives (`PMA` (Xie et al., 2025)), and target prioritization (`MALT` (Melamed et al., 2024)). Although `MALT` fixes its initial target priorities, its APGD-DLR backbone uses current logit rankings. These methods exploit class competition through scores, gradients, or boundaries; `RoHoAttack` investigates reciprocal changes between rival margin-gradient fields.

A.2 TOP-K AND ORDERED ADVERSARIAL ATTACKS

Untargeted top-K attacks exclude the source class from the leading predictions (Tursynbek et al., 2022), whereas ordered attacks enforce a target class sequence (Zhang & Wu, 2020). `QuadAttacK` (Paniagua et al., 2023) uses feature-space quadratic programming, while `RisingAttacK` (Paniagua et al., 2025) constructs image-space perturbations through the adversarial Jacobian and updates competing non-target classes during optimization. Both include single-target attacks at $K_{\text{ord}} = 1$. `RoHoAttack-O` uses reciprocal rival-gradient responses to guide exploration while retaining the target sequence and evaluating the projected ordered objective.

A.3 EVALUATION DOMAINS AND ROBUST DEFENSES

Most compared attacks emphasize natural-image benchmarks, whereas `Prompt2Perturb` (Medghalchi et al., 2025) uses text-guided diffusion for medical-image attacks. Domain coverage is distinct from evaluating defended models, which requires accounting for the defended computation. Expectation over transformation (EOT) (Athalye et al., 2018b) estimates gradients across stochastic realizations, while backward pass differentiable approximation (BPDA) (Athalye et al., 2018a) supplies surrogate backward derivatives. Our evaluation spans natural and medical images and compares standard and adaptive attacks against `RPF` (Dong & Xu, 2023), `CTRW` (Ma et al., 2023), `HyCAS` (Dhar et al., 2026b), and `HySCAN` (Dhar et al., 2026a).

A.4 RESEARCH GAPS

Table 3 highlights three gaps motivating our study. First, existing attacks exploit margins, boundaries, or ranking constraints, while reciprocal rival-gradient responses offer additional information for direction generation. `RoHoAttack` probes these responses, refines their residual, and evaluates the resulting candidates through projected lookahead, connecting interaction-aware exploration with the adversarial effect achieved under the perturbation constraint.

Second, evaluation coverage remains uneven across domains. Most compared studies emphasize natural images, whereas `Prompt2Perturb` focuses on medical images, leaving limited evidence within these studies from a common evaluation spanning both domains and defended models. We evaluate `RoHoAttack` on two natural-image and four medical-image benchmarks, covering non-robust classifiers and robust defenses through standard and adaptive attack comparisons.

Third, source displacement and ordered prediction manipulation require different objectives, motivating investigation of whether the same interaction-based search can support both. `RoHoAttack` uses active rivals to guide exploration without changing the source-displacement objective; `RoHoAttack-O` retains a fixed target sequence and evaluates the achieved ordered objective. Its ordered evaluation covers ImageNet, leaving extension to defended models and medical images as an open direction.

Table 3: **Comparison of attack objectives, rival modeling, and evaluation coverage.** ✓ indicates capabilities or evaluations reported in the cited studies; ✗ indicates otherwise. Dynamic rival modeling includes updated competitor identities, score rankings, or boundary choices. Our entries reflect experiments reported in this paper.

	Attack objectives		Rival modeling		Reported evaluation		
Method	**Top-1 displacement**	**Ordered top-K ($K_{\text{ord}} > 1$)**	**Dynamic rival modeling**	**Reciprocal rival-gradient interactions**	**Defended models**	**Natural images**	**Medical images**
Top-1 attacks and attack ensembles							
`PGD (CE)` (Madry et al., 2018)	✓	✗	✗	✗	✓	✓	✗
`APGD-DLR` (Croce & Hein, 2020b)	✓	✗	✓	✗	✓	✓	✗
`C&W` (Carlini & Wagner, 2017)	✓	✗	✓	✗	✓	✓	✗
`DeepFool` (Moosavi-Dezfooli et al., 2016)	✓	✗	✓	✗	✓	✓	✗
`FAB` (Croce & Hein, 2020a)	✓	✗	✓	✗	✓	✓	✗
`APAA` (Yuan et al., 2024)	✓	✗	✗	✗	✓	✓	✗
`PMA` (Xie et al., 2025)	✓	✗	✓	✗	✓	✓	✗
`MALT` (Melamed et al., 2024)	✓	✗	✓	✗	✓	✓	✗
`Prompt2Perturb` (Medghalchi et al., 2025)	✓	✗	✗	✗	✗	✗	✓
`AutoAttack` (Croce & Hein, 2020b)	✓	✗	✓	✗	✓	✓	✗
Ordered top-K attacks							
`QuadAttacK` (Paniagua et al., 2023)	✓	✓	✗	✗	✗	✓	✗
`RisingAttacK` (Paniagua et al., 2025)	✓	✓	✓	✗	✗	✓	✗
Our attacks							
`RoHoAttack`	✓	✗	✓	✓	✓	✓	✓
`RoHoAttack-O`	✓	✓	✓	✓	✗	✓	✗

B Additional Method

B.1 Active Loss and Source Displacement

We characterize successful source displacement among candidates that satisfy the image-domain and perturbation constraints.

Proposition 3 (Active loss exactly encodes source displacement)**.** *Let $x^\star \in \mathcal{B}_p(x, \epsilon)$ be a feasible candidate and $K \in \{1, \dots, C-1\}$. The active loss in Eq. 5 satisfies*

$$\mathcal{L}_K(x^\star; y) = \max_{a \in \mathcal{Y} \setminus \{y\}} \left[\ell_a(x^\star) - \ell_y(x^\star)\right]. \tag{26}$$

This candidate belongs to the source-displacement region exactly when its active loss is positive:

$$x^\star \in \mathcal{S}_p(x, y) \quad \Longleftrightarrow \quad \mathcal{L}_K(x^\star; y) > 0. \tag{27}$$

Proof. By definition, $\mathcal{A}_K(x^\star; y)$ contains the indices of the K largest non-source logits. Since $K \geq 1$, this set contains at least one class attaining the largest non-source logit, including when several classes tie. Therefore,

$$\max_{a \in \mathcal{A}_K(x^\star; y)} \ell_a(x^\star) = \max_{a \in \mathcal{Y} \setminus \{y\}} \ell_a(x^\star).$$

Subtracting the source logit $\ell_y(x^\star)$ proves Eq. 26.

Consequently, $\mathcal{L}_K(x^\star; y) > 0$ holds exactly when some non-source class a satisfies $\ell_a(x^\star) > \ell_y(x^\star)$. Since $x^\star$ is feasible by assumption, this is precisely the defining condition for $x^\star \in \mathcal{S}_p(x, y)$, proving Eq. 27. □

At any feasible candidate, the loss identity is independent of K; active-rival selection therefore controls candidate-direction generation rather than the scalar objective. A tie with the source class alone does not satisfy the strict success criterion. The pairwise generator requires $K \geq 2$, whereas the loss identity also holds for $K = 1$.

B.2 Constrained Adversarial Optimization

We provide the perturbation norm, the minimum-perturbation formulation, and the correspondence between image-space and perturbation-space search. The notation follows Section 2.1; the dependence of $\mathcal{S}_p(x, y)$ on the radius ϵ is implicit.

Perturbation norm. For $\delta \in \mathbb{R}^D$, let $(\delta)_j$ denote its j-th coordinate. We measure perturbation size by

$$\|\delta\|_p = \begin{cases} \left(\sum_{j=1}^{D} |(\delta)_j|^p\right)^{1/p}, & 1 \leq p < \infty, \\ \max_{1 \leq j \leq D} |(\delta)_j|, & p = \infty. \end{cases} \tag{28}$$

The index j refers to an input coordinate, whereas i in δ_i indexes attack iterations.

Minimum-perturbation formulation. Using this measure, we express minimum-perturbation search as

$$\min_{\delta \in \mathbb{R}^D} \|\delta\|_p \quad \text{s.t.} \quad x^\star = x + \delta \in [l, u]^D, \quad \mathcal{L}_K(x^\star; y) > 0. \tag{29}$$

The constraint requires successful source displacement, while the objective measures the perturbation needed to achieve it. Because success is expressed by a strict inequality, the optimal value is interpreted as an infimum if a minimum is not attained. For a given radius ϵ, Eq. 7 instead maximizes the active rival loss over $\Delta_p(x, \epsilon)$, which is the objective used by the projected search.

Image-space and perturbation-space correspondence. The substitution $x^\star = x + \delta$ relates the two feasible regions:

$$\delta \in \Delta_p(x, \epsilon) \quad \Longleftrightarrow \quad x + \delta \in \mathcal{B}_p(x, \epsilon). \tag{30}$$

It also preserves the loss, so the image-space and perturbation-space formulations have identical feasible objective values.

Proof. If $\delta \in \Delta_p(x, \epsilon)$, then $x + \delta \in [l, u]^D$ and $\|\delta\|_p \leq \epsilon$. Setting $x^\star = x + \delta$ gives $\|x^\star - x\|_p = \|\delta\|_p \leq \epsilon$, hence $x^\star \in \mathcal{B}_p(x, \epsilon)$.

Conversely, for any $x^\star \in \mathcal{B}_p(x, \epsilon)$, the perturbation $\delta = x^\star - x$ satisfies $x + \delta = x^\star \in [l, u]^D$ and $\|\delta\|_p \leq \epsilon$, hence $\delta \in \Delta_p(x, \epsilon)$. These substitutions are inverses, establishing a one-to-one correspondence. Since $\mathcal{L}_K(x^\star; y) = \mathcal{L}_K(x + \delta; y)$, corresponding candidates also have identical loss values. □

Accordingly, the main-text objective can equivalently be written as

$$\max_{x^\star \in \mathcal{B}_p(x, \epsilon)} \mathcal{L}_K(x^\star; y). \tag{31}$$

For $\delta \in \Delta_p(x, \epsilon)$, the corresponding image is feasible, so Proposition 3 gives

$$x + \delta \in \mathcal{S}_p(x, y) \quad \Longleftrightarrow \quad \mathcal{L}_K(x + \delta; y) > 0.$$

Thus, membership in $\Delta_p(x, \epsilon)$ establishes feasibility, while a positive active loss establishes successful source displacement.

C PROOFS FOR ROHOATTACK

All proofs are stated with respect to the active rival loss $\mathcal{L}_K$, the source-displacement region $\mathcal{S}_p(x, y)$, the active pair policy $\mathcal{P}_K$, the cross-response residual $H_{a,b}$, and the projection-aligned lookahead update defined in Section 2.

C.1 PROOF OF PROPOSITION 1

Proof. At iteration i, the active rival set $\mathcal{A}_K(x_i^\star; y)$ contains exactly K non-source classes. The active pair policy is defined as

$$\mathcal{P}_K(x_i^\star; y) = \{(a, b) : a, b \in \mathcal{A}_K(x_i^\star; y),\ a < b\}\,. \tag{32}$$

Thus, $\mathcal{P}_K(x_i^\star; y)$ is the set of unordered pairs drawn from K elements. Its cardinality is

$$|\mathcal{P}_K(x_i^\star; y)| = \binom{K}{2}. \tag{33}$$

If pairwise interactions were constructed over all non-source classes, the corresponding non-source class pool would have cardinality $C - 1$, and the number of unordered all-class non-source pairs would be

$$\binom{C-1}{2}. \tag{34}$$

Therefore, active-pair generation reduces the pairwise direction-generation search by the factor

$$\frac{\binom{K}{2}}{\binom{C-1}{2}}. \tag{35}$$

It remains to show that this active-pair restriction does not alter the active loss evaluated on projected candidates. For any projected candidate z, Proposition 3 gives

$$\mathcal{L}_K(z; y) = \max_{a \in \mathcal{Y} \setminus \{y\}} [\ell_a(z) - \ell_y(z)]\,. \tag{36}$$

Hence, although $\mathcal{P}_K$ restricts which pairwise interaction directions are generated, the loss used to score every projected candidate remains the exact global source-vs-rival margin loss. Therefore, active-pair search reduces the direction-generation space without changing the source-displacement objective used for candidate selection. □

C.2 PROOF OF PROPOSITION 2

Proof. Let $x = x_i^\star$ for compactness. By assumption, the projection in Eq. 14 is locally inactive for the two probes. Therefore, for sufficiently small h,

$$x_i^a = x + hB_a(x), \qquad x_i^b = x + hB_b(x). \tag{37}$$

Since U_b is differentiable at x, its first-order Taylor expansion along $B_a(x)$ gives

$$U_b(x + hB_a(x)) = U_b(x) + hJ_{U_b}(x)B_a(x) + r_b(h), \tag{38}$$

where $\|r_b(h)\|/h \to 0$ as $h \to 0$. Similarly,

$$U_a(x + hB_b(x)) = U_a(x) + hJ_{U_a}(x)B_b(x) + r_a(h), \tag{39}$$

where $\|r_a(h)\|/h \to 0$ as $h \to 0$.

Substituting these expansions into the finite-difference residual in Eq. 15 gives

$$H_{a,b}(x) = \frac{U_b(x + hB_a(x)) - U_b(x)}{h} - \frac{U_a(x + hB_b(x)) - U_a(x)}{h} \tag{40}$$

$$= \frac{hJ_{U_b}(x)B_a(x) + r_b(h)}{h} - \frac{hJ_{U_a}(x)B_b(x) + r_a(h)}{h} \tag{41}$$

$$= J_{U_b}(x)B_a(x) - J_{U_a}(x)B_b(x) + \frac{r_b(h) - r_a(h)}{h}. \tag{42}$$

Since $\|r_b(h)\|/h \to 0$ and $\|r_a(h)\|/h \to 0$, the final term is $o(1)$. If the Jacobians J_{U_a} and J_{U_b} are locally Lipschitz, the Taylor remainders satisfy $r_a(h) = \mathcal{O}(h^2)$ and $r_b(h) = \mathcal{O}(h^2)$, and hence

$$H_{a,b}(x) = J_{U_b}(x)B_a(x) - J_{U_a}(x)B_b(x) + \mathcal{O}(h). \tag{43}$$

This proves Eq. 16 under the standard finite-difference smoothness condition. Under the weaker assumption of differentiability alone, the same statement holds with an $o(1)$ remainder.

