SKELETON KEYS: ESTIMATING RANDOMIZED MODEL SELECTION DISTRIBUTIONS WITH LIMITED QUERIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Advances in adversarial machine learning have demonstrated that randomly selecting between defenses at inference time can further boost robustness to evasion attacks, in the white-box setting. However, it is an open question as to how an attacker could learn the model selection probability distribution in this setting. In this paper, we propose a new methodology, skeleton keys, for estimating model selection probability distributions. Skeleton keys are special inputs that leak probability distribution information to the attacker. Our contributions are as follows: First, we develop the first ever skeleton key generation algorithm and define the security properties necessary for an input to serve as a skeleton key. Second, based on skeleton key queries, we derive a likelihood estimator applicable to any defense distribution. We calculate the confidence bounds on this estimator in terms of the number of queries. Third and lastly, we empirically show the validity of our approach using state-of-the-art randomized defense ensembles. Our experiments are conducted using TRADES adversarially trained models, FAT adversarially trained models, MAMBA models and Spiking Neural Networks on CIFAR-10. Code for all our experiments is available anonymously here.

1 INTRODUCTION

Traditional adversarial machine learning has focused on the concept of the evasion attack Tramer et al. (2020); Mahmood et al. (2021a); Croce & Hein (2020) i.e., how to create adversarial examples that are misclassified with high confidence. In this paper, we consider a fundamentally different problem, how can the attacker estimate a probability distribution associated with the defense. We assume the defender can randomly select between multiple different models according to a certain probability distribution. What advantage does the defender gain from using multiple models? It has been shown that using multiple models deterministically does not yield robustness Tramer et al. (2020); Mahmood et al. (2021a). However, recent work has shown that using multiple models in a randomized fashion does yield near state-of-the-art performance, when the defender's probability distribution is computed using a game theoretic framework Mahmood et al. (2025). However, several fundamental open questions remain. How can inputs (queries) be generated such that they leak information about the distribution? Can a defense probability distribution estimator be derived from query responses? Empirically, what would advantage would an attacker gain if they used an estimated probability distribution as opposed to a uniform attack distribution strategy? To the best of our knowledge, we are the first to investigate the development of an estimator and an input generation algorithm (skeleton keys), to solve this problem. Specifically, our paper is organized around the following contributions:

1. **Skeleton Key Algorithm** - We formally define the properties an input query must have to leak information related to the defense probability distribution. We coin this type of input a *skeleton key*. We develop an algorithm for converting standard input images into skeleton keys in Section 2.
2. **Statistical Estimator Derivation** - We derive a maximum likelihood estimator for estimating any arbitrary defense probability distribution using skeleton key queries. We prove the negative log likelihood function associated with the estimator is convex, and therefore can be solved with standard gradient descent. Our theoretical derivations related to the estimator are given in Section 3.

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.

3. **Empirical Validation** - As a proof of concept, we validate our skeleton key algorithm and maximum likelihood estimator on different randomized defense distributions that select from a variety of SOTA CIFAR-10 adversarially trained models. We empirically demonstrate that the success rate of evasion attacks can be increased by as much as 10% on certain defense configuration when our estimated probability distribution is used. Experimentally results are presented in Section 4.

2 SKELETON KEY ALGORITHM

In this section we give the problem formulation, define the properties of a skeleton key and develop an algorithm for generating skeleton keys.

Problem Formulation. A defender has access to set of trained models (defenses) $D = \{d_1, d_2, \ldots, d_n\}$, where $n \geq 2$. Each time an input x is submitted to the defender, the defender randomly select one model d_j to produce the corresponding output: $y = d_j(x)$, where $j \sim \mathbf{P_D} = (p_1, p_2, \ldots, p_n)$, and $\sum_{i=1}^{n} p_i = 1, p_i \geq 0$. We denote $\mathbf{P_D}$ as the defense probability distribution. In this paper, we assume a white-box threat model where the attacker has access to the set of models D, a set of clean input examples X and (limited) query access to the defense. The goal of the attacker is to estimate $\mathbf{P_D}$. The estimation of $\mathbf{P_D}$ is important because if the attacker knows which models are used more frequently, they can adapt their evasion attacks to target on the more frequently used models in D. The estimation of $\mathbf{P_D}$ is a critical step in the attack process for randomized ensembles.

