THE MEAN IS NOT ENOUGH: DEFEATING QUERY-BASED ATTACKS WITH LEARNED DISAGREEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Many defenses against query-based adversarial example (AE) attacks are vulnerable to Expectation over Transformation (EOT): by averaging repeated responses, an attacker can estimate the defense's mean response and optimize directly against it. But recovering the mean is useful only when AEs crafted against the mean transfer reliably across the predictors. We present DISSENT, which suppresses this transferability by learning a distribution of predictors that agree on clean inputs but *disagree along adversarial directions*. Its learning objective discourages AEs crafted against one sampled predictor from transferring to another, while preserving agreement on benign inputs. At inference, each query is answered by one randomly selected predictor in a single forward pass, avoiding the computational cost of evaluating multiple predictors for voting or averaging. Our theoretical analysis explains how learned disagreement can sustain robustness even when the attacker knows the mean. We evaluate DISSENT on CIFAR-10 and ImageNet with different architectures. On CIFAR-10 with ResNet-50, for example, DISSENT retains 59.5% robust accuracy under EOT, compared with 23.1% for the strongest evaluated baseline. Further evaluations against adaptive attacks, including those crafted on method-aware surrogates, demonstrate its robustness.

1 INTRODUCTION

Query-based adversarial attacks rely on a simple premise: repeated model responses provide a sufficiently stable signal to guide the attack (Ilyas et al., 2018; Andriushchenko et al., 2020; Chen et al., 2020a; Chen & Gu, 2020). One class of defenses challenges this premise by introducing per-query randomness, so that repeated attack queries receive different responses (Qin et al., 2021; Nguyen et al., 2024; Byun et al., 2022). An adaptive attacker, however, can account for this variability through Expectation over Transformation (EOT), using repeated evaluations to estimate the defense's mean response or expected attack objective (Athalye et al., 2018a;b). To isolate what remains once this estimation succeeds, grant the attacker exact knowledge of the mean response and let it find a perturbation that fools that mean. Under per-query sampling, the defense returns not this mean but the prediction of one sampled predictor, which may still classify the perturbed input correctly. Exact mean recovery removes the estimation problem but leaves a distinct transfer problem: reliable evasion requires the perturbation to transfer broadly across the predictor distribution.

Once the attacker can recover the mean, response variability alone is no longer the relevant defense objective. What matters instead is whether an adversarial example optimized against that mean transfers to the predictor sampled for the next query. Existing randomized query defenses vary responses to disrupt attack feedback (Qin et al., 2021; Nguyen et al., 2024), whereas this question begins after repeated querying has already stabilized that feedback. Related ensemble defenses promote diversity across their members (Yang et al., 2020; 2021), but their deployed prediction is formed by an ensemble rule; here, each query is answered by one sampled predictor rather than by an aggregate. Per-query sampling therefore calls for a more specific robustness structure: predictors should agree on benign inputs but respond differently along adversarial directions, so that a perturbation that fools the mean or one sampled predictor need not remain adversarial after resampling.

Guided by this principle, we introduce DISSENT, which makes selective adversarial disagreement a trainable property of a predictor distribution. Its training objective preserves agreement on clean inputs while discouraging adversarial failures from being shared across samples from that distribution.

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.

Table 1: Comparison of defense properties and robustness under adaptive attacks. *Mean EOT* averages repeated responses before each attack step (Table 3); *Hard-label EOT* stabilizes decision feedback through repeated queries; and *Transfer* crafts perturbations offline on surrogate models (Table 5). For attack columns, ✓ indicates retained protection, × indicates substantial compromise, and — denotes that Mean EOT is not applicable to a deterministic defense. Low inference cost denotes one forward pass per query without voting or averaging.

Defense	Stateless	Score-based EOT	Hard-label EOT	Transfer	Low inference cost
QPA (Li et al., 2025)	×	✓	✓	×	×
IG-BNN (Doan et al., 2022)	✓	—	×	×	×
Adv-BNN (Liu et al., 2019)	✓	×	✓	×	×
AAA (Chen et al., 2022)	✓	—	×	×	✓
RND (Qin et al., 2021)	✓	×	×	×	✓
RFD (Nguyen et al., 2024)	✓	×	✓	×	✓
DISSENT	✓	✓	✓	✓	✓

We instantiate this distribution with low-rank adapters on a frozen base network and answer each query with one randomly selected predictor in a single forward pass, without voting or averaging. To ask whether this mechanism survives perfect mean recovery, we analyze an attacker with exact access to the mean response and its gradient. The analysis identifies attack-aligned disagreement as the quantity that can separate fooling the mean from fooling the predictor that actually answers the query, and further relates surrogate-defense directional misalignment to transfer. These predictions are then tested directly on the trained network (Section 5.4).

We therefore evaluate whether DISSENT remains robust after systematically removing alternative explanations for its observed robustness. Repeated-query EOT stabilizes stochastic feedback, while offline transfer eliminates reliance on target-query feedback; method-aware surrogates further test whether the advantage survives when the attacker builds its surrogate from the published defense design. Across CIFAR-10 and ImageNet with multiple architectures, the robustness persists under these stronger tests; on CIFAR-10 with ResNet-50, DISSENT retains 59.5% robust accuracy under Square-EOT, compared with 23.1% for the strongest evaluated baseline in this setting. Table 1 summarizes the main attack settings together with DISSENT's one-forward-pass inference cost.

Our contributions are the following:

- We formulate robustness after mean recovery for per-query sampled defenses: fooling the mean is insufficient unless the perturbation remains reliably adversarial under predictor resampling. DISSENT operationalizes this principle by preserving agreement on clean inputs while discouraging adversarial failures from being shared across sampled predictors.
- We analyze this regime by granting the attacker exact access to the mean response and its gradient, showing how attack-aligned disagreement can sustain single-query robustness and how surrogate-defense directional misalignment can limit transfer; we test these predictions on the trained network.
- We stress-test DISSENT under repeated-query EOT, offline transfer, and method-aware surrogate attacks, and show that its robustness persists as the attacker is given progressively stronger mechanisms for adapting to the defense, while retaining one-forward-pass inference without voting or averaging.

2 RELATED WORK

2.1 QUERY-BASED AND TRANSFER-BASED ATTACKS

Query-based attacks use returned prediction scores or labels to iteratively construct adversarial examples, including score-based attacks such as NES and Square and decision-based attacks such as HopSkipJump and RayS (Chen et al., 2017; Ilyas et al., 2018; Andriushchenko et al., 2020; Brendel et al., 2018; Chen et al., 2020c;a; Chen & Gu, 2020; Simon-Gabriel et al., 2021). Against randomized defenses, repeated querying can stabilize this feedback: EOT averages responses to optimize a

mean or expected attack objective (Athalye et al., 2018a;b), while analogous repetition can stabilize hard-label decisions (Chen et al., 2020a; Simon-Gabriel et al., 2021). Transfer-based attacks instead bypass target-query feedback altogether by crafting perturbations on surrogate models and submitting only the resulting adversarial examples (Papernot et al., 2017; Liu et al., 2017), commonly using gradient-based attacks and techniques that enhance surrogate-to-target transfer (Madry et al., 2018; Dong et al., 2018; Xie et al., 2019; Dong et al., 2019).

2.2 Stateful and stateless query defenses

Stateful defenses identify attacks from relationships among queries, including similarity to stored inputs (Chen et al., 2020b; Li et al., 2022), query provenance (Li et al., 2025), and patterns in query updates (Park et al., 2025). They can detect an attack as it is being constructed, but require retaining and processing information across requests. Perturbations crafted offline and submitted without related target queries do not expose the construction sequence on which such detection relies (Chen et al., 2020b). We focus on stateless defenses that seek robust predictions without retaining or relying on query history.

Stateless defenses reshape returned scores (Chen et al., 2022; Fu et al., 2026; Dashtbani et al., 2026), optimize inputs at inference (Bokobza & Mirsky, 2025), or randomize inputs or intermediate features (Byun et al., 2022). Label-preserving score transformations leave the hard-label interface unchanged, while counter-sample optimization requires per-query gradients. RND and RFD randomize inputs (Qin et al., 2021) and intermediate features (Nguyen et al., 2024), respectively, and can affect both scores and labels with a single forward pass. DISSENT retains this inference setting while learning predictors that agree on clean inputs but disagree along adversarial directions, reducing the transferability on which mean-based attacks depend.

2.3 Bayesian defenses

Bayesian neural networks model parameter uncertainty through a posterior, but posterior uncertainty is not itself a robustness mechanism: posterior inference alone does not guarantee adversarial robustness (Feng et al., 2024; Arce et al., 2025a;b). Adversarially robust Bayesian methods therefore commonly combine posterior modeling with adversarial training or additional robustness objectives (Zhang et al., 2021; Liu et al., 2019; Doan et al., 2022). Among the baselines evaluated in this work, Adv-BNN and IG-BNN aggregate multiple posterior predictions at inference (Liu et al., 2019; Doan et al., 2022). A broader literature studies diversity-promoting ensembles and randomized or certified prediction (Pang et al., 2019; Kariyappa & Qureshi, 2019; Yang et al., 2020; 2021; Cohen et al., 2019; Lecuyer et al., 2019), many of which rely on aggregation or repeated model evaluations at inference. In contrast, DISSENT answers each query with a single predictor sampled from a learned distribution, while its training explicitly suppresses cross-predictor adversarial transfer and preserves agreement on clean inputs.

3 Threat Model

We adopt a standard method-aware black-box threat model commonly used in query-based defense evaluations, in which deployment-instance state, including the target base weights, trained adapter parameters, and realized predictor pool, remains private.

System and defender. We consider a prediction service that returns either a vector of class scores (*score-based*) or only the top-1 label (*decision-based*). The defender learns a distribution of predictors through low-rank adapters to a frozen base network (Section 4.2). The defense is *stateless*: it retains no query history and independently selects one predictor for each request. A single forward pass through the selected predictor produces the response, without voting or averaging.

Adversary. The adversary is *method-aware*: it knows the defense design and training procedure, but accesses the deployed model only through its prediction interface. The target's base weights, trained adapter parameters and sampled predictor pool remain private. Target-specific configuration choices, such as adapter placement and hyperparameters, are not assumed known, although the adversary may use published configurations or search over alternatives when training independent

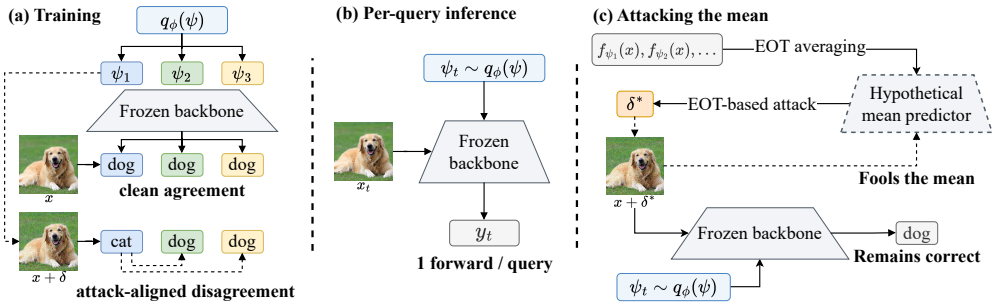

Figure 1: Overview of DISSENT. **(a)** Training preserves clean agreement while penalizing adversarial failures that transfer across sampled predictors. **(b)** One randomly selected predictor answers each query in a single forward pass. **(c)** An EOT-based attack can fool the hypothetical mean response, yet a newly sampled predictor may remain correct.

surrogates. It may choose queries adaptively and repeat evaluations at the same input to apply EOT. It may also craft perturbations offline using white-box surrogates, including independently trained instances of DISSENT, without querying the target during construction. Offline computation is unrestricted. Its goal is *untargeted evasion*: given a correctly classified input x, it seeks a perturbation δ to maximize the probability that $x + \delta$ is misclassified over the defense's randomness. Unless stated otherwise, $\|\delta\|_\infty \leq \epsilon$ with $\epsilon = 0.05$ in $[0, 1]$ pixel space. For query-based attacks, the default nominal budgets are 10^4 queries per input in the score-based setting and 2×10^4 in the decision-based setting. With m EOT repetitions per point, the interface-query budget is m times the nominal budget. Detailed attack settings, clean-input eligibility, and success criteria are specified in Section 5.1. Configurations and checkpoints released are provided for reproducibility and post-hoc verification; they are not part of the attacker knowledge assumed for a deployed black-box instance.

4 METHOD

DISSENT learns a distribution q_ϕ over predictors f_ψ that preserves agreement on clean inputs while suppressing adversarial transfer across sampled predictors. We first formulate this learning objective (Section 4.1), then instantiate the distribution with stochastic low-rank adapters to a frozen base network (Section 4.2). Finally, we analyze a stylized linear model in which the attacker knows the mean exactly (Section 4.3).

4.1 LEARNING AGAINST CROSS-PREDICTOR TRANSFER

Let q_ϕ be a trainable distribution over predictor-specific parameters ψ, with each $\psi \sim q_\phi$ defining a predictor f_ψ. On clean inputs, sampled predictors should remain consistent; under attack, an adversarial example optimized against one predictor should not reliably fool the others. We learn ϕ by minimizing

$$\mathcal{L}(\phi) = \mathbb{E}_{\psi \sim q_\phi}\big[\ell_{\mathrm{CE}}(f_\psi(x), y)\big] + \beta D_{\mathrm{KL}}\big(q_\phi \,\|\, p\big) + \lambda_{\mathrm{v}} \mathcal{V}(x) + \lambda_{\mathrm{c}} \mathcal{C}(x, y), \tag{1}$$

averaged over training pairs (x, y). At each step, we draw M independent predictors from q_ϕ. The cross-entropy term maintains clean accuracy and the KL term regularizes the learned distribution toward p.

Let $\mathcal{Y}$ denote the set of class labels. The clean-consistency term

$$\mathcal{V}(x) = \frac{1}{|\mathcal{Y}|} \sum_{k \in \mathcal{Y}} \mathrm{Var}_{\psi \sim q_\phi}[\mathrm{softmax}_k(f_\psi(x))] \tag{2}$$

encourages agreement in predictive probabilities on clean inputs.

To suppress cross-predictor transfer, for each sampled predictor f_{ψ_i} we construct an adversarial perturbation δ_i and evaluate it on the remaining predictors:

$$\mathcal{C}(x, y) = \frac{1}{M(M-1)} \sum_{i \neq j} \ell_{\mathrm{CE}}\big(f_{\psi_j}(x + \delta_i), y\big), \quad \delta_i = \arg\max_{\|\delta\|_\infty \leq \epsilon} \ell_{\mathrm{CE}}(f_{\psi_i}(x + \delta), y). \tag{3}$$

We approximate the inner maximization with a few PGD steps. Here f_{ψ_i} acts as the attack source, while f_{ψ_j} is trained to remain correct on the resulting perturbation. Thus $\mathcal{C}$ does not maximize predictor disagreement directly; instead, when a perturbation exposes a vulnerability of f_{ψ_i}, it discourages that vulnerability from being shared by the other sampled predictors. Together with $\mathcal{V}$, which suppresses variation on clean inputs, this biases predictor differences toward attack-induced directions rather than benign inputs. Empirically, the trained distribution exhibits this *attack-aligned disagreement*, which we examine in Section 5.4.

4.2 Stochastic Low-Rank Predictor Distribution

The objective above only requires a trainable distribution of predictors. We instantiate q_ϕ with stochastic low-rank adapters on a shared frozen base network.