For the specialization $B_r(x) = U_r(x)$, $r \in \{a, b\}$, with locally inactive probe projections, the expansion becomes

$$H_{a,b}(x) = J_{U_b}(x)U_a(x) - J_{U_a}(x)U_b(x) + \mathcal{O}(h). \tag{44}$$

By definition of the Lie bracket for vector fields,

$$[U_a, U_b](x) = J_{U_b}(x)U_a(x) - J_{U_a}(x)U_b(x). \tag{45}$$

Therefore,

$$H_{a,b}(x) = [U_a, U_b](x) + \mathcal{O}(h), \tag{46}$$

which proves Eq. 17. □

C.3 Proof of Lemma 1

Proof. First consider the exact projection case $\rho = 0$. For any nonzero vector u, the Euclidean projection of v onto u is

$$\mathrm{Proj}_u(v) = \frac{\langle v, u\rangle}{\|u\|_2^2} u. \tag{47}$$

By construction,

$$U_b^{\perp} = U_b - \mathrm{Proj}_{U_a}(U_b). \tag{48}$$

Therefore,

$$\langle U_b^{\perp}, U_a\rangle = \langle U_b, U_a\rangle - \left\langle \frac{\langle U_b, U_a\rangle}{\|U_a\|_2^2} U_a, U_a \right\rangle \tag{49}$$

$$= \langle U_b, U_a\rangle - \frac{\langle U_b, U_a\rangle}{\|U_a\|_2^2}\|U_a\|_2^2 \tag{50}$$

$$= 0. \tag{51}$$

Thus, $U_b^{\perp}$ is orthogonal to U_a.

The refined residual is

$$\widetilde{H}_{a,b} = H_{a,b} - \mathrm{Proj}_{U_a}(H_{a,b}) - \mathrm{Proj}_{U_b^{\perp}}(H_{a,b}). \tag{52}$$

Taking the inner product with U_a gives

$$\left\langle \widetilde{H}_{a,b}, U_a \right\rangle = \langle H_{a,b}, U_a\rangle - \left\langle \mathrm{Proj}_{U_a}(H_{a,b}), U_a \right\rangle - \left\langle \mathrm{Proj}_{U_b^{\perp}}(H_{a,b}), U_a \right\rangle. \tag{53}$$

The first two terms cancel because $\mathrm{Proj}_{U_a}(H_{a,b})$ is the Euclidean projection of $H_{a,b}$ onto U_a. The third term is zero because $\mathrm{Proj}_{U_b^{\perp}}(H_{a,b})$ is parallel to $U_b^{\perp}$, and $U_b^{\perp} \perp U_a$. Hence,

$$\left\langle \widetilde{H}_{a,b}, U_a \right\rangle = 0. \tag{54}$$

Similarly, taking the inner product with $U_b^{\perp}$ gives

$$\left\langle \widetilde{H}_{a,b}, U_b^{\perp} \right\rangle = \left\langle H_{a,b}, U_b^{\perp} \right\rangle - \left\langle \mathrm{Proj}_{U_a}(H_{a,b}), U_b^{\perp} \right\rangle - \left\langle \mathrm{Proj}_{U_b^{\perp}}(H_{a,b}), U_b^{\perp} \right\rangle. \tag{55}$$

The middle term is zero because $\mathrm{Proj}_{U_a}(H_{a,b})$ is parallel to U_a and $U_a \perp U_b^{\perp}$. The first and third terms cancel because $\mathrm{Proj}_{U_b^{\perp}}(H_{a,b})$ is the Euclidean projection of $H_{a,b}$ onto $U_b^{\perp}$. Therefore,

$$\left\langle \widetilde{H}_{a,b}, U_b^{\perp} \right\rangle = 0. \tag{56}$$

For $\rho > 0$, the projection operator

$$\mathrm{Proj}_u^\rho(v) = \frac{\langle v, u\rangle}{\|u\|_2^2 + \rho} u \tag{57}$$

is a regularized projection. For any fixed nonzero u and v,

$$\mathrm{Proj}_u^\rho(v) \to \mathrm{Proj}_u(v) \qquad \text{as } \rho \to 0. \tag{58}$$

Thus, the regularized refinement converges continuously to the exact orthogonal residual separation as $\rho \to 0$. For positive ρ, the refinement is numerically stabilized by avoiding division by small squared norms, while preserving the same residual-separation principle up to the regularization error. □

C.4 Proof of Theorem 1

Proof. We prove each property in order.

Feasibility. By definition,

$$\mathcal{C}_i^{\mathrm{RoHo}} = \left\{ \Pi_{\mathcal{B}_p(x,\epsilon)} \left(x_i^\star + \gamma \epsilon \psi(d) \right) : d \in \mathcal{D}_K^{\mathrm{H}}(x_i^\star; y),\, \gamma \in \bar{\Gamma}_i \right\} \cup \{x_i^\star\}. \tag{59}$$

Every nontrivial lookahead candidate is explicitly mapped by $\Pi_{\mathcal{B}_p(x,\epsilon)}$ into $\mathcal{B}_p(x,\epsilon)$. Moreover, $x_i^\star \in \mathcal{B}_p(x,\epsilon)$ by the definition of the current projected iterate. Therefore,

$$z \in \mathcal{C}_i^{\mathrm{RoHo}} \quad \Rightarrow \quad z \in \mathcal{B}_p(x,\epsilon). \tag{60}$$

Candidate-wise optimality. The update rule defines

$$x_{i+1}^\star = \arg\max_{z \in \mathcal{C}_i^{\mathrm{RoHo}}} \mathcal{L}_K(z; y). \tag{61}$$

Since $\mathcal{C}_i^{\mathrm{RoHo}}$ is finite and nonempty, the maximum is attained. Thus, $x_{i+1}^\star$ is candidate-wise optimal over the evaluated projected candidate family.

Monotone evaluated loss progress. The candidate family contains the current iterate $x_i^\star$ explicitly. Equivalently, the zero step $\gamma = 0$ also maps each direction to the current iterate after projection. Therefore,

$$x_i^\star \in \mathcal{C}_i^{\mathrm{RoHo}}. \tag{62}$$

Since $x_{i+1}^\star$ maximizes $\mathcal{L}_K$ over $\mathcal{C}_i^{\mathrm{RoHo}}$,

$$\mathcal{L}_K(x_{i+1}^\star; y) = \max_{z \in \mathcal{C}_i^{\mathrm{RoHo}}} \mathcal{L}_K(z; y) \geq \mathcal{L}_K(x_i^\star; y). \tag{63}$$

This proves monotone evaluated loss progress.

Dominance over ablated candidate subsets. Let $\mathcal{C}_i^{\mathrm{abl}} \subseteq \mathcal{C}_i^{\mathrm{RoHo}}$ be any nonempty evaluated ablation candidate family. Because $\mathcal{C}_i^{\mathrm{abl}}$ is a subset of $\mathcal{C}_i^{\mathrm{RoHo}}$,

$$\max_{z \in \mathcal{C}_i^{\mathrm{RoHo}}} \mathcal{L}_K(z; y) \geq \max_{z \in \mathcal{C}_i^{\mathrm{abl}}} \mathcal{L}_K(z; y). \tag{64}$$

Using the candidate-wise optimality identity,

$$\mathcal{L}_K(x_{i+1}^\star; y) = \max_{z \in \mathcal{C}_i^{\mathrm{RoHo}}} \mathcal{L}_K(z; y), \tag{65}$$

we obtain

$$\mathcal{L}_K(x_{i+1}^\star; y) \geq \max_{z \in \mathcal{C}_i^{\mathrm{abl}}} \mathcal{L}_K(z; y). \tag{66}$$

This completes the proof. □

C.5 Theory and proof of Corollary 1

Corollary 1 (Active-loss crossing implies robust-performance reduction)**.** *For a feasible endpoint $x_T^\star$, $\mathcal{L}_K(x_T^\star; y) > 0$ implies $x_T^\star \in \mathcal{S}_p(x, y)$. For attacks A and B evaluated on the same classifier and dataset $\{(x_n, y_n)\}_{n=1}^N$, define*

$$\mathrm{RP}(\mathsf{A}) = \frac{1}{N}\sum_{n=1}^{N} \mathbf{1}\left[\mathcal{L}_K(x_{\mathsf{A},n}^\star; y_n) \le 0\right]. \tag{67}$$

If $\mathcal{L}_K(x_{\mathsf{A},n}^\star; y_n) \ge \mathcal{L}_K(x_{\mathsf{B},n}^\star; y_n)$ for every n, then $\mathrm{RP}(\mathsf{A}) \le \mathrm{RP}(\mathsf{B})$.

Proof. First, suppose RoHoAttacK returns $x_T^\star$ with

$$\mathcal{L}_K(x_T^\star; y) > 0. \tag{68}$$

Since RoHoAttacK always returns a projected candidate, $x_T^\star \in \mathcal{B}_p(x, \epsilon)$. By Proposition 3,

$$\mathcal{L}_K(x_T^\star; y) > 0 \quad \Longleftrightarrow \quad x_T^\star \in \mathcal{S}_p(x, y). \tag{69}$$

Thus, the source class is displaced by at least one non-source rival.

Now consider two attacks A and B evaluated on the same dataset $\{(x_n, y_n)\}_{n=1}^N$. Assume that for every test example n,

$$\mathcal{L}_K(x_{\mathsf{A},n}^\star; y_n) \ge \mathcal{L}_K(x_{\mathsf{B},n}^\star; y_n). \tag{70}$$

For any example n, if

$$\mathcal{L}_K(x_{\mathsf{A},n}^\star; y_n) \le 0, \tag{71}$$

then the pointwise dominance assumption implies

$$\mathcal{L}_K(x_{\mathsf{B},n}^\star; y_n) \le \mathcal{L}_K(x_{\mathsf{A},n}^\star; y_n) \le 0. \tag{72}$$

Therefore,

$$\mathbf{1}\left[\mathcal{L}_K(x_{\mathsf{A},n}^\star; y_n) \le 0\right] \le \mathbf{1}\left[\mathcal{L}_K(x_{\mathsf{B},n}^\star; y_n) \le 0\right]. \tag{73}$$

Summing over $n = 1, \ldots, N$ and dividing by N gives

$$\frac{1}{N}\sum_{n=1}^{N} \mathbf{1}\left[\mathcal{L}_K(x_{\mathsf{A},n}^\star; y_n) \le 0\right] \le \frac{1}{N}\sum_{n=1}^{N} \mathbf{1}\left[\mathcal{L}_K(x_{\mathsf{B},n}^\star; y_n) \le 0\right]. \tag{74}$$

Using the definition of empirical source-preservation performance,

$$\mathrm{RP}(\mathsf{A}) \le \mathrm{RP}(\mathsf{B}). \tag{75}$$

The proof is complete. □

C.6 Cost of Active-Pair Projected Search

This appendix subsection gives the complete asymptotic cost analysis for RoHoAttacK. It can be cited from the main text or discussion when describing the robustness–cost trade-off.

Let $P = \binom{K}{2}$ denote the number of active rival pairs, $S = |\bar{\Gamma}_i|$ the number of signed lookahead steps including zero, T the number of attack iterations, and R the number of restarts. Let F and B denote the cost of one forward and one backward evaluation of the attacked model, respectively. Let $\Pi_p(D)$ denote the cost of projecting one candidate into the ℓ_p ball. For ℓ_2 and ℓ_∞ projection, $\Pi_p(D) = \mathcal{O}(D)$; for exact ℓ_1 projection, a sorting-based implementation gives $\Pi_1(D) = \mathcal{O}(D \log D)$.

Proposition 4 (Cost of active-pair projected loss search)**.** *At each iteration, RoHoAttacK evaluates*

$$|\mathcal{C}_i^{\mathrm{RoHo}}| \le 1 + PS = 1 + \binom{K}{2} S \tag{76}$$

projected candidates. A conservative asymptotic cost for R restarts and T iterations is

$$\mathcal{O}\left(RT\left[C \log K + \mathsf{F} + K\mathsf{B} + 2P\mathsf{B} + PD + PS\left(\mathsf{F} + C \log K + \Pi_p(D)\right)\right]\right). \tag{77}$$

For ℓ_2 and ℓ_∞ projection, $\Pi_p(D) = \mathcal{O}(D)$, so the dominant attack-specific dependence is

$$\mathcal{O}\left(RTK^2\left[\mathsf{B} + S\mathsf{F} + SC \log K + SD\right]\right), \qquad K \ll C. \tag{78}$$

Proof. At one iteration, active rival extraction requires one forward evaluation of the classifier and a top-K selection over C logits. This contributes

$$\mathcal{O}(\mathsf{F} + C \log K). \tag{79}$$

Computing the normalized active margin fields for the K active rivals requires approximately K backward evaluations, contributing

$$\mathcal{O}(K\mathsf{B}). \tag{80}$$

For each active pair $(a, b) \in \mathcal{P}_K(x_i^\star; y)$, RoHoAttacK evaluates the cross-responses $U_b(x_i^a)$ and $U_a(x_i^b)$ in the finite-difference cross-response residual. A conservative implementation therefore requires two additional backward evaluations per active pair, giving

$$\mathcal{O}(2P\mathsf{B}). \tag{81}$$

Residual construction and refinement are performed in input space. Since each residual vector has dimension D, this contributes

$$\mathcal{O}(PD). \tag{82}$$

Projected lookahead selection evaluates S signed steps for each of the P active pairs. Each projected candidate requires projection into the ℓ_p ball, a forward evaluation for logits, and a top-K extraction to compute $\mathcal{L}_K$. Thus the lookahead scoring cost is

$$\mathcal{O}\left(PS\left(\mathsf{F} + C \log K + \Pi_p(D)\right)\right). \tag{83}$$

Combining these terms gives the per-iteration cost

$$\mathcal{O}\left(C \log K + \mathsf{F} + K\mathsf{B} + 2P\mathsf{B} + PD + PS\left(\mathsf{F} + C \log K + \Pi_p(D)\right)\right). \tag{84}$$

Multiplying by T attack iterations and R restarts yields Eq. 77.

