WHEN COMPETITOR RANKINGS MISLEAD: DYNAMIC COMPETITOR DISCOVERY FOR UNTARGETED ADVERSARIAL ATTACKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Untargeted adversarial attacks need only cross the decision boundary of a single competing class, yet determining which competitor to pursue remains challenging. Class logits provide readily available guidance, but their rankings can change during optimization. Early target commitment or score-dependent gradient weighting can therefore place excessive emphasis on transient rankings and steer optimization toward less promising competitors. We introduce **Dynamic Competitor Discovery (DCD)**, which treats target selection as a coarse-to-fine process under uncertain competitor rankings. DCD begins with a broad candidate set, progressively narrows it as optimization proceeds, and selects a single target only after sufficient evidence has accumulated. **DCD-**m extends this process to multiple targets by repeating DCD while excluding previously selected competitors. Across ten benchmark models and four datasets, DCD with PGD, APGD, GAMA, FAB, and FMN improves the corresponding base attacks in 49 of 50 ℓ_∞ attack–model comparisons under equal or lower gradient budgets, with consistent gains also observed under ℓ_2. Applied after AutoAttack, APGD-DCD9 uncovers 88 additional adversarial examples across all ten models, revealing vulnerabilities missed by AutoAttack.

1 INTRODUCTION

Adversarial examples expose a fundamental weakness of deep neural networks: carefully crafted, often imperceptible perturbations can induce confident prediction errors Szegedy et al. (2014); Goodfellow et al. (2015). Reliable white-box attacks are therefore indispensable for uncovering hidden model vulnerabilities, assessing adversarial robustness, and supporting robust model training Madry et al. (2018); Uesato et al. (2018). In the untargeted setting, an adversarial example needs to cross only one competing decision boundary, yet identifying which boundary to pursue remains difficult because deep neural networks are highly nonlinear. The challenge grows rapidly with the number of classes: ImageNet already presents 999 competitors per input, while larger label spaces may contain tens of thousands Liu et al. (2022). Under a limited gradient budget, untargeted attacks must identify promising targets from many competitors using readily available evidence such as clean-input logits, rather than exhaustively searching the output space.

Existing untargeted attacks differ in how competitor evidence is translated into optimization preference, as illustrated in Figure 1. Loss-driven attacks such as APGD-CE Croce & Hein (2020b), shown in Figure 1(a), continuously translate current class logits into gradient weights without explicit target selection. Geometry-based methods such as MALT Melamed et al. (2024), shown in Figure 1(b), use clean-input gradient and boundary-distance estimates to select targets. Multi-target methods such as APGD-T Croce & Hein (2020b), shown in Figure 1(c), exploit clean-input logits to pre-rank competitors and sequentially attack a fixed set of top-ranked targets.

Competitor logits provide useful information, but the rankings they induce can vary across attack states Melamed et al. (2024). Existing attacks may place too much emphasis on early logits. Loss-driven attacks such as APGD-CE continuously convert current score differences into varied gradient weights, so early preferences can influence subsequent optimization. By contrast, fixed-target methods such as APGD-T and MALT select competitors using only information from the original input,

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.

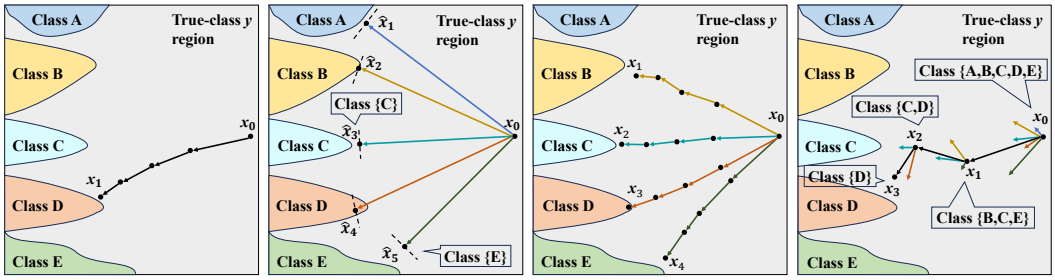


Figure 1: Varied strategies for choosing which competing class to attack. x_0 denotes the clean input. **(a) APGD-CE** adjusts competitor preference implicitly through the CE loss. **(b) MALT** estimates boundary distances at the clean input and selects one target (Class B). **(c) APGD-T** ranks competitors at the clean input and attacks a fixed set of top-ranked targets. **(d) DCD** repeatedly re-ranks competitors, shrinks the active Top-K set with mean-margin updates, and selects the final target after re-evaluation ($\{$A,B,C,D,E$\} \rightarrow \{$B,C,E$\} \rightarrow \{$C,D$\} \rightarrow$ Class D).

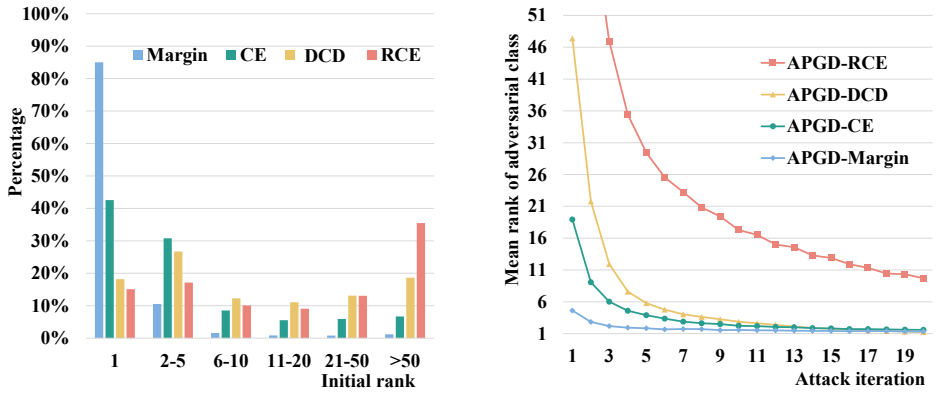


Figure 2: Competitor rankings vary across APGD objectives and evolve during optimization. Results are obtained on an ImageNet-trained ViT; among successful attacks, we track the final adversarial class, with rank 1 denoting the highest-scoring incorrect class. **(a) Initial rank distribution.** The final adversarial class is often not highly ranked at the original input. APGD-Margin favors initially top-ranked competitors, APGD-CE covers a broader range, and APGD-RCE reaches competitors that start much lower in the ranking. **(b) Rank evolution.** The mean rank of the final adversarial class improves only gradually, showing that early rankings may not reliably identify the competitor ultimately reached by the attack.

so errors in the initial ranking cannot be corrected during optimization. The key challenge is therefore to identify a promising competitor when the observed logit rankings are unreliable.

Figure 2 provides empirical evidence for this issue. The final adversarial class is often poorly ranked at the original input, indicating that initial rankings may not reveal which competitors will become favorable during optimization. Figure 2(b) shows that its rank improves only gradually. Thus, competitor rankings evolve during optimization, making early target commitment unreliable. These findings motivate a coarse-to-fine, revisable approach to target selection rather than one-shot ranking. Instead of committing early to a single competitor, the attack retains multiple plausible competitors and evaluates them jointly with uniform weighting. As optimization progresses, competitors are repeatedly re-ranked and the candidate set is progressively narrowed until a final target is selected.

Following this principle, we propose **Dynamic Competitor Discovery (DCD)**, a target-discovery method for untargeted attacks. At discovery iteration t, DCD forms the Top-K_t set from all eligible competitors, averages their pairwise margin gradients, and uses the resulting direction to update a shared discovery state. The support size K_t follows a decreasing schedule, from the full eligible set to eventually two competitors. After a fixed number of discovery iterations, DCD re-evaluates all eligible competitors at the final discovery state and selects the target with the highest margin. The base attack then restarts from the clean input and uses the remaining optimization budget for target-specific attack. To obtain multiple targets, DCD-m repeats the discovery stage while excluding previously selected targets, so each new target is identified from the remaining competitors rather than taken from a target list determined at a fixed clean-input. Experiments show that DCD-m captures a large share of its multi-target gain within the first two discovered targets, demonstrating effective use of a small target budget. DCD also strengthens multiple attack families under both ℓ_∞ and ℓ_2 threat models, showing that its effectiveness extends across different perturbation settings.

2 Related Work

Existing white-box attacks can be broadly categorized by how they identify and pursue competing classes. We organize representative approaches into three strategies relevant to competitor selection: adapting competitor preference during optimization, attacking multiple predefined targets, and selecting a target through a separate procedure. **(1) Trajectory-coupled competitor selection.** Standard untargeted attacks do not fix a target before optimization. PGD and APGD optimize objectives such as CE, DLR, or the logit margin, so competitor preference changes with the current class scores Goodfellow et al. (2015); Madry et al. (2018); Croce & Hein (2020b). GAMA gradually transitions from an auxiliary guidance term to a margin objective Sriramanan et al. (2020). Jitter Schwinn et al. (2023) perturbs output logits to encourage exploration and more diverse misclassification classes. FMN adapts the perturbation radius during margin optimization, while DeepFool and FAB use local boundary information to guide each update Moosavi-Dezfooli et al. (2016); Croce & Hein (2020a). Despite these differences, competitor preference is determined implicitly during optimization rather than through a separate target-selection step. **(2) Multi-target search.** Multi-target attacks pursue several target classes separately. MultiTargeted PGD uses pairwise objectives for different targets Gowal et al. (2019). Similarly, APGD-T, GAMA-MT, and FAB-T run separate optimizations for selected targets Croce & Hein (2020b); Sriramanan et al. (2020); Croce & Hein (2020a). Yamamura et al. (2024) further combine multi-target search with diversified directions and objectives, highlighting the importance of balancing diversification and intensification. Attacking multiple targets broadens boundary coverage. However, the coverage still depends on the chosen target set, while the computational cost grows with the number of targets. **(3) Dedicated target selection.** The Minimum-Margin (MM) attack uses Sequential Target Ranking Selection to identify a smaller set of promising classes for multi-target optimization Gao et al. (2022). MALT similarly ranks competitors using local boundary-distance estimates and selects the highest-ranked competitor as the attack target Melamed et al. (2024). MM and MALT improve competitor ranking, but still make selection decisions from potentially unreliable single-point evidence.

3 Problem Formulation

Let $z : [0,1]^d \to \mathbb{R}^C$ denote a C-class classifier on d-dimensional inputs, with logits $z(x) = [z_1(x), \ldots, z_C(x)]$, where $\mathcal{C} = \{1, \ldots, C\}$ is the class set. We consider a correctly classified clean input x_0 with label y. For an ℓ_p threat model with $p \in \{2, \infty\}$ and perturbation budget $\epsilon > 0$, the feasible set is

$$\mathcal{B}_\epsilon^p(x_0) = \left\{ x \in [0,1]^d : \|x - x_0\|_p \leq \epsilon \right\}. \tag{1}$$

Let $\mathcal{C}^- = \mathcal{C} \setminus \{y\}$ denote the set of incorrect classes, which we call *competitors*. For each competitor c, define the pairwise logit margin

$$m_c(x) = z_c(x) - z_y(x).$$

An untargeted attack succeeds whenever $\max_{c \in \mathcal{C}^-} m_c(x) > 0$ for $x \in \mathcal{B}_\epsilon^p(x_0)$. To describe the idealized promise of a competitor independently of the attack state, define its attainable margin

$$m_c^\star = \max_{x \in \mathcal{B}_\epsilon^p(x_0)} m_c(x). \tag{2}$$

Thus $m_c(x)$ is the currently observed margin, whereas $m_c^\star$ is its latent attainable counterpart over the threat set. The latter describes what is possible somewhere in the perturbation set, but does not imply that a finite-budget attack can attain the maximum. During optimization, the attacker must therefore reason from the current margins $m_c(x)$. The remaining question is how to use these state-dependent observations to allocate limited optimization budget across the $C-1$ competitors.

4 Dynamic Competitor Discovery

At the clean input x_0, the observed score ranking may not reveal which competing decision boundary is easiest to cross, so over-relying on it can lead to an irreversible targeting mistake (Section 4.1). DCD therefore uses scores only to narrow the candidate set, while assigning equal gradient weight to the retained competitors (Section 4.2). As optimization progresses, DCD repeatedly re-evaluates the scores and uses the updated rankings to progressively refine the candidate set before committing to any chosen target (Section 4.3).

4.1 State-Dependent Competitor Ranking

We first examine why the competitor ranking observed at a given attack state may not reveal which competitor has the largest attainable margin. Recall the latent reference $m_c^\star$ from equation 2. Given the current feasible state $x \in \mathcal{B}_\epsilon^p(x_0)$, define the *state-dependent shortfall* $\Delta_c(x) := m_c^\star - m_c(x) \geq 0$, equivalently $m_c(x) = m_c^\star - \Delta_c(x)$. For any pair of competitors $i, j \in \mathcal{C}^-$,

$$m_i(x) - m_j(x) = \big(m_i^\star - m_j^\star\big) - \big(\Delta_i(x) - \Delta_j(x)\big). \tag{3}$$

This identity separates the observed margin difference into two components: the difference in attainable margins and the difference in current shortfalls. Thus, a competitor with a lower observed margin can still have a larger attainable margin elsewhere in the perturbation set. Consequently, **current competitor rankings provide useful evidence, but their reliability cannot be determined from a single observation state**. This distinction exposes the core theoretical challenge addressed by DCD. Particularly, committing immediately to the Top-1 competitor may over-emphasize uncertain evidence. DCD instead preserves multiple competitors until stronger evidence emerges. Appendix A.1 analyzes how this distortion affects the resulting rankings.