Let f_0 be the frozen base classifier and $\mathcal{A}$ the adapted convolutional layers. For each $\ell \in \mathcal{A}$, a low-rank adapter (Hu et al., 2022) adds $\Delta W_\ell = B_\ell A_\ell$, where $B_\ell \in \mathbb{R}^{c_\ell^{\text{out}} \times r}$, $A_\ell \in \mathbb{R}^{r \times c_\ell^{\text{in}}}$, and $r \ll \min(c_\ell^{\text{in}}, c_\ell^{\text{out}})$. The adapter parameters $\psi = \{(A_\ell, B_\ell)\}_{\ell \in \mathcal{A}}$ define a predictor f_ψ that shares the frozen base.

Following variational treatments of low-rank adapters (Yang et al., 2024; Wang et al., 2024; Blundell et al., 2015), we parameterize $q_\phi(\psi)$ as a mean-field Gaussian with learned means and standard deviations for A_ℓ and B_ℓ, and use the Gaussian prior $p(\psi) = \mathcal{N}(0, \sigma_p^2 I)$. Samples are obtained by reparameterization, $A_\ell = \bar{A}_\ell + \sigma_\ell^A \odot \xi_\ell^A$ and $B_\ell = \bar{B}_\ell + \sigma_\ell^B \odot \xi_\ell^B$, with independent standard-Gaussian noise, making the objective in Equation (1) differentiable with respect to ϕ.

At deployment, we draw a finite pool of predictors from q_ϕ and route each query independently to one pool member. Only that predictor is evaluated, requiring one forward pass without voting or averaging. Implementation details and pool sizes are given in Sections B.1 and 5.1.

4.3 Mechanism Analysis with Exact Mean Access

To isolate the role of predictor disagreement after mean-estimation error has been removed, we analyze a stylized linear margin model in which the attacker has exact access to the mean margin and its gradient.

Let

$$s_\psi(x) = w^\top x + b, \qquad w = \bar{w} + \zeta, \qquad \mathbb{E}[\zeta] = 0, \quad \mathrm{Cov}(\zeta) = \Sigma, \tag{4}$$

where a positive margin denotes a correct prediction. The mean margin is $\bar{s}(x) = \bar{w}^\top x + b$, and for a clean input x_0 let $\bar{\mu}_0 = \bar{s}(x_0) > 0$. An attacker that minimizes this mean margin within $\|x - x_0\|_\infty \le \epsilon$ reaches

$$x^\star = x_0 + \epsilon u, \qquad u = -\operatorname{sign}(\bar{w}), \qquad \bar{s}(x^\star) = \bar{\mu}_0 - \epsilon \|\bar{w}\|_1. \tag{5}$$

Proposition 1 (Single-query robustness)**.** *Under the linear model of Assumption 1, assume that $\zeta^\top x^\star$ is Gaussian (Assumption 2). For $\sigma_\star > 0$, the probability that a randomly selected predictor classifies $x^\star$ correctly is*

$$R(\epsilon) = \Phi\left(\frac{\bar{s}(x^\star)}{\sigma_\star}\right) = \Phi\left(\frac{\bar{\mu}_0 - \epsilon\|\bar{w}\|_1}{\sqrt{x^{\star\top}\Sigma x^\star}}\right), \tag{6}$$

where $\sigma_\star^2 = x^{\star\top}\Sigma x^\star$ is the variance of predictor margins at the attacked point. The proof and zero-variance case are given in Section A.2.

Once the mean is fooled, $\bar{s}(x^\star) < 0$, increasing $\sigma_\star$ at a fixed mean margin raises $R(\epsilon)$ toward $1/2$. Thus, in this model, disagreement reduces transfer from the attacked mean to a randomly sampled predictor even though the attacker knows the mean and its gradient exactly.

For comparison, Gaussian input noise applied to a fixed linear predictor gives margin variance $\rho^2\|w\|_2^2$, independent of the input (Lemma 1), whereas a predictor distribution gives $x^\top \Sigma x$ and can therefore vary with the attack direction. Under the additional covariance assumptions of Propositions 2 and 3, the same model further relates aligned disagreement to a positive per-query robustness floor and relates surrogate-direction mismatch to reduced transfer. These are conditional properties of the stylized model rather than general guarantees for the nonlinear network. We examine their empirical implications in Section 5.4.

Table 2: Clean and robust accuracy (%) under standard black-box query attacks.

Architecture	Defense	CIFAR-10				ImageNet			
		Clean	NES	Square	RayS	Clean	NES	Square	RayS
WRN-28 / WRN-101	Undefended	95.63	0.84	0.11	0.00	82.27	13.99	1.46	2.34
	RND	92.76	38.03	61.43	66.84	80.59	57.87	71.50	75.74
	RFD	94.72	31.58	61.68	63.16	81.60	77.41	78.39	80.12
	AAA	95.63	63.27	57.89	17.99	82.35	81.52	82.00	54.39
	IG-BNN	87.81	13.78	1.14	0.55	80.10	23.76	9.74	2.99
	Adv-BNN	93.60	43.42	44.06	54.84	75.93	56.25	78.59	82.69
	DISSENT	91.44	**64.47**	**70.15**	**87.91**	83.98	**83.19**	**85.29**	**85.80**
ResNet-50	Undefended	91.89	19.61	0.22	0.00	79.77	3.51	1.00	0.00
	RND	87.22	67.78	18.46	58.62	79.30	50.19	64.03	72.12
	RFD	90.60	72.78	39.08	59.78	78.41	70.74	70.49	77.44
	AAA	91.28	72.77	37.47	9.29	80.23	81.26	63.10	43.56
	IG-BNN	94.32	18.77	2.97	0.53	78.04	22.44	11.15	3.80
	Adv-BNN	88.51	71.88	36.75	55.74	77.44	30.96	64.10	72.05
	DISSENT	88.71	**73.37**	**64.57**	**76.14**	78.35	**87.02**	**79.82**	**84.18**

5 EVALUATION

We organize the evaluation around four questions: **Q1:** How does DISSENT compare with existing defenses under standard score- and decision-based query attacks? **Q2:** Does its robustness persist when repeated queries are used to average away stochastic feedback? **Q3:** Does the advantage remain when the attacker bypasses target-query feedback entirely or knows the defense procedure? **Q4:** Do empirical diagnostics support the learned-disagreement mechanism analyzed in Section 4? After describing the common experimental setup (Section 5.1), Section 5.2 addresses the first question, Section 5.3 addresses the next two, and Section 5.4 examines the mechanism.

5.1 EXPERIMENTAL SETUP

Models, defenses, and attacks. We evaluate on CIFAR-10 and ImageNet with ResNet-50 and WideResNet backbones, using CIFAR-10 with ResNet-50 for adaptive evaluations and diagnostics. We compare DISSENT against RND (Qin et al., 2021), RFD (Nguyen et al., 2024), AAA (Chen et al., 2022), IG-BNN (Doan et al., 2022), and Adv-BNN (Liu et al., 2019). For DVERGE (Yang et al., 2020), we use official ResNet-20 checkpoints because our retraining attempts were unstable. DVERGE targets transfer robustness rather than query defense; we compare its original ensemble (DVERGE-Ens) and a per-query random-member extension introduced here (DVERGE-Samp) under transfer attacks in Section 5.3. For DVERGE, reproduction details and supplementary NES-EOT and Square-EOT evaluations are provided in Section B.5. Standard-query evaluations use NES and Square for score-based attacks and RayS for decision-based attacks; EOT and transfer adaptations are introduced in Section 5.3, with full configurations in Sections B.1 to B.3.

Evaluation protocol. We report conditional robust accuracy over inputs correctly classified by a single response of each defense. Following the majority-verification approach of RFD (Nguyen et al., 2024), randomized defenses use repeated responses only to verify attack success and determine final outcomes, with nine responses in the standard-query evaluations. This evaluator-side check reduces premature stopping due to stochastic label flips and does not alter the deployed prediction rule: each query to DISSENT is still answered by one sampled predictor in a single forward pass. Decision-based attacks separately use repeated queries to stabilize search feedback. Section B.3 gives the verification counts and other protocol details and reports the standard-query results with majority verification disabled throughout.

5.2 Q1: STANDARD QUERY ATTACKS

Across all four dataset–architecture configurations in Table 2, DISSENT achieves the highest robust accuracy under NES, Square, and RayS, retaining at least 64.5% on CIFAR-10 and 79.8% on

(a) NES-EOT

Defense	m=1	m=10	m=30
RND	$68.4_{\pm 1.07}$	$46.6_{\pm 0.98}$	$41.5_{\pm 1.25}$
RFD	$72.8_{\pm 0.30}$	$50.2_{\pm 1.63}$	$43.9_{\pm 1.21}$
Adv-BNN	$71.9_{\pm 0.13}$	$65.2_{\pm 0.89}$	$57.1_{\pm 1.02}$
DISSENT	$73.7_{\pm 0.63}$	$69.0_{\pm 0.97}$	$67.8_{\pm 0.99}$

(b) Square-EOT

Defense	m=1	m=10	m=30
RND	$18.5_{\pm 1.71}$	$10.7_{\pm 0.78}$	$10.1_{\pm 1.21}$
RFD	$39.6_{\pm 1.07}$	$29.1_{\pm 1.94}$	$21.4_{\pm 1.52}$
Adv-BNN	$36.8_{\pm 1.10}$	$36.4_{\pm 0.77}$	$23.1_{\pm 1.09}$
DISSENT	$62.8_{\pm 1.41}$	$61.9_{\pm 0.64}$	$59.5_{\pm 1.46}$

Table 3: Robust accuracy (%) under EOT attacks with m samples per query on CIFAR-10 and ResNet-50. Entries are mean±sample std over five runs with different seeds: the injected noise of RND and RFD; the pool draw or routing of Adv-BNN and DISSENT.

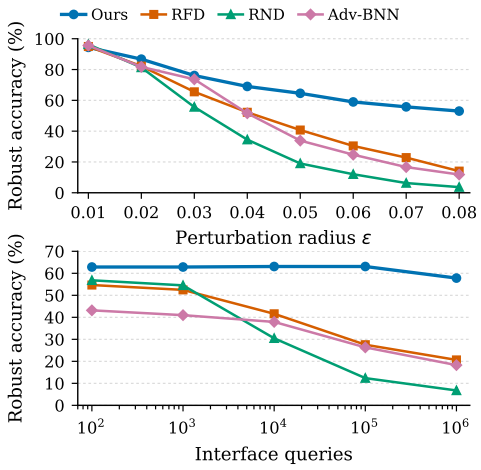

Figure 2: Robust accuracy under Square as the attacker raises the perturbation radius (top) and the total interface queries (bottom) on CIFAR-10 and ResNet-50.

ImageNet. Some baselines remain competitive under individual attacks, but none matches this consistency across both score- and decision-based interfaces. These results establish robustness under standard attacks but do not determine whether the advantage survives once an attacker explicitly accounts for the defense's stochasticity; **Q2** tests through repeated-query attacks that stabilize the observed feedback (Section 5.3).

5.3 Q2–Q3: ADAPTIVE AND TRANSFER ATTACKS

Q2 averages away stochastic feedback through repeated querying, while **Q3** bypasses feedback from the deployed target altogether and further considers method-aware surrogates.

EOT-stabilized score feedback. We evaluate both *EOT-mean* with NES and *EOT-loss* with Square (Athalye et al., 2018a;b), increasing the number of repeated responses per attack point from $m = 1$ to $m = 30$. As Table 3 shows, the randomized baselines degrade markedly as m increases, whereas DISSENT remains comparatively stable under both formulations. Thus, its robustness largely persists after stochastic score feedback is stabilized by repeated averaging.

Extended query and perturbation sweeps. We further stress the score-based evaluation to test whether the EOT results depend on the default query budget or perturbation radius. As Figure 2 shows, DISSENT degrades gradually as either increases, while the randomized baselines deteriorate much more sharply.

EOT-stabilized hard-label feedback. To test whether the EOT result extends to label-only interfaces, we evaluate ℓ_2 HopSkipJump (Chen et al., 2020a)[1] with an EOT-majority oracle and report REACH@p, the fraction of inputs driven to a single-query misclassification probability of at least p. Although the attack crosses the majority boundary on nearly all inputs, it reaches reliable misclassification much less often for DISSENT: REACH@0.9 is 14.4%, compared with 20.4–66.3% for the other stochastic defenses (Table 4). Thus, stabilizing hard-label feedback does not eliminate DISSENT's advantage; full attack and distance results are given in Section B.4.

These results are also consistent with the exact-mean mechanism analysis in Section 4.3: even eliminating mean-response estimation error does not by itself guarantee that an adversarial example optimized against the mean will transfer reliably to the predictor sampled for a query.

[1]PopSkipJump (Simon-Gabriel et al., 2021) did not converge reliably under our evaluation settings (ASR below 1% for RND, RFD, and DISSENT); we therefore report EOT-stabilized HopSkipJump.

Table 4: Decision-based robustness under HSJA-EOT on CIFAR-10 and ResNet-50. REACH@p is the fraction of inputs the attack drives to a single-forward misclassification probability $\geq p$; *lower is more robust*.

	Undefended	RND	RFD	AAA	IG-BNN	Adv-BNN	DISSENT
REACH@0.7	100.0	89.8	55.6	100.0	100.0	40.9	19.4
REACH@0.9	100.0	66.3	30.4	100.0	100.0	20.4	14.4

Table 5: Robust accuracy (%) under transfer attacks on CIFAR-10.

Defense	Single			Ensemble		
	PGD	MI-FGSM	MI-DI-TI	PGD	MI-FGSM	MI-DI-TI
RND	23.94	19.16	19.22	25.67	15.83	11.59
RFD	21.47	17.58	16.79	21.71	13.02	10.29
AAA	22.08	18.57	17.68	22.93	13.72	10.66
IG-BNN	32.70	27.19	21.20	36.18	27.63	14.12
Adv-BNN	35.14	29.87	28.39	37.66	26.36	20.34
DVERGE-Ens	79.72	67.64	49.73	87.49	69.58	53.83
DVERGE-Samp	77.97	66.96	48.35	84.91	67.29	53.41
DISSENT	85.84	82.50	70.40	88.36	84.72	72.55

Offline transfer without target feedback. **Q3** next removes target-query optimization altogether. If DISSENT benefited mainly from making the deployed feedback difficult to optimize, an attacker that constructs perturbations entirely offline on surrogate models should bypass that advantage. We generate perturbations using 50 steps of PGD, MI-FGSM, or MI-DI-TI on an independent AugMix-trained ResNeXt (Xie et al., 2017) (*single*) or an equal-weight ensemble of four independent models with different architectures (*ensemble*), and evaluate the resulting examples on the full CIFAR-10 test set. As Table 5 shows, DISSENT outperforms all evaluated baselines, including DVERGE-Ens and DVERGE-Samp, across all attacks and both surrogate settings. Under MI-DI-TI, it retains at least 70.4% robust accuracy, exceeding the strongest baseline in each setting by 18.7–20.7 percentage points. The advantage therefore persists when attack construction receives no feedback from the deployed target.

Method-aware surrogate transfer. Offline transfer still leaves open whether knowledge of the defense procedure enables stronger surrogates. We therefore train method-aware surrogates following the DISSENT training procedure on independently trained base models, without access to the deployed base or trained adapters, varying the loss weights, adapter depth, or backbone in accordance with the black-box threat model of Section 3. We attack these surrogates with PGD, MI-FGSM, and MI-DI-TI as above, using EOT with $m \in \{1, 10\}$ to account for predictor randomness. Table 6 summarizes the three surrogate groups, with full per-surrogate results in Table 17. Across all surrogate groups, attacks, and EOT settings, DISSENT retains 37.8–63.7% robust accuracy. Thus, under our threat model, knowledge of the training procedure alone does not eliminate the transfer gap.