The candidate count bound follows directly from the construction of $\mathcal{C}_i^{\mathrm{RoHo}}$. There are P active pairs and at most S signed steps per pair, together with the explicitly included current iterate. Therefore,

$$|\mathcal{C}_i^{\mathrm{RoHo}}| \le 1 + PS = 1 + \binom{K}{2} S. \tag{85}$$

Finally, for ℓ_2 and ℓ_∞ projection, $\Pi_p(D) = \mathcal{O}(D)$, and since

$$P = \binom{K}{2} = \mathcal{O}(K^2), \tag{86}$$

the dominant attack-specific dependence becomes

$$\mathcal{O}\left(RTK^2\left[\mathsf{B} + S\mathsf{F} + SC \log K + SD\right]\right). \tag{87}$$

Because $K \ll C$ in the intended active-rival regime, the pairwise interaction search is controlled by the active set size rather than by all-class pair enumeration. □

D Experimental Setup

Models under attack. We evaluate RoHoAttacK under a white-box threat model in which the attacked classifier is fixed and the adversary has access to logits, model parameters, and input gradients. For architecture-wise vulnerability analysis, we attack representative convolutional and transformer-based classifiers, including ResNet-50, DenseNet-121, ViT-B, and DEiT-B. This model set covers both local convolutional inductive biases and global self-attention architectures, and follows the model family used in recent ordered top-K ImageNet-scale attack evaluations (Paniagua et al., 2023; 2025). For robust-defense evaluation, we further attack representative randomized and robust architectures, including RPF, CTRW, HyCAS, and HySCAN. All models are evaluated in inference mode unless the corresponding defense explicitly requires stochastic inference; in that case, the defense randomness remains active during both attack generation and final evaluation. ***Results are reported as the mean across five different seeds.***

Data and attack targets. We evaluate RoHoAttacK on both natural-image and medical-domain benchmarks. The natural-image benchmarks include CIFAR-10 (Krizhevsky, 2009) and ImageNet-1K (Deng et al., 2009; Russakovsky et al., 2015), while the medical-domain benchmarks include NIH-CXR (Wang et al., 2017), HAM10000 (Tschandl et al., 2018), EyePACS (Cuadros & Bresnick, 2009; Kaggle and EyePACS, 2015), and NCT-CRC-HE-100K (Kather et al., 2018). For ImageNet-1K, we use validation images and sample a class-balanced subset of 1000 benign images, with one image per class. All 1000 benign images are correctly classified by the four ImageNet backbones used in this study, namely ResNet-50, DenseNet-121, ViT-B, and DEiT-B. This shared clean-correct subset ensures that attack success is measured from identical source-preserving starting points across convolutional and transformer architectures. For CIFAR-10, we use the official test splits and retain only clean samples that are correctly assigned to the ground-truth source class by the corresponding attacked model or defense. For NIH-CXR, HAM10000, EyePACS, and NCT-CRC-HE-100K, we use the held-out test partitions following the evaluation protocol of HyCAS (Dhar et al., 2026b).

For source-displacement evaluation, the attack is untargeted: given a source label y, the objective is to find a feasible adversarial candidate $x^\star$ for which at least one non-source class overtakes the source class. Thus, no fixed target class is specified in advance. Instead, RoHoAttacK dynamically constructs the active rival set $\mathcal{A}_K(x_i^\star; y)$ at each iteration and generates interaction-aware candidate directions from the currently competitive non-source classes.

For ordered top-K evaluation, we follow the ordered target-list protocol adopted by prior ordered top-K attacks (Paniagua et al., 2023; 2025). For each clean ImageNet image and ordered target size K_{ord}, an ordered target list $\mathcal{T}_{K_{\mathrm{ord}}} = (t_1, \ldots, t_{K_{\mathrm{ord}}})$ is sampled from non-source classes and kept fixed across competing attacks for the same image. When best/mean/worst ordered results are reported, five ordered target lists are sampled for each image and each K_{ord}. An ordered attack is successful only when the top-K_{ord} predicted classes of the adversarial candidate match the prescribed target order. We report ordered top-K results for $K_{\mathrm{ord}} \in \{1, 5, 10, 15, 20\}$, and use O_{30} and O_{60} to denote 30-step and 60-step ordered attack budgets, respectively. Accordingly, we denote the ordered top-K variants of RoHoAttacK as RoHoAttacK-O_{30} and RoHoAttacK-O_{60} for the 30-step and 60-step settings, respectively.

Metrics. For source-displacement and defense ℓ_∞ evaluation, the main budget is $\epsilon = 8/255$. Attack success is counted by the sign of the active rival loss:

$$\mathrm{ASR} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}\left[\mathcal{L}_K(x_n^\star; y_n) > 0\right], \tag{88}$$

where $\mathcal{L}_K(x_n^\star; y_n) > 0$ indicates source-class displacement. We also report the achieved active rival margin

$$M_K(x^\star; y) = \max_{a \in \mathcal{A}_K(x^\star; y)} \left[\ell_a(x^\star) - \ell_y(x^\star)\right], \tag{89}$$

where larger M_K indicates that the adversarial candidate has moved deeper into a non-source rival region. In our source-displacement formulation, M_K coincides with the active rival loss value used for maximization. To measure perturbation efficiency, we report ℓ_1, ℓ_2, and ℓ_∞ norms of the perturbation $\delta = x^\star - x$. For ℓ_∞ evaluation, the main budget is $\epsilon = 8/255$ unless stated otherwise; larger-budget results at $\epsilon = 16/255$ are reported separately in the appendix. Runtime is measured as average seconds per image under the same implementation and hardware setting for a given comparison.

For robust-defense evaluation, we report robust accuracy:

$$\mathrm{RA} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}\left[\mathcal{L}_K(x_n^\star; y_n) \leq 0\right]. \tag{90}$$

Lower robust accuracy indicates a stronger attack because fewer samples preserve the source class after adversarial evaluation. For ordered top-K evaluation, ASR is computed by exact satisfaction of the prescribed ordered target list, while perturbation norms and runtime are reported to measure attack efficiency.

Baselines. We compare RoHoAttacK with three groups of baselines. First, for standard white-box source-displacement evaluation, we compare against PGD, APGD-DLR, C&W, DeepFool, FAB, APAA, PMA, MALT, and Prompt2Perturb (Madry et al., 2018; Croce & Hein, 2020b; Carlini &

Wagner, 2017; Moosavi-Dezfooli et al., 2016; Croce & Hein, 2020a; Yuan et al., 2024). These baselines cover projected first-order optimization, adaptive projected attacks, boundary- and margin-based attacks, gradient-direction refinement, and recent high-potency perturbation methods.

Second, for robust-defense evaluation, we compare against standard and adaptive evaluation protocols, including PGD-20, APGD-20, AA-20, EOT-PGD (Athalye et al., 2018b;a), BPDA, and EOT-BPDA. This group tests whether randomized or non-smooth defense components can be bypassed by attacks that average over stochasticity or approximate non-differentiable operations.

Third, for ordered top-K evaluation, we compare RoHoAttacK-O against QuadAttacK and RisingAttacK (Paniagua et al., 2023; 2025). QuadAttacK optimizes ordered constraints through a feature-space quadratic program, while RisingAttacK optimizes ordered perturbations directly in image space through adversarial-Jacobian subspaces. RoHoAttacK-O differs from both by constructing active rival-margin interactions and selecting only the projected candidate that improves the ordered adversarial objective after projection.

Optimization details. All attacks are evaluated under the same feasible perturbation region used for final scoring. For RoHoAttacK, every candidate generated by the interaction-aware direction family is first mapped through the bounded direction map and then projected into $\mathcal{B}_p(x, \epsilon)$ before its loss is evaluated. The selected update is therefore not the candidate with the largest raw gradient magnitude, largest residual magnitude, or best pre-projection behavior; it is the feasible candidate that achieves the largest realized active-loss increase after projection. The zero step is included in the signed lookahead set, so the current iterate remains available as a candidate and a loss-decreasing update is not accepted. When random restarts are used, the final adversarial candidate is selected by the same loss-maximization criterion:

$$r^\star = \underset{r\in\{1,\dots,R\}}{\arg\max}\ \mathcal{L}_K(x_T^{\star,(r)}; y), \qquad x^\star = x_T^{\star,(r^\star)}. \tag{91}$$

This detail is important for fair evaluation because attack progress is measured only after feasibility, clipping, and projection have been enforced.

Evaluation details. Unless otherwise stated, RoHoAttacK uses active rival size $K_{\text{act}} = 5$, which gives $\binom{5}{2} = 10$ active rival pairs per iteration. The finite-difference probing scale is $h = \eta\epsilon$, and the signed lookahead schedule follows Eq. 21. All perturbation budgets are stated on the $[0, 1]$ pixel scale before dataset normalization, and adversarial candidates are clipped to the valid image range after projection. The same RoHoAttacK hyperparameters are used across models within each evaluation protocol, and the attack is not tuned separately for individual images.

For all baselines, we use official or publicly available implementations when available and follow their recommended hyperparameters under the same perturbation budget. For ordered top-K evaluation, competing attacks are initialized from the same clean image and use the same ordered target lists and optimization budgets. For randomized defenses, stochastic components remain active during evaluation, and adaptive baselines use their corresponding EOT or BPDA variants. All reported values are averaged over the evaluated samples; perturbation norms are computed on successful adversarial candidates, while ASR and robust accuracy account for both successes and failures.

E Algorithmic Implementation of RoHoAttacK

Algorithm 1 summarizes the full RoHoAttacK procedure. The attack begins from the clean input or a randomly initialized feasible point, repeatedly identifies the active non-source rivals, constructs interaction-aware directions from refined cross-response residuals, and selects the projected candidate that maximizes active rival-margin progress. Algorithms 2 and 3 detail the two main computational modules: pairwise interaction-aware direction generation and projected lookahead-margin selection.

F Additional Performance Comparison

Larger-budget evaluation at $\epsilon = 16/255$. Appendix Table 5 extends the white-box adversarial comparison to the larger $\epsilon = 16/255$ budget. RoHoAttacK remains consistently effective, achieving near-saturated ASR values between 0.9985 and 0.9999 across ResNet-50, DenseNet-121, ViT-B,

Algorithm 1: Rival Interaction Optimization for Holistic Attack (RoHoAttack)

Input: Classifier logits $\ell(\cdot)$; clean image-label pair (x, y); budget ϵ; norm p; active-rival size K; iterations T; base step α; finite-difference scale η; restarts R.

Output: Adversarial example $x^\star$.

$x_{\text{best}} \leftarrow x, \quad s_{\text{best}} \leftarrow M_K(x; y)$

for $r = 1$ **to** R **do**

  if $r = 1$ **then**

    $x_0^{\star,(r)} \leftarrow x$

  else

    Sample $\xi^{(r)}$ and set $x_0^{\star,(r)} \leftarrow \Pi_{\mathcal{B}_p(x,\epsilon)}(x + \xi^{(r)})$

  for $i = 0$ **to** $T - 1$ **do**

    Compute active rival set $\mathcal{A}_K(x_i^{\star,(r)}; y) = \text{TopK}_{a \in \mathcal{Y} \setminus \{y\}} \, \ell_a(x_i^{\star,(r)})$

    Construct active rival-pair set $\mathcal{P}_K(x_i^{\star,(r)}; y) = \{(a, b) : a, b \in \mathcal{A}_K(x_i^{\star,(r)}; y), a < b\}$

    $\mathcal{D}_K^{\text{H}}(x_i^{\star,(r)}; y) \leftarrow$ PAIRWISE-INTERACTION-GENERATOR$(x_i^{\star,(r)}, x, y, \mathcal{A}_K, \mathcal{P}_K, \epsilon, p, \eta)$

    $\alpha_i \leftarrow \alpha \left[0.10 + 0.90 \cdot \frac{1+\cos(\pi i/(T-1))}{2}\right]$

    $\bar{\Gamma}_i \leftarrow \{0, \pm\alpha_i, \pm 0.5\alpha_i, \pm 0.25\alpha_i\}$

    $x_{i+1}^{\star,(r)} \leftarrow$ PROJECTED-LOOKAHEAD-SELECTION$(x_i^{\star,(r)}, x, y, \mathcal{P}_K, \mathcal{D}_K^{\text{H}}, \bar{\Gamma}_i, \epsilon, p)$

  if $M_K(x_T^{\star,(r)}; y) > s_{\text{best}}$ **then**

    $x_{\text{best}} \leftarrow x_T^{\star,(r)}$

    $s_{\text{best}} \leftarrow M_K(x_T^{\star,(r)}; y)$

return $x^\star = x_{\text{best}}$

and DEiT-B. Compared with the highest-ASR non-RoHo baseline, it improves ASR by 3.2–4.6 percentage points, increases M_K by up to 30.8%, and reduces the lowest non-RoHo ℓ_1, ℓ_2, and ℓ_∞ norms by up to 8.2%, 10.5%, and 13.2%, respectively.

These results show that the advantage of RoHoAttacK is not restricted to a tight perturbation budget. At larger radius, many baselines can already reach the adversarial region, so the key distinction becomes how much adversarial margin they accumulate and how efficiently they use the available perturbation budget. RoHoAttacK continues to improve because projected lookahead-margin selection directly optimizes the achieved post-projection margin, while pairwise cross-response direction generation supplies a richer candidate family than single-gradient, boundary-local, or isolated class-wise updates. These results support combining reciprocal rival-gradient probing with projected lookahead: probing supplies interaction-aware directions, while lookahead evaluates their achieved adversarial margins after projection. Thus, increasing the budget does not eliminate the benefit of active rival interactions; it gives the interaction-aware search more feasible candidates from which to select the most damaging projected update.

Best/mean/worst ordered top-K reliability. Appendix Tables 6–9 provide the best/mean/worst decomposition for ordered top-K attacks. This analysis is important because average ASR alone can hide unstable optimization: an attack may perform well on easy target lists while degrading sharply on difficult target configurations. RoHoAttacK-O improves the mean regime while also improving the best and worst cases, especially under difficult Top-20 and Top-15 settings, indicating that the proposed interaction-aware search improves attack reliability rather than only average attack success.