A skeleton key is an input $\mathbf{s}$ that is submitted to the defense to obtain information about $\mathbf{P_D}$. The purpose of the skeleton key is to generate useful measurements to estimate the defense probability distribution. When a skeleton key $\mathbf{s}$ is submitted to the defense for evaluation, a class label $y^{(\mathbf{s})}$ is observed. In order for an input to qualify as a skeleton key, several important properties must be statisfied. We define each of the skeleton key constraints and their security implications below:

1. **Classwise Uniqueness** - Skeleton keys are designed to be specific to certain models. We denote skeleton key $\mathbf{s}_{i,j}$ as the i^{th} key designed to obtain information about model d_j. A skeleton key $\mathbf{s}_{i,j}$ is defined as valid for model d_j if only if the following condition is satisfied:
$$\exists y_a \in Y \quad \text{s.t.} \quad (d_j(\mathbf{s}_{i,j}) = y_a^{(\mathbf{s}_{i,j})}) \wedge (\forall d \in D \backslash d_j, d(\mathbf{s}_{i,j}) \neq y_a^{(\mathbf{s}_{i,j})}) \tag{1}$$
where Y is the set of all valid outputs (class labels) for a given dataset, D is the set of all defenses, and $\mathbf{s}_{i,j}$ is the i^{th} skeleton key designed to target the j^{th} model. Equation 1 specifies two important criteria for an input to be considered a skeleton key. First, the target model must produce a specific class label y_a when model d_j is used for evaluation. Second, all other models in the set D excluding d_j must produce class label $y \neq y_a$.
2. **ϵ-Bounded** - In adversarial machine learning evasion attacks, an adversarial example $x_{adv} = x + \delta$ should be visually indistinguishable from the original clean image x. This is usually enforced Goodfellow et al. (2015); Madry et al. (2018) by a norm constraint on the added noise δ such that $||\delta||_p \leq \epsilon$. For skeleton keys, to avoid visual human detection, we enforce the same constraint with $p = \infty$. I.e., all skeleton keys are generated starting from clean image x and follow the constraint $||x - \mathbf{s}||_\infty \leq \epsilon$. Our choice of ϵ is based on the values for datasets previously established in the literature Madry et al. (2018); Croce & Hein (2020); Mahmood et al. (2025).
3. **Single Query Use (SQU)** - A skeleton key $\mathbf{s}$ generated from clean input x, may be used to query the defense only once. We enforce this constraint because we consider the defender to be stateful Chen et al. (2020). Specifically, we assume the defender has a memory of past states and can detect the adversary under the following conditions. Let $T = \{\mathbf{s}_1, .., \mathbf{s}_q\}$ denote a sequences of q queries to the defense where an input $\mathbf{s}_t$
and the criteria for detection be defined as:
$$\forall \mathbf{s}_t \in T, \exists \mathbf{s}_{t'} \in T \backslash \{\mathbf{s}_t\} \quad \text{s.t.} \quad \phi(\mathbf{s}_t, \mathbf{s}_{t'}) \geq \zeta. \tag{2}$$
In Equation 2, $\phi(\cdot)$ is a similarity measure and ζ is the threshold for similar queries detection. For this memory-based detection system, submitting the same skeleton key twice within q queries results in the attack failing.

2.1 GENERATION ALGORITHM

Algorithm 1 presents our procedure for generating skeleton keys that satisfy the classwise uniqueness condition in Equation 1. Given a clean input pair (x, y), we designate one model M_t as the target model and treat the remaining set of models V as non-target models. Under our white-box threat model, the attacker has access to all the models and can compute gradients with respect to the input. Starting from x, we iteratively perturb the input using a main update equation for generating the skeleton key is: $\mathbf{s}^{(i+1)} = \mathbf{s}^{(i)} - \alpha \mathbf{f}_{\text{target}} + \beta \mathbf{f}_{\text{blend}}$. Here, $\mathbf{f}_{\text{blend}}$ denotes the aggregate perturbation direction obtained by summing the signed gradients of the non-target models, while $\mathbf{f}\text{target}$ denotes the signed gradient direction of the target model. The target-model gradient is gated by the predicted probability of the ground-truth class y, such that it is applied only when $M_t(s^{(i)})_y < 0.5$.