4.2 Top-K Support Selection and Mean-Margin Allocation

We reformulate target selection from choosing a single competitor to maintaining a set of plausible competitors under uncertain rankings. During discovery, current scores play two distinct roles:

$$\mathcal{U} \xrightarrow[\text{selection}]{m(x)} \mathcal{S}, \qquad \mathcal{S} \xrightarrow[\text{allocation}]{m(x)} w. \tag{4}$$

Selection determines which competitors remain active, while *allocation* determines how gradient weight is distributed among them. These decisions can use increasingly detailed score information. Support membership provides the coarsest information, ordinal allocation additionally uses within-support rank, and cardinal allocation further depends on the magnitudes of score differences. DCD uses current scores to determine the active support, then weights all retained competitors equally, avoiding direct dependence on potentially unreliable score gaps. Fixed-target methods instead commit to targets from a single observation, whereas CE-style attacks keep all competitors active and allocate gradient weight according to score differences. For a competitor set $\mathcal{U} \subseteq \mathcal{C}^-$, define

$$\mathcal{S}_K(x) = \mathrm{TopK}_K\{m_c(x) : c \in \mathcal{U}\}, \qquad 1 \leq K \leq |\mathcal{U}|, \tag{5}$$

where TopK_K gives the indices corresponding to the K largest values. Top-K selection turns the difficult task of identifying a single best competitor into the easier task of retaining a plausible candidate set. The competitors within this set need not be correctly ordered. A larger K keeps more competitors, reducing the need for precise ranking. A sufficient condition for exact Top-K recovery under the attainable-margin model is given in Appendix A.2.

For any nonempty support $\mathcal{S} \subseteq \mathcal{U}$, define the fixed-support mean-margin objective

$$F(x; \mathcal{S}) := \frac{1}{|\mathcal{S}|} \sum_{c \in \mathcal{S}} m_c(x). \tag{6}$$

Here $\mathcal{S}$ denotes an already selected support, which is fixed for the current update. Uniform within-support weighting then follows directly from

$$\nabla_x F(x;\mathcal{S}) = \frac{1}{|\mathcal{S}|}\sum_{c\in\mathcal{S}} \nabla_x m_c(x). \tag{7}$$

At any differentiable point, this is a first-order ascent direction for the fixed-support mean margin; the corresponding local statement is given in Appendix A.3. Conditional on the selected support, equation 7 is unaffected by either the ordering of the active competitors or the magnitudes of their score differences. Score-sensitive allocations make progressively greater use of the same within-support evidence. Appendix A.6 analyzes rank-based and cardinal score-weighted alternatives and isolates the additional pairwise feedback introduced by unequal allocation. For context, Appendix A.6 relates the fixed-support mean-margin formulation to RCE, margin, and DLR.

4.3 Coarse-to-Fine Support Refinement and Deferred Target Commitment

Mean-margin optimization avoids resolving competitor ordering within a fixed support, but the support must ultimately be reduced to a single target. The key is to narrow the support progressively while postponing final target commitment. This motivates re-observation together with coarse-to-fine support contraction. DCD applies this coarse-to-fine idea one discovery step at a time. For a discovery phase of T_d steps, let x_t denote the current input, with x_0 equal to the clean input. At each step $t = 1, \ldots, T_d$, let K_t denote the size of the active support. The current margins at x_{t-1} define $\mathcal{S}_t = \mathrm{TopK}_{K_t}\{m_c(x_{t-1}) : c \in \mathcal{U}\}$, and, given this support, the discovery direction is

$$g_t = \nabla_x F(x_{t-1};\mathcal{S}_t) = \frac{1}{K_t}\sum_{c\in\mathcal{S}_t} \nabla_x m_c(x_{t-1}). \tag{8}$$

Here $\mathcal{S}_t$ remains fixed for the current update and is recomputed at the next observation. Each discovery step follows $x_{t-1} \to \mathcal{S}_t \to g_t \to x_t$. At x_t, all candidate competitors are re-evaluated before forming the next active support, or before selecting the final target when $t = T_d$. Thus, current scores determine support membership, while gradient allocation within the selected candidates remains uniform. A competitor excluded at one step remains eligible and can be reconsidered later.

This coarse-to-fine contraction is motivated by a reliability hypothesis. For an active set, we assume that a smaller mean-margin deficit implies less variation in the competitors' shortfalls from their attainable margins. Because pairwise ranking distortion depends on differences between these shortfalls, lower variation yields a tighter bound for resolving competitor rankings. Under this hypothesis, increasing the fixed-support mean margin tightens the reliability bound, while reducing the Top-K support further lowers the threshold for resolving pairwise rankings among the retained competitors. Appendix A.4 formalizes the hypothesis and derives the pairwise ranking bound. Appendix A.5 further shows that, under a local affine model, sufficient gradient alignment guarantees recovery of the attainable Top-K competitors.

Driven by this reliability view, DCD realizes deferred commitment through coarse-to-fine support contraction. Re-observation makes this refinement state-adaptive. Early in discovery, a broad support keeps uncertain competitors active. As optimization progresses, the support is reduced while all eligible competitors are re-evaluated. The default implementation therefore follows the non-increasing sequence $K_1 \geq K_2 \geq \cdots \geq K_{T_d}$, gradually concentrating the discovery update on fewer competitors. Because all eligible competitors are re-evaluated, a competitor excluded earlier can still be reconsidered before final target selection. The canonical implementation uses the full eligible set at the first step, $K_1 = |\mathcal{U}|$, and ends at $K_{T_d} = K_f = 2$. After the final iteration, every competitor is re-evaluated and the target is selected as $\hat{c} = \arg\max_{c\in\mathcal{U}} m_c(x_{T_d})$. Once discovery is complete, the base attack restarts from the clean input x_0 and optimizes the targeted margin $m_{\hat{c}}(x) = z_{\hat{c}}(x) - z_y(x)$. The discovery iterations therefore update the information used for target selection, while the subsequent targeted attack restarts from the clean input.

Algorithms 1 and 2 provide implementation details for DCD and its multi-target extension. In Algorithm 1, $\Pi_{\mathcal{B}_\varepsilon^p(x_0)}$ denotes projection onto the feasible set, and $\mathcal{D}_p$ denotes the norm-specific direction, with $\mathcal{D}_\infty(g) = \mathrm{sign}(g)$ and $\mathcal{D}_2(g) = g/\|g\|_2$ for $g \neq 0$, and $\mathcal{D}_2(0) = 0$. Lines 1–3 specify the final support size and the geometric support schedule. At each discovery step, Line 4 recomputes the Top-K_t support over the candidate set, and Line 5 averages the margin gradients.

Algorithm 1 Dynamic Competitor Discovery

Require: Input x_0, true label y, candidate set $\mathcal{U} \subseteq \mathcal{C}^-$, radius ϵ, norm $p \in \{2, \infty\}$, budget $T_d \geq 2$
Ensure: Discovered target $\hat{c}$
1: $K_f \leftarrow \min\{2, |\mathcal{U}|\}$
2: **for** $t = 1, \ldots, T_d$ **do**
3: $\quad K_t \leftarrow \max\{K_f, \lceil |\mathcal{U}|^{1-\frac{t-1}{T_d-1}} \rceil\}$
4: $\quad \mathcal{S}_t \leftarrow \mathrm{TopK}_{K_t}\{m_c(x_{t-1}) : c \in \mathcal{U}\}$ {refresh over all candidate competitors}
5: $\quad g_t \leftarrow \frac{1}{K_t} \sum_{c \in \mathcal{S}_t} \nabla_x m_c(x_{t-1})$ {uniform within-support allocation}
6: $\quad \beta_t \leftarrow 4 - 3\frac{t-1}{T_d-1}$
7: $\quad x_t \leftarrow \Pi_{\mathcal{B}^p_\epsilon(x_0)}(x_{t-1} + \beta_t \epsilon \mathcal{D}_p(g_t))$
8: **end for**
9: $\hat{c} \leftarrow \arg\max_{c \in \mathcal{U}} m_c(x_{T_d})$ {point-valued commitment after endpoint re-evaluation}
10: **return** $\hat{c}$

Algorithm 2 DCD-m with Base Attack

Require: Clean input x_0, true label y, base attack $\mathcal{A}$, radius ϵ, norm $p \in \{2, \infty\}$, target number $1 \leq m \leq C-1$, total per-target budget T, DCD budget per target $2 \leq T_d < T$
Ensure: Adversarial example or attack failure
1: $\mathcal{U}_1 \leftarrow \mathcal{C}^-$; $\quad T_a \leftarrow T - T_d$
2: **for** $r = 1, \ldots, m$ **do**
3: $\quad \hat{c}_r \leftarrow \mathrm{DCD}(x_0, y, \mathcal{U}_r, \epsilon, p, T_d)$ {Algorithm 1}
4: $\quad \mathcal{U}_{r+1} \leftarrow \mathcal{U}_r \setminus \{\hat{c}_r\}$
5: $\quad x_r^{\mathrm{adv}} \leftarrow \mathcal{A}(x_0, y, \hat{c}_r; T_a)$
6: $\quad$ **if** $\arg\max_c z_c(x_r^{\mathrm{adv}}) \neq y$ **then**
7: $\quad\quad$ **return** x_r^{adv}
8: $\quad$ **end if**
9: **end for**
10: **return** Attack failure

Lines 6–7 apply the linearly decaying step size and project the normalized update back onto the perturbation set. After the final discovery update, Line 9 re-evaluates all candidates and selects the highest-margin competitor. Alternative support schedules are defined in Appendix B, while alternative within-support weighting rules are discussed in Appendix A.6 and evaluated in Table 12.

Algorithm 2 specifies the multi-target extension. Line 1 initializes the full candidate set and splits the per-target budget into discovery and attack budgets. For each target attempt, Line 3 runs a fresh DCD procedure on the remaining candidate set, Line 4 removes the discovered target, and Line 5 restarts the base attack from x_0 using the remaining budget. Lines 6–8 terminate immediately upon a successful attack; otherwise, the next iteration performs a new discovery run over the reduced candidate set. Thus, later targets are rediscovered rather than taken from a fixed initial ranking. With at most m target attempts, the worst-case computational cost scales as $O(mT)$ for $m \leq C-1$.

5 Experiments

5.1 Settings

Models, datasets, and threat models. We evaluate DCD on 16 classifiers across CIFAR-10, CIFAR-100, ImageNet, and ImageNet-12K, spanning natural trained DNNs, robust CNNs and transformers, a k-WTA gradient-masking stress test, and classifiers with a large output space. For the main ℓ_∞ evaluation, we use $\epsilon = 8/255$ on CIFAR-10 and CIFAR-100, $\epsilon = 4/255$ on ImageNet, and $\epsilon = 0.15/255$ on ImageNet-12K. We evaluate 10,000 images on CIFAR-10, CIFAR-100, and ImageNet-12K and 5,000 on ImageNet, using the same random seed for all attacks. We additionally test ℓ_2 generalization on CIFAR-10 and ImageNet. Exact model identifiers, architectures, sources, computational environments and threat-model details are given in Table 5 and Appendix C.1

Table 1: **DCD across attacks and models.** Robust accuracy (%); lower is better. Gray rows indicate DCD variants, bold values mark the best result within each attack block, and Avg. reports the unweighted mean over the 10 benchmark models.

Method	CIFAR-10		CIFAR-100		ImageNet			ImageNet-12K			Avg.
	Carm.	Rebu.	Gowal	Rice	Engs.	Salm.	MIM.	CXN	CXC	ViTC	
Clean Acc.	89.69	92.23	69.15	53.83	63.48	64.68	79.92	43.54	49.08	48.81	65.44
PGD-CE	64.25	70.82	41.08	22.06	35.64	41.28	62.36	19.30	19.16	21.07	39.70
PGD-Margin	62.36	68.72	39.37	21.09	35.10	40.12	61.72	17.74	18.67	20.16	38.51
PGD-DCD1	**61.49**	**67.80**	**37.57**	**19.82**	**31.60**	**37.72**	**59.94**	**16.00**	**16.50**	**18.60**	**36.70**
APGD-CE	61.75	69.44	40.02	20.59	33.22	39.92	61.32	17.15	16.37	18.27	37.80
APGD-Margin	60.66	67.72	38.83	20.26	33.68	39.12	61.02	16.34	16.71	17.57	37.19
APGD-DCD1	**59.80**	**66.78**	**37.01**	**19.08**	**30.36**	**36.82**	**59.40**	**14.75**	**14.89**	**16.48**	**35.54**
GAMA	59.81	66.91	37.37	19.20	30.72	36.92	59.42	14.97	15.32	17.09	35.77
GAMA-DCD1	**59.78**	**66.86**	**37.09**	**19.13**	**30.50**	**36.82**	**59.40**	**14.87**	**15.18**	**16.96**	**35.66**
FAB	60.95	67.63	37.72	19.55	**31.18**	37.46	60.04	16.48	18.71	21.44	37.12
FAB-DCD1	**60.48**	**67.25**	**37.47**	**19.41**	31.26	**37.40**	**59.90**	**16.31**	**18.25**	**20.83**	**36.86**
FMN	61.18	67.92	38.82	20.40	34.70	39.64	61.74	16.70	17.16	19.22	37.75
FMN-DCD1	**60.57**	**67.15**	**37.25**	**19.35**	**32.56**	**38.04**	**61.20**	**15.12**	**15.39**	**18.41**	**36.50**

Attack configuration. Unless stated otherwise, iterative white-box attacks use a per-trajectory budget of $T = 100$ iterations. For DCD-enhanced attacks, each trajectory allocates $T_d = 15$ iterations to discovery and the remaining $T_a = 85$ to the corresponding targeted base attack; after discovery, the base attack restarts from the clean input, so the discovery process selects the target without providing an optimization warm start. The discovery step size decreases linearly from $\beta_1 = 4$ to $\beta_{15} = 1$. DCD-1 uses one discovered target, whereas DCD-m sequentially discovers and pursues up to m distinct targets. For multi-target comparisons, we set $m = 9$ by default and, whenever a direct comparison is possible, match both the maximum number of target trajectories and the optimization budget per target; Figure 3 explicitly varies m. MALT considers at most 100 candidate classes during target ranking. For the ℓ_2 evaluations, all methods use the same per-trajectory budget of $T = 100$, with one restart for single-trajectory attacks and matched target budgets for APGD-T and APGD-DCD-m. Table 2 reports the resulting gradient and forward-evaluation costs.