5.4 Q4: MECHANISM DIAGNOSTICS

Q4 asks whether the trained predictor distribution exhibits the empirical signatures suggested by the learning objective and the stylized analysis in Section 4. We examine whether cross-vulnerability training yields EOT-persistent robustness, whether the resulting predictor distribution exhibits attack-aligned disagreement, and whether directional mismatch is associated with reduced transfer. Additional linearization and assumption checks are provided in Sections A.6 and A.8.

Role of cross-vulnerability training. We first test whether robustness that survives EOT depends on the cross-vulnerability term $\mathcal{C}$ rather than on predictor randomness alone. Across the three full-depth adapter placements, $\lambda_c = 0$ yields substantial robustness at $m = 1$ but much larger degradation at $m = 30$; EOT-surviving robustness emerges consistently only once λ_c is sufficiently large

Table 6: Robust accuracy (%) against method-aware surrogates on CIFAR-10 and ResNet-50. Mean over surrogates and EOT-m settings; minimum in parentheses.

Surrogate	PGD	MI-FGSM	MI-DI-TI
Different training loss weights	54.86 (54.59)	42.99 (42.76)	42.29 (42.06)
Different adapter depth	53.58 (52.87)	42.97 (42.58)	43.39 (42.37)
Different backbone	59.28 (56.94)	47.81 (46.06)	40.65 (37.76)

Defense	R		Gradient var.	
	Pred.	Emp.	u	clean ($\times 10^{-4}$)
Undefended	0.0898	0.0898	0.00	0.0
RND	0.0689	0.0688	3.60	105.0
RFD	0.0912	0.0914	5.27	1.5
DISSENT	0.3595	0.3599	5.52	3.9

Table 7: Mechanism diagnostics on CIFAR-10 and ResNet-50. Pred. and Emp. R are the input-averaged robustness predicted by Proposition 1 and observed at $x^\star$, respectively. Gradient variance is measured along the attack direction u and random clean directions.

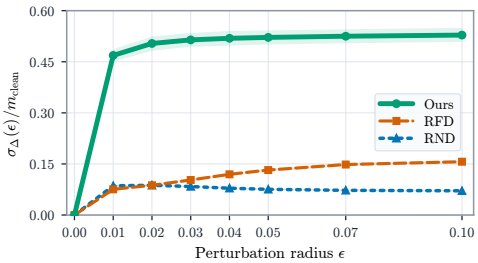

Figure 3: Attack-induced disagreement versus perturbation radius on CIFAR-10 and ResNet-50: $\sigma_\Delta(\epsilon) = \mathrm{std}_\psi[\mu_\psi(x^\star(\epsilon)) - \mu_\psi(x)]$, with $x^\star(\epsilon) = x + \epsilon u$, normalized by each defense's median clean margin.

(approximately 2 in our sweep), with a corresponding clean-accuracy trade-off. The full ablation is reported in Section B.7.

Calibration and attack-aligned disagreement. At each attacked input $x^\star$, substituting the measured mean and standard deviation of the predictor margins into Equation (6) yields a probability that agrees with the empirical fraction of correct predictors to within 4×10^{-4} for every evaluated defense (Table 7). Because both use measurements at the same attacked points, this calibrates the Gaussian moment-to-probability relation rather than independently predicting robust accuracy. For DISSENT, the margin-gradient variance along the attack direction exceeds that along random clean directions by more than four orders of magnitude, while Figure 3 shows that attack-induced margin disagreement rises rapidly and remains near 0.53, compared with 0.08 for RND and 0.15 for RFD. Although RFD has comparable directional gradient variance, its finite-radius disagreement is much smaller, showing that local variance alone does not characterize the attack-induced structure associated with Proposition 2 and the aligned-disagreement objective in Section 4.1.

Directional mismatch and transfer. Finally, we use diagnostic probes with different cross-vulnerability weights, adapter depths, or backbones. The loss-weight probes retain similar decision-direction alignment to the deployment, with angles of 73.7°–76.9° and transfer rates of 0.49–0.55. Structural changes in adapter placement or backbone increase the decision-direction angle to approximately 84°–89° and reduce transfer to 0.37–0.46. This separation across probe families supports the qualitative prediction of Proposition 3: greater directional mismatch between the surrogate and deployed predictor distribution suppresses adversarial transfer. Detailed results are provided in Section A.7.3.

6 CONCLUSION

We presented DISSENT, a stateless defense that counters score- and decision-based query attacks by learning disagreement along adversarial directions while preserving agreement on clean inputs. Our analysis explains how learned disagreement can sustain robustness even when the mean response is known and how directional mismatch affects surrogate transfer. Our experiments demonstrate substantial robustness gains under EOT and transfer attacks, with each query requiring only one forward pass. Future work will investigate robustness under more complex attack–defense scenarios.

Ethics Statement

This work studies adversarial attacks and defenses for machine learning prediction interfaces. All attacks are conducted in controlled experimental settings on standard benchmark datasets and research models for the purpose of evaluating and improving adversarial robustness. The experiments do not involve human subjects, personal or sensitive data, or attacks on deployed third-party systems.

Adversarial attack techniques have inherent dual-use potential, as methods developed to evaluate defenses may also inform attempts to evade machine learning systems. However, the attacks considered in this work are established methods from the adversarial machine learning literature, and we use them to provide rigorous and reproducible evaluation of defensive mechanisms. We encourage the use of these techniques only for authorized research, evaluation, and testing of machine learning systems.

AI Use Statement

Generative AI tools were used solely to assist with the writing and language polishing of this manuscript, including improving grammar, clarity, and phrasing. They were not used to formulate the research problem, develop the methodology or theoretical results, design experiments, implement the proposed method, analyze experimental results, or draw scientific conclusions. All AI-assisted text was reviewed and edited by the authors, who take full responsibility for the content of the manuscript.

Reproducibility Statement

We have taken several steps to make our results reproducible. Section A states every theoretical result with its assumptions and gives complete proofs, and Sections A.7 and A.8 describe the measurements that test these results and assumptions on the trained defense. Section B.1 identifies the base checkpoint, adapter placement, and training hyperparameters of every configuration (Table 11); and Section B.3 specifies the attack settings, query budgets, EOT accounting, surrogate models, and evaluation protocol, including how attack success is judged. These artifacts are released to reproduce and verify the reported experiments and do not alter the black-box attacker knowledge defined in Section 3. The EOT experiments report the mean and standard deviation over five runs that vary the attack and defense seeds. All datasets are publicly available. To preserve anonymity during review, we provide the code in this Anonymous Repository, which contains the training and evaluation code for DISSENT, together with a small set of checkpoints and trained LoRA adapters that allow the evaluation to be run directly.

References

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

Pablo G Arce, Roi Naveiro, and David Ríos Insua. Evasion attacks against Bayesian predictive models. In *Proceedings of the Forty-first Conference on Uncertainty in Artificial Intelligence*, volume 286, pp. 184–202. PMLR, 2025a.

Pablo G Arce, Roi Naveiro, and David Ríos Insua. A unifying Bayesian framework for adversarial robustness. *arXiv preprint arXiv:2510.09288*, 2025b.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pp. 274–283. PMLR, 2018a.

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pp. 284–293. PMLR, 2018b.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pp. 1613–1622. PMLR, 2015.

Alex Bogun, Dimche Kostadinov, and Damian Borth. Saliency diversified deep ensemble for robustness to adversaries. In *The AAAI-22 Workshop on Adversarial Machine Learning and Beyond*, 2022. URL https://openreview.net/forum?id=wGkmGrDsco8.

Roey Bokobza and Yisroel Mirsky. Counter-samples: A stateless strategy to neutralize black-box adversarial attacks. *ACM Transactions on Intelligent Systems and Technology*, 16(4):1–23, 2025.

Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations*, 2018.

Junyoung Byun, Hyojun Go, and Changick Kim. On the effectiveness of small input noise for defending against query-based black-box attacks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3051–3060, 2022.

Jianbo Chen, Michael I Jordan, and Martin J Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 ieee symposium on security and privacy (sp)*, pp. 1277–1294. IEEE, 2020a.

Jinghui Chen and Quanquan Gu. Rays: A ray searching method for hard-label adversarial attack. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1739–1747, 2020.

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

Sizhe Chen, Zhehao Huang, Qinghua Tao, Yingwen Wu, Cihang Xie, and Xiaolin Huang. Adversarial attack on attackers: Post-process to mitigate black-box score-based query attacks. *Advances in Neural Information Processing Systems*, 35:14929–14943, 2022.

Steven Chen, Nicholas Carlini, and David Wagner. Stateful detection of black-box adversarial attacks. In *Proceedings of the 1st ACM Workshop on Security and Privacy on Artificial Intelligence*, pp. 30–39, 2020b.

Weilun Chen, Zhaoxiang Zhang, Xiaolin Hu, and Baoyuan Wu. Boosting decision-based black-box adversarial attacks with random sign flip. In *European Conference on Computer Vision*, pp. 276–293. Springer, 2020c.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pp. 1310–1320. PMLR, 2019.

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

Hamid Dashtbani, Mehdi Dousti Gandomani, and AmirMahdi Sadeghzadeh. Random logit scaling: Defending deep neural networks against black-box score-based adversarial example attacks. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=CXafPv4aAG.

Bao Gia Doan, Ehsan M Abbasnejad, Javen Qinfeng Shi, and Damith C Ranasinghe. Bayesian learning with information gain provably bounds risk for a robust adversarial defense. In *International Conference on Machine Learning*, pp. 5309–5323. PMLR, 2022.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9185–9193. IEEE, 2018.

Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 4307–4316. IEEE, 2019.

Yunzhen Feng, Tim GJ Rudner, Nikolaos Tsilivis, and Julia Kempe. Attacking bayes: On the adversarial robustness of bayesian neural networks. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=C6wj17VBnu.

Yanzhang Fu, Jizhou Luo, and Zizheng Guo. Dashed line defense: Plug-and-play defense against adaptive score-based query attacks. In *The 29th International Conference on Artificial Intelligence and Statistics*, 2026. URL https://openreview.net/forum?id=lYqi9b9i6o.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A simple data processing method to improve robustness and uncertainty. In *International Conference on Learning Representations*, 2020.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *International conference on machine learning*, pp. 2137–2146. PMLR, 2018.

Sanjay Kariyappa and Moinuddin K Qureshi. Improving adversarial robustness of ensembles with diversity training. *arXiv preprint arXiv:1901.09981*, 2019.

Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 656–672. IEEE, 2019.

Huiying Li, Shawn Shan, Emily Wenger, Jiayun Zhang, Haitao Zheng, and Ben Y Zhao. Blacklight: Scalable defense for neural networks against {Query-Based}{Black-Box} attacks. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 2117–2134, 2022.

Shaofei Li, Ziqi Zhang, Haomin Jia, Yao Guo, Xiangqun Chen, and Ding Li. Query provenance analysis: Efficient and robust defense against query-based black-box attacks. In *2025 IEEE Symposium on Security and Privacy (SP)*, pp. 1641–1656. IEEE, 2025.

Xuanqing Liu, Yao Li, Chongruo Wu, and Cho-Jui Hsieh. Adv-BNN: Improved adversarial defense through robust Bayesian neural network. In *International Conference on Learning Representations*, 2019.

Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *International Conference on Learning Representations*, 2017.

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

Hung Quang Nguyen, Yingjie Lao, Tung Pham, Kok-Seng Wong, and Khoa Doan. Understanding the robustness of randomized feature defense against query-based adversarial attacks. In *International Conference on Learning Representations*, volume 2024, pp. 54019–54039, 2024.

Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. In *International Conference on Machine Learning*, pp. 4970–4979. PMLR, 2019.

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

Jeonghwan Park, Niall McLaughlin, and Ihsen Alouani. Mind the gap: Detecting black-box adversarial attacks in the making through query update analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10235–10243, 2025.

Ruoxi Qin, Linyuan Wang, Xuehui Du, Shuxiao Ma, Xingyuan Chen, and Bin Yan. An adversarial transferability metric based on svd of jacobians to disentangle the correlation with robustness. *Applied Intelligence*, 53(10):11636–11653, 2023.

Zeyu Qin, Yanbo Fan, Hongyuan Zha, and Baoyuan Wu. Random noise defense against query-based black-box attacks. *Advances in Neural Information Processing Systems*, 34:7650–7663, 2021.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

Carl-Johann Simon-Gabriel, Noman Ahmed Sheikh, and Andreas Krause. Popskipjump: Decision-based attack for probabilistic classifiers. In *International Conference on Machine Learning*, pp. 9712–9721. PMLR, 2021.

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

Yibin Wang, Haizhou Shi, Ligong Han, Dimitris Metaxas, and Hao Wang. Blob: Bayesian low-rank adaptation by backpropagation for large language models. *Advances in neural information processing systems*, 37:67758–67794, 2024.

Christopher Wiedeman and Ge Wang. Disrupting adversarial transferability in deep neural networks. *Patterns*, 3(5), 2022.

Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 2725–2734. IEEE, 2019.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 5987–5995. IEEE, 2017.

Adam Yang, Maxime Robeyns, Xi Wang, and Laurence Aitchison. Bayesian low-rank adaptation for large language models. In *international conference on learning representations*, volume 2024, pp. 1812–1842, 2024.

Huanrui Yang, Jingyang Zhang, Hongliang Dong, Nathan Inkawhich, Andrew Gardner, Andrew Touchet, Wesley Wilkes, Heath Berry, and Hai Li. Dverge: diversifying vulnerabilities for enhanced robust generation of ensembles. *Advances in Neural Information Processing Systems*, 33: 5505–5515, 2020.

Zhuolin Yang, Linyi Li, Xiaojun Xu, Shiliang Zuo, Qian Chen, Pan Zhou, Benjamin I. Rubinstein, Ce Zhang, and Bo Li. TRS: Transferability reduced ensemble via promoting gradient diversity and model smoothness. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17642–17655. Curran Associates, Inc., 2021.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference*, 2016.

Jiaru Zhang, Yang Hua, Zhengui Xue, Tao Song, Chengyu Zheng, Ruhui Ma, and Haibing Guan. Robust bayesian neural networks by spectral expectation bound regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3815–3824, 2021.

A Theoretical Analysis and Empirical Validation

This appendix states the analysis of Section 4.3 in full and tests it on the trained defense. Every result answers one question: **when an example is crafted against a fixed function, how likely is it to fool the predictor that answers the query?** Section A.1 fixes the model and assumptions. Sections A.2 to A.4 take the fixed function to be the mean response: single-query robustness, the contrast with input randomization, and the floor under aligned disagreement. Section A.5 takes it to be a surrogate or a differentiable proxy. Section A.6 connects the linear model to the trained network, Section A.7 tests each result on it, and Section A.8 checks the modeling approximations.

A.1 Setup and Assumptions

This section fixes the model, the attack, and the assumptions shared by all results that follow; each later result states only the conditions specific to it. We use the notation of Section 4.3 and the perturbation set of the main text, the ℓ_∞ ball $\{x : \|x - x_0\|_\infty \le \epsilon\}$ around a correctly classified input x_0. The only exception is Section A.5, which works in ℓ_2 for the reason given there.

Assumption 1 (Linear model)**.** Each predictor ψ has margin $s_\psi(x) = w^\top x + b$, where $w = \bar{w} + \zeta$ with $\mathbb{E}[\zeta] = 0$ and $\mathrm{Cov}(\zeta) = \Sigma \succeq 0$, and the bias b is fixed. A positive margin denotes a correct prediction. The mean response is $\bar{s}(x) = \bar{w}^\top x + b$, with clean margin $\bar{\mu}_0 = \bar{s}(x_0) > 0$.

Real predictors are nonlinear. Section A.6 maps each quantity of Assumption 1 to a gradient statistic of the trained network, and Section A.8 checks the local linearity on which this mapping relies.