The update step works by combining two objectives in the form of $\mathbf{f}_{\text{blend}}$ and $\mathbf{f}_{\text{target}}$. One objective is to meet the second condition originally outlined in Equation 1: $\forall v \in \mathcal{V} \backslash m_t, v(\mathbf{s}) \neq y$, i.e., all non-target models in the set $\mathcal{V}$ should produce the wrong class label. To accomplish this, in Algorithm 1, f_{blend} aggregates gradients from the non-target models to encourage their predictions to differ from that of the target model. Our second objective is to satisfy the first condition outlined in Equation 1: $M_t(\mathbf{s}) = y$, where $M_t(\mathbf{s})$ is the softmax function of target model t given the skeleton key $\mathbf{s}$, i.e., the target model produces the correct class label. In Algorithm 1 this is realized using f_{target}. The weighting factors α and β balance these two objectives. After each update, the perturbed input is projected onto the l_∞-norm ball of radius $\epsilon_{\max}$ around x and clipped to the valid input range $[0, 1]$. The resulting $s^{(N)}$ is a candidate skeleton key and is considered valid when the target model produces a class label that is not produced by any of the non-target models, as defined in Equation 1. In practice α and β are hyperparameters that can be optimized. For simplicity, in our implementation we used $\alpha, \beta = \frac{1}{2N} \cdot \epsilon_{max}$.

Algorithm 1 Skeleton Key Generation Algorithm

Input: clean sample and label (x, y), target model m_t, target model weighting factor α, the set of non-target models $\mathcal{V}$, non-target models weighting factor β, number of iterations N and maximum perturbation $\epsilon_{\max}$.
Output: final skeleton key $\mathbf{s}^{(N)}$
$\mathbf{s}^{(0)} \leftarrow x$
for $i = 0, \ldots, N-1$ **do**
  $\mathbf{f}_{\text{blend}} \leftarrow \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \text{sign}\left(\frac{\partial L_v}{\partial \mathbf{s}^{(i)}}\right)$
  $\mathbf{f}_{\text{target}} \leftarrow \text{sign}\left(\frac{\partial L_t}{\partial \mathbf{s}^{(i)}}\right) \cdot \mathbb{1}\left[M_t(\mathbf{s}^{(i)})_y < 0.5\right]$
  $\mathbf{s}^{(i+1)} \leftarrow \mathbf{s}^{(i)} - \alpha \mathbf{f}_{\text{target}} + \beta \mathbf{f}_{\text{blend}}$
  $\mathbf{s}^{(i+1)} \leftarrow P\left(\mathbf{s}^{(i+1)}, x, \epsilon_{\max}\right)$
  $\mathbf{s}^{(i+1)} \leftarrow \text{clip}\left(\mathbf{s}^{(i+1)}, 0, 1\right)$
end for
return $\mathbf{s}^{(N)}$

3 STATISTICAL ESTIMATOR DERIVATION

For each model d_j, we assume a set of exclusive skeleton keys $[s_{i,j}]_{i=1}^{m_j}$, where m_j is the number of skeleton keys corresponding to model d_j. Let $X_{i,j} = g(s_{i,j}, D, \mathbf{P_D}) \in \{0, 1\}$ denote the binary observation associated with the i^{th} skeleton key for model d_j, where

$$X_{i,j} = \begin{cases} 1, & \text{if the input (skeleton key) is evaluated using } d_j, \\ 0, & \text{otherwise.} \end{cases}$$