Table 2: Gradient (G) and forward (F) evaluations.

Method	G/F
APGD-CE	101/101
APGD-DCD1	101/102
FAB	900/201
FAB-DCD1	100/187
FMN	100/102
FMN-DCD1	100/103
MultiTargeted	900/909
APGD-T	909/909
APGD-DCD9	909/918
MM+	909/909
MALT CIFAR-10	918/918
MALT CIFAR-100	1008/1008
MALT ImageNet/12K	1009/1009
GAMA-MT	900/918
FAB-T	900/1809
FAB-DCD9	900/1683

5.2 Main Results

DCD strengthens diverse base attacks. Table 1 shows that DCD strengthens diverse base attacks across the 10 benchmark models. Relative to the baselines in the same attack family (PGD, APGD, GAMA, FAB, and FMN), DCD1 lowers robust accuracy in 49 of 50 attack–model pairs; the only exception is FAB on Engs., where FAB-DCD1 is weaker by 0.08 percentage points (pp). The gains are substantial for PGD, APGD, and FMN: APGD-DCD1 improves over APGD-CE by 2.26 pp on average, while PGD-DCD1 and FMN-DCD1 improve by 3.00 and 1.25 pp, respectively. The consistent performance across optimizers shows that DCD is not tied to a particular algorithm. Appendix C.4 (Table 8) shows that DCD remains effective under the ℓ_2 threat model, consistently strengthening the corresponding base attacks across the evaluated models. These results demonstrate that the benefits of DCD generalize beyond the ℓ_∞ setting.

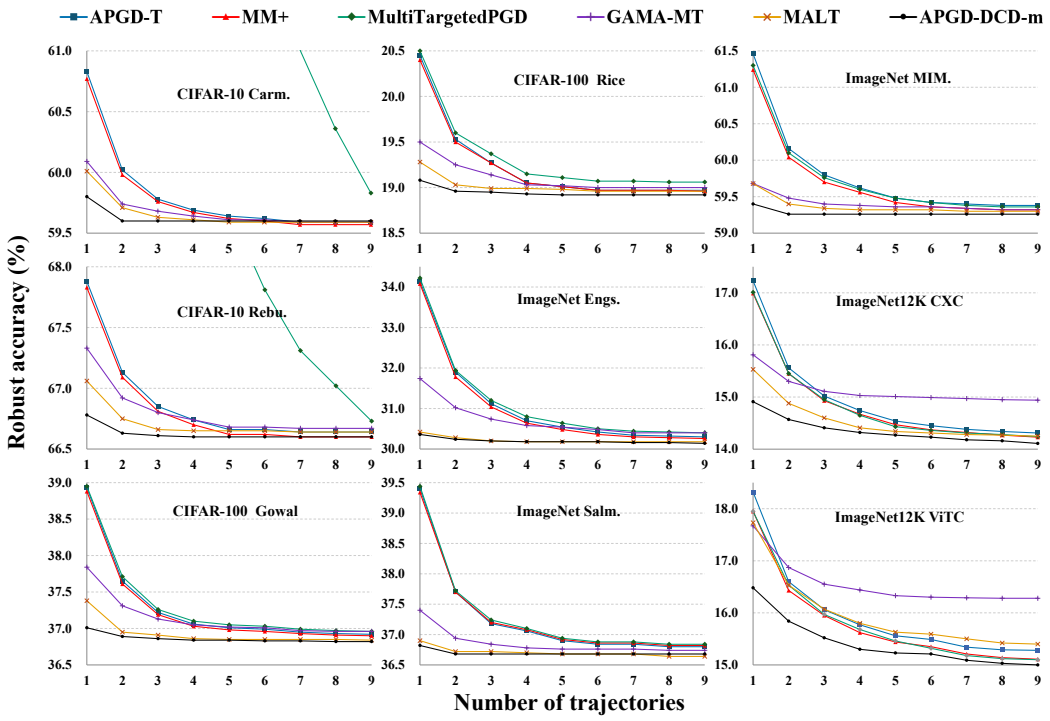


Figure 3: Multi-target scaling. Robust accuracy subject to the maximum allowed number of target trajectories; lower is better.

DCD efficiently exploits multiple target trajectories. Figure 3 compares APGD-DCD-m with APGD-T, MM+, MultiTargeted, GAMA-MT, and MALT across nine models. On seven of the nine models, APGD-DCD-m remains at or below all competing methods throughout the full trajectory range. The two exceptions are Carm. and Salm., where DCD is best or competitive through the first four trajectories. More importantly, APGD-DCD-m obtains most of its multi-target benefit with few trajectories. On Carm. and MIM., two DCD trajectories essentially reach the final $m = 9$ performance; on most other models, three to four trajectories capture the large majority of the gain, whereas several baselines stabilize only around seven to eight trajectories. On Gowal and Engs., two DCD trajectories already match or surpass APGD-T with nine fixed targets, suggesting that DCD can identify vulnerable competitors using far less search budget.

The exact $m = 9$ endpoint values in Appendix C.2 (Table 6) provide a quantitative complement to the scaling behavior in Figure 3. APGD-DCD9 and FAB-DCD9 each match or improve their fixed-target counterparts on 9/10 models, with the only degradation being 0.01 pp on CIFAR-10 Carm. CIFAR-10 has only nine incorrect classes, so a nine-target setting already evaluates every possible target, leaving little room for further gains from improved target selection. In contrast, APGD-DCD9 improves over APGD-T by $0.10/0.05$ pp on CIFAR-100, $0.16/0.12/0.12$ pp on ImageNet, and $0.24/0.20/0.28$ pp on ImageNet-12K. Thus, the advantage becomes more pronounced when exhaustive target search is infeasible. Among the evaluated multi-target attacks, APGD-DCD9 attains the best or tied-best robust accuracy on eight of the ten benchmark models. MALT is marginally stronger in a few cases, but its target-ranking stage requires additional gradient evaluations—up to about 100 in higher-class settings (Table 2). The k-WTA results are reported separately as a gradient-masking stress test in Appendix C.5 (Table 9), outside the standard regime for gradient-based DCD.

Complementarity with AutoAttack. Table 3 applies APGD-DCD9 only to examples that remain robust after AutoAttack. DCD9 further reduces robust accuracy on all 10 evaluated models, discovering 88 additional adversarial examples in total among the AutoAttack survivors. The additional successes are especially pronounced on CXC and ViTC, which have large output spaces, where APGD-DCD9 finds 20 and 26 additional adversarial examples, respectively. Because these exam-

Table 3: **Additional adversarial examples beyond AutoAttack.** Robust accuracy (%) before and after applying APGD-DCD9 to examples that remain robust to AutoAttack; lower is better. "Extra" reports the number of additional adversarial examples found by APGD-DCD9.

	CIFAR-10		CIFAR-100		ImageNet			ImageNet-12K		
Method	Carm.	Rebu.	Gowal	Rice	Engs.	Salm.	MIM.	CXN	CXC	ViTC
AUTO	59.56	66.63	36.90	18.92	30.24	36.76	59.28	14.27	14.12	14.90
AUTO+DCD9	**59.51**	**66.58**	**36.82**	**18.89**	**30.14**	**36.64**	**59.26**	**14.18**	**13.92**	**14.64**
Extra	5	5	8	3	5	6	1	9	20	26

ples remain robust against standard AutoAttack, the gains demonstrate genuinely complementary coverage, making DCD a strong complementary attack mechanism.

Objective-specific failure modes. Table 7 in Appendix C.3 compares APGD-DCD1 separately with APGD-CE, APGD-Margin, and APGD-RCE on six models. For each comparison, we count adversarial examples found by APGD-DCD1 but missed by the compared method, and vice versa. Across the six models, APGD-DCD1 finds 1178, 970, and 4264 adversarial examples missed by APGD-CE, APGD-Margin, and APGD-RCE, respectively. Conversely, APGD-CE, APGD-Margin, and APGD-RCE find only 103, 100, and 41 examples missed by APGD-DCD1. The target-agreement analysis helps explain why the compared attacks fail. For each example found only by APGD-DCD1, we compare DCD's selected target with the competitor ranked highest by the compared attack. The two classes match in 73.5% of the APGD-CE comparisons and 75.6% of the APGD-RCE comparisons, but in only 0.5% of the APGD-Margin comparisons. This suggests that APGD-CE and APGD-RCE often identify the same promising competitor as DCD but fail to exploit it effectively, whereas APGD-Margin usually focuses on a different competitor. The large differences in target agreement suggest that these attacks fail for different reasons. For APGD-CE and APGD-RCE, high agreement implies that many failures involve the same competitor selected by DCD. However, gradient weight remains distributed across multiple classes, limiting optimization toward that target. In contrast, APGD-Margin rarely selects the same competitor as DCD, suggesting that its failures more often stem from target selection. DCD addresses both failure sources sequentially: its discovery stage identifies a vulnerable competitor, after which the base attack restarts from x_0 and commits the remaining budget to that target. The corresponding analysis therefore shows that DCD benefits from both more effective target discovery and stronger target-centric optimization.

Mechanism Ablations of DCD. We further analyze the key design choices of DCD through a series of ablation studies in Appendix C.8. The ablations identify progressive support contraction as the strongest mechanism-level factor, with dynamic re-observation providing an additional benefit and concentrated score-dependent allocation consistently hurting performance. These findings suggest that DCD's gains mainly arise from adaptive competitor discovery and coarse-to-fine support refinement. Sensitivity studies further show that DCD is stable to the discovery-step coefficients and to the discovery budget (Appendices C.6 and C.7).

6 Conclusion

We introduced **Dynamic Competitor Discovery (DCD)**, which treats target selection as a process that should evolve with adversarial optimization. Our key observation is that scores and rankings at a single attack state are useful, but not fully reliable. Trusting them too strongly can lock the attack onto an initial rankings error. DCD therefore uses ranking information with limited commitment. It keeps a set of plausible competitors, gradually narrows this set as optimization progresses, and assigns them equal gradient weight within the active support. This allows the attack to use score information without repeatedly amplifying small, state-dependent ranking differences. Across the evaluated benchmarks, DCD strengthens diverse base attacks, works effectively with multi-target budgets, and finds adversarial examples missed by existing attacks. More broadly, our results suggest that adversarial target selection should depend not only on which competitor currently looks strongest, but also on how much that temporary evidence should be trusted.

AI USE STATEMENT

Generative AI tools were used to assist with proofreading, improving the readability of the manuscript, checking presentation consistency, and identifying potentially relevant literature. All referenced works were independently verified by the authors. Generative AI tools were not used to generate experimental results or determine the scientific conclusions of this work. The authors reviewed all AI-assisted content and take full responsibility for the final manuscript.

REPRODUCIBILITY STATEMENT

We provide the complete DCD algorithms in Section 4, implementation details and support schedules in Appendix B, and model, threat-model, evaluation, and computational-environment details in Appendix C.1. Additional hyperparameter sensitivity studies and ablations are reported in Appendices C.6–C.8.

REFERENCES

Maximilian Augustin, Alexander Meinke, and Matthias Hein. Adversarial robustness on in- and out-distribution improves explainability. In *European Conference on Computer Vision (ECCV)*, pp. 228–245, 2020.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2818–2829, 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 (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2196–2205, 2020a.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2206–2216, 2020b.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: A standardized adversarial robustness benchmark. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Logan Engstrom, Andrew Ilyas, Hadi Salman, Shibani Santurkar, and Dimitris Tsipras. Robustness (python library), 2019. URL `https://github.com/MadryLab/robustness`.

Ruize Gao, Jiongxiao Wang, Kaiwen Zhou, Feng Liu, Binghui Xie, Gang Niu, Bo Han, and James Cheng. Fast and reliable evaluation of adversarial robustness with minimum-margin attack. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pp. 7144–7163, 2022.

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

Sven Gowal, Jonathan Uesato, Chongli Qin, Po-Sen Huang, Timothy Mann, and Pushmeet Kohli. An alternative surrogate loss for pgd-based adversarial testing. *arXiv preprint arXiv:1910.09338*, 2019.

Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11976–11986, 2022.

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

Odelia Melamed, Gilad Yehudai, and Adi Shamir. MALT powers up adversarial attacks. In *Advances in Neural Information Processing Systems*, volume 37, pp. 44366–44392, 2024. doi: 10.52202/079017-1408.

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 (CVPR)*, pp. 2574–2582, 2016.

Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A. Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*, 2021.

Leslie Rice, Eric Wong, and J. Zico Kolter. Overfitting in adversarially robust deep learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8093–8104. PMLR, 2020.

Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Leo Schwinn, René Raab, An Nguyen, Dario Zanca, and Bjoern Eskofier. Exploring misclassifications of robust neural networks to enhance adversarial attacks. *Applied Intelligence*, 53:19843–19859, 2023. doi: 10.1007/s10489-023-04532-5.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Gaurang Sriramanan, Sravanti Addepalli, Arya Baburaj, and R. Venkatesh Babu. Guided adversarial attack for evaluating and enhancing adversarial defenses. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6105–6114, 2019.

Jonathan Uesato, Brendan O'donoghue, Pushmeet Kohli, and Aaron Oord. Adversarial risk and the dangers of evaluating against weak attacks. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5025–5034. PMLR, 2018.

Chang Xiao, Peilin Zhong, and Changxi Zheng. Enhancing adversarial defense by k-winners-take-all. In *International Conference on Learning Representations*, 2020.

Xiaoyun Xu, Shujian Yu, Zhuoran Liu, and Stjepan Picek. Mimir: Masked image modeling for mutual information-based adversarial robustness. In *Proceedings of the 33rd Annual Network and Distributed System Security Symposium (NDSS)*, 2026. doi: 10.14722/ndss.2026.241813.