Attack against the mean. We grant the adversary exact access to the mean response $\bar{s}$ and its gradient, which EOT can only estimate from repeated queries (Athalye et al., 2018b). Because $\bar{s}$ is linear, its minimizer within the perturbation set moves every coordinate against the sign of $\bar{w}$:

$$x^\star = \arg\min_{\|x - x_0\|_\infty \le \epsilon} \bar{s}(x) = x_0 + \epsilon u, \qquad \bar{s}(x^\star) = \bar{\mu}_0 - \epsilon\|\bar{w}\|_1, \tag{7}$$

where $u = -\operatorname{sign}(\bar{w})$ is the attack direction. We call this attacker the mean-gradient adversary; it is what a mean-based EOT attack approaches as its estimate of the mean becomes exact. The mean is fooled once ϵ exceeds $\epsilon_0 = \bar{\mu}_0 / \|\bar{w}\|_1$. Holding the exact gradient that gradient masking would hide, this adversary rules out masking as the source of any robustness derived below (Athalye et al., 2018a).

Robustness measure. Each query at $x^\star$ is still answered by one randomly selected predictor. The quantity of interest is the single-query robustness $R(\epsilon) = \Pr_\psi[s_\psi(x^\star) > 0]$, the probability that the selected predictor classifies $x^\star$ correctly. Equivalently, $1 - R(\epsilon)$ is the probability that the adversarial example crafted against the mean transfers to that predictor. Probabilities are over the selection of the predictor, either from the learned distribution or uniformly from the deployed pool. Section A.5 evaluates the same quantity when the example is crafted on a surrogate instead of the mean.

Assumption 2 (Gaussian projection)**.** At every attack point x considered, the projection $\zeta^\top x$ is Gaussian.

This holds exactly when ζ is Gaussian. Section A.8 checks it on the deployed pool, where $\zeta^\top x^\star$ is observed directly as the deviation of each predictor's margin from the mean margin.

A.2 Single-Query Robustness

We first ask how likely the adversarial example crafted against the mean is to fool the predictor that answers the query. Proposition 1 answers this in closed form.

Proposition (Single-query robustness; Proposition 1 restated)**.** *Under Assumptions 1 and 2, let $x^\star$ be the point in equation 7, and suppose $\sigma_\star > 0$. The probability that a randomly selected predictor classifies $x^\star$ correctly is*

$$R(\epsilon) = \Phi\left(\frac{\bar{s}(x^\star)}{\sigma_\star}\right) = \Phi\left(\frac{\bar{\mu}_0 - \epsilon\|\bar{w}\|_1}{\sqrt{x^{\star\top}\Sigma x^\star}}\right), \tag{8}$$

where Φ is the standard normal CDF and $\sigma_\star^2 = x^{\star\top}\Sigma x^\star$ measures margin disagreement across predictors at $x^\star$.

Proof. Since $w = \bar{w} + \zeta$ and b is fixed, a randomly selected predictor has margin $s_\psi(x^\star) = \bar{s}(x^\star) + \zeta^\top x^\star$ at the attacked point, where $\zeta^\top x^\star$ has mean zero and variance $x^{\star\top}\Sigma x^\star = \sigma_\star^2$. Therefore

$$R(\epsilon) = \Pr\left[\zeta^\top x^\star > -\bar{s}(x^\star)\right] = 1 - \Phi\left(-\frac{\bar{s}(x^\star)}{\sigma_\star}\right) = \Phi\left(\frac{\bar{s}(x^\star)}{\sigma_\star}\right), \tag{9}$$

where the second equality uses Assumption 2. Substituting $\bar{s}(x^\star)$ from equation 7 gives the second form. If $\sigma_\star = 0$, then $\zeta^\top x^\star = 0$ almost surely, every predictor returns the mean margin, and $R(\epsilon) = \mathbb{1}[\bar{s}(x^\star) > 0]$. □

The zero-variance case corresponds to a deterministic model: the adversarial example crafted against the mean transfers with certainty once $\epsilon > \epsilon_0$. With disagreement, robustness depends on two quantities. The adversary controls the numerator $\bar{s}(x^\star)$, which turns negative beyond ϵ_0; it does not target the denominator $\sigma_\star$, the disagreement at the point it reaches. Once the mean is fooled, $R(\epsilon) < 1/2$ and, at a fixed mean margin, R increases with $\sigma_\star$: disagreement at $x^\star$ is what keeps the example from transferring. Section A.7.1 tests this prediction on the deployed pool, both in aggregate and input by input.

Where the mean is still correct, the same disagreement pulls R toward $1/2$ from above and costs single-query accuracy. Disagreement is therefore useful only if it is low on clean inputs and high along adversarial directions, the split that the clean-consistency term $\mathcal{V}$ and the cross-vulnerability term $\mathcal{C}$ of the training objective are designed to produce. How $\sigma_\star$ evolves as the attack proceeds then determines robustness at larger radii: input noise contributes a variance that does not grow with the attack (Section A.3), whereas disagreement aligned with the decision direction grows with it and keeps R above a positive floor (Section A.4).

A.3 Comparison with Input Randomization

Section 4.3 contrasts learned disagreement with input randomization. This section derives the input-noise counterpart of Proposition 1 in the setting of Section A.1.

An input-noise defense evaluates a single fixed classifier with margin $s(x) = w^\top x + b$ at $x + n$, where $n \sim \mathcal{N}(0, \rho^2 I)$ is drawn afresh for each query and $s(x_0) = \mu_0 > 0$. Its mean response is $\mathbb{E}_n[s(x+n)] = s(x)$, the classifier itself, so the mean-gradient adversary attacks s directly and reaches $x^\star = x_0 - \epsilon\,\mathrm{sign}(w)$ with $s(x^\star) = \mu_0 - \epsilon\|w\|_1$. The margin returned at $x^\star$ is $s(x^\star) + w^\top n$, where $w^\top n \sim \mathcal{N}(0, \rho^2\|w\|_2^2)$.

Lemma 1 (Input-noise robustness)**.** *A single query at $x^\star$ is classified correctly with probability*

$$R_{\mathrm{inp}}(\epsilon) = \Phi\left(\frac{\mu_0 - \epsilon\|w\|_1}{\rho\|w\|_2}\right). \tag{10}$$

Proof. The argument of Proposition 1 applies with $\zeta^\top x^\star$ replaced by $w^\top n$, which is exactly Gaussian with variance $\rho^2\|w\|_2^2$. □

Proposition 1 and lemma 1 share their form and differ only in the denominator. Input noise contributes $\rho^2\|w\|_2^2$, the same at every input and every radius, whereas a distribution of predictors contributes $x^{\star\top}\Sigma x^\star$, which depends on where the attack moves. Two consequences follow. First, input noise admits no floor: R_{inp} decreases to zero as ϵ grows, and reaching $R_{\mathrm{inp}} \le \alpha < 1/2$ takes only a fixed extra radius beyond the radius $\mu_0/\|w\|_1$ at which the classifier is fooled:

$$\epsilon \;\ge\; \frac{\mu_0}{\|w\|_1} + |\Phi^{-1}(\alpha)|\,\frac{\rho\|w\|_2}{\|w\|_1}. \tag{11}$$

Second, the only way to lengthen this extra radius is to raise $\rho\|w\|_2$, which lowers single-query accuracy on clean inputs, $\Phi(\mu_0/(\rho\|w\|_2))$, through the same quantity: the variance is paid on clean and attacked inputs alike. A distribution of predictors can instead concentrate its disagreement along the attack direction and keep it low on clean inputs (Section A.4). What matters is therefore not where the randomness is injected but whether the resulting disagreement depends on the input and its direction (Lemma 1).

A.4 Robustness under Aligned Disagreement

Proposition 1 leaves open how $\sigma_\star$ behaves as the attack proceeds. Input noise keeps it fixed, so robustness vanishes beyond a fixed extra radius (Section A.3). This section identifies the covariance under which disagreement keeps pace with the attack and single-query robustness stays above a positive floor. For an arbitrary input x we write $R(x) = \Pr_\psi[s_\psi(x) > 0]$ and $\sigma(x)^2 = x^\top \Sigma x$, so that $R(\epsilon) = R(x^\star)$ and $\sigma_\star = \sigma(x^\star)$, and we denote the decision direction by $\hat{w} = \bar{w}/\|\bar{w}\|_2$.

Proposition 2 (Robustness floor under aligned disagreement)**.** *Under Assumptions 1 and 2, suppose that $b > 0$ and that $\Sigma \succeq \tau^2 \hat{w}\hat{w}^\top$ for some $\tau > 0$. Then, at every input x, a randomly selected predictor is correct with probability*

$$R(x) > \Phi\big(-\|\bar{w}\|_2/\tau\big). \tag{12}$$

If $\Sigma = \tau^2 \hat{w}\hat{w}^\top$, the bound is tight: along the mean-gradient attack, $R(\epsilon)$ decreases from $1/2$ at ϵ_0 toward $\Phi(-\|\bar{w}\|_2/\tau)$ as $\epsilon \to \infty$.

Proof. Write $a = \hat{w}^\top x$, so that $\bar{s}(x) = \|\bar{w}\|_2\, a + b$ and, since $\Sigma \succeq \tau^2 \hat{w}\hat{w}^\top$, $\sigma(x) \geq \tau|a|$. Applied at x instead of $x^\star$, the argument of Proposition 1 gives $R(x) = \Phi(\bar{s}(x)/\sigma(x))$ whenever $\sigma(x) > 0$.

If $\bar{s}(x) \geq 0$, then $R(x) \geq 1/2 > \Phi(-\|\bar{w}\|_2/\tau)$: either $\sigma(x) > 0$ and $\Phi(\bar{s}(x)/\sigma(x)) \geq 1/2$, or $\sigma(x) = 0$, which forces $a = 0$ and hence $\bar{s}(x) = b > 0$, so every predictor is correct.

If $\bar{s}(x) < 0$, then $a < -b/\|\bar{w}\|_2 < 0$ and $\sigma(x) \geq \tau|a| > 0$. Dividing the negative margin $\bar{s}(x)$ by the larger quantity $\sigma(x)$ can only raise the ratio, so

$$\frac{\bar{s}(x)}{\sigma(x)} \geq \frac{\|\bar{w}\|_2\, a + b}{\tau|a|} = -\frac{\|\bar{w}\|_2}{\tau} + \frac{b}{\tau|a|} > -\frac{\|\bar{w}\|_2}{\tau},$$

which gives equation 12.

If $\Sigma = \tau^2 \hat{w}\hat{w}^\top$, the first inequality holds with equality. Along the mean-gradient attack, $a = \hat{w}^\top x^\star = \hat{w}^\top x_0 - \epsilon\|\bar{w}\|_1/\|\bar{w}\|_2$ decreases linearly in ϵ and equals $-b/\|\bar{w}\|_2$ at ϵ_0, where $R = 1/2$. Beyond ϵ_0, $|a|$ grows without bound, so $R(\epsilon) = \Phi\big(-\|\bar{w}\|_2/\tau + b/(\tau|a|)\big)$ decreases toward $\Phi(-\|\bar{w}\|_2/\tau)$. □

Both conditions have a direct reading. At $x = 0$ the random part $\zeta^\top x$ vanishes and all predictors return the same margin b; taking this reference point to be the clean input x_0, on which the predictors are trained to agree, gives $b = \bar{\mu}_0 > 0$. The condition $\Sigma \succeq \tau^2 \hat{w}\hat{w}^\top$ requires every direction to carry disagreement at least τ times its component along $\hat{w}$, the only component of a perturbation that moves the mean. Together they make disagreement grow with the attack: pushing the mean below zero requires moving along $\hat{w}$, and $\sigma(x) \geq \tau|\hat{w}^\top x|$ grows with that movement, so wherever the mean is fooled, $\sigma(x)$ exceeds the fixed fraction $\tau/\|\bar{w}\|_2$ of $|\bar{s}(x)|$. In the linear model, a larger radius therefore deepens the mean margin without driving R toward zero, whereas input noise loses all robustness beyond a fixed extra radius (Lemma 1).

Because the bound holds at every input, it does not depend on how the adversary reaches its final point.

Corollary 1 (Floor against every adversary)**.** *Under the conditions of Proposition 2, no adversary can reduce single-query robustness below $\Phi(-\|\bar{w}\|_2/\tau)$, whatever objective it optimizes and whatever its access to the mean response and its gradient. In particular, the mean-gradient adversary of equation 7, which holds both exactly, satisfies $R(\epsilon) > \Phi(-\|\bar{w}\|_2/\tau)$ for every ϵ.*

Proof. Every attack ends at some input, and Proposition 2 bounds R at every input. □

Remark 1 (Robustness is not gradient masking). Gradient masking defends by denying the adversary a usable gradient (Athalye et al., 2018a). Corollary 1 holds with the exact gradient in the adversary's hands, so the floor cannot come from masked or obfuscated gradients; it comes from disagreement along the direction the attack must traverse. The floor is also consistent with the standard test for stochastic gradients: robustness that stems from randomized gradients disappears once the adversary averages enough responses at the same point (Athalye et al., 2018a;b), whereas here the averaged response is granted exactly and R still stays above the floor.

The floor depends on Σ only through τ, so disagreement outside the decision direction does not raise it. Where the mean is fooled, such disagreement can only increase R; where the mean is correct, including on clean inputs, it lowers R toward $1/2$ (Section A.2). Isotropic disagreement $\Sigma = s_\perp^2 I$ illustrates the trade: it satisfies Proposition 2 with $\tau = s_\perp$, but spends the same variance on every direction. Since the average variance along a random direction is $\mathrm{tr}(\Sigma)/d$ for any Σ, where d is the input dimension, a fixed amount of disagreement along random directions allows τ^2 to range from that amount, for isotropic Σ, up to d times it, for Σ concentrated on $\hat{w}$. The comparison in Section 5.4 between gradient variance along the attack direction and along random directions is the empirical counterpart of this concentration.

The floor is a statement about the linear model and bounds per-query robustness; at any finite radius R lies strictly above it. Real predictors are linear only locally (Sections A.6 and A.8), so the floor describes perturbations within that regime rather than arbitrarily large ones, which can turn the input into a natural image of another class on which all predictors agree. Its empirical counterparts are per-query measurements: the single-query robustness of the deployed pool at the attacked point (Section 5.4), and the hard-label attack of Table 15, which is not constrained in radius and, as expected of such an attack, crosses the majority boundary on nearly all inputs, yet drives far fewer of them to a misclassification probability of 0.9.

A.5 Surrogate Transfer under Directional Mismatch

Proposition 1 concerns an example crafted against our mean. A transfer adversary instead crafts its example against a surrogate it has built and submits it to the deployed defense. Under the conditions of Proposition 2, Corollary 1 already bounds its success by one minus the floor; this section shows how transfer depends on the direction the surrogate chooses. We work in ℓ_2 here, where a surrogate's attack direction coincides with its decision direction, so that the angle between decision directions, the quantity measured in Section A.7.3, is what enters the analysis; for an ℓ_∞ attack the same statements hold with θ the angle between $\hat{w}$ and the perturbation direction and ϵ the ℓ_2 length of the perturbation. As in the reading of Proposition 2, we place the reference point of the linear model at the clean input, $x_0 = 0$, so that $b = \bar{\mu}_0$ and disagreement is measured from the clean input.

A surrogate with unit decision direction $\hat{w}_{\mathrm{sur}}$ attacks along it, reaching $x^\star = -\epsilon \hat{w}_{\mathrm{sur}}$. Its alignment with our decision direction is $\cos\theta = \hat{w}^\top \hat{w}_{\mathrm{sur}}$, and its example transfers when the selected predictor misclassifies it, that is, when $\bar{s}(x^\star) + \zeta^\top x^\star < 0$.