Under this construction, $X_{i,j} \sim \text{Bernoulli}(p_j)$, where p_j denotes the probability that the defender selects model d_j. We assume that the binary observations are independent random variables. Therefore, the aggregate number of observations where d_j is selected is: $Z_j = \sum_{i=1}^{m_j} X_{i,j} \sim \text{Binomial}(m_j, p_j)$. For simplicity, we consider the balanced-query setting $m_1 = m_2 = \cdots = m_n =$

m, which corresponds to the case where, in the absence of prior information regarding the defense probability distribution , the attacker allocates an equal number of queries to each defense model. Therefore, $Z_j \sim \text{Binomial}(m, p_j), \forall j \in \{1, 2, \dots n\}$. A simple estimator for the probability of selecting defense d_j is the empirical proportion:

$$\tilde{p}_j = \frac{1}{m} \sum_{i=1}^{m} X_{i,j} \quad \forall j \in \{1, 2, \dots n\}. \tag{3}$$

The estimators presented in Equation 3 are unbiased, however they are derived from marginal observations and therefore do not, in general, enforce the constraint $\sum_{j=1}^{n} \tilde{p}_j = 1$. This arises from the fact that the estimators are constructed independently for each model and thus do not fully exploit the dependence structure induced by the shared underlying probability space.

To obtain a statistically rigorous estimator that leverages the full probability space, we derive the maximum likelihood estimator (MLE). This formulation additionally enables asymptotic analysis of the estimator variance and distribution, which is also useful to solve for the number of independent queries m required to estimate the defense probability distribution $\mathbf{P_D}$ with a given confidence level.

The likelihood function for the defense selection probabilities $\mathbf{P_D}$ is

$$L(\mathbf{P_D}) = \prod_{j=1}^{n} \binom{m}{Z_j} p_j^{Z_j} (1 - p_j)^{m - Z_j} \propto \prod_{j=1}^{n} p_j^{Z_j} (1 - p_j)^{m - Z_j}, \tag{4}$$

where the proportionality follows from omitting the binomial coefficient terms $\prod_{j=1}^{n} \binom{m}{Z_j}$, which are constant with respect to $\mathbf{P_D}$ and therefore do not affect parameter estimation. Taking the logarithm yields the log-likelihood, up to an additive constant as:

$$\ell(\mathbf{P_D}) = \sum_{j=1}^{n} \left[Z_j \log p_j + (m - Z_j) \log(1 - p_j) \right]. \tag{5}$$

To explicitly enforce the constraint, we eliminate the last coordinate by defining the reduced parameterization

$$p_n = 1 - \sum_{j=1}^{n-1} p_j. \tag{6}$$

Substituting Equation 6 into Equation 5, we obtain the reduced log-likelihood

$$\begin{aligned} \ell(p_1, \dots, p_{n-1}) = &\left(\sum_{j=1}^{n-1} \Big[Z_j \log p_j + (m - Z_j) \log(1 - p_j) \Big] \right) \\ &+ Z_n \log \left(1 - \sum_{j=1}^{n-1} p_j \right) + (m - Z_n) \log \left(\sum_{j=1}^{n-1} p_j \right). \end{aligned} \tag{7}$$

The maximum likelihood estimator is defined as

$$\hat{\mathbf{p}} = \underset{\mathbf{p} \in \mathcal{P}}{\operatorname{argmax}}\ \ell(p_1, \dots, p_{n-1}), \quad \mathcal{P} = \left\{ \mathbf{p} \in [0, 1]^{n-1} : \sum_{j=1}^{n-1} p_j \leq 1 \right\}. \tag{8}$$

Equivalently, this can be written as a minimization problem over the negative log-likelihood:

$$\hat{\mathbf{p}} = \underset{\mathbf{p} \in \mathcal{P}}{\operatorname{argmin}}\ -\ell(p_1, \dots, p_{n-1}). \tag{9}$$

The first-order optimality conditions are given by

$$\frac{\partial \ell(\mathbf{p})}{\partial p_j} = \frac{Z_j}{p_j} - \frac{m - Z_j}{1 - p_j} - \frac{Z_n}{1 - \sum_{k=1}^{n-1} p_k} + \frac{m - Z_n}{\sum_{k=1}^{n-1} p_k} = 0, \quad \forall j = 1, \dots, n-1. \tag{10}$$

The resulting nonlinear system does not admit a closed-form solution. However, if the function is convex, an global optimal solution can be obtained through standard optimization procedures like gradient descentBoyd & Vandenberghe (2004).