Keiichiro Yamamura, Issa Oe, Nozomi Hata, Hiroki Ishikura, and Katsuki Fujisawa. Appropriate balance of diversification and intensification improves performance and efficiency of adversarial attacks. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=mK6TwmInTg.

A Additional Analysis of Dynamic Competitor Discovery

This appendix develops the theoretical argument in the same order as the DCD design. State-dependent ranking distortion motivates retaining a Top-K set of plausible competitors, while uniform mean-margin allocation avoids making an additional score-sensitive commitment within that support. A local reliability hypothesis then gives a coarse-to-fine interpretation by relating the supported mean-margin deficit and support size to sufficient ranking-resolution bounds. We finally provide a complementary geometric justification in an idealized affine regime. Supporting analyses of score-sensitive weighting, temperature effects on CE objectives, and relations to standard attack objectives are collected at the end to keep the main argument focused. Throughout, the analysis is local and explanatory: its purpose is to clarify the design choices behind DCD, not to establish convergence or optimal target identification.

A.1 State-Dependent Ranking Distortion

equation 3 can be written as

$$\underbrace{m_i - m_j}_{\text{observed score difference}} = \underbrace{m_i^\star - m_j^\star}_{\text{latent attainable-margin signal}} - \underbrace{(\Delta_i - \Delta_j)}_{\text{state-dependent distortion}}, \tag{9}$$

where all state-dependent quantities are evaluated at the current discovery state. The observed ordering need not match the attainable-margin ordering:

$$m_i(x) > m_j(x) \quad \not\Rightarrow \quad m_i^\star > m_j^\star. \tag{10}$$

For a non-tied pair, the two orderings agree exactly when

$$\big(m_i - m_j\big)\big(m_i^\star - m_j^\star\big) > 0. \tag{11}$$

A sufficient condition is

$$|\Delta_i - \Delta_j| < |m_i^\star - m_j^\star|, \tag{12}$$

although it is not necessary because the sign of the distortion also matters.

These identities make precise the observation-error interpretation used in the main text. Current score gaps can be informative when relative shortfall distortion is modest, but their reliability may vary across competitor pairs, discovery states, architectures, and training procedures. Because $m_c^\star$ and $\Delta_c(x)$ are latent, the attack cannot assess this reliability directly.

This uncertainty motivates retaining a set of plausible competitors instead of committing to a single competitor from a fixed ranking. DCD refreshes this Top-K_t support from the scores observed at each discovery iteration. In DCD-m, a target selected in a completed run is additionally removed from the next eligible set $\mathcal{U}_{r+1}$ (Section 4.3).

A.2 Top-K Support Recovery

Cross-boundary Top-K recovery. Recall from equation 3 that, for any pair $i, j \in \mathcal{U}$,

$$m_i(x) - m_j(x) = \big(m_i^\star - m_j^\star\big) - \big(\Delta_i(x) - \Delta_j(x)\big).$$

This identity shows how state-dependent differences in shortfall can distort the latent attainable-margin ranking.

For $1 \le K < |\mathcal{U}|$, let

$$\mathcal{T}_K^\star := \mathrm{TopK}_K\{m_c^\star : c \in \mathcal{U}\}, \tag{13}$$

and let

$$\gamma_K^\star := m_{(K)}^\star - m_{(K+1)}^\star, \tag{14}$$

where $m_{(1)}^\star \ge \cdots \ge m_{(|\mathcal{U}|)}^\star$ are the attainable margins sorted over $\mathcal{U}$. Assume $\gamma_K^\star > 0$, so that the latent Top-K set is unique.

Exact recovery of $\mathcal{T}_K^\star$ is guaranteed if every competitor in the latent Top-K set outranks every competitor outside it:

$$m_i(x) > m_j(x), \qquad \forall i \in \mathcal{T}_K^\star,\ j \notin \mathcal{T}_K^\star. \tag{15}$$

By equation 3, it is sufficient that

$$m_i^\star - m_j^\star > \Delta_i(x) - \Delta_j(x), \qquad \forall i \in \mathcal{T}_K^\star,\ j \notin \mathcal{T}_K^\star. \tag{16}$$

Since $m_i^\star - m_j^\star \ge \gamma_K^\star$ for every such cross-boundary pair, the compact condition

$$\gamma_K^\star > \max_{\substack{i \in \mathcal{T}_K^\star \\ j \notin \mathcal{T}_K^\star}} \big(\Delta_i(x) - \Delta_j(x)\big) \tag{17}$$

implies

$$\mathrm{TopK}_K\{m_c(x) : c \in \mathcal{U}\} = \mathcal{T}_K^\star. \tag{18}$$

Thus, exact Top-K membership depends only on relative distortion across the Top-K boundary, rather than on the full ordering of all competitors. The stronger all-pairs condition

$$\gamma_K^\star > \max_{i,j \in \mathcal{U}} |\Delta_i(x) - \Delta_j(x)| \tag{19}$$

is an immediate conservative corollary. Because $m_c^\star$ and $\Delta_c(x)$ are latent, these conditions characterize ranking recovery rather than providing an operational certificate to the attack.

These conditions make precise why DCD can defer commitment to a single competitor: Top-K membership depends on cross-boundary comparisons, not on correctly ordering the competitors within the retained support. Once a plausible support has been retained, the next question is how to allocate the discovery update within that support without introducing another score-sensitive commitment.

A.3 Uniform Allocation over the Retained Support

DCD addresses this allocation question with the fixed-support mean-margin objective $F(x;\mathcal{S})$ from equation 6. Conditional on the selected support, its gradient weights all retained competitors equally and is therefore independent of their within-support ordering and score-gap magnitudes. The following local properties characterize this direction and its relation to Top-K support selection.

Variational form and local gradient. To characterize support selection and support value jointly, define, for $1 \le K \le |\mathcal{U}|$,

$$\begin{aligned} \Phi_K(x) &:= \max_{\substack{\mathcal{S} \subseteq \mathcal{U} \\ |\mathcal{S}| = K}} F(x;\mathcal{S}) \\ &= F(x;\mathcal{S}_K(x)) = \frac{1}{K} \sum_{c \in \mathcal{S}_K(x)} m_c(x), \end{aligned} \tag{20}$$

where $\mathcal{S}_K(x) = \mathrm{TopK}_K\{m_c(x) : c \in \mathcal{U}\}$. The variational identity holds because the mean of any K margins is maximized by selecting the K largest ones.

For $K < |\mathcal{U}|$, suppose the K-th and $(K+1)$-th largest margins are distinct at x. By continuity of the margins, the active Top-K support is unchanged in a sufficiently small neighborhood of x. Hence, in that neighborhood,

$$\Phi_K(x') = F(x';\mathcal{S}_K(x)). \tag{21}$$

At a differentiable point,

$$\nabla_x \Phi_K(x) = \nabla_x F(x;\mathcal{S}_K(x)) = \frac{1}{K} \sum_{c \in \mathcal{S}_K(x)} \nabla_x m_c(x). \tag{22}$$

For $K = |\mathcal{U}|$, the support is fixed by construction and the same gradient identity holds. At a support-boundary tie, Φ_K need not be ordinarily differentiable, so the gradient identity above is restricted to non-tied states when $K < |\mathcal{U}|$.

Norm-specific local ascent. For any fixed nonempty support $\mathcal{S}$, let

$$g := \nabla_x F(x; \mathcal{S}). \tag{23}$$

For $p \in \{2, \infty\}$, let q denote the dual norm index and define

$$\mathcal{D}_p(g) \in \arg\max_{\|d\|_p \le 1} \langle g, d \rangle. \tag{24}$$

By duality,

$$\max_{\|d\|_p \le 1} \langle g, d \rangle = \|g\|_q. \tag{25}$$

Therefore, at a differentiable point and for sufficiently small $\eta > 0$,

$$F(x + \eta \mathcal{D}_p(g); \mathcal{S}) = F(x; \mathcal{S}) + \eta \|g\|_q + o(\eta), \tag{26}$$

where $o(\eta)/\eta \to 0$ as $\eta \to 0$.

For the ℓ_∞ threat model,

$$\mathcal{D}_\infty(g) = \operatorname{sign}(g), \tag{27}$$

and therefore

$$F(x + \eta \operatorname{sign}(g); \mathcal{S}) = F(x; \mathcal{S}) + \eta \|g\|_1 + o(\eta). \tag{28}$$

For the ℓ_2 threat model, when $g \neq 0$,

$$\mathcal{D}_2(g) = \frac{g}{\|g\|_2}, \tag{29}$$

and

$$F\left(x + \eta \frac{g}{\|g\|_2}; \mathcal{S}\right) = F(x; \mathcal{S}) + \eta \|g\|_2 + o(\eta). \tag{30}$$

For $g = 0$, we take $\mathcal{D}_2(0) = 0$, consistent with the DCD update. Thus, on a fixed support, the pre-projection DCD direction is a first-order steepest-ascent direction under the corresponding threat-model norm. When the Top-K support remains locally unchanged, the same statement applies to Φ_K.

The actual DCD update additionally applies projection and recomputes the active support across iterations. Accordingly, the local result above does not imply monotonic increase of the fixed- or changing-support objective along the full finite-step DCD trajectory.

The two endpoints of the Top-K value function are $\Phi_{|\mathcal{U}|}(x) = F(x; \mathcal{U})$, the full-support mean margin related to reverse cross-entropy (RCE) in Appendix A.6, and $\Phi_1(x) = \max_{c \in \mathcal{U}} m_c(x)$, the ordinary untargeted max-margin objective restricted to $\mathcal{U}$. The canonical support schedule (equation 91) contracts to $K_f = 2$, retaining a small multi-competitor discovery support late in the trajectory. Under refreshed DCD, this terminal support controls only the final discovery gradients: target commitment separately re-evaluates all candidate competitors at the endpoint.

These local properties characterize the discovery update once a support has been selected. The remaining question is why that support should narrow as discovery proceeds. We address this next through a local reliability model.

A.4 Coarse-to-Fine Support Contraction under a Local Reliability Hypothesis

We now formalize the local reliability model underlying coarse-to-fine support contraction. The pairwise shortfall difference $\Delta_i - \Delta_j$ acts as the state-dependent ranking distortion. For the eligible set $\mathcal{U}$, define the mean shortfall

$$\bar{\Delta}(x) = \frac{1}{|\mathcal{U}|} \sum_{c \in \mathcal{U}} \Delta_c(x). \tag{31}$$

A common shift of all shortfalls changes every observed margin equally and therefore does not affect ranking. For any nonempty support $\mathcal{S} \subseteq \mathcal{U}$ of size K, define the centered-shortfall dispersion

$$\mathsf{Err}_{\mathcal{S}}(x) = \left(\frac{1}{K} \sum_{c \in \mathcal{S}} \big(\Delta_c(x) - \bar{\Delta}(x)\big)^2 \right)^{1/2}, \tag{32}$$

where the centering is taken over $\mathcal{U}$, rather than recomputed within $\mathcal{S}$. For the same fixed support, use the fixed-support mean margin $F(x;\mathcal{S})$ from equation 6. Let $[u]_+ := \max\{u, 0\}$. For the family of observation states and nonempty supports under consideration, suppose there exists a *shared* nondecreasing function $h : \mathbb{R}_+ \to \mathbb{R}_+$ such that, for every such pair $(x, \mathcal{S})$,

$$\mathsf{Err}_{\mathcal{S}}(x) \le h([-F(x;\mathcal{S})]_+)\,. \tag{33}$$

This is a *modeling hypothesis*, not a consequence of the attainable-margin decomposition. It formalizes the design intuition that states with a smaller supported mean-margin deficit may admit a tighter bound on relative ranking distortion.

Under this hypothesis,

$$\max_{c \in \mathcal{S}} \big|\Delta_c(x) - \bar{\Delta}(x)\big| \le \sqrt{K}\, \mathsf{Err}_{\mathcal{S}}(x) \le \sqrt{K}\, h([-F(x;\mathcal{S})]_+)\,. \tag{34}$$

Hence, for any $i, j \in \mathcal{S}$, Cauchy–Schwarz gives

$$\begin{aligned} \big|\big(m_i(x) - m_j(x)\big) - \big(m_i^\star - m_j^\star\big)\big| &= |\Delta_i(x) - \Delta_j(x)| \\ &= \big|\big(\Delta_i(x) - \bar{\Delta}(x)\big) - \big(\Delta_j(x) - \bar{\Delta}(x)\big)\big| \\ &\le \sqrt{2}\left[\big(\Delta_i(x) - \bar{\Delta}(x)\big)^2 + \big(\Delta_j(x) - \bar{\Delta}(x)\big)^2\right]^{1/2} \\ &\le \sqrt{2K}\, h([-F(x;\mathcal{S})]_+)\,. \end{aligned} \tag{35}$$

Therefore, the sufficient condition

$$|m_i^\star - m_j^\star| > \sqrt{2K}\, h([-F(x;\mathcal{S})]_+) \quad \Longrightarrow \quad \operatorname{sign}(m_i - m_j) = \operatorname{sign}(m_i^\star - m_j^\star) \tag{36}$$

connects a smaller mean-margin deficit to a tighter sufficient bound for pairwise ranking resolution.

The same bound also yields a direct coarse-to-fine interpretation of Top-K contraction. Let

$$\mathcal{S}_K(x) = \mathrm{TopK}_K\{m_c(x) : c \in \mathcal{U}\}, \qquad \Phi_K(x) = F(x; \mathcal{S}_K(x)).$$

For $K' < K$, the average of the largest K' margins cannot be smaller than the average of the largest K margins, and hence

$$\Phi_{K'}(x) \ge \Phi_K(x). \tag{37}$$

Therefore,

$$[-\Phi_{K'}(x)]_+ \le [-\Phi_K(x)]_+.$$

Because the same nondecreasing h applies across the supports under consideration and $K' < K$, the sufficient ranking-resolution bound

$$B_K(x) := \sqrt{2K}\, h([-\Phi_K(x)]_+) \tag{38}$$

satisfies

$$B_{K'}(x) \le B_K(x). \tag{39}$$

Thus, under the local reliability hypothesis, contracting a Top-K support at a fixed observation state yields a no-larger sufficient bound for resolving pairwise rankings among the retained competitors. This gives the coarse-to-fine rationale: a broad support is useful early, when premature exclusion is risky, whereas a smaller support permits finer ranking decisions once the discovery state is more informative.