The reliability gain follows from two design choices. First, the active rival-pair policy avoids both all-class enumeration and single-rival ascent by focusing on locally competitive non-source classes. Second, the projected lookahead rule includes the current iterate in the candidate set and accepts only candidates that improve the projected adversarial margin. This prevents the attack from committing to geometrically plausible but margin-ineffective directions, which is particularly useful when the

Algorithm 2: Pairwise Interaction-Aware Direction Generation

Input: Current iterate $x_i^\star$; clean input x; label y; active rivals $\mathcal{A}_K$; active pairs $\mathcal{P}_K$; budget ϵ; norm p; finite-difference scale η; stability constants $\rho, \varepsilon_{\text{num}}$.

Output: Interaction-aware direction family $\mathcal{D}_K^{\text{H}}(x_i^\star; y)$.

$h \leftarrow \eta\epsilon$

foreach $r \in \mathcal{A}_K(x_i^\star; y)$ **do**

  $g_r(x_i^\star) \leftarrow \nabla_s m_r(s; y)|_{s=x_i^\star}$

  $U_r(x_i^\star) \leftarrow \dfrac{g_r(x_i^\star)}{\sqrt{\frac{1}{D}\sum_{j=1}^{D} g_{r,j}(x_i^\star)^2 + \rho}}$

  $x_i^r \leftarrow \Pi_{\mathcal{B}_p(x,\epsilon)}\left(x_i^\star + h \cdot \psi(U_r(x_i^\star))\right)$

$\mathcal{D}_K^{\text{H}}(x_i^\star; y) \leftarrow \emptyset$

foreach $(a, b) \in \mathcal{P}_K(x_i^\star; y)$ **do**

  Evaluate $U_b(x_i^a)$ and $U_a(x_i^b)$ using the normalized margin-field rule

  $H_{a,b}(x_i^\star) \leftarrow \dfrac{U_b(x_i^a) - U_b(x_i^\star)}{h} - \dfrac{U_a(x_i^b) - U_a(x_i^\star)}{h}$

  $U_b^\perp \leftarrow U_b(x_i^\star) - \text{Proj}_{U_a(x_i^\star)}(U_b(x_i^\star))$

  $\widetilde{H}_{a,b} \leftarrow H_{a,b} - \text{Proj}_{U_a(x_i^\star)}(H_{a,b}) - \text{Proj}_{U_b^\perp}(H_{a,b})$

  if $\|\widetilde{H}_{a,b}\|_2 > \varepsilon_{\text{num}}$ **then**

    $d_{a,b}(x_i^\star) \leftarrow \widetilde{H}_{a,b}$

  else

    $d_{a,b}(x_i^\star) \leftarrow \frac{1}{2}\left(U_a(x_i^\star) + U_b(x_i^\star)\right)$

  $\mathcal{D}_K^{\text{H}}(x_i^\star; y) \leftarrow \mathcal{D}_K^{\text{H}}(x_i^\star; y) \cup \{d_{a,b}(x_i^\star)\}$

return $\mathcal{D}_K^{\text{H}}(x_i^\star; y)$

Algorithm 3: Projected Lookahead-Margin Selection

Input: Current iterate $x_i^\star$; clean input x; label y; active pairs $\mathcal{P}_K$; directions $\mathcal{D}_K^{\text{H}}$; signed step set $\bar{\Gamma}_i$; budget ϵ; norm p.

Output: Next adversarial iterate $x_{i+1}^\star$.

$z_{\text{best}} \leftarrow x_i^\star$

$q_{\text{best}} \leftarrow M_K(x_i^\star; y)$

foreach $(a, b) \in \mathcal{P}_K(x_i^\star; y)$ **do**

  Retrieve $d_{a,b}(x_i^\star) \in \mathcal{D}_K^{\text{H}}(x_i^\star; y)$

  foreach $\gamma \in \bar{\Gamma}_i$ **do**

    $z_{a,b}^\gamma \leftarrow \Pi_{\mathcal{B}_p(x,\epsilon)}\left(x_i^\star + \gamma\epsilon\psi(d_{a,b}(x_i^\star))\right)$

    $q \leftarrow M_K(z_{a,b}^\gamma; y)$

    if $q > q_{\text{best}}$ **then**

      $z_{\text{best}} \leftarrow z_{a,b}^\gamma$

      $q_{\text{best}} \leftarrow q$

return $x_{i+1}^\star = z_{\text{best}}$

feasible ordered top-K region is narrow. Consequently, RoHoAttacK-O attains equal or higher ASR while reducing perturbation norms across best, mean, and worst regimes.

F.1 Qualitative Analysis on HAM10000

Figure 4 compares adversarial examples generated by PMA, EOT-PGD, EOT-BPDA, AutoAttack, and RoHoAttack from the same HAM10000 image. The upper row presents the clean and attacked images, their predicted classes, and magnified regions for inspecting local appearance changes. The

Table 4: **Ordered top-K attack effectiveness comparison on ImageNet-1K with 60 optimization steps.** Mean ASR, perturbation norms, and average computation time (s/image) for `QuadAttacK`, `RisingAttacK`, and `RoHoAttack-O60` across four classifiers and $K_{\text{ord}} \in \{1, 5, 10, 15, 20\}$.

(a) ResNet-50

Top-K	Method	Mean ASR↑	$\ell_1\downarrow$	$\ell_2\downarrow$	$\ell_\infty\downarrow$	Time (s/img)↓
Top-20	QuadAttacK$_{60}$	0.8344	10.0891	3133.6199	0.1079	**3.4039**
	RisingAttacK$_{60}$	0.8306	3.7474	1101.1521	0.0281	6.7267
	RoHoAttacK-O$_{60}$	**0.9130**	**3.2524**	**954.8126**	**0.0248**	10.7627
Top-15	QuadAttacK$_{60}$	0.9440	8.3368	2600.7510	0.0822	**3.4839**
	RisingAttacK$_{60}$	0.9868	3.0150	878.9222	0.0233	5.1634
	RoHoAttacK-O$_{60}$	**0.9930**	**2.6532**	**773.4515**	**0.0209**	8.2614
Top-10	QuadAttacK$_{60}$	0.9866	6.5228	2044.5753	0.0576	3.7396
	RisingAttacK$_{60}$	0.9936	2.0825	602.1784	0.0167	**3.3991**
	RoHoAttacK-O$_{60}$	**0.9966**	**1.8326**	**529.9170**	**0.0150**	5.9834
Top-5	QuadAttacK$_{60}$	0.9968	4.0029	1261.2314	0.0309	4.5257
	RisingAttacK$_{60}$	0.9558	1.1534	330.1495	0.0098	**1.8225**
	RoHoAttacK-O$_{60}$	**0.9986**	**1.0150**	**290.5316**	**0.0088**	7.2411
Top-1	QuadAttacK$_{60}$	0.9996	1.4443	467.1178	0.0083	5.3373
	RisingAttacK$_{60}$	0.9992	0.6144	165.8517	0.0064	**0.6114**
	RoHoAttacK-O$_{60}$	**1.0000**	**0.5407**	**145.9495**	**0.0057**	8.5397

(b) DenseNet-121

Top-K	Method	Mean ASR↑	$\ell_1\downarrow$	$\ell_2\downarrow$	$\ell_\infty\downarrow$	Time (s/img)↓
Top-20	QuadAttacK$_{60}$	0.8340	11.6266	3583.3589	0.1268	**4.0066**
	RisingAttacK$_{60}$	0.9812	5.9921	1744.8239	0.0468	8.2901
	RoHoAttacK-O$_{60}$	**0.9930**	**5.2730**	**1535.4450**	**0.0421**	13.2642
Top-15	QuadAttacK$_{60}$	0.9866	9.2713	2884.2755	0.0887	**3.8963**
	RisingAttacK$_{60}$	**1.0000**	4.3657	1252.9889	0.0359	6.2878
	RoHoAttacK-O$_{60}$	**1.0000**	**3.8418**	**1102.6302**	**0.0315**	10.0605
Top-10	QuadAttacK$_{60}$	0.9986	6.7558	2123.4894	0.0545	**3.8256**
	RisingAttacK$_{60}$	**1.0000**	2.6903	759.1986	0.0235	4.2223
	RoHoAttacK-O$_{60}$	**1.0000**	**2.3675**	**668.0948**	**0.0210**	6.7557
Top-5	QuadAttacK$_{60}$	0.9998	3.9671	1258.1706	0.0264	3.8644
	RisingAttacK$_{60}$	0.9994	1.2169	331.6714	0.0119	**2.2643**
	RoHoAttacK-O$_{60}$	**1.0000**	**1.0709**	**291.8708**	**0.0107**	6.1830
Top-1	QuadAttacK$_{60}$	**1.0000**	1.5191	503.0047	0.0070	3.0413
	RisingAttacK$_{60}$	**1.0000**	0.7001	177.1356	0.0085	**0.8046**
	RoHoAttacK-O$_{60}$	**1.0000**	**0.6161**	**155.8793**	**0.0063**	4.8661

(c) ViT-B

Top-K	Method	Mean ASR↑	$\ell_1\downarrow$	$\ell_2\downarrow$	$\ell_\infty\downarrow$	Time (s/img)↓
Top-20	QuadAttacK$_{60}$	0.7828	7.9108	2393.0875	0.0815	**5.0069**
	RisingAttacK$_{60}$	0.9864	3.7609	1007.6887	0.0360	15.8230
	RoHoAttacK-O$_{60}$	**0.9924**	**3.3096**	**886.7661**	**0.0318**	25.3168
Top-15	QuadAttacK$_{60}$	0.8404	6.2661	1893.5173	0.0620	**4.7622**
	RisingAttacK$_{60}$	0.9988	2.8751	753.1852	0.0284	11.9841
	RoHoAttacK-O$_{60}$	**0.9994**	**2.4924**	**651.2189**	**0.0250**	19.1746
Top-10	QuadAttacK$_{60}$	0.9130	4.5246	1374.2282	0.0410	**4.6368**
	RisingAttacK$_{60}$	0.9936	1.9915	508.8791	0.0206	8.2583
	RoHoAttacK-O$_{60}$	**0.9990**	**1.7525**	**447.8136**	**0.0185**	13.2133
Top-5	QuadAttacK$_{60}$	0.9980	3.6439	1128.3054	0.0288	**4.3981**
	RisingAttacK$_{60}$	0.5712	1.1650	292.6494	0.0128	4.4038
	RoHoAttacK-O$_{60}$	**0.9990**	**1.0252**	**257.5315**	**0.0115**	7.0461
Top-1	QuadAttacK$_{60}$	0.9998	1.5736	509.7575	0.0081	2.6007
	RisingAttacK$_{60}$	0.9388	0.4365	96.0745	0.0060	**1.2715**
	RoHoAttacK-O$_{60}$	**1.0000**	**0.3841**	**84.5456**	**0.0054**	4.1611

(d) DEiT-B

Top-K	Method	Mean ASR↑	$\ell_1\downarrow$	$\ell_2\downarrow$	$\ell_\infty\downarrow$	Time (s/img)↓
Top-20	QuadAttacK$_{60}$	0.9612	7.6974	2343.5441	0.0735	**4.1868**
	RisingAttacK$_{60}$	0.9956	2.9174	781.2607	0.0282	15.8331
	RoHoAttacK-O$_{60}$	**0.9976**	**2.5673**	**687.5094**	**0.0250**	25.3330
Top-15	QuadAttacK$_{60}$	0.9750	6.0671	1852.4958	0.0544	**3.9525**
	RisingAttacK$_{60}$	**1.0000**	2.2015	573.3811	0.0223	11.9810
	RoHoAttacK-O$_{60}$	**1.0000**	**1.9373**	**504.5754**	**0.0198**	19.1696
Top-10	QuadAttacK$_{60}$	0.9762	4.3693	1346.6326	0.0353	**3.8755**
	RisingAttacK$_{60}$	0.9996	1.5076	379.6582	0.0162	8.2610
	RoHoAttacK-O$_{60}$	**1.0000**	**1.3267**	**334.0992**	**0.0146**	13.2176
Top-5	QuadAttacK$_{60}$	0.9984	3.3975	1064.7252	0.0243	**3.4381**
	RisingAttacK$_{60}$	0.9992	1.0575	254.5953	0.0121	4.4027
	RoHoAttacK-O$_{60}$	**1.0000**	**0.9306**	**224.0439**	**0.0109**	7.0443
Top-1	QuadAttacK$_{60}$	**1.0000**	1.3910	459.6084	0.0063	2.9955
	RisingAttacK$_{60}$	0.9794	0.3340	68.5738	0.0052	**1.2708**
	RoHoAttacK-O$_{60}$	**1.0000**	**0.2939**	**60.3449**	**0.0047**	4.7928

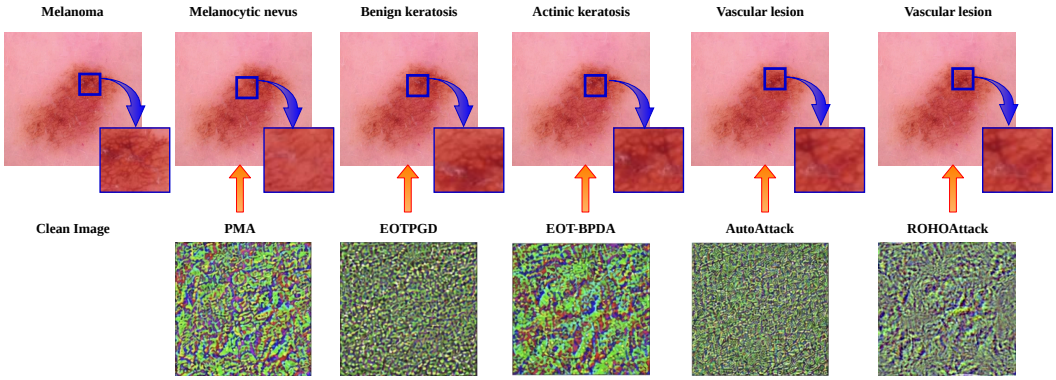

Figure 4: **Qualitative comparison of adversarial attacks on HAM10000. Top:** Clean and adversarial images with predicted classes and magnified regions. **Bottom:** Perturbation visualizations for `PMA`, `EOT-PGD`, `EOT-BPDA`, `AutoAttack`, and `RoHoAttack`. The perturbations alter the clean image and induce incorrect predictions. `RoHoAttack` refines reciprocal rival-field responses and selects projected candidates by achieved adversarial loss, here changing the prediction from melanoma to vascular lesion. This interaction-aware search strengthens attack effectiveness over the evaluated baselines, as supported by Fig. 3.

lower row visualizes the corresponding perturbations, showing how each attack modifies the input to influence the classifier's decision.