Negative Log Likelihood Proof of Convexity: We prove that $-\ell(\mathbf{p})$ (the negative of Equation 7) is convex. A twice continuously differentiable function is convex over a convex domain if and only if its Hessian matrix is positive semidefinite everywhere on that domain Boyd & Vandenberghe (2004). Since the negative log likelihood is twice differentiable on the interior of $\mathcal{P}$, we study its Hessian structure.

The Hessian of $-\ell(\mathbf{p})$ is an $(n-1)\times(n-1)$ matrix that can be decomposed into two components:

$$\nabla^2(-\ell(\mathbf{p})) = -\nabla^2(\ell(\mathbf{p})) = \mathcal{D}(\mathbf{p}) + c(\mathbf{p})\mathbf{1}\mathbf{1}^T, \tag{11}$$

where $\mathcal{D}(\mathbf{p})$ is a diagonal matrix with entries

$$\mathcal{D}_{jj} = \frac{Z_j}{p_j^2} + \frac{m - Z_j}{(1-p_j)^2}, \quad \forall j = 1, \ldots, n-1. \tag{12}$$

and

$$c(\mathbf{p}) = \frac{Z_n}{\left(1 - \sum_{k=1}^{n-1} p_k\right)^2} + \frac{m - Z_n}{\left(\sum_{k=1}^{n-1} p_k\right)^2} \geq 0. \tag{13}$$

In Equation 11, $\mathbf{1}\mathbf{1}^T$ is a rank-one positive semidefinite matrix and $\mathcal{D}(\mathbf{p})$ is positive semidefinite because all diagonal entries are nonnegative over the feasible domain. Therefore, the Hessian is a sum of two positive semidefinite matrices:

$$\mathcal{D}(\mathbf{p}) \succeq 0, \quad c(\mathbf{p})\mathbf{1}\mathbf{1}^T \succeq 0, \tag{14}$$

which implies

$$\nabla^2(-\ell(\mathbf{p})) \succeq 0. \tag{15}$$

Hence, the negative log-likelihood is convex over $\mathcal{P}$, and any local minimizer is also a global minimizer Boyd & Vandenberghe (2004).

3.1 Asymptotic Variance of the Skeleton Key Estimator

In Equation 7, we gave the log-likelihood function to be used in the maximum likelihood estimator (MLE). We now characterize the statistical efficiency of the (MLE) for the defense distribution $\mathbf{P_D} = (p_1, \ldots, p_n)$. In particular, we analyze the asymptotic variance, which directly governs the number of skeleton key queries required to estimate the underlying distribution with a prescribed accuracy. Under standard regularity conditions, the MLE is asymptotically normal Casella & Berger (2021), satisfying:

$$\hat{\mathbf{p}} \overset{approx}{\sim} \mathcal{N}\big(\mathbf{p}, \mathcal{I}^{-1}(\mathbf{p})\big), \tag{16}$$

where $\mathcal{I}(\mathbf{p})$ denotes the Fisher Information matrix induced by the skeleton key queries. From the structure of the likelihood, each model component d_j contributes independently and so does the constraint. Consequently, the Fisher Information matrix takes the form:

$$\mathcal{I}(\mathbf{p}) = \mathbf{E}(-\nabla^2(\ell(\mathbf{p}))) = m\left[\mathrm{diag}\left(\frac{1}{p_1(1-p_1)}, \ldots, \frac{1}{p_{n-1}(1-p_{n-1})}\right) + \frac{1}{p_n(1-p_n)}\mathbf{1}\mathbf{1}^T\right], \tag{17}$$

where $p_n = 1 - \sum_{j=1}^{n-1} p_j$ and $\mathbf{1}$ is the $(n-1)$-dimensional all-ones vector. Inverting Equation 17 using the Sherman–Morrison–Woodbury identity yieldsWoodbury (1950).