The preceding argument considers support contraction at a fixed observation state. For a fixed support $\mathcal{S}$, the same hypothesis gives the complementary effect: the bound tightens as the supported mean margin improves. Because $[-F(x;\mathcal{S})]_+$ is nonincreasing in $F(x;\mathcal{S})$ and h is nondecreasing, maximizing the fixed-support mean margin over the threat set also minimizes this hypothesized upper bound over that set. Together, these observations motivate the coarse-to-fine design from two

directions: optimization can tighten the reliability bound for a fixed support, while support contraction can further tighten the ranking-resolution bound among the retained competitors. Neither statement implies monotonic tightening along the practical normalized, projected DCD trajectory, which changes supports and takes finite steps.

Although the latent shortfalls in equation 33 are not directly observable, the mechanism ablations in Appendix C.8 and Table 12 exhibit the qualitative pattern suggested by this coarse-to-fine view. Fixed-width Top-K discovery is substantially weaker than progressive contraction, and larger terminal supports progressively weaken the attack. In contrast, moderate changes to the initial support width or the precise contraction schedule have comparatively small effects. This agreement is only qualitative: these ablations do not directly test the latent reliability relation in equation 33.

A.5 Complementary Geometric View: Approximate Alignment and Attainable Top-K Recovery

The preceding reliability analysis allows the observed competitor ordering to differ from the latent ordering induced by the threat set. We next consider a more structured affine regime and give an idealized robustness result for the alignment argument. Exact agreement on a maximizing direction is unnecessary: it is enough for the competitor gradients to be sufficiently aligned with a common feasible direction relative to the attainable-margin separation.

This result is deliberately limited: it is neither a universal assumption on neural networks nor a convergence statement for finite-step DCD. It characterizes a simple geometric regime in which imperfect common-direction alignment still permits recovery of the attainable Top-K membership. Assume that every eligible competitor margin is affine over the threat set,

$$m_c(x) = m_c(x_0) + a_c^\top (x - x_0), \qquad c \in \mathcal{U}. \tag{40}$$

Let q be the dual norm index of p. Suppose there exists a common direction v, together with nonnegative alignment tolerances $\{\eta_c\}_{c\in\mathcal{U}}$, such that

$$\|v\|_p = 1, \qquad 0 \le \|a_c\|_q - a_c^\top v \le \eta_c, \qquad \forall c \in \mathcal{U}. \tag{41}$$

The left inequality follows from Hölder's inequality and is included to make the approximation error explicit. We also assume that the entire ray segment remains within the input domain,

$$x_0 + \rho v \in [0,1]^d, \qquad 0 \le \rho \le \epsilon. \tag{42}$$

Proposition 1 (Approximate alignment and attainable Top-K recovery)**.** *Under equation 40–equation 42, let*

$$x(\rho) = x_0 + \rho v, \qquad 0 \le \rho \le \epsilon, \tag{43}$$

and define

$$B_c(\rho) := (\epsilon - \rho)\|a_c\|_q + \rho\eta_c. \tag{44}$$

Then, for every $c \in \mathcal{U}$,

$$0 \le \Delta_c(x(\rho)) \le B_c(\rho). \tag{45}$$

For any $1 \le K < |\mathcal{U}|$ with $\gamma_K^\star > 0$, if

$$\gamma_K^\star > \max_{i\in\mathcal{T}_K^\star} B_i(\rho), \tag{46}$$

then

$$\mathcal{S}_K(x(\rho)) = \mathcal{T}_K^\star. \tag{47}$$

In particular, at the endpoint it is sufficient that

$$\gamma_K^\star > \epsilon \max_{i\in\mathcal{T}_K^\star} \eta_i \tag{48}$$

to recover the attainable Top-K set. For $K = 1$, this gives the corresponding sufficient condition for recovering the unique optimal target.

Moreover, for any nonempty fixed support $\mathcal{S} \subseteq \mathcal{U}$, its mean-margin gradient

$$g_{\mathcal{S}} := \nabla_x F(x;\mathcal{S}) = \frac{1}{|\mathcal{S}|}\sum_{c\in\mathcal{S}} a_c \tag{49}$$

satisfies

$$0 \le \|g_{\mathcal{S}}\|_q - g_{\mathcal{S}}^\top v \le \frac{1}{|\mathcal{S}|}\sum_{c\in\mathcal{S}} \eta_c. \tag{50}$$

Thus v achieves a directional derivative within the average alignment tolerance of the steepest-ascent value for every fixed-support mean-margin objective.

Proof. For any $x \in \mathcal{B}_\epsilon^p(x_0)$, Hölder's inequality gives

$$\begin{aligned} m_c(x) - m_c(x_0) &= a_c^\top (x - x_0) \\ &\le \|a_c\|_q \|x - x_0\|_p \\ &\le \epsilon \|a_c\|_q. \end{aligned} \tag{51}$$

Hence

$$m_c^\star \le m_c(x_0) + \epsilon\|a_c\|_q. \tag{52}$$

Along equation 43, equation 41 gives

$$\begin{aligned} m_c(x(\rho)) &= m_c(x_0) + \rho a_c^\top v \\ &\ge m_c(x_0) + \rho\big(\|a_c\|_q - \eta_c\big). \end{aligned} \tag{53}$$

Combining the last two displays yields

$$\begin{aligned} \Delta_c(x(\rho)) &= m_c^\star - m_c(x(\rho)) \\ &\le (\epsilon - \rho)\|a_c\|_q + \rho\eta_c = B_c(\rho). \end{aligned} \tag{54}$$

Nonnegativity follows from the definition of $m_c^\star$, proving equation 45.

Now fix $i \in \mathcal{T}_K^\star$ and $j \notin \mathcal{T}_K^\star$. By the attainable-margin decomposition,

$$\begin{aligned} m_i(x(\rho)) - m_j(x(\rho)) &= \big(m_i^\star - m_j^\star\big) - \big(\Delta_i(x(\rho)) - \Delta_j(x(\rho))\big) \\ &\ge \gamma_K^\star - \Delta_i(x(\rho)) \\ &\ge \gamma_K^\star - B_i(\rho). \end{aligned} \tag{55}$$

Therefore equation 46 makes every cross-boundary difference strictly positive, which proves equation 47. Setting $\rho = \epsilon$ gives $B_c(\epsilon) = \epsilon\eta_c$, yielding equation 48. The $K = 1$ statement follows as a special case.

Finally,

$$\begin{aligned} g_{\mathcal{S}}^\top v &= \frac{1}{|\mathcal{S}|}\sum_{c\in\mathcal{S}} a_c^\top v \\ &\ge \frac{1}{|\mathcal{S}|}\sum_{c\in\mathcal{S}} \big(\|a_c\|_q - \eta_c\big). \end{aligned} \tag{56}$$

Hölder's inequality and the triangle inequality give, respectively,

$$g_{\mathcal{S}}^\top v \le \|g_{\mathcal{S}}\|_q, \qquad \|g_{\mathcal{S}}\|_q \le \frac{1}{|\mathcal{S}|}\sum_{c\in\mathcal{S}} \|a_c\|_q. \tag{57}$$

Consequently,

$$\begin{aligned} 0 &\le \|g_{\mathcal{S}}\|_q - g_{\mathcal{S}}^\top v \\ &\le \frac{1}{|\mathcal{S}|}\sum_{c\in\mathcal{S}} \big(\|a_c\|_q - a_c^\top v\big) \\ &\le \frac{1}{|\mathcal{S}|}\sum_{c\in\mathcal{S}} \eta_c, \end{aligned} \tag{58}$$

which proves equation 50. □

The exact shared-maximizing-direction condition is the zero-error special case of Proposition 1. If $\eta_c = 0$ for every $c \in \mathcal{U}$, then $a_c^\top v = \|a_c\|_q$ for every competitor. In that case $x_0 + \epsilon v$ attains the Hölder upper bound simultaneously for all competitors, so

$$m_c^\star = m_c(x_0) + \epsilon\|a_c\|_q, \qquad \Delta_c(x(\rho)) = (\epsilon - \rho)\|a_c\|_q, \tag{59}$$

and $m_c(x(\epsilon)) = m_c^\star$ for all c. Likewise, equation 50 becomes equality,

$$g_{\mathcal{S}}^\top v = \|g_{\mathcal{S}}\|_q = \frac{1}{|\mathcal{S}|}\sum_{c\in\mathcal{S}}\|a_c\|_q, \tag{60}$$

recovering the original exact shared-direction argument.

The approximate-alignment condition is intentionally idealized. For $p = 2$, the normalized defect $(\|a_c\|_2 - a_c^\top v)/\|a_c\|_2$ equals $1 - \cos\theta_c$ for nonzero a_c, so small tolerance corresponds to small angular disagreement with the common direction. The proposition shows that exact directional agreement is unnecessary: attainable Top-K membership remains recoverable whenever its separation exceeds the residual shortfall induced by imperfect alignment. It does not imply that the practical normalized, projected DCD trajectory follows v, reaches the endpoint $x_0 + \epsilon v$, or is globally optimal on nonlinear neural networks. Instead, it provides an affine robustness extension of the alignment argument that complements the state-dependent reliability analysis above.

A.6 Supporting Analyses: Score-Sensitive Allocation and Objective Relations

With the core theoretical narrative in place, we now collect supporting analyses that compare uniform allocation with score-sensitive alternatives, examine temperature and logit-scale effects, and relate the mean-margin formulation to standard attack objectives. These results provide additional context for the preceding design choices but are not needed for the coarse-to-fine reliability argument.

Once selection forms the active support, allocation maps the retained competitors to normalized discovery-gradient weights. The alternatives below follow the score-information hierarchy introduced in equation 4: mean-margin allocation uses support membership only, Rank-Margin adds ordinal rank, and T-CE/CE adds cardinal score-gap magnitude.

Score-dependent commitments from transient evidence. The selection–allocation view also clarifies where standard attacks place commitment to the current scores. APGD-T mainly commits in the selection stage: its target ordering is formed from the clean input, so subsequent targeted optimization inherits the ranking distortion observed at x_0 without revising that ordering during the targeted run.

APGD-CE instead keeps all competitors active but uses observed score gaps to control allocation. For its normalized softmax allocation w^{CE}, any pair $i, j \in \mathcal{U}$ satisfies

$$\log\frac{w_i^{\mathrm{CE}}(x)}{w_j^{\mathrm{CE}}(x)} = m_i(x) - m_j(x) = m_i^\star - m_j^\star - \big(\Delta_i(x) - \Delta_j(x)\big). \tag{61}$$

Thus the same state-dependent distortion that perturbs ranking also perturbs the relative optimization weight assigned to different competitors.

A CW-style max-margin objective makes a harder within-state allocation decision: it selects $\arg\max_{c\in\mathcal{U}} m_c(x)$, which need not equal $\arg\max_{c\in\mathcal{U}} m_c^\star$, and locally concentrates the update on the observed maximizer. Together, these comparisons motivate DCD's use of uncertain scores for revisable support membership while avoiding a second, stronger score-dependent commitment within the selected support.

General allocation operator. For a fixed support $\mathcal{S}$ of size K, let $g_c(x) = \nabla_x m_c(x)$. We write the direction induced by any normalized nonnegative allocation w as

$$G(x; \mathcal{S}, w) := \sum_{c\in\mathcal{S}} w_c(x;\mathcal{S}) g_c(x), \qquad w_c \geq 0, \qquad \sum_{c\in\mathcal{S}} w_c = 1. \tag{62}$$

At discovery step t, this becomes

$$g_t = G(x_{t-1}; \mathcal{S}_t, w_t), \qquad \sum_{c \in \mathcal{S}_t} w_{c,t} = 1. \tag{63}$$

The weights may depend on the current scores and support. Canonical DCD uses $w_{c,t} = 1/K_t$. When x and $\mathcal{S}$ are fixed below, we use the shorthand $G_w := G(x; \mathcal{S}, w)$. For a differentiable scalar objective J, its coefficients $\partial J / \partial m_c$ induce the corresponding allocation after normalization when needed.

Mean margin. For mean margin,

$$w_c = \frac{1}{K}, \qquad G_w = \frac{1}{K} \sum_{c \in \mathcal{S}} g_c. \tag{64}$$

Conditional on $\mathcal{S}$, the weights are invariant to both score ordering and gap magnitude.

Rank-Margin. With $m_{(1)} \geq \cdots \geq m_{(K)}$, Rank-Margin uses

$$w^{\mathrm{Rank}}_{(r)} = \frac{2(K - r + 1)}{K(K+1)}, \qquad r = 1, \ldots, K. \tag{65}$$

Treating the ordering as locally fixed, it corresponds to

$$F_{\mathrm{Rank}} = \sum_{r=1}^{K} \frac{2(K - r + 1)}{K(K+1)} m_{(r)}. \tag{66}$$

Equivalently,

$$F_{\mathrm{Rank}} = \sum_{j=1}^{K} \frac{2j}{K(K+1)} \left(\frac{1}{j} \sum_{r=1}^{j} m_{(r)} \right). \tag{67}$$

The coefficients sum to one, so Rank-Margin is a convex mixture of Top-j mean-margin objectives and uses ordinal, but not cardinal, score information.