Lemma 2 (General transfer)**.** *Under Assumptions 1 and 2, the example transfers to a randomly selected predictor with probability*

$$P_{\mathrm{trans}}(\epsilon) = \Phi\left(\frac{\epsilon \|\bar{w}\|_2 \cos\theta - \bar{\mu}_0}{\epsilon \sqrt{\hat{w}_{\mathrm{sur}}^\top \Sigma\, \hat{w}_{\mathrm{sur}}}} \right).$$

Proof. At $x^\star = -\epsilon \hat{w}_{\mathrm{sur}}$ the mean margin is $\bar{s}(x^\star) = \bar{\mu}_0 - \epsilon \|\bar{w}\|_2 \cos\theta$ and the disagreement is $\sigma(x^\star) = \epsilon \sqrt{\hat{w}_{\mathrm{sur}}^\top \Sigma \hat{w}_{\mathrm{sur}}}$. The argument of Proposition 1 gives $P_{\mathrm{trans}} = \Pr[\zeta^\top x^\star < -\bar{s}(x^\star)] = \Phi(-\bar{s}(x^\star)/\sigma(x^\star))$. □

Transfer is governed by two quantities. The numerator carries the alignment: only the component of the perturbation along $\hat{w}$ moves the mean, so a surrogate at angle θ needs radius $\epsilon/\cos\theta$ to move it as far as an aligned attack of radius ϵ. The denominator is our disagreement along the direction the surrogate chose, which the surrogate does not control. At $\theta = 0$ the surrogate attacks along our own decision direction, and Lemma 2 reduces to $1 - R(\epsilon)$ for the ℓ_2 version of Proposition 1: even an example built on our exact mean still faces the disagreement of the predictor that answers. A method-aware surrogate, trained with our procedure on its own base network and differing from our deployed configuration in its loss weights, adapter depth, or backbone, crafts its examples against its own mean response. Its decision direction need not match ours, and the disagreement that decides transfer is ours, along its direction, not its own.

Proposition 3 (Large-radius transfer limit under decision-direction mismatch)**.** *Under Assumptions 1 and 2, let the defense covariance be aligned plus isotropic, $\Sigma = \tau^2 \hat{w}\hat{w}^\top + s_\perp^2 I$ with $\tau, s_\perp > 0$, where $s_\perp^2$ is the residual disagreement outside the decision direction. An example crafted*

by a surrogate at angle θ transfers with a probability that increases with the radius and approaches

$$P^{\infty}_{\text{trans}}(\theta) = \Phi\left(\frac{\|\bar{w}\|_2 \cos\theta}{\sqrt{\tau^2 \cos^2\theta + s_\perp^2}}\right) \tag{13}$$

from below, so $P^{\infty}_{\text{trans}}(\theta)$ bounds transfer at every radius. Both this bound and the transfer probability at every radius decrease strictly as θ grows.

Proof. Here $\hat{w}_{\text{sur}}^\top \Sigma \hat{w}_{\text{sur}} = \tau^2 \cos^2\theta + s_\perp^2 =: D(\theta)^2$, so by Lemma 2

$$P_{\text{trans}}(\epsilon) = \Phi\left(\frac{\|\bar{w}\|_2 \cos\theta}{D(\theta)} - \frac{\bar{\mu}_0}{\epsilon\, D(\theta)}\right).$$

The second term is negative and vanishes as $\epsilon \to \infty$, so P_{trans} increases with ϵ and approaches $P^{\infty}_{\text{trans}}(\theta)$ from below. For the dependence on θ, write $c = \cos\theta \in [0,1]$ and let $g(c) = (\epsilon\|\bar{w}\|_2 c - \bar{\mu}_0)/(\epsilon\sqrt{\tau^2 c^2 + s_\perp^2})$ be the argument of Φ. Then

$$\frac{dg}{dc} = \frac{\epsilon\|\bar{w}\|_2\, s_\perp^2 + \bar{\mu}_0 \tau^2 c}{\epsilon\,(\tau^2 c^2 + s_\perp^2)^{3/2}} > 0,$$

and the argument of $P^{\infty}_{\text{trans}}$, $\|\bar{w}\|_2 c/\sqrt{\tau^2 c^2 + s_\perp^2}$, has derivative $\|\bar{w}\|_2 s_\perp^2/(\tau^2 c^2 + s_\perp^2)^{3/2} > 0$. Both arguments increase with c and hence decrease strictly in θ on $[0^\circ, 90^\circ]$. □

This covariance satisfies the condition of Proposition 2, and the two results agree: even a surrogate aligned with our mean ($\theta = 0$) is bounded by $\Phi(\|\bar{w}\|_2/\sqrt{\tau^2 + s_\perp^2}) < 1$, which leaves robustness above the floor $\Phi(-\|\bar{w}\|_2/\tau)$. At $\theta = 90^\circ$ the bound equals $1/2$, but transfer reaches it only in the limit of infinite radius: at radius ϵ, a perturbation orthogonal to $\hat{w}$ does not move the mean and transfers with probability $\Phi(-\bar{\mu}_0/(\epsilon s_\perp))$, through our residual disagreement alone. Between the two, misalignment reduces transfer at every radius because it cuts the mean movement in proportion to $\cos\theta$, while the disagreement it induces falls more slowly, since its isotropic part does not depend on direction. Section A.7.3 tests this trend on the trained network.

In a multiclass, multilayer network these mismatches compound. A surrogate with adapters at a different depth forms its gradient statistics in a different representation and propagates them through a different inter-layer Jacobian, which rotates its decision direction, and the margin gradient itself varies with the class pair. The diagnostic probes of Section A.7.3 bear this out: restricting the adapters to the last one or two blocks, with the base and objective unchanged, already moves θ beyond 84°, nearly as far as replacing the backbone.

Differentiable proxies. Backward Pass Differentiable Approximation (BPDA) circumvents a defense by replacing a non-differentiable or obfuscated component with a differentiable proxy, thereby recovering a usable gradient (Athalye et al., 2018a). DISSENT has no such component: every predictor is smooth and exactly differentiable, and the interface applies no non-differentiable transform to a fixed network. What remains for BPDA is to replace the per-query selection by a deterministic proxy and attack the proxy.

Corollary 2 (Differentiable proxies reduce to transfer)**.** *Suppose the adversary replaces the per-query selection $\psi \sim q_\phi$ by a deterministic differentiable proxy $\tilde{f}$ and attacks $\tilde{f}$ with its gradient. Its example is that of a fixed surrogate with decision direction $\hat{w}_{\tilde{f}} = \nabla\tilde{f}/\|\nabla\tilde{f}\|$, so it transfers with the probability of Lemma 2 at $\cos\theta = \hat{w}^\top \hat{w}_{\tilde{f}}$. The proxy $\tilde{f} = \bar{s}$ recovers the attack on our mean, with success $1 - R(\epsilon)$, and under the conditions of Proposition 2 no proxy lowers robustness below its floor.*

Proof. A gradient attack on a fixed proxy moves along the proxy's decision direction, so it reaches $x^\star = -\epsilon\hat{w}_{\tilde{f}}$, the setting of Lemma 2 with $\hat{w}_{\text{sur}} = \hat{w}_{\tilde{f}}$. For $\tilde{f} = \bar{s}$, $\hat{w}_{\tilde{f}} = \hat{w}$ and Lemma 2 gives $1 - R(\epsilon)$. The last claim is Corollary 1. □

Remark 2. BPDA and straight-through estimators recover a gradient that a defense hides. DISSENT hides none: the adversary may differentiate any predictor exactly. What it cannot do is make the predictor it differentiates be the predictor that answers the next query. Whatever proxy it differentiates, it attacks a fixed function and faces the transfer problem of Lemma 2.

A.6 Connection to Neural Predictors

The analysis above is stated for linear predictors. This section gives its correspondence with the trained network and the measurements that the correspondence enables.

Linearizing each predictor at the clean input x_0 gives $s_\psi(x) \approx s_\psi(x_0) + g_\psi^\top (x - x_0)$ with $g_\psi = \nabla_x s_\psi(x_0)$. This is the model of Assumption 1 with its reference point at the clean input, as in Sections A.4 and A.5. The mean and covariance of the slopes play the roles of $\bar{w}$ and Σ, $\bar{w} \mapsto \bar{g} := \mathbb{E}_\psi[g_\psi]$ and $\Sigma \mapsto \Sigma_g := \mathrm{Cov}_\psi(g_\psi)$, and the intercepts are the predictors' clean margins, which the clean-consistency term drives toward a common value, the fixed bias $b = \bar{\mu}_0$. The mean-gradient attack of equation 7 moves along $u = -\operatorname{sign}(\bar{g})$, and the disagreement it induces at radius ϵ, the attack-induced disagreement of Section 5.4, has standard deviation $\epsilon\sqrt{u^\top \Sigma_g u}$.

The correspondence has no free parameters: every quantity in the analysis is a statistic of the margins and gradients of the deployed predictors. Each result is therefore testable on the trained network at the level at which it is stated. Proposition 1 is a closed-form prediction and is tested input by input; the mean margin and disagreement it takes as inputs are read from the measured margins at $x^\star$, which include any residual clean offset and curvature, so this test does not depend on the linearization (Section A.7.1). Proposition 2 rests on disagreement concentrated along the attack direction and implies that disagreement grows as the attack proceeds; both are measured, through the gradient variance along u relative to random directions and through the attack-induced disagreement (Section A.7.2). Proposition 3 states a monotone relation between the decision-direction angle and transfer, which the diagnostic probes test under controlled changes of that angle (Section A.7.3). The linearization itself is local, and Section A.8 measures how much of each predictor's margin variation along the attack direction it accounts for.

Multiclass margins. With more than two classes, the margin of predictor ψ is the correct-versus-runner-up margin $\mu_\psi(x) = s_\psi^{(y)}(x) - \max_{k \neq y} s_\psi^{(k)}(x)$, which is positive exactly when ψ is correct. All results above apply to this margin unchanged in form, with $\bar{g}$ and Σ_g the mean and covariance of its gradient; this is the margin used in all measurements of Sections A.7.1 to A.7.3.

A.7 Empirical Validation of the Analysis

This section tests the analysis on the trained defense, following the three questions of Section 5.4. Unless stated otherwise the setting is that of Table 7: CIFAR-10 with ResNet-50, 256 test inputs, and the $K = 50$ predictors of the deployed pool. For each input, the attacked point $x^\star = x_0 + \epsilon u$ with $\epsilon = 0.05$ is formed from the mean margin gradient at the clean input, as in equation 7, and all margins are the correct-versus-runner-up margins of Section A.6. Each result is tested at the level at which it is stated: Proposition 1 as a numerical prediction (Section A.7.1), Proposition 2 through the concentration and growth of disagreement it requires (Section A.7.2), and Proposition 3 through the monotone relation between decision-direction angle and transfer (Section A.7.3). Section A.8 checks the modeling approximations themselves.

A.7.1 Predicting Single-Query Robustness

At each attacked point we measure the mean margin and the disagreement across the K predictors, insert them into Proposition 1, and compare the predicted robustness with the fraction of predictors that classify $x^\star$ correctly. Averaged over inputs, prediction and measurement agree to within 4×10^{-4} for every defense (Table 8); for DISSENT, the prediction is 0.3595 and the measured robustness 0.3599.

Figure 4 repeats the comparison input by input. The markers follow the diagonal for all three randomized defenses, without the systematic curvature that cancelling errors would produce, so the aggregate agreement reflects a per-input prediction.

The panels also show what the aggregate numbers hide. For RND the per-input robustness is close to 0 or 1: its disagreement at the attacked point is small compared with the margin the attack removes, as Lemma 1 predicts when the noise is small relative to that margin, so Φ saturates and each input is either certainly lost or certainly safe. Its aggregate robustness of 0.069 therefore counts the inputs the attack fails to reach rather than a per-query survival probability. DISSENT instead populates the

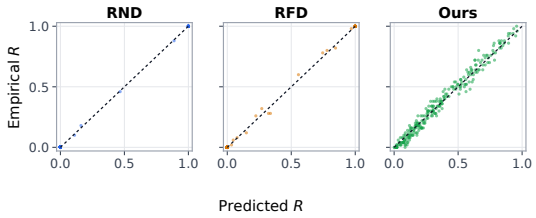

Figure 4: Per-input calibration of Proposition 1. For each of 256 inputs we form the attacked point $x^\star$, evaluate the $K = 50$ predictors of the deployed pool at that point, and compute two numbers from the same evaluation: the predicted robustness $\Phi(\bar{s}(x^\star)/\sigma_\star)$ from the measured mean margin and disagreement, and the empirical robustness, the fraction of predictors that still classify $x^\star$ correctly. Each marker is one input; the dashed line is $y = x$. The undefended model is omitted because it is deterministic ($\sigma_\star = 0$), so every input sits at 0 or 1 by construction; its aggregate agreement is reported in Table 7.

Table 8: Full version of Table 7 (CIFAR-10, ResNet-50; ℓ_∞, $\epsilon = 0.05$; 256 inputs, $K = 50$). Pred. R follows Proposition 1 from the measured mean margin and disagreement at $x^\star$, and Emp. R is the fraction of predictors that classify $x^\star$ correctly, both averaged over inputs. $\|\bar{g}\|$ is the norm of the mean margin gradient at the clean input. The last two columns are the variances of the margin gradient projected onto the unit attack direction and onto random clean directions. The undefended model is deterministic, so its disagreement vanishes.

Defense	Pred. R	Emp. R	$\|\bar{g}\|$	Var. along u	Var. along clean dir.
Undefended	0.0898	0.0898	45.85	0.00	0
RND	0.0689	0.0688	17.26	3.60	1.05×10^{-2}
RFD	0.0912	0.0914	45.58	5.27	1.50×10^{-4}
DISSENT	0.3595	0.3599	5.46	5.52	3.92×10^{-4}

whole diagonal: its aggregate robustness of 0.36 comes from inputs that survive each query with intermediate probability, the regime in which disagreement at $x^\star$ is comparable to the mean margin and the denominator of Proposition 1 decides the outcome.

A.7.2 DISAGREEMENT ALONG ADVERSARIAL DIRECTIONS

Proposition 2 requires disagreement concentrated along the direction the attack takes and growing as the attack proceeds. The directional columns of Table 8 measure the first property at the clean input. For DISSENT, the gradient variance along the attack direction exceeds that along random directions by four orders of magnitude. RND illustrates the alternative analyzed in Section A.3: it spends 27× more variance on random directions than DISSENT while achieving less along the attack direction.

Figure 3 measures the second property through the attack-induced disagreement $\sigma_\Delta(\epsilon)$, the spread across predictors of the change in margin from x_0 to $x^\star(\epsilon)$. This is the quantity the linear model identifies with the effect of the attack, $\epsilon\sqrt{u^\top \Sigma_g u}$ (Section A.6); the spread of the margins themselves also contains the clean offset between predictors, which carries no information about the attack and which the attack compresses toward zero. For DISSENT, σ_Δ rises rapidly and stays near 0.53 of the median clean margin from $\epsilon = 0.02$ through the largest radius tested. RND, whose disagreement is fixed by the injected noise, settles near 0.08 and, if anything, decays; RFD grows slowly to about 0.15, more than three times below DISSENT at every radius tested.

The growth saturates beyond $\epsilon \approx 0.02$. This saturation is consistent with the increasing departures from local linearity observed over wider neighborhoods of the clean input (Section A.8). What robustness at a given radius requires is disagreement at $x^\star$ comparable to the mean margin, and at $\epsilon = 0.05$, well beyond the saturation point, Section A.7.1 confirms this input by input. DISSENT therefore shows both properties that Proposition 2 requires; RND's disagreement is far less concentrated and does not grow, and RFD's grows far more slowly.