$$\mathcal{I}^{-1}(\mathbf{p}) = \frac{1}{m}\left(\mathrm{diag}(v_1, \ldots, v_{n-1}) - c_{\mathbf{p}}\mathbf{v}\mathbf{v}^T\right), \tag{18}$$

where:

$$v_j = p_j(1-p_j), \qquad \mathbf{v} = (v_1, \ldots, v_{n-1}), \qquad c_{\mathbf{p}} = \frac{1}{\sum_{j=1}^{n} p_j(1-p_j)}.$$

Therefore, the asymptotic variance of each component estimator is given by

$$\mathrm{Var}(\hat{p}_j) = \frac{1}{m}\, p_j(1-p_j)\left(1 - \frac{p_j(1-p_j)}{\sum_{k=1}^{n} p_k(1-p_k)}\right). \tag{19}$$

Equation 19 highlights that estimation uncertainty of the defense probability distribution component p_j decreases as number of skeleton key queries m increases.

3.2 Confidence Interval and Minimum Number of Skeleton Key Queries

Using the asymptotic normality in Equation 16, a $100(1-\alpha)\%$ confidence interval for p_j is approximated by

$$\mathrm{CI}_j = \hat{p}_j \pm z_{\alpha/2}\sqrt{\mathrm{Var}(\hat{p}_j)}, \tag{20}$$

where $z_{\alpha/2}$ is the standard Gaussian quantile such that $P(Z > z_{\alpha/2}) = \alpha/2$. The corresponding confidence interval width is therefore

$$\omega_j = 2z_{\alpha/2}\sqrt{\frac{1}{m}\,p_j(1-p_j)\left(1 - \frac{p_j(1-p_j)}{\sum_{k=1}^{n} p_k(1-p_k)}\right)}. \tag{21}$$

Rearranging Equation 21 provides a lower bound on the number of skeleton-key queries required to achieve a desired precision ω:

$$m \;\geq\; \frac{4z_{\alpha/2}^2}{\omega^2}\;\max_{\mathbf{p}}\left[p_j(1-p_j)\left(1 - \frac{p_j(1-p_j)}{\sum_{k=1}^{n} p_k(1-p_k)}\right)\right]. \tag{22}$$

4 Experimental Results

Models: We adversarially train six CIFAR-10 models spanning four architectural families: WRN-28-10 (Wide Residual Network) Zagoruyko & Komodakis (2016), ViT-L/16 (Vision Transformer) Dosovitskiy (2020), CaiT-XXS-12 Touvron et al. (2021), MambaVision-T Hatamizadeh & Kautz (2025), and two SEW-ResNet-18 spiking neural networks Xu et al. (2025) (DM-R18 and FAT-R18). Five of the six models follow the diffusion-augmented adversarial training approach of Wang et al. (2023). Each training batch combines real CIFAR-10 images with class-conditional EDM-generated images, with generated images comprising 70% of the batch. We use 1M generated images for WRN-28-10, ViT-L/16, CaiT-XXS-12, and MambaVision-T, and 10M for DM-R18. All images are augmented using random cropping and horizontal flipping. These five models are trained with the TRADES objective (Zhang et al., 2019) ($\beta = 5$), using label smoothing of 0.1. Adversarial examples during training are generated using a 10-step L_∞ PGD attack with $\epsilon = 8/255$ and a step size of $2/255$. Additional training details can be found in the appendix.

Choice of adversarially trained models. We deliberately include models from four different architectural families: convolutional networks, vision transformers, state-space models, and spiking neural networks. Prior work has shown that adversarial examples can transfer poorly across models with substantially different architectures (Mahmood et al., 2021b; Xu et al., 2025). Architectural diversity is therefore important for evaluating whether an attack remains effective across heterogeneous defenses rather than exploiting vulnerabilities specific to a single model family.