The clean image is classified as melanoma. After perturbation, `PMA` changes the prediction to melanocytic nevus, `EOT-PGD` to benign keratosis, and `EOT-BPDA` to actinic keratosis. Both `AutoAttack` and `RoHoAttack` produce a vascular-lesion prediction. These outcomes illustrate

Table 5: **Attack effectiveness comparison against non-robust classifiers on ImageNet-1K at $\epsilon = 16/255$.** ASR, active rival margin M_K, perturbation norms, and average computation time (s/image) across CNN and transformer architectures.**Green** denotes best performance.

(a) ResNet-50

Attack	ASR↑	M_K ↑	ℓ_1 ↓	ℓ_2 ↓	ℓ_∞ ↓	Time↓
PGD	0.9418	4.4826	8.7362	3895.4187	0.0627	**0.4382**
APGD-DLR	0.9627	5.1134	7.6284	3428.7316	0.0627	0.8564
C&W	0.9365	4.6318	6.4825	2814.5269	0.0496	4.8537
DeepFool	0.9036	3.0761	4.3967	1806.5142	0.0389	1.1265
FAB	0.9321	3.8847	4.6128	1884.7631	0.0413	1.5368
APAA	0.9584	5.2159	7.1426	3198.6274	0.0562	0.7514
PMA	0.9639	5.3817	6.2348	2692.5718	0.0483	1.3846
MALT	0.9678	5.4924	5.9472	2538.7645	0.0469	1.1729
Prompt2Perturb	0.9467	5.0375	5.2861	2254.6183	0.0435	6.2748
RoHoAttacK	**0.9999**	**7.1846**	**4.1263**	**1652.4879**	**0.0346**	10.4627

(b) DenseNet-121

Attack	ASR↑	M_K ↑	ℓ_1 ↓	ℓ_2 ↓	ℓ_∞ ↓	Time↓
PGD	0.9364	4.3728	8.9481	4016.3827	0.0627	**0.4694**
APGD-DLR	0.9582	4.9826	8.1053	3654.2189	0.0627	0.9468
C&W	0.9291	4.4915	6.8276	2946.5173	0.0512	5.2739
DeepFool	0.8968	2.9184	4.5289	1855.3264	0.0398	1.2358
FAB	0.9257	3.7642	4.7463	1962.8841	0.0425	1.6974
APAA	0.9541	5.0867	7.3842	3335.7162	0.0574	0.8427
PMA	0.9608	5.2419	6.5173	2836.4195	0.0496	1.4426
MALT	0.9645	5.3627	6.2154	2672.5038	0.0481	1.2384
Prompt2Perturb	0.9422	4.9068	5.4827	2356.4928	0.0448	6.5871
RoHoAttacK	**0.9997**	**6.9274**	**4.1847**	**1695.7436**	**0.0351**	10.9365

(c) ViT-B

Attack	ASR↑	M_K ↑	ℓ_1 ↓	ℓ_2 ↓	ℓ_∞ ↓	Time↓
PGD	0.9216	3.9725	9.1364	4148.5726	0.0627	**0.5038**
APGD-DLR	0.9472	4.6931	8.3248	3762.6917	0.0627	0.9842
C&W	0.9208	4.1837	7.1063	3062.7845	0.0526	5.5174
DeepFool	0.8823	2.7046	4.6815	1938.5126	0.0407	1.2967
FAB	0.9139	3.4628	4.8924	2045.8361	0.0434	1.7835
APAA	0.9495	4.9186	7.6184	3484.6392	0.0581	0.8731
PMA	0.9543	5.0836	6.7462	2965.4138	0.0505	1.5127
MALT	0.9571	5.1984	6.4389	2794.6821	0.0492	1.2845
Prompt2Perturb	0.9346	4.7325	5.7641	2486.5372	0.0456	7.0346
RoHoAttacK	**0.9988**	**6.6418**	**4.3126**	**1764.5293**	**0.0359**	11.4826

(d) DEiT-B

Attack	ASR↑	M_K ↑	ℓ_1 ↓	ℓ_2 ↓	ℓ_∞ ↓	Time↓
PGD	0.9184	3.9142	8.8627	3998.4635	0.0627	**0.4876**
APGD-DLR	0.9437	4.6125	8.0186	3616.8249	0.0627	0.9374
C&W	0.9162	4.1048	6.7342	2916.3824	0.0517	5.1842
DeepFool	0.8795	2.7629	4.3983	1807.6425	0.0401	1.1847
FAB	0.9106	3.5213	4.5927	1885.2746	0.0426	1.6345
APAA	0.9453	4.8075	7.2368	3298.6147	0.0572	0.8269
PMA	0.9498	4.9562	6.4826	2847.4926	0.0497	1.4653
MALT	0.9526	5.0837	6.1265	2661.7384	0.0484	1.2431
Prompt2Perturb	0.9308	4.6289	5.4173	2354.7815	0.0449	6.7185
RoHoAttacK	**0.9985**	**6.3725**	**4.0368**	**1618.7429**	**0.0348**	10.9718

ResNet-50

Table 6: **Best-, mean-, and worst-case comparison of ordered top-K attacks on ImageNet-1K with ResNet-50.** ASR and perturbation norms over five sampled target lists per image and ranking length, with average computation time (s/image) for 30- and 60-step attacks.

Top-K	Method	Best				Mean				Worst				Time
		ASR↑	ℓ_1 ↓	ℓ_2 ↓	ℓ_∞ ↓	ASR↑	ℓ_1 ↓	ℓ_2 ↓	ℓ_∞ ↓	ASR↑	ℓ_1 ↓	ℓ_2 ↓	ℓ_∞ ↓	(s/img)↓
Top-20	QuadAttacK$_{60}$	0.9560	9.7368	3030.3407	0.0984	0.8344	10.0891	3133.6199	0.1079	0.6500	10.4514	3239.8729	0.1183	**3.4039**
	RisingAttacK$_{60}$	0.9500	3.3880	992.4068	0.0257	0.8306	3.7474	1101.1521	0.0281	0.6520	4.1005	1207.9307	0.0305	6.7267
	RoHoAttacK-O$_{60}$	**0.9760**	**2.9137**	**853.4698**	**0.0226**	**0.9130**	**3.2524**	**954.8126**	**0.0248**	**0.7070**	**3.5264**	**1038.8204**	**0.0268**	10.7627
	QuadAttacK$_{30}$	0.2620	9.0111	2824.4299	0.0839	0.0978	9.0948	2850.0433	0.0858	0.0080	9.1756	2874.5937	0.0876	**1.7264**
	RisingAttacK$_{30}$	0.2040	3.4129	1000.3753	0.0264	0.0666	3.4854	1022.5585	0.0269	0.0010	3.5609	1045.7229	0.0274	3.7216
	RoHoAttacK-O$_{30}$	**0.4020**	**2.8668**	**840.3153**	**0.0227**	**0.2778**	**3.0672**	**899.8515**	**0.0242**	**0.1880**	**2.9912**	**878.4072**	**0.0236**	5.9546
Top-15	QuadAttacK$_{60}$	0.9870	7.9013	2470.1343	0.0729	0.9440	8.3368	2600.7510	0.0822	0.8460	8.8010	2739.1058	0.0932	**3.4839**
	RisingAttacK$_{60}$	0.9990	2.6335	763.6941	0.0208	0.9868	3.0150	878.9222	0.0233	0.9610	3.4246	1003.1229	0.0260	5.1634
	RoHoAttacK-O$_{60}$	**1.0000**	**2.2648**	**656.7769**	**0.0183**	**0.9930**	**2.6532**	**773.4515**	**0.0209**	**0.9810**	**2.9452**	**862.6857**	**0.0229**	8.2614
	QuadAttacK$_{30}$	0.7540	7.5962	2380.3027	0.0674	0.4922	7.8296	2451.8036	0.0717	0.2090	8.0427	2517.2334	0.0759	**1.7382**
	RisingAttacK$_{30}$	0.8310	2.7910	812.1335	0.0219	0.5856	2.9944	873.3877	0.0234	0.2970	3.2057	936.9361	0.0249	2.8794
	RoHoAttacK-O$_{30}$	**0.9110**	**2.3444**	**682.1921**	**0.0188**	**0.7110**	**2.6351**	**768.5812**	**0.0207**	**0.4370**	**2.6928**	**787.0263**	**0.0214**	4.6070
Top-10	QuadAttacK$_{60}$	0.9970	6.1074	1917.6238	0.0504	0.9866	6.5228	2044.5753	0.0576	0.9660	6.9893	2186.1273	0.0666	3.7396
	RisingAttacK$_{60}$	0.9980	1.8077	519.1006	0.0149	0.9936	2.0825	602.1784	0.0167	0.9800	2.3735	690.1587	0.0187	**3.3991**
	RoHoAttacK-O$_{60}$	**1.0000**	**1.5546**	**446.4265**	**0.0131**	**0.9966**	**1.8326**	**529.9170**	**0.0150**	**0.9830**	**2.0412**	**593.5365**	**0.0165**	5.9834
	QuadAttacK$_{30}$	0.9520	6.0248	1893.2294	0.0491	0.8460	6.3547	1994.8023	0.0544	0.6600	6.6744	2093.2491	0.0598	**1.7593**
	RisingAttacK$_{30}$	0.9410	1.9697	568.1198	0.0160	0.8064	2.1748	630.0922	0.0175	0.5990	2.3743	690.3320	0.0188	1.7965
	RoHoAttacK-O$_{30}$	**0.9870**	**1.6545**	**477.2206**	**0.0138**	**0.9060**	**1.9138**	**554.4811**	**0.0158**	**0.7400**	**1.9944**	**579.8789**	**0.0162**	2.8744
Top-5	QuadAttacK$_{60}$	**1.0000**	3.6813	1161.6413	0.0264	0.9968	4.0029	1261.2314	0.0309	0.9900	4.3529	1369.4046	0.0362	4.5257
	RisingAttacK$_{60}$	0.9890	0.9567	270.4779	0.0085	0.9558	1.1534	330.1495	0.0098	0.8980	1.3547	391.1246	0.0112	**1.8225**
	RoHoAttacK-O$_{60}$	**1.0000**	**0.8228**	**232.6110**	**0.0075**	**0.9986**	**1.0150**	**290.5316**	**0.0088**	**0.9930**	**1.1650**	**336.3672**	**0.0099**	7.2411
	QuadAttacK$_{30}$	0.9880	3.6540	1153.1996	0.0261	0.9590	3.9539	1246.4929	0.0300	0.8950	4.2646	1343.3064	0.0344	2.1458
	RisingAttacK$_{30}$	0.9860	1.2737	361.3176	0.0110	0.9504	1.4693	420.0254	0.0124	0.8910	1.6607	477.3890	0.0138	**0.9517**
	RoHoAttacK-O$_{30}$	**0.9940**	**1.0699**	**303.5068**	**0.0095**	**0.9840**	**1.2325**	**351.8154**	**0.0105**	**0.9750**	**1.3950**	**401.0068**	**0.0119**	3.4333
Top-1	QuadAttacK$_{60}$	**1.0000**	1.1548	381.4498	0.0057	0.9996	1.4443	467.1178	0.0083	0.9980	1.7222	550.7556	0.0110	5.3373
	RisingAttacK$_{60}$	**1.0000**	0.4136	104.7782	0.0049	0.9992	0.6144	165.8517	0.0064	0.9990	0.8364	233.5616	0.0079	**0.6114**
	RoHoAttacK-O$_{60}$	**1.0000**	**0.3557**	**90.1093**	**0.0043**	**1.0000**	**0.5407**	**145.9495**	**0.0057**	**1.0000**	**0.7193**	**200.8630**	**0.0070**	8.5397
	QuadAttacK$_{30}$	0.9980	1.1521	380.6522	0.0057	0.9772	1.4244	461.1199	0.0080	0.9340	1.6861	540.0446	0.0105	2.6411
	RisingAttacK$_{30}$	**1.0000**	0.6218	163.8710	0.0066	0.9986	0.9155	251.6174	0.0088	0.9950	1.2054	338.3424	0.0110	**0.3201**
	RoHoAttacK-O$_{30}$	**1.0000**	**0.5223**	**137.6516**	**0.0049**	**1.0000**	**0.7626**	**209.3175**	**0.0072**	**1.0000**	**1.0125**	**284.2076**	**0.0090**	4.2258

that different perturbation patterns can displace the same source prediction, exposing sensitivity in the classifier's decision despite the broadly similar appearance of the displayed images. The magnified regions support inspection of local changes, while the predicted labels reveal their effect on classification.

`RoHoAttack` constructs these perturbations through interaction-aware exploration. It probes how movement associated with one active rival changes another rival's normalized margin-gradient field,

DenseNet-121

Table 7: **Best-, mean-, and worst-case comparison of ordered top-K attacks on ImageNet-1K with DenseNet-121.** ASR and perturbation norms over five sampled target lists per image and ranking length, with average computation time (s/image) for 30- and 60-step attacks.