Table 9: Diagnostic probes: decision-direction mismatch versus transfer (CIFAR-10, ResNet-50 target; ℓ_∞, $\epsilon = 0.05$, PGD on the probe, 50 steps). Loss-weight probes vary the cross-vulnerability weight λ_c while keeping the target base, architecture, and adapter placement fixed; adapter-depth probes vary where the adapters are placed on the same base; backbone probes are the method-aware backbone surrogates of Table 6. θ is the angle between the mean margin gradients of the probe and deployment at the clean input, averaged over inputs; *Transfer* is the per-query success rate against the deployed pool; *Probe clean* is the probe's own clean accuracy.

Probe differs in	θ	Transfer	Probe clean
Cross-vulnerability weight ($\lambda_c = 1$)	73.71°	0.492	0.934
Cross-vulnerability weight ($\lambda_c = 2$)	75.44°	0.547	0.887
Cross-vulnerability weight ($\lambda_c = 3$)	76.93°	0.504	0.867
Adapter depth (layer4)	85.58°	0.427	0.953
Adapter depth (layer3–4)	84.74°	0.398	0.953
Backbone (WRN-28-10)	89.03°	0.374	0.938
Backbone (VGG-16)	88.58°	0.457	0.875
Backbone (ViT-B/16)	88.70°	0.450	0.957

A.7.3 Directional Mismatch and Transfer

Proposition 3 proves that, under the model assumptions, transfer at a fixed radius decreases as a surrogate's decision direction turns away from that of the deployed predictor distribution. We test this prediction using controlled diagnostic probes that vary one source of surrogate mismatch at a time: the cross-vulnerability weight, the adapter depth, or the backbone. The loss-weight and adapter-depth probes hold the target base checkpoint fixed, allowing us to isolate configuration-induced changes in the decision direction; the backbone probes are the method-aware backbone surrogates of Table 6, each trained on its own base. For each probe, we measure θ, the angle between its mean margin gradient and that of the deployment at the clean input, together with the per-query rate at which PGD examples crafted on the probe transfer to the deployed predictor pool.

Loss-weight probes. We first vary only the cross-vulnerability weight λ_c and retrain the probe while keeping the base checkpoint, architecture, adapter placement, and remaining training setup fixed. Across $\lambda_c \in \{1, 2, 3\}$, the mean decision-direction angle remains in a relatively narrow range of 73.71°–76.93°, and the corresponding transfer rates cluster between 0.492 and 0.547. These probes establish a controlled moderate-mismatch regime in which changes to the training objective alter the learned predictor distribution while leaving the first-order decision direction at a comparable level of alignment.

Structural directional mismatch. Changing the adapter placement or backbone produces a substantially larger directional shift. The adapter-depth probes move θ beyond 84°, with transfer rates of 0.398–0.427, while changing the backbone moves θ to approximately 89° and yields transfer rates of 0.374–0.457. Thus, relative to the loss-weight probes, structural changes consistently move the surrogate into a larger-mismatch regime and are accompanied by a further reduction in transfer. Notably, restricting adaptation to the last one or two residual stages is already sufficient to produce nearly the same directional separation and transfer reduction as replacing the backbone entirely.

Agreement with Proposition 3. All probes are evaluated under the same perturbation budget, $\epsilon = 0.05$, providing a fixed-budget test of the directional-mismatch effect predicted by Proposition 3. Across probe families, probes whose mean decision directions remain around 74°–77° transfer at approximately 0.50, whereas structural changes that increase the mismatch beyond 84° yield lower transfer rates of 0.374–0.457. The same shift appears whether the additional mismatch is induced by adapter placement or by a change of backbone, showing that the effect is not specific to a single source of surrogate variation. Taken together, these measurements support the mechanism captured by Proposition 3: greater separation between the surrogate and deployed decision directions suppresses adversarial transfer. The aligned-plus-isotropic model provides a tractable analytic form for this effect, while the neural-network measurements reproduce the same qualitative dependence across independently constructed probe families.

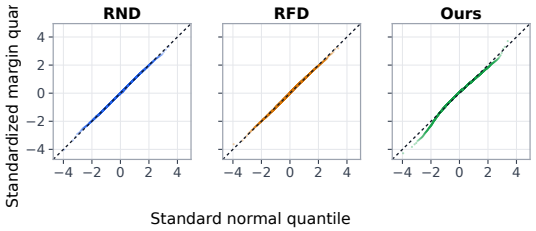

Figure 5: Gaussian projection of predictor margins at the attacked point (Assumption 2). For each input we standardize the margins of the $K = 50$ predictors at $x^\star$ by their own mean and standard deviation and pool across the 256 inputs (12,800 values per defense); the dashed line is the standard normal. RND, whose margin perturbation is to first order the Gaussian projection of its input noise, serves as a reference for the procedure.

Table 10: Local linearity of predictor margins along the attack direction, testing Assumption 1. For each of 128 inputs and 10 sampled predictors we evaluate the margin on a symmetric grid of points $x + t\,u$ along the attack direction, fit a line, and record its R^2 (1,280 curves per row). Rows differ in the half-width of the neighborhood. *Low-R^2 fraction* is the share of curves with $R^2 < 0.5$; for that subset we also refit a quadratic and report its median R^2, which indicates whether the departures reflect smooth curvature or noise, and the median clean margin, which indicates whether they concentrate near the decision boundary.

Half-width	Linear fit R^2			Low-R^2 subset		
	p10	median	p90	fraction	quad. R^2	clean margin
0.5ϵ	0.285	0.856	0.977	15.6%	0.796	2.54
0.3ϵ	0.580	0.955	0.994	8.4%	0.930	2.83

A.8 Checks of the Modeling Assumptions

The analysis uses two approximations: the Gaussian projection of Assumption 2 and the local linearity behind Assumption 1. We check both in the setting of Section A.7.

Gaussian projection. Figure 5 pools the predictor margins at $x^\star$, standardized per input, and compares them with the standard normal. All three defenses follow the reference line through the bulk of the distribution, and DISSENT departs mildly in both tails, toward slightly heavier tails than normal. RND, whose margin perturbation is to first order the Gaussian projection of its input noise, follows the line throughout, confirming that the procedure introduces no departure of its own. Tail departures affect $\Phi(\bar{s}/\sigma_\star)$ only where its argument is large, that is, where the predicted robustness is near 0 or 1 and is set by the mean margin alone; the per-input calibration of Figure 4 shows no systematic deviation there. The Gaussian projection therefore converts the measured mean and disagreement into a probability accurately wherever the disagreement matters.

Local linearity. Table 10 measures how much of each predictor's margin variation along the attack direction a straight line accounts for, over windows whose width is comparable to the perturbation itself. At half-width 0.3ϵ the median curve is linear to $R^2 = 0.955$, with 90% of curves above 0.580; at half-width 0.5ϵ the median falls to 0.856 and the low-R^2 fraction nearly doubles, as expected of a local approximation. The remaining departures are smooth curvature rather than noise: at half-width 0.3ϵ, a quadratic fits the low-R^2 curves with median R^2 0.930, and their median clean margin, 2.83, is close to the overall median of 2.98, so they are not concentrated near the decision boundary. The linearization is used to interpret the measured quantities through the gradient statistics of Section A.6; the test of Proposition 1 in Figure 4 uses measured margins and does not rely on it.

Table 11: Implementation and adapter-training settings of DISSENT for each configuration, reported for reproducibility. λ_c, λ_v, and β weight the cross-vulnerability, clean-consistency, and KL terms of the training objective. Depth lists the residual layers (groups for WRN-28-10) that carry adapters; coverage lists the adapted convolutions within each adapted block, by their index inside the block: †`conv1/3` in layers 1–2 and `conv1/2/3` in layers 3–4; ‡`conv1/2` and the shortcut convolutions; §stochastic BN affine parameters in addition. The optimizer is AdamW throughout.

Setting	CIFAR-10		ImageNet	
	WRN-28-10	ResNet-50	WRN-101-2	ResNet-50
Adapter rank r	10	4	1	2
Depth	stem, groups 2–3	layers 1–4	layers 1–4	layers 1–4
Coverage	`conv1/2`‡	mixed†	`conv1/2/3`§	mixed†
Pool size K	50	50	20	20
Samples/step M	3	2	2	2
Epochs	10	8	3	2
Learning rate	2.3×10^{-4}	2.5×10^{-4}	4.0×10^{-5}	4.5×10^{-5}
$\lambda_c, \lambda_v, \beta$	$(4.0, 0.065, 0.05)$	$(1.9, 0.036, 0.18)$	$(1.15, 0.018, 0.80)$	$(1.20, 0.018, 0.70)$

B ADDITIONAL EXPERIMENTAL DETAILS AND RESULTS

This appendix collects the material deferred from Section 5: exact baseline and attack configurations, the adapter placement per architecture, default-configuration numbers for the tunable defenses, and full per-architecture result tables.

B.1 IMPLEMENTATION DETAILS FOR OUR DEFENSE

This section gives the settings needed to reproduce each configuration of DISSENT: the base checkpoints, the adapter and training settings, and the serving implementation. Table 11 describes the reference configurations used to reproduce the experiments. These disclosures do not expand the deployment-time adversary knowledge defined in Section 3. In the evaluated black-box deployment, the target base weights, learned adapter parameters, realized pool, and instance-specific configuration remain unavailable through the prediction interface.

Base checkpoints. Every configuration freezes a clean base network that is not adversarially trained; Section B.8 attaches the defense to an adversarially trained base instead. In deployment the base should remain private, as for any model served as a black box; a defender starting from a public checkpoint can fine-tune it.

Adapter and training settings. Depth lists the residual layers (groups for WRN-28-10) that carry adapters, and coverage lists the convolutions adapted within them. For each configuration, Table 11 gives the adapter rank r, the depth and coverage, the number M of predictors sampled per training step, and the optimization settings; λ_c, λ_v, and β weight the cross-vulnerability, clean-consistency, and KL terms of the training objective. For the ImageNet WRN-101-2 configuration we additionally make the affine parameters of the batch-normalization layers stochastic, drawing their perturbation from an 8-dimensional latent variable with scale $\sigma_{\mathrm{BN}} = 3.5\times10^{-4}$ and training it with the adapters under the same objective; the other three configurations keep these parameters fixed.

Serving. At deployment we draw a pool of K predictors once from the learned distribution and store the sampled low-rank factors of each. Each query is routed independently and uniformly to one pool entry, whose factors are applied alongside the frozen base in a single forward pass, with no voting or averaging. The deployment thus stores the base, the trained adapter distribution, and the factors of each pool entry, about 1 MB per entry for the CIFAR-10 ResNet-50 configuration. Section B.7.2 compares the pool with drawing a fresh predictor for every query and reports latency and storage.

Table 12: Baseline configurations used in Section 5. Clean and robust accuracies for every method are reported in the main result tables.

Defense	Configuration	Source & notes
RND	Input-space Gaussian noise with $\sigma = 0.02$ in $[0, 1]$ pixel space, followed by clipping; identical on CIFAR-10 and ImageNet.	Project-native implementation following (Qin et al., 2021); no trained checkpoint required.
RFD	Feature-space Gaussian noise, with the noise strength and perturbed layers of the official implementation.	Project-native reproduction of (Nguyen et al., 2024), following its official implementation.
AAA	Official AAALinear defaults: attractor interval 6, reverse step 1, 100 optimization iterations, calibration-loss weight 5, learning rate 0.1, temperature 1, input quantization enabled.	Port of the official implementation (Chen et al., 2022). Deterministic, so a single response is used for judging.
IG-BNN	Ten particles initialized from the clean base with small perturbations and trained with the information-gain objective and SVGD-style repulsion; inference returns the mean particle logit.	Reproduction of (Doan et al., 2022) on the clean base; the adversarially trained base for which it is designed is evaluated in Section B.8. Deterministic at inference, so repeated queries are collapsed while the EOT cost is still charged to the interface-query count.
Adv-BNN	Bayesian convolution, batch-normalization, and linear layers with Gaussian posteriors (prior std 0.08, init std 0.001), initialized from the clean base and adversarially trained with PGD-10 at the corresponding radius, as in the original method; inference averages the softmax over 20 posterior samples.	Reproduction of (Liu et al., 2019), starting from the clean base. Stochastic at inference, so EOT is not collapsed.

B.2 Baseline Configurations and Reproduction

We follow each baseline's original configuration where possible and use a project-native reproduction when no public checkpoint exists for the required backbone; Table 12 lists the settings. RND uses input noise of standard deviation $\sigma = 0.02$ on every dataset, and RFD uses the configuration of its official implementation; the resulting clean-accuracy drops remain within the approximate operating points of Section 5.1 (Table 2). In the main tables, every defense is built on the same clean base rather than an adversarially trained one. IG-BNN is designed to start from an adversarially trained network, and Section B.8 evaluates it on such a base. The two Bayesian baselines also differ in how they use their posterior at inference, which matters for EOT accounting: IG-BNN averages the logits of a fixed particle set and is therefore deterministic across repeated queries, whereas Adv-BNN samples its posterior at every forward pass and remains stochastic.

B.3 Attack Configurations and Evaluation Protocol

Attack settings. Table 13 lists every attack reported in Sections 5.2 and 5.3. All attacks are untargeted and, except for HopSkipJump (Section B.4), operate in the ℓ_∞ ball of radius $\epsilon = 0.05$ in $[0, 1]$ pixel space. Query budgets follow Section 3: 10^4 queries per input for the score-based attacks and 2×10^4 for RayS; under EOT with m responses per point, the interface-query budget is m times the nominal budget. NES, Square, and RayS use the default hyperparameters of their public implementations. The transfer attacks run 50 steps with step size $\epsilon/10$ on the surrogate, with the remaining hyperparameters of MI-FGSM, DI, and TI at their reference defaults (Dong et al., 2018; Xie et al., 2019; Dong et al., 2019). The single surrogate is the AugMix-trained ResNeXt (Croce et al., 2020; Hendrycks et al., 2020; Xie et al., 2017). The ensemble combines, with equal weights, four independent CIFAR-10 models: ResNet56 (He et al., 2016), VGG-16 (Simonyan & Zisserman, 2015), and MobileNet-V2 (Sandler et al., 2018), and the WRN-28 (Zagoruyko & Komodakis, 2016). These surrogates all use architectures different from the target's; surrogates that share its architecture on

Table 13: Attack configurations. *Queries / steps* is the nominal interface budget per input for query attacks and the number of optimization steps for transfer attacks; under EOT with m responses per point the interface queries are m times the nominal budget. *Inputs* is the evaluation subset. The metric is robust accuracy, except REACH@p for HSJA-EOT (Section B.4).

Attack	Feedback	Queries / steps	Inputs	Reported in	Notes
NES	score	10^4	1,000	Table 2	
Square	score	10^4	1,000	Table 2	
RayS	hard label	2×10^4	200	Table 2	
NES-EOT	score	10^4	1,000	Table 3	EOT-mean, $m \in \{1, 10, 30\}$
Square-EOT	score	10^4	1,000	Table 3 and fig. 2	EOT-loss, $m \in \{1, 10, 30\}$
HSJA-EOT	hard label	1.2×10^4	200	Table 4	ℓ_2, EOT-majority oracle
PGD	none	50 steps	full test set	Tables 5 and 6	step size $\epsilon/10$
MI-FGSM	none	50 steps	full test set	Tables 5 and 6	step size $\epsilon/10$
MI-DI-TI	none	50 steps	full test set	Tables 5 and 6	step size $\epsilon/10$

an independently trained base are covered by the method-aware setting (Section B.6), whose closest group matches our configuration except for the loss weights.