GaME Payoff Matrix Construction and Defender Strategy for optimal $\mathbf{P_D}$. We use the Game Theoretic Mixed Experts (GaME) framework (Mahmood et al., 2025) to determine the probability distribution $\mathbf{P_D}$ for our adversarially trained models in a subset of our experiments. GaME formulates the interaction between a set of attacks and defenses as a finite zero-sum game, where the payoff matrix represents the robust accuracy of each defense against each attack strategy. Solving the resulting game yields a defender mixed strategy λ^D, which specifies the probability of selecting each defense. To construct the attack set, we randomly select 1,000 class-balanced samples (100 per class) from the CIFAR-10 training set. Further details relating to our GaME experiments are given in the appendix.

4.1 Skeleton Key Generation Yield Rate

We randomly select 2,000 class-balanced samples from CIFAR-10 (200 samples per class) and use the same samples for every target model. For each model in turn, we designate that model as the target and treat all remaining models as non-targets. We run the skeleton key attack for 100 iterations with a maximum l_∞ perturbation of $\epsilon_{\max} = 0.031$. Table 2 summarizes the resulting number of valid and non-valid keys and the corresponding yield rates.

As shown in Table 2, the attack produces working skeleton keys for every target model in both configurations, although the yield varies considerably across models. WRN-28-10 has the highest

Table 1: Clean and robust accuracy of the six adversarially trained models on the CIFAR-10.

MODEL	CLEAN ACCURACY	ROBUST ACCURACY
CaiT-XXS	86.16%	56.64%
DM-R18	66.76%	61.10%
FAT-R18	73.15%	48.50%
MambaVision-T	91.15%	63.48%
ViT-L/16	93.03%	65.14%
WRN-28-10	91.02%	65.60%

Table 2: Skeleton key Generation algorithm evaluation for each of the six models. All skeleton keys use 100 iterations and a maximum l_∞ perturbation of $\epsilon_{\max} = 0.031$.

MODEL	VALID	NON-VALID	YIELD RATE
CaiT-XXS	15	1985	0.75%
DM-R18	37	1963	1.85%
FAT-R18	28	1972	1.40%
MambaVision-T	16	1984	0.80%
ViT-L/16	43	1957	2.15%
WRN-28-10	103	1897	5.15%

yield rate at 5.15%, followed by ViT-L/16 at 2.15%, while the remaining models have yield rates below 2%. The most important takeaway from this set of experiments is the yield rate is non-zero for all models. This means our proposed skeleton key generation algorithm can create input queries to measure the defense probability distribution for all models.

4.2 Skeleton Key Estimator Experiments

Estimator Metric. The Kullback–Leibler (KL) divergence is often used as a relative measure of how close one probability distribution is to a target distribution, it is neither symmetric nor bounded, making it less suitable for directly quantifying the absolute difference between two distributions. Therefore, we use the Jensen–Shannon divergence (JSD) Lin (1991), a symmetric and bounded measure of the difference between two probability distributions to assess how closely an MLE metric when evaluated with $m = 400$, approximates the ground-truth probability of defense. For the MLE evaluated at $m = 400$, denoted by $\hat{\mathbf{P}}_{m=400}$, and the ground-truth distribution, denoted by $\mathbf{P_D}$, the Jensen–Shannon divergence is defined as:

$$\text{JSD}\left(\hat{\mathbf{P}}_{m=400} \,\|\, \mathbf{P_D}\right) = \frac{1}{2} KL\left(\hat{\mathbf{P}}_{m=400} \,\|\, M\right) + \frac{1}{2} KL\left(\mathbf{P_D} \,\|\, M\right), \tag{23}$$

where $M = \frac{1}{2}\left(\hat{\mathbf{P}}_{m=400} + \mathbf{P_D}\right)$. When the logarithm is taken to base 2, the JSD is bounded as follows: $0 \leq \text{JSD}\left(\hat{\mathbf{P}}_{m=400} \,\|\, \mathbf{P_D}\right) \leq 1$. A value of 0 indicates that the MLE and ground-truth distributions are identical, while larger values indicate greater discrepancy. The observed JSD values are small and therefore it suggests a reasonable similarity between the maximum likelihood defense probabilities and the ground truth defense probabilities.