Top-K	Method	Best				Mean				Worst				Time
		ASR↑	$\ell_1\downarrow$	$\ell_2\downarrow$	$\ell_\infty\downarrow$	ASR↑	$\ell_1\downarrow$	$\ell_2\downarrow$	$\ell_\infty\downarrow$	ASR↑	$\ell_1\downarrow$	$\ell_2\downarrow$	$\ell_\infty\downarrow$	(s/img)↓
Top-20	QuadAttacK_{60}	0.9730	11.0446	3412.9160	0.1140	0.8340	11.6266	3583.3589	0.1268	0.5840	12.1967	3749.9306	0.1403	**4.0066**
	RisingAttacK_{60}	0.9970	5.1080	1480.5618	0.0406	0.9812	5.9921	1744.8239	0.0468	0.9450	6.9129	2020.8591	0.0532	8.2901
	RoHoAttacK-O_{60}	**1.0000**	**4.3929**	**1273.2831**	**0.0357**	**0.9930**	**5.2730**	**1535.4450**	**0.0421**	**0.9650**	**5.9451**	**1737.9388**	**0.0468**	13.2642
	QuadAttacK_{30}	0.1240	9.7962	3054.7057	0.0914	0.0330	9.8564	3072.6790	0.0923		*Failed*			**2.0206**
	RisingAttacK_{30}	0.7320	5.7318	1664.7979	0.0456	0.4500	6.1377	1786.5613	0.0485	0.1460	6.5680	1915.5893	0.0516	4.6070
	RoHoAttacK-O_{30}	**0.8120**	**4.8147**	**1398.4302**	**0.0392**	**0.6100**	**5.4012**	**1572.1739**	**0.0437**	**0.3260**	**5.5171**	**1609.0950**	**0.0444**	7.3712
Top-15	QuadAttacK_{60}	0.9970	8.6591	2701.7797	0.0774	0.9866	9.2713	2884.2755	0.0887	0.9610	9.8890	3067.5058	0.1011	**3.8963**
	RisingAttacK_{60}	**1.0000**	3.8050	1086.1460	0.0318	**1.0000**	4.3657	1252.9889	0.0359	**1.0000**	4.9640	1431.3256	0.0402	6.2878
	RoHoAttacK-O_{60}	**1.0000**	**3.2723**	**934.0856**	**0.0280**	**1.0000**	**3.8418**	**1102.6302**	**0.0315**	**1.0000**	**4.2690**	**1230.9400**	**0.0354**	10.0605
	QuadAttacK_{30}	0.7780	8.3044	2599.4287	0.0720	0.5088	8.6281	2697.9823	0.0771	0.2380	8.9352	2791.7948	0.0821	**1.8919**
	RisingAttacK_{30}	0.9900	4.2188	1207.4247	0.0350	0.9362	4.7380	1362.7350	0.0387	0.8350	5.2677	1521.6817	0.0425	3.5501
	RoHoAttacK-O_{30}	**0.9960**	**3.5438**	**1014.2367**	**0.0301**	**0.9798**	**4.1694**	**1199.2068**	**0.0336**	**0.9150**	**4.4249**	**1278.2126**	**0.0365**	5.6802
Top-10	QuadAttacK_{60}	**1.0000**	6.2172	1957.3529	0.0469	0.9986	6.7558	2123.4894	0.0545	0.9970	7.3050	2291.5986	0.0629	**3.8256**
	RisingAttacK_{60}	**1.0000**	2.3212	649.6995	0.0208	**1.0000**	2.6903	759.1986	0.0235	**1.0000**	3.0749	874.1637	0.0263	4.2223
	RoHoAttacK-O_{60}	**1.0000**	**1.9962**	**558.7416**	**0.0183**	**1.0000**	**2.3675**	**668.0948**	**0.0210**	**1.0000**	**2.6444**	**751.7808**	**0.0231**	6.7557
	QuadAttacK_{30}	0.9900	6.1801	1946.4525	0.0464	0.9392	6.6701	2098.0095	0.0531	0.8380	7.1563	2248.2525	0.0602	**1.8918**
	RisingAttacK_{30}	0.9980	2.4880	698.6799	0.0221	0.9880	2.9210	827.1606	0.0253	0.9690	3.3607	958.0512	0.0285	2.2937
	RoHoAttacK-O_{30}	**1.0000**	**2.0899**	**586.8911**	**0.0190**	**1.0000**	**2.5705**	**727.9013**	**0.0225**	**1.0000**	**2.8230**	**804.7630**	**0.0245**	3.6699
Top-5	QuadAttacK_{60}	**1.0000**	3.6045	1143.1525	0.0226	0.9998	3.9671	1258.1706	0.0264	0.9990	4.3422	1377.3171	0.0305	3.8644
	RisingAttacK_{60}	**1.0000**	1.0122	270.9332	0.0103	0.9994	1.2169	331.6714	0.0119	0.9980	1.4422	398.9840	0.0136	**2.2643**
	RoHoAttacK-O_{60}	**1.0000**	**0.8705**	**233.0026**	**0.0091**	**1.0000**	**1.0709**	**291.8708**	**0.0107**	**1.0000**	**1.2403**	**343.1262**	**0.0120**	6.1830
	QuadAttacK_{30}	0.9980	3.6019	1142.1894	0.0226	0.9924	3.9526	1253.5745	0.0262	0.9820	4.3229	1371.6348	0.0303	1.8502
	RisingAttacK_{30}	**1.0000**	1.3896	378.3550	0.0135	0.9982	1.6603	457.9204	0.0156	0.9940	1.9524	544.3561	0.0178	**1.2082**
	RoHoAttacK-O_{30}	**1.0000**	**1.1673**	**317.8182**	**0.0116**	**1.0000**	**1.4216**	**392.0683**	**0.0137**	**1.0000**	**1.6400**	**457.2591**	**0.0153**	2.9603
Top-1	QuadAttacK_{60}	**1.0000**	1.1724	397.8131	0.0049	**1.0000**	1.5191	503.0047	0.0070	**1.0000**	1.8544	606.7211	0.0093	3.0413
	RisingAttacK_{60}	**1.0000**	0.4732	112.4793	0.0063	**1.0000**	0.7001	177.1356	0.0085	**1.0000**	0.9566	251.8730	0.0108	**0.8046**
	RoHoAttacK-O_{60}	**1.0000**	**0.4070**	**96.7322**	**0.0043**	**1.0000**	**0.6161**	**155.8793**	**0.0063**	**1.0000**	**0.8227**	**216.6108**	**0.0082**	4.8661
	QuadAttacK_{30}	**1.0000**	1.1728	397.9329	0.0049	0.9960	1.5144	501.4779	0.0070	0.9880	1.8426	603.0238	0.0092	1.5519
	RisingAttacK_{30}	**1.0000**	0.7032	175.2027	0.0084	**1.0000**	1.0708	280.0300	0.0116	**1.0000**	1.4523	390.3135	0.0149	**0.4255**
	RoHoAttacK-O_{30}	**1.0000**	**0.5907**	**147.1703**	**0.0042**	**1.0000**	**0.9214**	**239.2618**	**0.0063**	**1.0000**	**1.2199**	**327.8633**	**0.0079**	2.4830

then refines the reciprocal response residual into a candidate direction. Projected lookahead evaluates the resulting feasible candidates and retains the one with the highest achieved active rival loss. Refreshing the active rivals after each update adapts this search to the evolving class competition. This coordination connects perturbation generation with its realized adversarial effect, supporting effective source-class displacement under the perturbation constraint.

The displayed melanoma-to-vascular-lesion transition illustrates the vulnerability exposed by `RoHoAttack`. Since all displayed attacks change the source prediction, this individual example does not establish their relative effectiveness. The comparative evidence is provided by Fig. 3, where `RoHoAttack` achieves lower robust accuracy than the evaluated standard and adaptive attacks across RPF, CTRW, HyCAS, and HySCAN on HAM10000. The qualitative visualization therefore complements the dataset-level evaluation by showing how adversarial perturbations translate into altered medical-image predictions.

G Ablation Study

We ablate RoHoAttacK to identify how each design choice contributes to adversarial progress. The study follows the structure of the proposed method: active top-K rival-gradient search, pairwise rival interaction search, reciprocal cross-response residual generation, residual refinement, and projected lookahead-margin selection. Unless otherwise stated, results are averaged over ResNet-50, DenseNet-121, ViT-B, and DEiT-B under the same ℓ_p-bounded protocol used in Table 1. Higher ASR and M_K indicate greater adversarial progress, while lower ℓ_p norms indicate more efficient use of the perturbation budget. For robust-defense evaluation, lower robust accuracy indicates higher attack potency.

Core component analysis. Table 10 shows that RoHoAttacK improves progressively as the attack moves from isolated top-1 margin ascent to active, interaction-aware, and projection-aligned optimization. The top-1 variant yields the weakest adversarial progress because it ignores simultaneous competition among non-source classes. Introducing active top-K rivals improves ASR and M_K by focusing the search on classes closest to overtaking the source. Pairwise probing further

ViT-B

Table 8: **Best-, mean-, and worst-case comparison of ordered top-K attacks on ImageNet-1K with ViT-B.** ASR and perturbation norms over five sampled target lists per image and ranking length, with average computation time (s/image) for 30- and 60-step attacks.

Top-K	Method	Best				Mean				Worst				Time
		ASR↑	$\ell_1\downarrow$	$\ell_2\downarrow$	$\ell_\infty\downarrow$	ASR↑	$\ell_1\downarrow$	$\ell_2\downarrow$	$\ell_\infty\downarrow$	ASR↑	$\ell_1\downarrow$	$\ell_2\downarrow$	$\ell_\infty\downarrow$	(s/img)↓
Top-20	QuadAttacK$_{60}$	0.9710	7.4811	2268.2729	0.0748	0.7828	7.9108	2393.0875	0.0815	0.4910	8.3426	2519.7231	0.0884	**5.0069**
	RisingAttacK$_{60}$	0.9980	3.0033	792.6934	0.0294	0.9864	3.7609	1007.6887	0.0360	0.9560	4.5804	1241.9193	0.0432	15.8230
	RoHoAttacK-O$_{60}$	**1.0000**	**2.5828**	**681.7163**	**0.0259**	**0.9924**	**3.3096**	**886.7661**	**0.0318**	**0.9760**	**3.9391**	**1068.0506**	**0.0380**	25.3168
	QuadAttacK$_{30}$	0.0020	6.3770	1992.7502	0.0533	0.0004	6.3770	1992.7502	0.0533		*Failed*			**2.6210**
	RisingAttacK$_{30}$	0.7380	4.5276	1221.1645	0.0436	0.4956	4.9482	1343.2135	0.0473	0.2490	5.3892	1471.1873	0.0511	7.9615
	RoHoAttacK-O$_{30}$	**0.8180**	**3.8032**	**1025.7782**	**0.0375**	**0.6620**	**4.2531**	**1149.2845**	**0.0410**	**0.3890**	**4.5269**	**1235.7973**	**0.0439**	12.7384
Top-15	QuadAttacK$_{60}$	0.9730	5.8803	1780.9271	0.0558	0.8404	6.2661	1893.5173	0.0620	0.5980	6.6687	2011.4965	0.0684	**4.7622**
	RisingAttacK$_{60}$	**1.0000**	2.3069	593.4747	0.0235	0.9988	2.8751	753.1852	0.0284	0.9940	3.5373	939.6227	0.0343	11.9841
	RoHoAttacK-O$_{60}$	**1.0000**	**1.9839**	**510.3882**	**0.0207**	**0.9994**	**2.4924**	**651.2189**	**0.0250**	**0.9970**	**3.0421**	**808.0755**	**0.0302**	19.1746
	QuadAttacK$_{30}$	0.0240	4.7795	1490.0706	0.0382	0.0056	4.7982	1495.8188	0.0385		*Failed*			**2.4245**
	RisingAttacK$_{30}$	0.9270	3.4401	908.2324	0.0339	0.7510	3.8944	1038.7394	0.0379	0.5310	4.4471	1197.6996	0.0427	6.0305
	RoHoAttacK-O$_{30}$	**0.9620**	**2.8897**	**762.9152**	**0.0292**	**0.8510**	**3.1817**	**847.2894**	**0.0325**	**0.6710**	**3.7356**	**1006.0677**	**0.0367**	9.6488
Top-10	QuadAttacK$_{60}$	0.9900	4.1855	1274.5181	0.0363	0.9130	4.5246	1374.2282	0.0410	0.7510	4.8729	1476.9954	0.0459	**4.6368**
	RisingAttacK$_{60}$	**1.0000**	1.5872	397.2939	0.0169	0.9936	1.9915	508.8791	0.0206	0.9750	2.4423	634.1043	0.0247	8.2583
	RoHoAttacK-O$_{60}$	**1.0000**	**1.3650**	**341.6728**	**0.0149**	**0.9990**	**1.7525**	**447.8136**	**0.0185**	**0.9950**	**2.1004**	**545.3297**	**0.0217**	13.2133
	QuadAttacK$_{30}$	0.0810	3.4468	1078.1791	0.0256	0.0252	3.4999	1094.6987	0.0261		*Failed*			**2.3034**
	RisingAttacK$_{30}$	0.9010	2.3233	599.0848	0.0240	0.7112	2.6247	684.1576	0.0267	0.4810	2.9492	775.8982	0.0297	4.1602
	RoHoAttacK-O$_{30}$	**0.9360**	**1.9516**	**503.2312**	**0.0206**	**0.8112**	**2.1835**	**568.1873**	**0.0230**	**0.6210**	**2.4773**	**651.7545**	**0.0255**	6.6563
Top-5	QuadAttacK$_{60}$	**1.0000**	3.2461	1010.5003	0.0241	0.9980	3.6439	1128.3054	0.0288	0.9930	4.0423	1246.0157	0.0338	**4.3981**
	RisingAttacK$_{60}$	0.8280	0.9934	245.3329	0.0111	0.5712	1.1650	292.6494	0.0128	0.2910	1.3395	340.9352	0.0144	4.4038
	RoHoAttacK-O$_{60}$	**1.0000**	**0.8543**	**210.9863**	**0.0098**	**0.9990**	**1.0252**	**257.5315**	**0.0115**	**0.9960**	**1.1520**	**293.2043**	**0.0127**	7.0461
	QuadAttacK$_{30}$	0.8120	3.0924	968.7317	0.0221	0.5024	3.2930	1029.8490	0.0242	0.1780	3.4820	1087.7387	0.0261	**2.1108**
	RisingAttacK$_{30}$	0.8420	1.3886	345.5126	0.0153	0.5980	1.6101	406.4644	0.0174	0.3310	1.8357	468.6344	0.0195	2.2197
	RoHoAttacK-O$_{30}$	**0.9220**	**1.1664**	**290.2306**	**0.0132**	**0.6980**	**1.3587**	**343.0915**	**0.0151**	**0.4710**	**1.5420**	**393.6529**	**0.0168**	3.5515
Top-1	QuadAttacK$_{60}$	**1.0000**	1.2537	410.8221	0.0059	0.9998	1.5736	509.7575	0.0081	0.9990	1.9042	612.7661	0.0106	2.6007
	RisingAttacK$_{60}$	0.9940	0.2978	60.3734	0.0045	0.9388	0.4365	96.0745	0.0060	0.8260	0.6048	139.9744	0.0078	**1.2715**
	RoHoAttacK-O$_{60}$	**1.0000**	**0.2561**	**51.9211**	**0.0040**	**1.0000**	**0.3841**	**84.5456**	**0.0054**	**1.0000**	**0.5201**	**120.3780**	**0.0069**	4.1611
	QuadAttacK$_{30}$	**1.0000**	1.2541	410.9606	0.0059	0.9958	1.5681	508.0591	0.0081	0.9900	1.8935	609.5179	0.0105	1.3040
	RisingAttacK$_{30}$	0.9950	0.4270	90.9070	0.0060	0.9362	0.6578	149.1661	0.0086	0.8180	0.9318	219.6820	0.0115	**0.6417**
	RoHoAttacK-O$_{30}$	**1.0000**	**0.3587**	**76.3619**	**0.0051**	**0.9992**	**0.5789**	**130.3188**	**0.0073**	**0.9960**	**0.7827**	**184.5329**	**0.0090**	2.0864

improves the result, showing that adversarial directions can arise from interactions between locally competitive margin fields rather than from either margin alone. The cross-response residual captures differences between reciprocal margin-gradient changes, while refinement suppresses components aligned with the original rival gradients. Finally, projected lookahead-margin selection gives the most favorable ASR–norm–RA trade-off because each candidate is judged only after projection into the feasible ℓ_p region.