T-CE and support-restricted CE. Suppressing the common state argument, for $\tau > 0$ define

$$F^{\mathrm{T}}_{\tau,\mathcal{S}} = \tau \log \left(\frac{1}{K} \sum_{c \in \mathcal{S}} \exp\left(\frac{m_c}{\tau} \right) \right), \tag{68}$$

with derivatives

$$\frac{\partial F^{\mathrm{T}}_{\tau,\mathcal{S}}}{\partial m_c} = w_c^{(\tau)} = \frac{\exp(m_c/\tau)}{\sum_{j \in \mathcal{S}} \exp(m_j/\tau)}. \tag{69}$$

Support-restricted CE is

$$L_{\mathrm{CE},\mathcal{S}} = \log \left(e^{z_y} + \sum_{c \in \mathcal{S}} e^{z_c} \right) - z_y = \log \left(1 + \sum_{c \in \mathcal{S}} e^{m_c} \right). \tag{70}$$

Its gradient is

$$\nabla_x L_{\mathrm{CE},\mathcal{S}} = \frac{\sum_{j \in \mathcal{S}} e^{m_j}}{1 + \sum_{j \in \mathcal{S}} e^{m_j}} \sum_{c \in \mathcal{S}} w_c^{\mathrm{CE}} \nabla_x m_c, \tag{71}$$

where the leading scalar is positive and the normalized CE profile is

$$w_c^{\mathrm{CE}} = \frac{\exp(m_c)}{\sum_{j \in \mathcal{S}} \exp(m_j)} = \frac{\exp(z_c)}{\sum_{j \in \mathcal{S}} \exp(z_j)}. \tag{72}$$

At $\tau = 1$, this is exactly the normalized T-CE profile. The scalar objectives are related by

$$L_{\mathrm{CE},\mathcal{S}} = \log\left(1 + K \exp\left(F^{\mathrm{T}}_{1,\mathcal{S}}\right)\right), \tag{73}$$

so their gradients are collinear before implementation-specific aggregation or normalization. The implementation dispatches T-CE with $\tau = 1$ through the historical CE branch to preserve the exact CE-DCD update path.

The temperature limits are

$$\lim_{\tau\to\infty} w_c^{(\tau)} = \frac{1}{K}, \quad \lim_{\tau\to\infty} F^{\mathrm{T}}_{\tau,\mathcal{S}} = \frac{1}{K}\sum_{c\in\mathcal{S}} m_c, \qquad \lim_{\tau\to 0^+} F^{\mathrm{T}}_{\tau,\mathcal{S}} = \max_{c\in\mathcal{S}} m_c. \tag{74}$$

Thus $\tau \uparrow$ removes dependence on within-support gap magnitude, while $\tau \downarrow$ sharpens score-driven specialization.

Signal–distortion decomposition. Since $m_c = m_c^\star - \Delta_c$, the exponential allocation has the pairwise decomposition

$$\log \frac{w_i^{(\tau)}}{w_j^{(\tau)}} = \frac{m_i - m_j}{\tau} = \frac{m_i^\star - m_j^\star}{\tau} - \frac{\Delta_i - \Delta_j}{\tau}. \tag{75}$$

Hence temperature scales both latent preference and shortfall distortion by the same factor $1/\tau$. The effective sharpness depends on $(m_i - m_j)/\tau$, so the same numerical temperature can induce different weighting profiles when model score scales differ.

Global logit-scale invariance. Consider a reparameterization of the logits

$$\widetilde{z}_c(x) = a\, z_c(x) + b(x), \qquad a > 0, \tag{76}$$

where a is a constant and $b(x)$ is shared by all classes. Then $\widetilde{m}_c(x) = a\, m_c(x)$, so the Top-$K$ support and within-support ranking are unchanged. For both mean-margin and Rank-Margin allocation, the allocation weights are therefore unchanged, while each margin gradient is scaled by a. Consequently, for $p \in \{2, \infty\}$,

$$\mathcal{D}_p\Big(\nabla_x \widetilde{F}(x;\mathcal{S})\Big) = \mathcal{D}_p(\nabla_x F(x;\mathcal{S}))\,. \tag{77}$$

Thus, under the same initialization, support schedule, projection, and tie-breaking rule, canonical DCD produces the same discovery iterates and support sequence under equation 76; endpoint re-evaluation also selects the same competitor.

In contrast, at a fixed numerical temperature,

$$\widetilde{w}_c^{(\tau)} = \frac{\exp(a m_c/\tau)}{\sum_{j\in\mathcal{S}} \exp(a m_j/\tau)} = w_c^{(\tau/a)}, \tag{78}$$

so finite-temperature score allocation is generally not invariant to global logit rescaling. Its weighting profile is preserved if the temperature is rescaled simultaneously as $\widetilde{\tau} = a\tau$.

Together, these observations suggest a useful qualitative interpretation, but not an operational diagnostic. Finite-temperature weighting may help when transient gaps are sufficiently aligned with latent susceptibility; intermediate temperatures may be useful when signal and distortion coexist; and high temperature may be preferable when gap information is weak. These regimes are interpretive rather than observable classifications available to the attack. In particular, a model-dependent temperature optimum may reflect score reliability, score scale, local geometry, or some combination of them.

Local pairwise feedback. The following proposition isolates the effect of unequal pair weights.

Proposition 2 (Pairwise coefficient decomposition)**.** *Let $G_w = G(x;\mathcal{S},w)$, and let $\widetilde{w}$ agree with w except that $\widetilde{w}_i = \widetilde{w}_j = (w_i + w_j)/2$. Then*

$$\langle g_i - g_j, G_w\rangle - \langle g_i - g_j, G_{\widetilde{w}}\rangle = \frac{w_i - w_j}{2}\|g_i - g_j\|_2^2. \tag{79}$$

Proof. By construction, only the pair (i, j) is changed, so

$$G_w - G_{\widetilde{w}} = \frac{w_i - w_j}{2}(g_i - g_j). \tag{80}$$

Taking the inner product with $g_i - g_j$ gives equation 79. □

For a differentiable pairwise gap and a raw local step $x^+ = x + \eta G_w$,

$$\begin{aligned} m_i(x^+) - m_j(x^+) = m_i(x) - m_j(x) + \eta\langle g_i - g_j, G_{\widetilde{w}}\rangle \\ + \eta\frac{w_i - w_j}{2}\|g_i - g_j\|_2^2 + o(\eta). \end{aligned} \tag{81}$$

For mean margin, $w_i - w_j = 0$. For Rank-Margin, let r_c denote the rank position of competitor c within $\mathcal{S}$; if $r_i < r_j$,

$$w_i^{\mathrm{Rank}} - w_j^{\mathrm{Rank}} = \frac{2(r_j - r_i)}{K(K+1)} > 0, \tag{82}$$

so the extra pairwise term depends on rank distance but not gap magnitude. For T-CE, equation 75 shows that both the observed gap and temperature control the preference.

For the monotone score-based allocation rules above, when

$$(m_i - m_j)(m_i^\star - m_j^\star) > 0, \tag{83}$$

the score-sensitive term reinforces an ordering consistent with the latent reference; when the signs disagree, it reinforces the mismatch. The equalized term in equation 81 can have either sign, so this is only a local feedback characterization, not a statement that unequal or equal weighting yields a better attack update.

Temperature-dependent mean and spread. Let

$$\bar{m} = \frac{1}{K}\sum_{c\in\mathcal{S}} m_c, \qquad \widetilde{m}_c = m_c - \bar{m}. \tag{84}$$

Then

$$F_{\tau,\mathcal{S}}^{\mathrm{T}} - \bar{m} = \tau \log\left(\frac{1}{K}\sum_{c\in\mathcal{S}} e^{\widetilde{m}_c/\tau}\right) \geq 0. \tag{85}$$

The inequality follows from Jensen's inequality, with equality iff all supported margins are equal. When $\max_{c\in\mathcal{S}} |\widetilde{m}_c|/\tau \ll 1$,

$$\begin{aligned} F_{\tau,\mathcal{S}}^{\mathrm{T}} - \bar{m} &= \frac{1}{2\tau}\,\mathrm{Var}_{c\in\mathcal{S}}(m_c) + O\left(\frac{\left(\max_{c\in\mathcal{S}} |\widetilde{m}_c|\right)^3}{\tau^2}\right), \\ \mathrm{Var}_{c\in\mathcal{S}}(m_c) &:= \frac{1}{K}\sum_{c\in\mathcal{S}} \widetilde{m}_c^2. \end{aligned} \tag{86}$$

Using empirical variance and covariance over $c \in \mathcal{S}$,

$$\mathrm{Var}_{c\in\mathcal{S}}(m_c) = \mathrm{Var}_{c\in\mathcal{S}}(m_c^\star) + \mathrm{Var}_{c\in\mathcal{S}}(\Delta_c) - 2\,\mathrm{Cov}_{c\in\mathcal{S}}(m_c^\star, \Delta_c). \tag{87}$$

Thus observed score dispersion mixes variation in latent attainable margins with state-dependent shortfall variation and their covariance.

Finally, we relate the fixed-support mean-margin formulation to standard attack objectives.

Relation to reverse cross-entropy. Let $\mathcal{U} \subseteq \mathcal{C}^-$ be any nonempty eligible competitor set. Recall the fixed-support mean-margin objective $F(x; \mathcal{S})$. With $\mathrm{CE}(z, c) = -z_c + \log\sum_{j\in\mathcal{C}} \exp z_j$, reverse cross-entropy over the eligible labels satisfies

$$\begin{aligned} \mathcal{L}_{\mathrm{RCE}}(z, y; \mathcal{U}) &= \mathrm{CE}(z, y) - \frac{1}{|\mathcal{U}| + 1}\left(\mathrm{CE}(z, y) + \sum_{c\in\mathcal{U}} \mathrm{CE}(z, c)\right) \\ &= \frac{|\mathcal{U}|}{|\mathcal{U}| + 1}\left(\frac{1}{|\mathcal{U}|}\sum_{c\in\mathcal{U}} z_c - z_y\right) = \frac{|\mathcal{U}|}{|\mathcal{U}| + 1} F(x; \mathcal{U}). \end{aligned} \tag{88}$$

Table 4: Support size K_t during DCD competitor discovery. Here, $t = 1, \ldots, T_d$ denotes the discovery step and $T_d = 15$ is the total number of discovery steps.

t	CIFAR-10 ($C-1=9$)	CIFAR-100 ($C-1=99$)	ImageNet ($C-1=999$)	ImageNet-12K ($C-1=11820$)
1	9	99	999	11820
2	8	72	610	6050
3	7	52	373	3097
4	6	37	228	1585
5	5	27	139	811
6	5	20	85	416
7	4	14	52	213
8	3	10	32	109
9	3	8	20	56
10	3	6	12	29
11	2	4	8	15
12	2	3	5	8
13	2	2	3	4
14	2	2	2	2
15	2	2	2	2

Thus, the full eligible-support mean-margin objective is equivalent to RCE up to a positive scale. At the other endpoint, any singleton support $\mathcal{S} = \{c\}$ gives

$$F(x; \{c\}) = z_c(x) - z_y(x) = m_c(x), \tag{89}$$

the standard pairwise logit-margin objective. These are endpoint identities of the fixed-support mean-margin instantiation. More generally, DCD retains state-dependent support selection while allowing other within-support allocation rules.

Relation to DLR. For the standard untargeted DLR loss Croce & Hein (2020b), suppose the current input is correctly classified and the relevant logits are not tied. Let π order the logits in decreasing order, so $\pi_1 = y$, let $a = \pi_2$, and set $s = z_y - z_{\pi_3} > 0$. Then

$$\mathcal{L}_{\mathrm{DLR}} = \frac{m_a}{s}, \qquad \nabla_x \mathcal{L}_{\mathrm{DLR}} = \frac{1}{s} g_a - \frac{m_a}{s^2} \nabla_x s. \tag{90}$$

This decomposition separates the one-hot top-competitor contribution in the numerator from the normalization contribution in the denominator. We therefore do not identify DLR with max-margin in the main analysis.

Scope of the local analysis. Equation equation 81 describes the raw aggregated gradient before normalization, projection, momentum, adaptive updates, or EOT. These operations can change the realized dynamics, especially at finite step sizes. Likewise, objective-level collinearity between CE and unit-temperature T-CE does not by itself imply identical updates after aggregation and normalization. The latent-margin decomposition is interpretive. Accordingly, the analysis supports the design distinction between support selection and score-sensitive allocation, but does not establish convergence, ranking preservation, optimal target identification, optimal allocation, or optimal temperature selection.

The weighting ablations in Appendix C.8 and Table 12 provide a qualitative consistency check for this interpretation. High- and moderate-temperature T-CE configurations, whose allocations are closer to uniform weighting, remain close to the default DCD performance, whereas performance degrades at lower temperatures as score-dependent allocation becomes increasingly concentrated. This pattern is consistent with DCD's conservative choice to limit within-support reliance on transient score-gap magnitudes, but it does not establish uniform allocation as optimal.

B Implementation Details and Support Schedules

This section records the implementation details needed to reproduce the canonical DCD discovery phase and to distinguish it from the schedule and handoff variants evaluated later. DCD progressively contracts the active support while keeping the eligible competitor universe fixed within each run. For a fixed run r, let $\mathcal{U} = \mathcal{U}_r$ denote the eligible set passed to Algorithm 1; DCD-m initializes $\mathcal{U}_1 = \mathcal{C}^-$ and removes each selected target before the next run. We suppress the run index below. Let K_0 denote the initial-support cardinality parameter and set $K_f = \min\{2, K_0\}$; the canonical setting uses $K_0 = |\mathcal{U}|$.

For $T_d \geq 2$, the canonical cardinality schedule is

$$K_t = \max\left\{K_f, \left\lceil K_0^{\,1-\frac{t-1}{T_d-1}} \right\rceil\right\}, \qquad t = 1, \ldots, T_d. \tag{91}$$

Thus the underlying geometric-style curve decays toward one while the default implementation clips the support at Top-2 whenever at least two initially active competitors remain. The resulting canonical support sizes are shown in Table 4. A linear cardinality schedule is

$$K_t = \left\lceil K_0 + (K_f - K_0)\frac{t-1}{T_d - 1} \right\rceil, \qquad t = 1, \ldots, T_d. \tag{92}$$

These schedules control only the number of active competitors. Fixed-cardinality, narrow-initial-support, alternative-terminal-support, and linear-contraction variants are evaluated in Appendix C.8.