Evaluation protocol. Score-based attacks use a fixed subset of 1,000 test inputs and decision-based attacks a fixed subset of 200, shared across defenses; transfer attacks use the full test set. An input is clean-eligible if the defense classifies it correctly, and robust accuracy is the fraction of clean-eligible inputs that remain correctly classified under attack. For a randomized defense, the final correctness outcome is determined by a strict majority of 9 independent responses; for a deterministic defense, by its single response. The majority enters only the reported metric: it changes neither the queries the attacker issues nor the forward passes the defense performs, and the deployed defense never votes. Two measurements use a different rule and state it where they appear: the hard-label results of Section B.4 report the fraction of inputs driven to a misclassification probability of at least p (REACH@p), and the diagnostics of Section A.7 use fractions of pool predictors.

EOT and repeated runs. NES-EOT averages the logits of m responses before computing its loss (EOT-mean), whereas the gradient-free Square-EOT averages the margin losses of m responses (EOT-loss); both are run with $m \in \{1, 10, 30\}$. EOT is a no-op for deterministic defenses; for IG-BNN the repeated queries are collapsed while their cost is still charged. The EOT results of Table 3 are the mean and sample standard deviation over five runs: the default seeds, two further attack seeds that change the sampling inside NES and Square, and two further defense seeds that change the injected noise of RND and RFD and the pool draw and routing of DISSENT. All other results, including those of Table 2, come from a single run with the default seeds and may therefore differ slightly from the five-run means of Table 3.

Evaluation without majority verification To assess sensitivity to the verification protocol, we rerun the standard-query comparison with majority verification disabled throughout, retaining the clean-eligible subsets and random seeds of the corresponding main-table runs. NES and Square use single-response success checks, and RayS also uses single-response decisions during search. Final outcomes are determined from one response. This comparison jointly changes early stopping, the search feedback used by RayS, and outcome assessment. Table 14 reports the four randomized defenses. Across all twelve dataset–architecture–attack combinations, DISSENT retains the highest conditional accuracy among these defenses. In these single runs, every entry differs from its corresponding majority-based result in Table 2 by at most 3.29 percentage points. The advantage over the evaluated randomized baselines therefore persists under the single-response attack protocol.

B.4 FULL EOT-HOPSKIPJUMP RESULTS

Table 15 extends the reliability summary of Table 4 to three misclassification levels and reports the perturbation size at each.

Table 14: Conditional robust accuracy (%) under standard query attacks with majority verification disabled throughout, including success checks, final outcome assessment, and the search decisions of RayS. Results are single runs on the same clean-eligible subsets as Table 2. Only randomized defenses are shown. WideResNet denotes WRN-28-10 on CIFAR-10 and WRN-101-2 on ImageNet.

		CIFAR-10			ImageNet		
Architecture	Defense	NES	Square	RayS	NES	Square	RayS
WideResNet	RND	41.17	60.67	64.71	59.85	70.38	73.37
	RFD	34.11	62.32	61.58	76.56	79.12	78.95
	Adv-BNN	42.57	46.20	55.91	58.33	77.19	81.41
	DISSENT	**63.22**	**71.06**	**84.62**	**84.05**	**83.68**	**86.39**
ResNet-50	RND	69.15	20.64	56.90	52.33	62.89	70.30
	RFD	71.90	38.31	60.87	69.86	71.25	75.61
	Adv-BNN	70.31	38.98	54.64	33.50	62.82	73.29
	DISSENT	**74.63**	**62.86**	**77.27**	**85.48**	**80.98**	**83.54**

Table 15: Full EOT-HopSkipJump results on CIFAR-10 with ResNet-50 (200 inputs). Each entry is REACH@p, the percentage of inputs driven to a single-query misclassification probability of at least p (lower is more robust), followed in parentheses by L_2@p, the smallest normalized perturbation $\|\delta\|_2/\sqrt{d}$ achieving it, averaged over those inputs (larger is more robust). $p = 0.5$ is the majority boundary the oracle targets. L_2@p is comparable across defenses only at similar REACH@p. For the deterministic defenses the misclassification probability is 0 or 1, so both quantities are essentially independent of p.

Defense	$p = 0.5$	$p = 0.7$	$p = 0.9$
Undefended	100.0 (0.162)	100.0 (0.162)	100.0 (0.162)
AAA	100.0 (0.151)	100.0 (0.150)	100.0 (0.150)
IG-BNN	100.0 (0.154)	100.0 (0.154)	100.0 (0.154)
RND	100.0 (0.068)	89.8 (0.072)	66.3 (0.081)
RFD	100.0 (0.067)	55.6 (0.080)	30.4 (0.094)
Adv-BNN	98.5 (0.058)	40.9 (0.069)	20.4 (0.085)
DISSENT	97.8 (0.063)	19.4 (0.087)	14.4 (0.131)

Attack and oracle. We run the ℓ_2 version of HopSkipJump (Chen et al., 2020a) on 200 CIFAR-10 inputs against the ResNet-50 configuration of each defense, with a nominal budget of 1.2×10^4 oracle calls. For the randomized defenses (RND, RFD, Adv-BNN, and DISSENT), each oracle call samples 15 responses at the queried point and reports the point as adversarial if at least 8 of them are misclassified, a strict majority; the interface-query budget is therefore 15 times the nominal budget, 1.8×10^5. The deterministic defenses (the undefended base, AAA, and IG-BNN) return the same response to repeated queries and use a single response per call.

Metrics. HopSkipJump minimizes the perturbation size rather than working within a fixed radius, so it eventually crosses the majority boundary on almost every input, and success alone says little about a defense. We therefore measure how reliably the attack misclassifies. The single-query misclassification probability of a point is estimated from $N = 100$ fresh responses drawn independently of the attack's oracle; for DISSENT these are uniform draws from the pool with replacement, an unbiased estimate of the misclassification probability of a single deployed query, and every randomized defense uses the same estimator. REACH@p is the fraction of inputs for which the attack finds a point whose estimated misclassification probability is at least p, and L_2@p is the smallest normalized perturbation $\|\delta\|_2/\sqrt{d}$ that achieves it, averaged over the inputs that reach p. At $p = 0.5$, the level the oracle targets, the point is selected among the candidates the attack retains, excluding its initialization; at $p \in \{0.7, 0.9\}$, it is found by a ray search along the final optimization direction, with the probability re-estimated at every probe. Since L_2@p averages only over reached inputs, it is comparable across defenses only at similar REACH@p.

Results. The undefended base fixes the reference point: HopSkipJump reaches every input at every level, so any residual robustness reflects the defense rather than a weak attack. The deterministic defenses expose no margin of consistency. Their misclassification probability at any point is 0 or 1, so once the attack pays the boundary distance, the input is adversarial at every level: REACH@p is 100% throughout, and L_2@p is essentially independent of p, the small differences coming from the two search procedures. A large distance in this regime measures the raw boundary distance rather than resistance to the attack. AAA's distance, 0.150–0.151, is close to that of the undefended base, 0.162, consistent with its label-preserving post-processing, which leaves the hard-label boundary in place and offers no protection against a decision-based attack.

The randomized defenses behave differently. Near the clean boundary their randomness already flips the majority label, so L_2@0.5 is small and comparable, between 0.058 and 0.068. Driving an input to consistent misclassification requires larger perturbations, and fewer inputs remain reachable as p grows. This growth is steepest for DISSENT: its distance roughly doubles, from 0.063 at $p = 0.5$ to 0.131 at $p = 0.9$, against increases of at most 47% for the other randomized defenses, and its REACH@0.9 of 14.4% is the lowest of all defenses. The attack thus crosses the majority boundary of DISSENT about as easily as those of the other randomized defenses, but rarely turns the crossing into a reliable misclassification: fooling the majority, like fooling the mean, is not enough to fool the answering predictor reliably (Section A.4).

B.5 DVERGE REPRODUCTION AND SUPPLEMENTARY EVALUATION

Scope and reproduction. DVERGE was designed to improve ensemble robustness against transfer attacks rather than to defend against query-based attacks (Yang et al., 2020). We exclude it from the main cross-dataset and cross-architecture comparison because our attempts to retrain it beyond its original CIFAR-10/ResNet-20 setting did not yield sufficiently stable or reproducible robustness. This difficulty is consistent with prior reports of substantial training variability in DVERGE and with the architecture-specific nature of its original implementation (Yang et al., 2020; Wiedeman & Wang, 2022; Bogun et al., 2022; Qin et al., 2023). We therefore use the official ResNet-20 checkpoints with the default configuration. Their architecture differs from the ResNet-50 and WideResNet backbones of our main query evaluations. We compare these checkpoints under the common transfer attacks in Table 5 and provide a separate ResNet-20 comparison under score-based EOT here.

Deployment and comparison. DVERGE-Ens follows the original method and averages members' softmax probabilities. DVERGE-Samp independently and uniformly selects one of the same trained members for each query, including EOT repetitions. This sampling extension is introduced here and is not part of DVERGE's original design. For this comparison, we train DISSENT on the same clean ResNet-20 base provided by the official DVERGE implementation. Both DVERGE-Samp and DISSENT answer each query with one predictor in a single forward pass. This comparison examines whether directly replacing DVERGE's aggregation with sampling can match the robustness of our learned predictor distribution.

Evaluation protocol. Following Section B.3, we evaluate NES-EOT and Square-EOT on the same fixed subset of $1{,}000$ CIFAR-10 test inputs, using an ℓ_∞ radius of $\epsilon = 0.05$, a nominal query budget of 10^4, and $m \in \{1, 10, 30\}$ responses per attack point. NES-EOT averages logits before computing the loss, whereas Square-EOT averages per-response margin losses. EOT repetitions are counted as interface queries. Clean eligibility is determined by a single response. For randomized defenses, majority verification with nine responses is used for attack stopping and final outcome assessment; this evaluator-side verification does not alter the deployed inference rule. Robust accuracy is conditional on clean eligibility.

Results. Table 16 shows that sampling substantially improves DVERGE's robustness over its original ensemble deployment. Nevertheless, DISSENT achieves higher robust accuracy at every tested m under both attacks, with similar clean accuracy. At $m = 30$, its gains over DVERGE-Samp are 12.90 percentage points under Square-EOT and 8.76 under NES-EOT. These results indicate that the gains of DISSENT are not attributable to sampled inference alone: applying the same inference principle to DVERGE does not recover the robustness obtained by our learned attack-aligned disagreement.

Table 16: Clean accuracy and conditional robust accuracy (%) on CIFAR-10 with ResNet-20. Here m is the number of responses per attack point. DVERGE-Ens is deterministic, so only $m = 1$ is reported; dashes indicate omitted redundant EOT evaluations.

Defense	Clean	NES-EOT			Square-EOT		
		$m = 1$	$m = 10$	$m = 30$	$m = 1$	$m = 10$	$m = 30$
DVERGE-Ens	90.81	25.81	–	–	4.84	–	–
DVERGE-Samp	89.22	77.35	57.46	52.49	73.78	70.81	66.27
DISSENT	88.72	**89.35**	**71.30**	**61.25**	**85.65**	**81.94**	**79.17**

B.6 METHOD-AWARE SURROGATE RESULTS

Table 17 reports each method-aware surrogate of Section 5.3 separately; Table 6 condenses each group to the mean and the minimum over its surrogates and both values of m.

Surrogate construction. Method-aware means knowledge of the defense family and training procedure, not access to the deployed instance. Therefore each surrogate must instantiate the method independently rather than reuse the target base or trained adapters. The surrogates form three groups, ordered by decreasing agreement with the deployed configuration. The first matches it in everything except the loss weights, the closest replication of our deployment short of its trained weights. The second keeps the ResNet-50 architecture but places the adapters at a different depth: a near match at layers 1–4, layers 3–4, or layer 4 only. The third changes the backbone to WRN-28-10, VGG-16, or ViT-B/16; these three surrogates are also the backbone probes of Section A.7.3, where their decision-direction angles are reported.

Attack settings. Perturbations are crafted on each surrogate with PGD, MI-FGSM, and MI-DI-TI under the settings of Section B.3, averaging each gradient step over $m \in \{1, 10\}$ predictors sampled from the surrogate to account for its own randomness, and are evaluated on the full CIFAR-10 test set against the deployed ResNet-50 defense.

Results. Across all surrogates, attacks, and values of m, robust accuracy ranges from 37.76% to 63.69%, and raising m from 1 to 10 changes it by at most 1.83 points for any surrogate and attack. Closer agreement with our configuration buys the attacker little: the closest replication, identical except for its loss weights and base, keeps robust accuracy above 42% under every attack and performs about the same as the surrogates that differ in adapter depth. Together with the diagnostic probes of Section A.7.3, where same-base probes occupy a substantially more aligned decision-direction regime than different-backbone probes, this indicates that knowledge of the training procedure and a closely matched configuration is not enough to build a strongly transferring surrogate without our base.

B.7 ABLATION STUDIES

We examine how adapter placement and the cross-vulnerability weight affect robustness, and how the sampling implementation affects accuracy and inference cost. These experiments use CIFAR-10 with ResNet-50, the default perturbation radius and query budgets of Section 3, and the evaluation protocol of Section 5.1, with NES at $m = 1$ and Square at $m \in \{1, 30\}$. Each entry is a single run with the default seeds (Section B.3), so small differences between neighboring settings should not be over-read.

B.7.1 ADAPTER PLACEMENT AND CROSS-VULNERABILITY WEIGHT

Experimental setup. Table 18 crosses six adapter placements with the cross-vulnerability weight $\lambda_c \in \{0, 1, 2, 3\}$, using the depth and coverage defined in Section B.1. With 1×1 convolutions, the depth covers layer 4, layers 3–4, layers 1–2, or layers 1–4. At full depth, we also add the 3×3 convolutions of layers 3–4, which gives the mixed coverage of Table 11, or those of all layers. The remaining training settings follow Table 11.

Table 17: Robust accuracy (%) against each method-aware surrogate (CIFAR-10, ResNet-50 defense, full test set); Table 6 summarizes the three groups. Each surrogate applies our training procedure to its own independently trained base and differs from the deployed configuration in the factor named; groups are ordered by decreasing agreement with that configuration. Perturbations are crafted with PGD, MI-FGSM, and MI-DI-TI, averaging over m sampled predictors of the surrogate per step. Higher is more robust.

Surrogate	m	PGD	MI-FGSM	MI-DI-TI
Different training loss weights				
Otherwise identical	1	55.12	43.21	42.06
	10	54.59	42.76	42.51
Different adapter depth				
Near match (layer1–4)	1	54.85	43.37	42.37
	10	54.79	43.12	42.78
Partial (layer3–4)	1	52.87	43.04	43.71
	10	53.09	42.98	44.05
Last block only (layer4)	1	53.01	42.71	43.51
	10	52.88	42.58	43.92
Different backbone				
WRN-28-10	1	63.69	49.28	37.99
	10	61.86	49.01	37.76
VGG-16	1	56.97	48.02	44.45
	10	56.94	48.11	44.45
ViT-B/16	1	58.83	46.06	39.54
	10	57.36	46.36	39.72

Adapter depth. With coverage fixed to 1×1 convolutions, full-depth adaptation achieves the highest Square-EOT robust accuracy at every tested weight. At $\lambda_c = 2$, extending adaptation from layer 4 to layers 1–4 raises robust accuracy at $m = 30$ from 22.8% to 57.2%. Intermediate placements do not consistently interpolate between these results: adapting layers 3–4 can perform worse than adapting layer 4 alone. These findings support distributing adapters throughout the network and show that the effectiveness of cross-vulnerability training depends on where adaptation is allowed.