Effect of active rival size. Table 11 shows that increasing K from 1 to 5 produces clear gains in ASR, M_K, perturbation efficiency, and robust-defense degradation. This confirms that a single-rival search underuses the local multi-class geometry. Beyond $K = 5$, performance nearly saturates: increasing K from 5 to 10 improves ASR by only 0.05 percentage points while increasing the pair count from 10 to 45. Thus, $K = 5$ provides a practical balance between attack potency and candidate-search cost.

Effect of projected lookahead-margin selection. Table 12 supports evaluating candidate quality after projection. A large cross-response residual does not guarantee adversarial progress after bounded mapping and projection, while pre-projection scoring overlooks their effect on the resulting candidate. The full rule evaluates both orientations at multiple scales within the feasible ℓ_p region and retains the current iterate as a zero-step candidate. For deterministic classifiers, this ensures non-decreasing active loss. Among the evaluated variants, the full rule achieves the highest ASR and M_K with the lowest perturbation norms.

Effect on robust-defense evaluation. Table 13 shows progressively lower robust accuracy as active rival optimization, reciprocal probing, and projected selection are combined. The active rival-gradient search reduces robust accuracy relative to the leading evaluated baseline, with further reductions from pairwise cross-response residuals. The full `RoHoAttack` achieves the lowest defense-averaged robust accuracy on each of the six natural and medical benchmarks. These results support combining interaction-aware direction generation with projection-aligned selection to expose vulnerabilities in defended models.

DEiT-B

Table 9: **Best-, mean-, and worst-case comparison of ordered top-K attacks on ImageNet-1K with DeiT-B.** ASR and perturbation norms over five sampled target lists per image and ranking length, with average computation time (s/image) for 30- and 60-step attacks.

Top-K	Method	Best				Mean				Worst				Time
		ASR↑	$\ell_1\downarrow$	$\ell_2\downarrow$	$\ell_\infty\downarrow$	ASR↑	$\ell_1\downarrow$	$\ell_2\downarrow$	$\ell_\infty\downarrow$	ASR↑	$\ell_1\downarrow$	$\ell_2\downarrow$	$\ell_\infty\downarrow$	(s/img)↓
Top-20	QuadAttacK$_{60}$	0.9990	7.2048	2199.2260	0.0665	0.9612	7.6974	2343.5441	0.0735	0.8540	8.2263	2499.1404	0.0811	**4.1868**
	RisingAttacK$_{60}$	**1.0000**	2.3202	611.1048	0.0230	0.9956	2.9174	781.2607	0.0282	0.9850	3.5733	968.9946	0.0339	15.8331
	RoHoAttacK-O$_{60}$	**1.0000**	**1.9954**	**525.5501**	**0.0202**	**0.9976**	**2.5673**	**687.5094**	**0.0250**	**0.9880**	**3.0730**	**833.3354**	**0.0298**	25.3330
	QuadAttacK$_{30}$	0.0160	6.2491	1950.0525	0.0524	0.0032	6.2491	1950.0525	0.0524		*Failed*			**2.1503**
	RisingAttacK$_{30}$	0.8560	3.5031	946.6329	0.0338	0.6348	3.8373	1045.3953	0.0366	0.3810	4.1863	1148.5782	0.0395	7.9624
	RoHoAttacK-O$_{30}$	**0.9360**	**2.9426**	**795.1716**	**0.0291**	**0.7348**	**3.3314**	**904.1626**	**0.0322**	**0.5210**	**3.5165**	**964.8057**	**0.0340**	12.7398
Top-15	QuadAttacK$_{60}$	**1.0000**	5.5928	1713.0918	0.0481	0.9750	6.0671	1852.4958	0.0544	0.9100	6.5599	1997.5902	0.0611	**3.9525**
	RisingAttacK$_{60}$	**1.0000**	1.7594	448.6124	0.0184	**1.0000**	2.2015	573.3811	0.0223	**1.0000**	2.7017	715.2705	0.0266	11.9810
	RoHoAttacK-O$_{60}$	**1.0000**	**1.5131**	**385.8067**	**0.0162**	**1.0000**	**1.9373**	**504.5754**	**0.0198**	**1.0000**	**2.3235**	**615.1326**	**0.0234**	19.1696
	QuadAttacK$_{30}$	0.1350	4.9541	1548.1232	0.0383	0.0338	4.9874	1558.1460	0.0386		*Failed*			**2.0234**
	RisingAttacK$_{30}$	0.9870	2.7227	714.8364	0.0273	0.9278	3.1490	838.2295	0.0310	0.8140	3.5773	963.3399	0.0346	6.0263
	RoHoAttacK-O$_{30}$	**0.9930**	**2.2871**	**600.4626**	**0.0235**	**0.9732**	**2.7219**	**725.3367**	**0.0266**	**0.8940**	**3.0049**	**809.2055**	**0.0298**	9.6421
Top-10	QuadAttacK$_{60}$	0.9980	3.9545	1222.8665	0.0305	0.9762	4.3693	1346.6326	0.0353	0.9150	4.7999	1475.4252	0.0406	**3.8755**
	RisingAttacK$_{60}$	**1.0000**	1.1885	291.5197	0.0132	0.9996	1.5076	379.6582	0.0162	0.9980	1.8825	484.4438	0.0195	8.2610
	RoHoAttacK-O$_{60}$	**1.0000**	**1.0221**	**250.7069**	**0.0116**	**1.0000**	**1.3267**	**334.0992**	**0.0146**	**1.0000**	**1.6190**	**416.6217**	**0.0172**	13.2176
	QuadAttacK$_{30}$	0.3540	3.4965	1097.7370	0.0249	0.1298	3.5782	1123.3760	0.0256	0.0100	3.6597	1148.9145	0.0263	**1.9552**
	RisingAttacK$_{30}$	0.9800	1.7993	458.0865	0.0191	0.9200	2.1465	556.2741	0.0222	0.8000	2.5030	658.8665	0.0253	4.1613
	RoHoAttacK-O$_{30}$	**0.9860**	**1.5114**	**384.7927**	**0.0164**	**0.9648**	**1.8423**	**477.2476**	**0.0197**	**0.8800**	**2.1025**	**553.4479**	**0.0218**	6.6581
Top-5	QuadAttacK$_{60}$	**1.0000**	2.9872	940.1374	0.0201	0.9984	3.3975	1064.7252	0.0243	0.9950	3.8203	1192.7754	0.0288	**3.4381**
	RisingAttacK$_{60}$	**1.0000**	0.8108	189.0044	0.0096	0.9992	1.0575	254.5953	0.0121	0.9970	1.3365	329.6166	0.0148	4.4027
	RoHoAttacK-O$_{60}$	**1.0000**	**0.6973**	**162.5438**	**0.0084**	**1.0000**	**0.9306**	**224.0439**	**0.0109**	**1.0000**	**1.1494**	**283.4703**	**0.0130**	7.0443
	QuadAttacK$_{30}$	0.9700	2.9571	932.7132	0.0197	0.7794	3.2526	1024.0607	0.0225	0.4770	3.5452	1114.4266	0.0252	**1.7718**
	RisingAttacK$_{30}$	0.9760	1.0890	264.4306	0.0125	0.8800	1.3450	334.9398	0.0149	0.7110	1.6168	410.8617	0.0174	2.2165
	RoHoAttacK-O$_{30}$	**1.0000**	**0.9148**	**222.1217**	**0.0108**	**0.9400**	**1.1527**	**286.3053**	**0.0129**	**0.7910**	**1.3581**	**345.1238**	**0.0150**	3.5464
Top-1	QuadAttacK$_{60}$	**1.0000**	1.1376	381.0736	0.0047	**1.0000**	1.3910	459.6084	0.0063	**1.0000**	1.6659	545.2655	0.0080	2.9955
	RisingAttacK$_{60}$	0.9990	0.2450	46.1936	0.0041	0.9794	0.3340	68.5738	0.0052	0.9420	0.4472	97.7384	0.0065	**1.2708**
	RoHoAttacK-O$_{60}$	**1.0000**	**0.2107**	**39.7265**	**0.0036**	**1.0000**	**0.2939**	**60.3449**	**0.0047**	**1.0000**	**0.3846**	**84.0550**	**0.0057**	4.7928
	QuadAttacK$_{30}$	**1.0000**	1.1372	380.9527	0.0047	0.9994	1.3899	459.3060	0.0063	0.9970	1.6631	544.4684	0.0080	1.4404
	RisingAttacK$_{30}$	**1.0000**	0.3367	67.5065	0.0052	0.9772	0.5249	114.2980	0.0073	0.9240	0.7524	171.9542	0.0099	**0.6426**
	RoHoAttacK-O$_{30}$	**1.0000**	**0.2828**	**56.7055**	**0.0040**	**1.0000**	**0.4619**	**100.5822**	**0.0057**	**1.0000**	**0.6320**	**144.4415**	**0.0069**	2.3046

Table 10: **Comparison of `RoHoAttack` component variants on ImageNet-1K.** Progressive inclusion of active rival selection, pairwise search, cross-response residuals, refinement, and projected lookahead, with results averaged over ResNet-50, DenseNet-121, ViT-B, and DeiT-B.

Variant	Design component					Evaluation metric				
	Active-K	Pairwise	Residual	Refine	PLM	ASR↑	$M_K\uparrow$	$\ell_2\downarrow$	$\ell_\infty\downarrow$	RA↓
Top-1 margin ascent	–	–	–	–	–	0.9585	4.66	1329.4	0.0251	67.5
Active rival-gradient search	✓	–	–	–	–	0.9711	5.03	1256.8	0.0235	65.3
Active pairwise averaging	✓	✓	–	–	–	0.9796	5.29	1198.7	0.0226	63.8
Raw cross-response residual	✓	✓	✓	–	–	0.9867	5.52	1148.3	0.0218	62.6
Refined residual w/o PLM	✓	✓	✓	✓	–	0.9888	5.62	1126.5	0.0214	61.8
RoHoAttacK	✓	✓	✓	✓	✓	**0.9951**	**5.90**	**1070.6**	**0.0201**	**60.4**

Effect on ordered top-K manipulation. Table 14 shows progressive improvements in ordered top-K attack effectiveness on ImageNet-1K, most pronounced at Top-20 and Top-15. With the target sequence fixed, active rival selection adapts exploration to current class competition, while pairwise cross-response residuals provide interaction-aware directions. Projected lookahead evaluates the achieved ordered objective over feasible candidates, including the current iterate. The full `RoHoAttack-O` achieves the highest mean ASR and lowest mean ℓ_∞ norm among the evaluated variants, supporting the complementary contributions of these components.

H Discussion

RoHoAttacK is designed to address a specific limitation in current adversarial evaluation: conventional top-1 and ranked-output attacks often optimize isolated losses, fixed ranking constraints, feature-space objectives, or local gradient directions, but they do not explicitly exploit the interaction geometry among active non-source classes that are closest to overtaking the source class. The central objective of RoHoAttacK is therefore not merely to induce top-1 failure, but to construct a projection-aligned adversarial evaluator that identifies locally competitive rivals, extracts

Table 11: Sensitivity to the number of active rivals K. Pair count denotes the number of active rival interactions, $\binom{K}{2}$, evaluated by RoHoAttacK on ImageNet-1K. Increasing K improves attack potency until performance saturates; the default $K = 5$ balances adversarial strength and candidate-search cost.

Active rivals K	Pair count	ASR↑	M_K ↑	ℓ_2 ↓	ℓ_∞ ↓	RA↓	Cost
$K = 1$	0	0.9625	4.87	1328.4	0.0249	66.9	**1.00**×
$K = 2$	1	0.9796	5.39	1198.3	0.0227	64.7	1.21×
$K = 3$	3	0.9900	5.70	1109.6	0.0211	62.0	1.46×
$K = 5$	10	0.9951	5.90	1070.6	0.0201	60.4	1.77×
$K = 7$	21	0.9954	5.93	1068.9	0.0200	60.2	2.44×
$K = 10$	45	**0.9956**	**5.95**	**1067.5**	**0.0200**	**60.1**	3.27×

Table 12: **Comparison of candidate-selection rules on ImageNet-1K.** Ablation of orientation search, multiple step scales, zero-step inclusion, and post-projection scoring, averaged over four classifiers. Mono. denotes the percentage of iterations with non-decreasing active rival margin.