The default support remains dynamic and revisable:

$$\mathcal{S}_t = \mathrm{TopK}_{K_t}\{m_c(x_{t-1}) : c \in \mathcal{U}\}, \qquad t = 1, \ldots, T_d. \tag{93}$$

Thus monotone contraction of the cardinality does not imply nested membership,

$$K_{t+1} \leq K_t \quad \not\Rightarrow \quad \mathcal{S}_{t+1} \subseteq \mathcal{S}_t, \qquad t = 1, \ldots, T_d - 1. \tag{94}$$

A competitor outside the current support can re-enter when its relative margin improves. The *Nested Pruning* ablation instead restricts later Top-K operations to the preceding support:

$$\begin{aligned} \mathcal{S}_1^{\mathrm{nest}} &= \mathrm{TopK}_{K_1}\{m_c(x_0) : c \in \mathcal{U}\}, \\ \mathcal{S}_t^{\mathrm{nest}} &= \mathrm{TopK}_{K_t}\{m_c(x_{t-1}) : c \in \mathcal{S}_{t-1}^{\mathrm{nest}}\}, \qquad t = 2, \ldots, T_d. \end{aligned} \tag{95}$$

This nested-support restriction is an empirical ablation rather than the default DCD mechanism.

The discovery-step schedule is shared across runs. Its canonical coefficient is

$$\beta_t = 4 - 3\frac{t-1}{T_d - 1}, \qquad t = 1, \ldots, T_d, \tag{96}$$

which decreases linearly from 4 to 1. Using the same normalized direction $\mathcal{D}_p$ defined in the main text and initializing at the clean input x_0, the canonical discovery update is

$$x_t = \Pi_{\mathcal{B}_\epsilon^p(x_0)}\left(x_{t-1} + \beta_t \epsilon\, \mathcal{D}_p(g_t)\right), \qquad t = 1, \ldots, T_d. \tag{97}$$

The observation loop is therefore

$$x_{t-1} \longrightarrow \mathcal{S}_t \longrightarrow g_t \longrightarrow x_t, \qquad t = 1, \ldots, T_d. \tag{98}$$

The support-cardinality and step schedules are implementation choices. Discovery-step sensitivity is reported in Appendix C.6, while support-cardinality ablations are reported in Appendix C.8.

C Experimental Details and Additional Analyses

This section provides exact model information, implementation details, and supporting experiments and ablations that complement the experimental protocol summarized in the main text.

Table 5: **Models and abbreviations.** Model abbreviations used in the main text, corresponding architectures, source papers, and evaluation threat models with default attack radii.

Dataset	Threat	Abbr.	Model	Source
CIFAR-10	$\ell_\infty = 8/255$	Carm.	WRN-28-10	Carmon et al. (2019)
		Rebu.	WRN-70-16	Rebuffi et al. (2021)
		kwta	DenseNet-121 (k-WTA)	Xiao et al. (2020)
CIFAR-100	$\ell_\infty = 8/255$	Gowal	WRN-70-16	Gowal et al. (2020)
		Rice	PreActResNet-18	Rice et al. (2020)
ImageNet	$\ell_\infty = 4/255$	Engs.	ResNet-50	Engstrom et al. (2019)
		Salm.	ResNet-50	Salman et al. (2020)
		MIM.	Swin-L	Xu et al. (2026)
ImageNet-12K	$\ell_\infty = 0.15/255$	CXN	ConvNeXt-Nano	Liu et al. (2022)
		CXC	ConvNeXt-Base (CLIP)	Cherti et al. (2023)
		ViTC	ViT-B/16 (CLIP)	Cherti et al. (2023)
CIFAR-10	$\ell_2 = 0.5$	Augustin	ResNet-50	Augustin et al. (2020)
		Gowal	WRN-70-16	Gowal et al. (2020)
ImageNet	$\ell_2 = 0.2$	ViT	ViT-B/32	Dosovitskiy et al. (2021)
		VGG	VGG-19	Simonyan & Zisserman (2015)
		Efficient	EfficientNet-B0	Tan & Le (2019)

C.1 Full Experimental Setup

Models, datasets, and threat models. Table 5 lists the model abbreviations used throughout the paper together with their architectures, source papers, evaluation threat models, and perturbation radii. RobustBench checkpoints are used where indicated Croce et al. (2021). The ImageNet-12K models are standard large-output-space classifiers rather than adversarially trained models; they are included to test target discovery when the number of competing classes is large. For CIFAR-10, CIFAR-100, and ImageNet-12K, we evaluate on 10,000 randomly sampled images; for ImageNet, we use 5,000 images. The same class-balanced ordering and random seed are used for all attacks so that every method is evaluated on exactly the same examples. The ℓ_2 experiments use the same dataset-level sample counts and sample-selection protocol. The k-WTA model is included as a gradient-masking stress test because its intentionally non-informative gradients fall outside the standard differentiable regime in which DCD is intended to operate.

Implementation environment. Experiments were run on Ubuntu 20.04.6 LTS with two Intel Xeon Gold 6330 CPUs and four NVIDIA GeForce RTX 4090 GPUs, using Python 3.11, PyTorch 2.3.0, and Torchvision 0.18.0.

C.2 Additional Multi-target Baseline Results

Table 6 complements Table 1 with the endpoint robust accuracies of the additional multi-target and target-selection baselines used in Figure 3. The table below reports the exact $m = 9$ results for APGD-T, APGD-DCD9, MultiTargeted, MM+, MALT, GAMA-MT, FAB-T, and FAB-DCD9.

C.3 Exclusive Adversarial Examples Relative to APGD Objectives

To determine whether DCD merely improves success on the same inputs or also uncovers failures missed by standard APGD objectives, Table 7 compares APGD-DCD1 with APGD-CE, APGD-Margin, and APGD-RCE on six models. For each pair, we count adversarial examples found exclusively by APGD-DCD1 and examples found exclusively by the other APGD variant. We also report how often the target discovered by DCD matches the corresponding competitor or final adversarial class used for the comparison.

Table 6: **Multi-target attack comparison.** Robust accuracy (%) under the reported $m = 9$ configurations. Lower is better. Avg. is the unweighted mean across the 10 benchmark models.

	CIFAR-10		CIFAR-100		ImageNet			ImageNet-12K			
Method	Carm.	Rebu.	Gowal	Rice	Engs.	Salm.	MIM.	CXN	CXC	ViTC	Avg.
APGD-T	59.59	66.64	36.92	18.97	30.30	36.80	59.38	14.48	14.31	15.28	35.27
APGD-DCD9	59.60	**66.60**	**36.82**	**18.92**	**30.14**	36.68	**59.26**	**14.24**	**14.11**	**15.00**	**35.14**
MultiTargeted	59.83	66.73	36.96	19.06	30.40	36.84	59.36	14.50	14.24	15.10	35.30
MM+	**59.57**	**66.60**	36.90	18.96	30.26	36.82	59.32	14.51	14.23	15.11	35.23
MALT	59.59	66.64	36.84	18.96	30.18	**36.64**	59.30	14.29	14.23	15.33	35.20
GAMA-MT	59.59	66.67	36.96	19.00	30.40	36.74	59.32	14.61	14.94	16.28	35.45
FAB-T	**60.20**	**67.09**	37.32	19.25	31.08	37.30	59.84	15.51	16.34	18.25	36.22
FAB-DCD9	60.21	**67.09**	**37.25**	**19.23**	**30.90**	**37.22**	**59.72**	**15.28**	**16.14**	**18.07**	**36.11**

Table 7: **Exclusive adversarial examples relative to other APGD objectives.** "DCD Only" and "Other Only" count examples attacked successfully by only one of the two methods. Within each exclusive subset, "Target Match" counts cases in which the target discovered by APGD-DCD1 agrees with the corresponding competitor/final class associated with the other APGD variant.

		DCD Only		Other Only	
Model	Compared	Exclusive	Match	Exclusive	Match
Gowal	APGD-CE	303	215	2	0
	APGD-Margin	194	0	12	0
	APGD-RCE	636	472	0	0
Rice	APGD-CE	157	113	6	1
	APGD-Margin	130	0	12	0
	APGD-RCE	795	589	0	0
Engs.	APGD-CE	145	103	2	0
	APGD-Margin	170	0	4	1
	APGD-RCE	732	537	0	0
Salm.	APGD-CE	158	127	3	0
	APGD-Margin	123	0	8	1
	APGD-RCE	619	484	0	0
CXC	APGD-CE	176	126	30	5
	APGD-Margin	207	2	27	4
	APGD-RCE	732	555	10	1
ViTC	APGD-CE	239	182	60	20
	APGD-Margin	146	3	37	8
	APGD-RCE	750	585	31	12
Total	APGD-CE	1178	866	103	26
Total	APGD-Margin	970	5	100	14
Total	APGD-RCE	4264	3222	41	13

Across every model and APGD objective in Table 7, the number of DCD-only adversarial examples is substantially larger than the number found only by the competing APGD variant. This pattern is especially pronounced relative to APGD-RCE, but is also consistent for APGD-CE and APGD-Margin. For APGD-CE and APGD-RCE, many of these exclusive DCD successes also show target agreement, indicating that DCD often discovers a competitor that remains directly relevant to the eventual attack outcome. The main result is therefore not merely a reshuffling of successes on the same inputs: DCD exposes additional vulnerabilities that standard APGD objectives miss.

Table 8: **Threat-model generalization under ℓ_2 perturbations.** Robust accuracy (%) for DCD-enhanced and corresponding baseline attacks. Lower is better. The best result within each attack family for each model is highlighted in bold.

	CIFAR-10		ImageNet		
Method	Augustin	Gowal	ViT	VGG	Efficient
PGD-CE	76.64	82.86	56.88	28.54	47.32
PGD-Margin	75.95	82.13	56.38	29.94	45.64
PGD-DCD1	**74.41**	**81.87**	**54.72**	**25.80**	**43.30**
APGD-CE	74.67	81.54	52.78	23.44	42.78
APGD-Margin	74.60	80.97	53.68	26.04	43.00
APGD-DLR	75.12	80.98	53.88	27.22	43.68
APGD-DCD1	**73.09**	**80.59**	**52.02**	**21.92**	**40.40**
FAB	74.31	80.93	52.14	23.16	41.18
FAB-DCD1	**73.35**	**80.82**	**52.04**	**22.74**	**40.74**
FMN	74.29	80.91	56.58	26.00	42.78
FMN-DCD1	**73.07**	**80.60**	**56.04**	**22.30**	**40.50**
APGD-T	72.90	**80.53**	50.30	21.34	39.80
APGD-DCD9	**72.89**	**80.53**	**49.74**	**21.12**	**39.64**

C.4 ℓ_2 Threat-Model Generalization

To test whether DCD continues to improve attacks under a different perturbation geometry, we use the ℓ_2 models and attack settings specified in Appendix C.1; here we report the resulting robust accuracies.

Table 8 shows the same overall pattern as the main ℓ_∞ experiments. PGD-DCD1 achieves lower robust accuracy than the standard PGD objectives on all five models, and APGD-DCD1 similarly improves over APGD-CE, APGD-Margin, and APGD-DLR throughout the table. DCD also strengthens FAB and FMN on every evaluated model, showing that the transfer to ℓ_2 is not specific to a single optimizer or attack objective. The multi-target result is consistent with the single-target comparisons: APGD-DCDm matches or improves APGD-T on all five models. Overall, the benefits of dynamic competitor discovery extend to the evaluated ℓ_2 settings across multiple attack families and both CIFAR-10 and ImageNet.

C.5 Gradient-Masking Stress Test

We separately evaluate the CIFAR-10 DenseNet-121 with k-WTA activations Xiao et al. (2020) using the same ℓ_∞ radius $\epsilon = 8/255$, 10,000-image evaluation set, and (T=100) per-trajectory attack budget as in the main CIFAR-10 experiments. Because the hard winner selection in k-WTA induces a highly non-smooth optimization landscape, we treat this model as a complementary stress test rather than as part of the standard benchmark. DCD uses local first-order information to identify promising competing classes, and therefore primarily targets settings in which such gradients remain informative for attack optimization. The k-WTA results in Table 9 provide a test of this assumption: unlike the consistent gains observed on the standard benchmark models, DCD does not uniformly improve its base attacks in this gradient-masked regime. This contrast helps delineate the operating regime in which dynamic competitor discovery is most effective.

C.6 Discovery-Step Sensitivity

Table 10 studies the endpoint coefficients of the discovery-step schedule. More generally, parameterizing the same linear schedule directly by its endpoints gives

$$\beta_t = \beta_1 + (\beta_{T_d} - \beta_1)\,\frac{t-1}{T_d - 1}, \qquad t = 1, \ldots, T_d, \tag{99}$$

Table 9: **k-WTA gradient-masking stress test.** Robust accuracy (%); lower is better. Gray rows denote DCD-based variants, and bold values indicate the best result within each attack block.

Method	Robust Acc.
Clean Acc.	79.09
PGD-CE	**45.78**
PGD-Margin	46.10
PGD-DCD1	46.43
APGD-CE	38.45
APGD-Margin	**38.36**
APGD-DCD1	40.14
GAMA	**39.08**
GAMA-DCD1	40.26
FAB	78.91
FAB-DCD1	**78.57**
FMN	**71.13**
FMN-DCD1	77.80
APGD-T	**31.51**
APGD-DCD9	32.15
FAB-T	**78.92**
FAB-DCD9	**78.92**

Table 10: APGD-DCD discovery-step schedule ablation (robust accuracy, %). For $T_d = 15$, β_1 and β_{15} denote the initial and terminal discovery-step coefficients in equation 99. AVG is the mean robust accuracy across the reported models; lower is better.