Convolution coverage. At $\lambda_c = 2$, the three full-depth coverage settings achieve 57.16%, 58.44%, and 60.63% robust accuracy under Square-EOT with $m = 30$. These differences are smaller than the improvement obtained by extending adaptation across all layers, and at $\lambda_c = 3$ the three settings again remain close, from 69.04% to 71.14%. Coverage matters more at other weights: at $\lambda_c = 1$, adding 3×3 convolutions throughout the network raises robust accuracy from 28.88% to 42.62%. Coverage and the cross-vulnerability weight therefore need to be chosen jointly.

Cross-vulnerability weight. Setting $\lambda_c = 0$ removes cross-vulnerability training while retaining the cross-entropy, KL, and clean-consistency terms. Across the three full-depth placements, raising λ_c from 0 to 2 lifts Square-EOT robust accuracy at $m = 30$ from 28.2–32.6% to 57.2–60.6%, at a cost of 2.6–5.0 points of clean accuracy, and $\lambda_c = 3$ reaches 69.0–71.1% with clean accuracy between 82.7% and 87.4%. The robustness obtained without cross-vulnerability training is of a different kind. At $\lambda_c = 0$ it is high at $m = 1$, where this setting gives the highest NES robust accuracy in every full-depth placement, but it does not survive averaging: Square robust accuracy falls from 52.6–54.4% at $m = 1$ to 28.2–32.6% at $m = 30$, whereas at $\lambda_c = 2$ it falls only from 61.2–65.3% to 57.2–60.6%. This is the pattern of the noise-based defenses in Table 3: randomness without structure obstructs gradient estimation at $m = 1$ but averages out. Cross-vulnerability training trades part of this non-adaptive robustness, notably under NES, for robustness that survives averaging, consistent with its role of placing disagreement along adversarial directions. The transition is not monotone at small weights: at $\lambda_c = 1$, robustness at $m = 1$ often falls below that at $\lambda_c = 0$, and in the full-depth placements the gain under averaging appears consistently only from $\lambda_c = 2$. A

Table 18: Adapter placement crossed with the cross-vulnerability weight λ_c (CIFAR-10, ResNet-50; ℓ_∞, $\epsilon = 0.05$; 1,000 inputs; single runs). 3×3 *deep* adds the 3×3 convolutions of layers 3–4, the mixed coverage of Table 11; 3×3 *all* adds them in all four layers. NES is run at m=1.

Depth	Coverage	λ_c	Clean	Square		NES
				m=1	m=30	
layer 4	1×1	0	93.50	12.83	6.84	39.36
		1	88.90	35.77	24.97	75.48
		2	91.80	28.10	22.77	71.13
		3	91.50	27.54	18.03	65.03
layers 3–4	1×1	0	93.50	20.32	10.59	51.34
		1	93.30	11.58	7.72	42.66
		2	93.30	6.86	5.04	30.44
		3	93.70	10.35	8.00	41.84
layers 1–2	1×1	0	93.20	48.07	28.11	85.30
		1	90.90	15.07	12.98	32.89
		2	87.50	27.89	25.60	44.00
		3	76.50	47.48	44.44	58.17
layers 1–4	1×1	0	93.70	52.61	28.18	85.70
		1	92.10	35.07	28.88	56.46
		2	88.70	63.13	57.16	76.89
		3	83.50	75.33	71.14	84.07
layers 1–4	1×1, 3×3 deep	0	93.30	53.38	30.01	87.03
		1	92.70	27.08	26.54	45.85
		2	88.30	61.16	58.44	70.78
		3	82.70	72.31	69.04	81.26
layers 1–4	1×1, 3×3 all	0	92.90	54.36	32.62	88.05
		1	92.20	50.76	42.62	63.88
		2	90.30	65.34	60.63	76.63
		3	87.40	72.31	69.53	81.46

weak weight perturbs the model enough to cost some clean accuracy and the incidental robustness of unstructured randomness, but not enough to shape disagreement along adversarial directions.

For the main comparison, we select full-depth adaptation with mixed coverage and $\lambda_c = 1.9$. This choice keeps the clean-accuracy cost within the approximate operating point that Section 5.1 uses for the randomized baselines, whereas larger weights buy robustness at a higher clean cost. The configuration reaches 88.71% clean accuracy, 64.57% and 61.52% robust accuracy under Square at $m = 1$ and $m = 30$, and 73.37% under NES (the $K = 50$ row of Table 20). The placements and weights used for the other architectures are listed in Table 11.

Mean–disagreement decomposition. The preceding sweep shows that sufficiently strong cross-vulnerability training is necessary for robustness to persist under EOT, but it leaves open an alternative explanation: increasing λ_c may primarily improve the attacked mean margin, rather than reducing the transferability of adversarial failures across sampled predictors. We therefore examine how the predictor distribution changes across the same λ_c sweep. For each input, we form the mean-gradient attacked point $x^\star$ used in our mechanism analysis (Equation (7)) and measure the mean correct-vs.-runner-up margin across the deployed pool, the standard deviation of that margin across predictors, and the fraction of predictors that remain correct. We also report the corresponding clean-side disagreement and a gradient cancellation ratio

$$\rho_g = \frac{\|\mathbb{E}_\psi[g_\psi]\|}{\mathbb{E}_\psi[\|g_\psi\|]},$$

where g_ψ denotes the per-predictor margin gradient used in this diagnostic. A smaller ρ_g indicates less directional agreement among predictors: their individual gradients cancel more strongly when averaged.

Table 19: Mean–disagreement decomposition across the cross-vulnerability weight λ_c. Statistics are measured in the same CIFAR-10/ResNet-50 setting as the λ_c ablation above, across an extended λ_c sweep, at the mean-gradient attacked point $x^\star$. Attack mean margin is the mean correct-vs.-runner-up margin across the deployed predictor pool at $x^\star$; negative values indicate that, on average across pool members, the correct-class logit is lower than the member-specific highest competing-class logit. Attack margin std. is the corresponding cross-predictor standard deviation, and p_{correct} is the fraction of pool predictors that remain correct at $x^\star$. Pred. Maj.-9 is obtained from each input's single-query correctness probability under nine independent predictor draws and then averaged over inputs. The gradient cancellation ratio is $\rho_g = \|\mathbb{E}_\psi[g_\psi]\|/\mathbb{E}_\psi[\|g_\psi\|]$; smaller values indicate greater directional inconsistency among predictor gradients.

λ_c	Clean mean margin	Attack mean margin	Attack margin std.	Attack p_{correct}	Pred. Maj.-9	Clean disagreement	Grad. cancel. ratio ρ_g
0	9.188	−3.616	1.082	7.6%	6.3%	1.5%	0.757
0.5	7.734	−0.153	0.149	9.4%	0.9%	3.4%	0.722
1	7.100	−0.168	0.194	17.4%	4.8%	5.2%	0.710
2	4.855	−0.080	0.324	44.8%	43.1%	8.4%	0.667
3	2.711	−0.084	0.400	44.4%	43.8%	13.2%	0.606

The two smallest weights are useful for separating effects that would otherwise be conflated. At $\lambda_c = 0$, the cross-vulnerability term is absent. Although this model exhibits the largest *absolute* attack-margin standard deviation in Table 19, its attacked mean is driven far into the incorrect region, to -3.616, and only 7.6% of sampled predictors remain correct. Thus, a large raw spread by itself is not the property targeted by DISSENT: without cross-vulnerability training, the variation is not organized so as to prevent the adversarial failure from being shared once the mean has moved far past the boundary. This is consistent with the ablation above, where the $\lambda_c = 0$ models can show substantial robustness before averaging but degrade strongly as stochastic feedback is stabilized (Table 18).

The $\lambda_c = 0.5$ setting shows the complementary failure mode of an insufficient cross-vulnerability weight. Here the attacked mean moves dramatically closer to the boundary, from -3.616 to -0.153, but the attack-margin standard deviation falls to 0.149 and single-query correctness remains only 9.4%. Hence, substantially reducing the displacement of the mean is not by itself sufficient to obtain the robustness observed at larger λ_c. We view this as a low-weight transition regime: the cross-vulnerability term is already strong enough to alter the learned decision geometry, but not yet strong enough to consistently shape the predictor distribution into the adversarially useful cross-predictor structure targeted by the objective. This also agrees with the non-monotonic low-weight behavior in the broader ablation above, where weak cross-vulnerability regularization can perturb the model without yet producing robustness that survives response averaging.

The transition from $\lambda_c = 0.5$ to $\lambda_c = 2$ more directly separates mean-side changes from cross-predictor effects. Across these settings, the attacked mean remains close to the decision boundary (-0.153, -0.168, and -0.080), whereas the attack-margin standard deviation increases from 0.149 to 0.194 and then to 0.324. Over the same range, single-query correctness rises from 9.4% to 17.4% and then to 44.8%, with the predicted majority-of-nine correctness reaching 43.1% at $\lambda_c = 2$. The gradient cancellation ratio also decreases from 0.722 to 0.667, indicating that the sampled predictors become less directionally aligned rather than merely sharing a uniformly weakened gradient. Together, these measurements argue against an explanation in which the gain is explained by changes in the attacked mean margin alone: high sampled robustness appears only once the attacked predictor distribution develops substantial cross-predictor variation as well.

Increasing λ_c further to 3 raises the attack-margin standard deviation from 0.324 to 0.400 and further lowers the gradient cancellation ratio from 0.667 to 0.606, yet p_{correct} is essentially unchanged (44.8% versus 44.4%). At the same time, clean disagreement increases from 8.4% to 13.2% and the clean mean margin decreases from 4.855 to 2.711. This saturation illustrates the intended trade-off of the objective: the goal is not to maximize predictor disagreement indiscriminately, but to induce enough disagreement along adversarial directions while preserving agreement on benign inputs. Overall, the sweep indicates that λ_c changes both the mean decision geometry and the

Table 20: Realizing the learned distribution at serving time (CIFAR-10, ResNet-50; ℓ_∞, $\epsilon = 0.05$; one forward pass per query at batch size 1; single runs). *per-query* draws a fresh predictor from the learned distribution for every query instead of routing among a pool; *Undefended* is the base, listed as a reference point. *Disk* is the on-disk footprint of the whole deployment: the frozen base (90.06 MB), the trained adapter distribution, and the pool. NES is run at m=1; the undefended base is deterministic, so its two Square columns coincide.

K	Clean	Square		NES	Latency (ms/query)	Disk (MB)
		m=1	m=30			
Undefended	91.89	0.22	0.22	19.61	2.496	90.06
5	84.54	57.72	58.65	75.03	3.859	117.52
10	87.85	59.34	56.83	70.73	3.866	122.59
20	87.62	64.16	60.27	74.32	3.864	132.74
50	88.71	64.57	61.52	73.37	3.958	163.18
100	88.13	64.25	59.36	71.96	3.946	213.91
per-query	88.29	62.36	58.05	74.60	8.355	112.44

predictor distribution, but changes to the mean alone do not explain the emergence of EOT-persistent robustness.

B.7.2 Pool Size, Sampling Strategy, and Inference Cost

Sampling implementations. We compare two implementations of the learned posterior of Section 4.2. Per-query sampling draws fresh adapter parameters for each request. Pool sampling draws K predictors once and independently selects one member for each request. Both implementations answer each query with one predictor in a single forward pass, without voting or averaging. Table 20 compares their accuracy, latency, and storage requirements.

Pool size. For $K \geq 20$, clean and robust accuracy vary modestly: robust accuracy stays within 3.5 points of per-query sampling across the evaluated attacks, above it under Square and slightly below it under NES. Small pools can be less representative of the learned distribution. We use $K = 50$ for the CIFAR-10 evaluations, within the range where a larger pool yields little additional change. These results support implementing the learned distribution with a finite pool, which the analysis of Section A.1 covers as well.

Latency and storage. At batch size 1, pool sampling takes approximately 3.9 ms per query across the tested pool sizes, since each request evaluates only one predictor. Per-query sampling additionally draws and materializes the adapter updates online, increasing latency from 3.96 ms at $K = 50$ to 8.36 ms. Each additional pool entry adds approximately 1 MB of disk storage. The complete deployment at $K = 50$ occupies 163.18 MB, compared with 112.44 MB for per-query sampling and 90.06 MB for the frozen base alone. The pool therefore reduces serving latency while keeping the additional storage compact.

Comparison with Bayesian baselines. The evaluated Adv-BNN and IG-BNN implementations aggregate multiple predictions per response and require 21.4 ms and 13.3 ms per query, respectively, and IG-BNN's stored particles occupy approximately 2.8 GB. Both sampling implementations of DISSENT have lower measured latency, consistent with their use of a single predictor for each response. The configurations of the Bayesian baselines are given in Section B.2.

B.8 Composability with Adversarial Training

Every configuration in Section 5 is built on a clean base. Adversarial training is the other established route to robustness, and a defender who already runs an adversarially trained network may prefer to build on it. This section checks that DISSENT composes with such a network, which changes neither its training procedure nor the single forward pass that answers a query; it is a composability check rather than a full re-evaluation of every baseline.

Table 21: Clean and robust accuracy (%) on the clean base and on a shared adversarially trained (AT) checkpoint (CIFAR-10, ResNet-50; ℓ_∞, $\epsilon = 0.05$; single runs). Each pair of rows gives a defense on the clean base and the same defense rebuilt on the AT checkpoint; the AT row is the undefended AT checkpoint. Parenthesized values are the change relative to the same defense on the clean base. NES is run at m=1; Square is reported at m=1 and m=30 responses per point.

Defense	Clean	NES	Square	
			m=1	m=30
Undefended	91.89	19.61	0.22	0.22
AT	82.17	93.44 (+73.83)	41.56 (+41.34)	41.56 (+41.34)
RFD	90.60	72.78	39.08	18.86
RFD +AT	80.46	97.67 (+24.89)	95.59 (+56.51)	91.79 (+72.93)
IG-BNN	94.32	18.77	2.97	2.97
IG-BNN +AT	82.46	94.09 (+75.32)	48.37 (+45.40)	48.37 (+45.40)
DISSENT	88.71	73.37	64.57	61.52
DISSENT +AT	80.61	97.69 (+24.32)	95.75 (+31.18)	92.35 (+30.83)

Construction. All AT rows of Table 21 share one adversarially trained ResNet-50 checkpoint that we train on CIFAR-10. Each defense is rebuilt on this checkpoint with the configuration it uses on the clean base: RFD recomputes its noise calibration on the AT base, IG-BNN initializes its particles from it, and DISSENT trains its adapters on it with the settings of Table 11. We include RFD, which its authors also combine with adversarial training (Nguyen et al., 2024), and IG-BNN, which is designed to start from an adversarially trained network; Adv-BNN already includes adversarial training in its main-table version (Section B.2). The attacks, query budgets, and evaluation subsets are those of Sections 5.2 and 5.3.

Results. On the clean base the spread between defenses is wide and grows as the attacker averages more responses: under Square at m=30, DISSENT retains 61.52% against 18.86% for RFD and 2.97% for IG-BNN. On the AT base the same comparison is compressed to 92.35%, 91.79%, and 48.37%. The AT base raises every defense, and it raises RFD more than DISSENT, so the two randomized defenses converge near saturation: under NES every AT-based configuration exceeds 93%, and under Square at m=1 both randomized defenses exceed 95%, whereas plain AT and IG-BNN+AT stay below 50%. DISSENT +AT is at least as accurate as RFD+AT in every column, clean accuracy included, although these margins are within run-to-run variation; the AT base leaves little room for differences between the two, and the clean base is where the defenses differ. Adversarial training costs between 8 and 12 points of clean accuracy here, which is why the main configuration is reported on a clean base throughout Section 5; a deployment that already runs an adversarially trained model can attach DISSENT to it without modification and obtains the AT-base numbers of Table 21.