Selection rule	Selection design				Evaluation metric				
	Signed	Multi-scale	Zero cand.	Post-proj.	ASR↑	M_K ↑	ℓ_2 ↓	ℓ_∞ ↓	Mono.↑
Max raw residual magnitude	–	–	–	–	0.9822	5.40	1190.9	0.0227	84.6
Max pre-projection margin	✓	–	–	–	0.9859	5.52	1160.4	0.0220	89.7
Projected single-step selection	–	–	✓	✓	0.9890	5.69	1124.7	0.0213	95.8
Signed multi-scale w/o zero candidate	✓	✓	–	✓	0.9923	5.80	1091.2	0.0206	97.4
Projected lookahead-margin selection	✓	✓	✓	✓	**0.9951**	**5.90**	**1070.6**	**0.0201**	**100.0**

interaction-aware directions from their pairwise margin fields, and accepts only the feasible candidate that maximizes adversarial progress under the same ℓ_p budget.

Why RoHoAttacK fulfills the adversarial vulnerability objective. The first design choice, active top-K rival selection, focuses candidate generation on the non-source classes most relevant to the current decision boundary. Refreshing this set adapts the search directions to evolving class competition while preserving the same source-displacement loss. Thus, K changes the search policy rather than the scalar objective.

The second design choice, pairwise interaction-aware direction generation, constructs search directions from reciprocal rival-gradient responses. A source-relative margin gradient describes the first-order variation of one rival against the source. `RoHoAttack` additionally probes how movement along one rival's bounded direction changes the other's normalized gradient. Refining the difference between these reciprocal responses attenuates components aligned with the original rival gradients. This provides additional directions for projected evaluation while restricting pairwise exploration to the active rival set. RoHoAttacK uses this principle to search for adversarial movement induced by pairwise rival interactions, thereby expanding the feasible direction family without enumerating all classes. This is the key reason the attack can uncover adversarial directions that single-loss, single-margin, or independently ranked-output attacks can leave underexplored.

The third design choice, projected lookahead-margin selection, determines which candidate should be trusted. RoHoAttacK does not accept a direction because it has high gradient magnitude, high residual magnitude, or favorable pre-projection behavior. Instead, each candidate is first mapped into the same ℓ_p-bounded feasible region imposed on the final adversarial example, and the selected update is the one that maximizes the achieved active rival margin after projection. Therefore, RoHoAttacK is optimal in a precise per-iteration candidate-wise sense: among the evaluated projected candidates, it selects the candidate with maximal feasible adversarial progress. Since the current iterate is included in the candidate set, the update is monotone over the evaluated family:

$$M_K(x^\star_{i+1}; y) \geq M_K(x^\star_i; y), \qquad \mathcal{L}_K(x^\star_{i+1}; y) \geq \mathcal{L}_K(x^\star_i; y). \tag{92}$$

This candidate-wise optimality does not claim global optimality of the non-convex adversarial problem; rather, it provides an objective-aligned optimal update over a rich interaction-aware candidate set at each iteration.

Why RoHoAttacK is effective against robust defenses. Robust defenses often reshape the loss landscape through stochasticity, smoothing, random projections, attention noise, or Lipschitz-constrained structures. Attacks that rely on a fixed surrogate loss or one dominant local gradient

Table 13: **Attack variant comparison against robust defenses across six benchmarks.** Robust accuracy (%) on CIFAR-10, ImageNet-1K, NIH-CXR, NCT-CRC-HE-100K, HAM10000, and EyePACS, averaged across RPF, CTRW, HyCAS, and HySCAN per dataset. Mean reports the average across datasets; lower is better.

Attack variant	CIFAR-10↓	ImageNet↓	NIH-CXR↓	NCT↓	HAM↓	EyePACS↓	Mean↓
Best existing evaluation attack	68.1	56.5	78.2	85.7	63.0	68.6	70.0
Top-1 margin ascent	65.2	55.1	77.6	84.8	61.6	67.0	68.6
Active rival-gradient search	63.5	53.9	75.9	82.9	60.1	65.5	67.0
Cross-response residual w/o PLM	62.0	52.5	72.6	80.3	58.4	63.0	64.8
RoHoAttacK	**59.2**	**48.3**	**67.4**	**74.4**	**54.1**	**58.7**	**60.4**

Table 14: **Ablation of `RoHoAttack-O` component variants for ordered top-K attacks on ImageNet-1K.** ASR across ranking lengths and mean ℓ_∞ perturbation norms under a 30-step budget, averaged over ResNet-50, DenseNet-121, ViT-B, and DeiT-B.

Variant	Top-20↑	Top-15↑	Top-10↑	Top-5↑	Top-1↑	Mean ASR↑	Mean ℓ_∞ ↓
Fixed ordered-target objective	0.386	0.704	0.781	0.842	0.956	0.734	0.0315
Active rival-ranking objective	0.462	0.765	0.833	0.875	0.975	0.782	0.0288
Active pairs + cross-response residual	0.526	0.832	0.886	0.899	0.989	0.826	0.0234
RoHoAttacK-O$_{30}$	**0.571**	**0.887**	**0.923**	**0.905**	**0.999**	**0.857**	**0.0207**

direction may underestimate such defenses when the most damaging feasible update is not aligned with that direction. RoHoAttacK addresses this by generating multiple rival-interaction directions and evaluating them only after projection. Consequently, even when a defense disperses gradients or alters the apparent local geometry, RoHoAttacK can still identify which active rival interaction remains most damaging inside the feasible perturbation region. This explains why the attack reduces robust accuracy more aggressively across stochastic and Lipschitz-constrained defenses: it evaluates the defense through an interaction-aware and projection-consistent adversarial search rather than through a single averaged or isolated update direction.

Why RoHoAttacK improves the ASR–norm trade-off. The empirical results show that RoHoAttacK improves ASR and active rival margin while reducing ℓ_p perturbation norms. This behavior follows directly from its selection rule. A large raw perturbation direction is useful only if it improves the adversarial margin after bounded mapping and projection; otherwise, it is discarded. Thus, RoHoAttacK does not spend the perturbation budget uniformly or greedily. Instead, it assigns each update to the pairwise rival direction that produces the largest feasible gain. This explains why RoHoAttacK can drive the adversarial iterate deeper into the competing-class region while requiring lower perturbation energy.

Connection to ordered top-K manipulation. Ordered top-K attacks must satisfy a fixed target sequence across multiple outputs, requiring coordinated changes in class margins. `RoHoAttack-O` retains this sequence while using active rivals to guide exploration and reciprocal gradient responses to construct interaction-aware candidates. Projected lookahead then selects the candidate with the highest achieved ordered objective, retaining the current iterate when no candidate improves it. The larger improvements observed at Top-20 and Top-15 support this combination when success requires coordinating longer target rankings.

Asymptotic cost analysis. Let C be the number of classes, D the input dimension, K the number of active rivals, $P = \binom{K}{2}$ the number of active rival pairs, $S = |\bar{\Gamma}_i|$ the number of signed lookahead steps, T the number of attack iterations, and R the number of restarts. Let F and B denote the cost of one forward and one backward evaluation of the attacked model or defense, respectively. Let $\Pi_p(D)$ denote the cost of projecting one candidate into the ℓ_p ball. For the commonly used ℓ_2 and ℓ_∞ cases, $\Pi_p(D) = \mathcal{O}(D)$; for exact ℓ_1 projection, this can be implemented with sorting-based cost $\mathcal{O}(D \log D)$.

At each iteration, active rival extraction requires one forward pass and a top-K selection over the class logits:

$$\mathcal{O}(\mathsf{F} + C \log K). \tag{93}$$

Computing normalized active margin fields costs $\mathcal{O}(K\mathsf{B})$. Pairwise interaction-aware direction generation evaluates reciprocal cross-responses for each active rival pair. In a conservative implementation, this costs

$$\mathcal{O}(2P\mathsf{B} + PD), \tag{94}$$

where the PD term accounts for residual construction and orthogonal refinement in input space. Projected lookahead-margin selection evaluates S signed candidates per active pair, giving

$$\mathcal{O}\big(PS(\mathsf{F} + C\log K + \Pi_p(D))\big). \tag{95}$$

Therefore, the total RoHoAttacK cost is

$$\mathcal{O}\left(RT\left[C\log K + \mathsf{F} + K\mathsf{B} + 2P\mathsf{B} + PD + PS(\mathsf{F} + C\log K + \Pi_p(D))\right]\right). \tag{96}$$

For ℓ_2 and ℓ_∞ evaluation, where $\Pi_p(D) = \mathcal{O}(D)$, and since $P = \binom{K}{2}$, the dominant attack-specific term scales as

$$\mathcal{O}\left(RTK^2(\mathsf{B} + S\mathsf{F} + SC\log K + SD)\right), \qquad K \ll C. \tag{97}$$

This is substantially more efficient than all-class pairwise interaction search, which would scale as $\mathcal{O}(C^2)$. For example, ImageNet-1K has $C = 1000$, while the default RoHoAttacK setting uses a small active rival set such as $K = 5$, producing only $P = 10$ active pairwise interactions rather than nearly half a million all-class pairs.

Robustness–cost effectiveness trade-off. The computational overhead of RoHoAttacK arises from its intended role as a high-potency adversarial evaluator: instead of following a single loss-gradient direction, it evaluates a compact set of active rival-interaction candidates and selects the feasible update with maximal projected adversarial progress. This cost is controlled by three design choices. First, active top-K filtering restricts the search to the locally consequential non-source classes, avoiding all-class enumeration. Second, interaction-aware direction generation is performed only over $\binom{K}{2}$ active rival pairs, not over all class pairs. Third, projected lookahead scoring evaluates a finite signed candidate set, which can be batched across pairs and step sizes.

Importantly, RoHoAttacK does not require a separate expectation-over-transformation objective or an additional stochastic averaging loop to evaluate randomized or stochastic defenses. The same active rival-margin objective, pairwise interaction-aware direction generation, and projected lookahead-margin selection are applied uniformly across standard models and robust defenses. For a randomized defense, one model evaluation simply corresponds to the defense's realized forward or backward computation, and this cost is already absorbed into F and B. Thus, stochasticity in the evaluated defense changes the cost of an individual model evaluation only as an implementation-level factor; it does not change the algorithmic structure of RoHoAttacK and does not introduce an additional multiplicative sampling factor. Consequently, the robustness–cost trade-off remains governed by the active-pair search size $P = \binom{K}{2}$ and the signed lookahead set size S, rather than by the full class count or by an external stochastic averaging procedure.

The ablation study further supports this design. Increasing K from 1 to 5 yields substantial gains in ASR, active rival margin, perturbation efficiency, and robust-defense degradation, whereas increasing K beyond 5 provides only marginal improvement while sharply increasing the pair count. Therefore, the default setting $K = 5$ provides a practical balance between attack potency and computational cost, retaining the main benefit of active rival interactions while keeping the evaluation tractable.

Applications. RoHoAttacK provides a practical adversarial evaluation tool for settings where model reliability must be assessed beyond clean accuracy or top-1 robustness. First, it can serve as a high-potency benchmark for comparing adversarial training, randomized defenses, smoothing-based architectures, Lipschitz-constrained models, stochastic inference mechanisms, and hybrid certified–empirical defenses under a common ℓ_p-bounded protocol. Second, it can support adversarial red-team evaluation of safety-critical vision systems, including medical image analysis, biometric verification, autonomous perception, fraud detection, and content moderation, where visually small perturbations may lead to clinically, financially, or operationally consequential decisions. Third, because RoHoAttacK explicitly targets active rival-class interactions, it is useful for auditing ranked-output systems such as retrieval, triage, recommendation, and multi-class decision-support pipelines, where reliability depends not only on the top-1 label but also on the ordering and confidence allocation across competing outputs. Finally, RoHoAttacK can serve as a diagnostic tool for

defense design: failure cases discovered by the attack can reveal which rival classes, local margins, or decision regions require stronger regularization or more robust architectural design.

Societal impact and responsible use. The primary societal value of RoHoAttacK is defensive. By exposing hidden vulnerabilities in both non-robust models and adversarially robust defenses, it helps developers identify failure modes before models are used in high-impact settings. This is especially important for domains such as clinical decision support, medical triage, identity verification, autonomous systems, fraud screening, and public-facing moderation, where incorrect predictions can create safety, financial, or fairness-related harms. RoHoAttacK therefore encourages a more rigorous robustness culture: a defense should not be considered reliable only because it withstands standard PGD-, AutoAttack-, EOT-, or BPDA-style evaluations, but should also be examined under interaction-aware and projection-aligned adversarial search.

At the same time, RoHoAttacK is a dual-use method. A more potent attack can be misused to degrade deployed systems if applied outside authorized evaluation settings. Accordingly, RoHoAttacK should be used for legitimate robustness assessment, controlled red-team evaluation, benchmark construction, and defense development. When applied to sensitive domains such as medical imaging, biometric verification, or identity-related systems, experiments should be conducted on approved datasets or authorized models, and reporting should emphasize aggregate robustness findings rather than actionable exploitation of specific deployed systems. In this sense, the intended contribution of RoHoAttacK is not to enable misuse, but to raise the standard for adversarial evaluation and motivate defenses that remain reliable under multi-class interaction-aware attacks.

Limitations and practical use. RoHoAttacK is intended as a high-potency adversarial evaluation method rather than a lightweight real-time attack routine. Its additional computation is appropriate when the goal is to reveal hidden failure modes of robust models, compare defenses under a more demanding white-box protocol, or generate adversarial examples for developing next-generation defenses. In practice, the default setting $K = 5$ with a small signed step set offers a favorable balance: it preserves the central benefit of active rival interactions while keeping the cost controlled through pair filtering and batched candidate evaluation. Future work may further reduce evaluation cost through adaptive pair pruning, early stopping based on projected margin saturation, and batched evaluation of interaction-aware candidates, while preserving the same active rival-margin and projection-aligned selection principle.