	CIFAR-10			CIFAR-100		ImageNet			
(β_1, β_{15})	Carm.	Rebu.	Augustin	Gowal	Rice	Engs.	Salm.	MIM.	AVG↓
$(5.0, 1.0)$	59.80	66.79	73.10	37.01	19.09	30.34	36.82	**59.40**	47.80
$(4.5, 1.0)$	59.80	66.81	73.10	37.00	19.11	30.36	36.80	**59.40**	47.80
$(4.0, 2.0)$	59.76	66.82	73.22	37.00	19.10	30.34	**36.76**	59.42	47.80
$(4.0, 1.5)$	59.78	66.82	73.13	37.00	19.10	30.36	36.78	**59.40**	47.80
$(4.0, 1.0)$ (default)	59.80	66.78	73.09	37.01	**19.08**	30.36	36.82	**59.40**	**47.79**
$(4.0, 0.5)$	59.78	66.77	**73.06**	37.01	19.13	30.46	36.84	59.42	47.81
$(4.0, 0.3)$	59.79	66.77	73.07	37.01	19.13	30.52	36.86	59.42	47.82
$(3.0, 2.0)$	59.76	66.82	73.20	**36.99**	19.10	30.34	**36.76**	59.42	47.80
$(3.0, 1.5)$	59.78	66.79	73.16	37.00	19.12	**30.32**	36.80	**59.40**	47.80
$(3.0, 1.0)$	59.81	**66.76**	73.13	37.00	19.13	30.38	36.80	**59.40**	47.80
$(3.0, 0.5)$	**59.74**	66.77	73.08	37.02	19.16	30.48	36.86	59.44	47.82
$(3.0, 0.3)$	59.75	66.78	73.09	37.03	19.17	30.56	36.92	59.46	47.85
$(2.0, 1.0)$	59.75	**66.76**	73.11	37.03	19.10	30.42	36.82	**59.40**	47.80
$(2.0, 0.5)$	59.77	66.78	73.10	37.04	19.16	30.46	36.90	59.42	47.83
$(2.0, 0.3)$	59.77	66.78	73.07	37.06	19.18	30.56	36.92	59.48	47.85
$(1.0, 0.5)$	59.80	66.78	73.09	37.06	19.14	30.54	36.92	59.42	47.84
$(1.0, 0.3)$	59.78	66.78	**73.06**	37.09	19.14	30.66	36.92	59.44	47.86

so that the nominal discovery update before projection is $\beta_t \epsilon \mathcal{D}_p(g_t)$. Thus, β_1 and β_{T_d} are respectively the initial (largest) and terminal (smallest) discovery-step coefficients, rather than the realized post-projection step lengths. For the default $T_d = 15$, the canonical setting is $(\beta_1, \beta_{15}) = (4, 1)$.

Across the tested endpoint choices, robust accuracy changes only slightly. The average varies from 47.79% to 47.86%, and the canonical $(4, 1)$ schedule obtains 47.79%. DCD therefore remains stable over a broad range of discovery-step coefficients and does not rely on precise tuning of these endpoints.

Table 11: APGD-DCD discovery-budget ablation under a fixed total budget $T = 100$ (robust accuracy, %). T_d denotes the number of iterations allocated to competitor discovery, while the base attack receives $T_a = T - T_d$ iterations. AVG is the mean robust accuracy across all reported models; lower is better.

	CIFAR-10			CIFAR-100		ImageNet			
T_d	Carm.	Rebu.	Augustin	Gowal	Rice	Engs.	Salm.	MIM.	AVG↓
$T_d = 2$	60.24	67.09	73.51	37.61	19.59	31.30	37.30	59.82	48.31
$T_d = 5$	59.81	66.79	73.11	37.07	19.17	30.42	36.86	59.48	47.84
$T_d = 10$	**59.77**	66.79	73.12	37.07	19.17	30.40	36.84	**59.40**	47.82
$T_d = 15$ (default)	59.80	**66.78**	**73.09**	**37.01**	**19.08**	**30.36**	**36.82**	**59.40**	**47.79**
$T_d = 20$	59.79	66.79	73.14	**37.01**	19.18	30.52	**36.82**	59.44	47.84
$T_d = 25$	59.81	66.80	73.12	37.02	19.17	**30.36**	**36.82**	59.42	47.82
$T_d = 30$	59.78	66.80	73.16	**37.01**	19.12	30.38	**36.82**	59.44	47.81

C.7 Discovery-Budget Allocation

Table 11 studies how the fixed per-target optimization budget is divided between competitor discovery and the subsequent targeted base attack. Rather than changing the total attack budget, we keep

$$T = T_d + T_a = 100, \tag{100}$$

where T_d iterations are allocated to DCD and $T_a = 100 - T_d$ iterations remain for the base attack. This ablation therefore tests the allocation of a fixed gradient budget between *discovering* a promising competitor and *optimizing* the corresponding boundary-specific attack.

A very short discovery phase ($T_d = 2$) is consistently weaker, with an average robust accuracy of 48.31%. Performance improves rapidly once more iterations are allocated to discovery and remains stable for $10 \leq T_d \leq 30$, where the average varies by only 0.04 percentage points. The default $T_d = 15$ obtains the lowest average robust accuracy, 47.79%, while leaving $T_a = 85$ iterations for the base attack. Thus, a relatively small portion of the total budget is sufficient for discovery, and a broad range of discovery–attack allocations performs similarly well.

C.8 DCD Design Ablations

We use APGD-DCD1 to isolate four mechanism-level choices in DCD: whether competitor rankings are recomputed as the discovery state moves, whether the active support contracts over time, how gradients are allocated within the selected support, and whether the base attack restarts from the clean input after discovery. Unless stated otherwise, each variant changes only the named component of the default DCD procedure. Table 12 reports robust accuracy; lower is better.

Dynamic re-ranking. To isolate the value of re-observing the competitor landscape, *Initial Ranking* computes the competitor ordering once at x_0 and uses this fixed ordering to form the Top-K_t support throughout discovery. The mean-margin gradient is still evaluated at the current state x_{t-1}, so this variant removes only dynamic re-ranking. *Nested Pruning* keeps dynamic re-ranking but makes pruning irreversible: once a class leaves the support, later Top-K_t selections are restricted to the remaining classes, as in equation 95.

Freezing the initial ranking raises the average robust accuracy from 35.15% to 35.70% ($+0.55$ pp) and weakens the attack on all eight reported models. By contrast, Nested Pruning remains essentially unchanged at 35.16%. These results indicate that the important ingredient is re-evaluating the relative competitor ordering at updated observation states; allowing a previously pruned class to re-enter is much less important on these benchmarks.

Progressive support contraction. We next test whether DCD needs a coarse-to-fine support schedule. The TKMM variants keep a fixed Top-K mean-margin support for the entire discovery phase. The K_0 variants change only the initial support size while keeping the full eligible set available for later re-ranking, whereas the K_f variants change the terminal support size. *Linear Shrink* keeps progressive contraction but replaces the default geometric-style cardinality schedule with the linear schedule in equation 92.

Table 12: Ablation study of APGD-DCD1 design choices (robust accuracy, %). Lower is better. Rows varying $K_0/|\mathcal{U}|$ modify only the initial active-support cardinality while keeping the full eligible universe $\mathcal{U}$ available for subsequent re-ranking. The CE rows correspond to the T-CE family defined in Appendix A.6. AVG is the mean robust accuracy across all eight reported models.

	CIFAR-10		CIFAR-100		ImageNet		ImageNet12K		
Method	Carm.	Rebu.	Gowal	Rice	Engs.	Salm.	CXC	ViTC	AVG↓
Standard	59.80	66.78	37.01	19.08	30.36	36.82	**14.89**	16.48	35.15
DLR obj.	59.82	66.80	37.05	19.13	30.38	36.82	14.99	16.70	35.21
Use Endpoint	59.86	66.82	37.05	19.11	30.38	36.78	14.91	**16.29**	35.15
Rank-Margin	59.80	66.90	37.20	19.20	30.64	36.88	14.95	16.57	35.27
Nested Pruning	59.80	66.78	37.01	19.08	30.36	36.82	14.90	16.52	35.16
Initial Ranking	60.01	67.09	37.54	19.44	31.56	37.44	15.62	16.87	35.70
Linear Shrink	59.81	66.80	37.20	19.25	30.70	36.88	15.12	16.56	35.29
FullSupportCE	59.78	66.86	37.34	19.15	30.54	36.92	15.09	16.56	35.28
TCE $\tau = 100$	59.73	66.78	37.00	19.12	30.32	36.82	14.92	16.44	**35.14**
TCE $\tau = 10$	59.76	66.76	37.00	19.11	30.38	36.82	14.92	16.45	35.15
TCE $\tau = 5$	59.75	66.75	37.01	19.12	30.34	36.82	14.91	16.40	**35.14**
TCE $\tau = 2$	59.79	66.75	37.03	19.14	30.42	36.86	14.92	16.47	35.17
TCE $\tau = 1.5$	59.76	66.75	37.04	19.12	30.38	36.88	14.94	16.42	35.16
TCE $\tau = 1$	59.74	66.74	37.06	**19.04**	30.42	36.86	15.02	16.50	35.17
TCE $\tau = 0.9$	**59.69**	**66.73**	37.07	19.06	30.42	36.82	15.02	16.53	35.17
TCE $\tau = 0.8$	59.70	66.75	37.08	19.09	30.46	36.84	15.12	16.54	35.20
TCE $\tau = 0.7$	59.72	66.75	37.11	19.12	30.64	36.94	15.25	16.59	35.27
TCE $\tau = 0.5$	59.78	66.82	37.26	19.30	31.00	37.06	15.47	16.75	35.43
TCE $\tau = 0.2$	60.06	67.16	37.88	19.77	32.28	37.70	16.12	17.24	36.03
$K_0/\|\mathcal{U}\| = 0.5$	59.78	66.80	37.00	19.11	30.30	36.82	14.94	16.42	35.15
$K_0/\|\mathcal{U}\| = 0.25$	59.85	66.84	37.00	19.15	30.32	36.82	14.93	16.45	35.17
$K_0/\|\mathcal{U}\| = 0.1$	60.66	67.72	**36.99**	19.09	**30.28**	**36.76**	14.94	16.50	35.37
$K_f = 3$	59.80	66.85	37.09	19.26	30.52	36.94	14.96	16.44	35.23
$K_f = 6$	59.97	67.07	37.44	19.45	30.80	37.10	15.13	16.58	35.44
$K_f = 9$	60.06	67.01	37.49	19.49	30.94	37.10	15.14	16.66	35.49
TKMM $K = 2$	60.54	67.57	38.52	20.09	33.84	39.10	16.71	17.82	36.77
TKMM $K = 3$	60.58	67.64	38.50	20.13	33.78	39.14	16.71	17.80	36.78
TKMM $K = 5$	60.63	67.63	38.58	20.13	33.78	39.18	16.72	17.79	36.80
TKMM $K = 9$	60.67	67.62	38.62	20.13	33.80	39.22	16.78	17.81	36.83

All fixed-width TKMM variants are substantially weaker, with average robust accuracy between 36.77% and 36.83%, more than 1.6 pp above the default 35.15%. In contrast, reducing the initial support to $K_0/|\mathcal{U}| = 0.5$ or 0.25 has almost no effect (35.15% and 35.17%), while the more aggressive 0.1 setting degrades to 35.37%. Increasing the terminal support from the default $K_f = 2$ to $3, 6, 9$ progressively worsens the average to 35.23%, 35.44%, and 35.49%. Linear Shrink reaches 35.29%. Together, these results show that the key requirement is to move from sufficiently broad early coverage to a small late-stage support; the exact initial width and shrink curve are comparatively flexible.

Within-support gradient allocation. Default DCD uses current scores to choose the active support but assigns equal weight to the retained competitors. *Rank-Margin* adds ordinal weighting based on the current within-support ranking (equation 65), while T-CE uses score-gap magnitudes through the temperature-controlled weights in equation 69; smaller τ produces more concentrated allocation. *FullSupportCE* removes the dynamic support restriction and applies CE-style weighting over the full eligible set. Finally, *DLR obj.* changes the discovery objective while leaving the remaining DCD procedure unchanged.

Rank-Margin increases the average robust accuracy to 35.27%. High- and moderate-temperature T-CE remains close to uniform weighting (e.g., 35.14% at $\tau = 100$ and $\tau = 5$), but performance degrades as the allocation becomes more concentrated, reaching 36.03% at $\tau = 0.2$. FullSupportCE obtains 35.28%, compared with 35.17% for support-restricted T-CE at $\tau = 1$, while the DLR objective remains close to the default at 35.21%. Thus, DCD is not tied to a unique discovery loss, but it benefits from avoiding strong score-dependent commitment within the active support. In this role, scores are more reliable for screening plausible competitors than for determining precise gradient weights.

Restart after discovery. The *Use Endpoint* variant keeps the target selected by DCD but initializes the subsequent base attack from the discovery endpoint x_{T_d}, instead of restarting from the clean input x_0. This isolates whether DCD mainly benefits from a favorable warm start or from the target identified during discovery.

Use Endpoint has the same average robust accuracy as the default method, 35.15%, and neither handoff strategy consistently dominates across models. The discovery trajectory therefore appears to be useful primarily for identifying the competitor to attack, rather than for providing a better initial perturbation. Restarting from x_0 cleanly separates discovery from target-specific optimization without reducing aggregate attack strength.

Summary. The mechanism ablations point to three consequential choices: re-ranking competitors at evolving observation states, progressively contracting the support, and avoiding highly concentrated score-dependent weighting within that support. In contrast, re-entry after pruning, the precise shrink curve, the specific discovery objective, and the endpoint handoff have comparatively small effects. This distinction supports DCD as a structured discovery mechanism while leaving several implementation details flexible.