Estimator Analysis. Table 3 shows that the skeleton key estimator closely approximates the ground-truth defense distribution using $m = 400$ observations. Across all tested probability settings, the estimated probabilities $\hat{\mathbf{P}}_D$ remain close to the corresponding ground-truth probabilities $\mathbf{P}_D$, with JSD values ranging from 1.51×10^{-4} to 3.20×10^{-3}. This outcome is observed across all evaluated model combinations and defense probability assignments. These results indicate that the estimator can approximate the defense probability distribution when the number of skeleton keys per model is $m = 400$.

Table 3: Estimate of $\mathbf{P_D}$ under different model configurations and probability distributions. Model abbreviations are, W = WRN-28-10, C = CaiT-XXS, V = ViT-L/16, M = MambaVision-T, D = DM-R18, and F = FAT-R18. JSD denotes Jensen–Shannon divergence computed using base-2 logarithms.

MODELS	$\mathbf{P}_\mathrm{D}$	$\hat{\mathbf{P}}_\mathrm{D}$	JSD
W,C,V,M,D,F	$[1, 0, 0, 0, 0, 0]$	$[.9995, .0001, .0001, .0001, .0001, .0001]$	2.50×10^{-4}
W,C,V,M,D,F	$[.5, 0, .25, .25, 0, 0]$	$[.4984, .0010, .2592, .2421, .0010, .0010]$	2.56×10^{-4}
W,C,V,M,D,F	$[.26, .07, .26, .17, .17, .07]$	$[.2350, .0605, .2455, .1926, .2086, .0578]$	3.20×10^{-3}
W,C,V,M,D,F	$[.167, .167, .167, .167, .166, .166]$	$[.1735, .1581, .1735, .1452, .1658, .1838]$	1.02×10^{-3}
C,V,M,D,F	$[0, .4813, .5187, 0, 0]$	$[.0001, .4824, .5173, .0001, .0001]$	1.51×10^{-4}

Table 4: Robust accuracy of randomized model ensemble under different attack settings. R_U denotes robust accuracy when the attacker assumes a uniform probability distribution. R_S denotes the robust accuracy when the attacker uses the skeleton key estimated distribution $\hat{\mathbf{P}}_\mathrm{D}$. Model abbreviations are: W = WRN-28-10, C = CaiT-XXS, V = ViT-L/16, M = MambaVision-T, D = DM-R18, and F = FAT-R18.

MODELS	$\mathbf{P}_\mathrm{D}$	$\hat{\mathbf{P}}_\mathrm{D}$	$\mathbf{R}_U$	$\mathbf{R}_S$	$\mathbf{\Delta}$
W,C,V,M,D,F	$[1, 0, 0, 0, 0, 0]$	$[.9995, .0001, .0001, .0001, .0001, .0001]$	76.50%	65.80%	10.70%
C,V,M,D,F	$[0, .4813, .5187, 0, 0]$	$[.0001, .4824, .5173, .0001, .0001]$	75.90%	65.90%	10.00%

Attack Metrics. We generate two different defense configurations based on the GaME framework. The first configuration is a six-models defense in which only the most robust model is used (WRN-28-10). The second configuration is a five model defense in which we select from the two most robust models (MambaVision-T and ViT-L/16). In terms of robustness for each model, we generate 1000 adversarial samples using the APGD Croce & Hein (2020) attack method for 100 steps and $\epsilon_{max} = 8/255$. APGD is an adaptive variant of Projected Gradient Descent that iteratively optimizes adversarial perturbations within a constrained l_∞-norm ball while automatically adjusting the step size to improve attack effectiveness.

We report the robustness to the 1000 APGD adversarial examples under two different attack configurations. As a baseline we use a naive attack that randomly select the model to generate the adversarial example with probability $\hat{\mathbf{P}}$, where $\hat{\mathbf{P}}$ belong to the Uniform distribution. We denote the robustness measured under this assumption as R_U. For our main result, we consider an attack that selects among the models used to generate the adversarial examples with the skeleton estimated probability distribution $\hat{\mathbf{P}_\mathbf{D}}$ ($m = 400$). We denote the robustness under the skeleton key estimator assumption as R_S. Lastly, for a simple metric, we compute the gain in using $\hat{\mathbf{P}_\mathbf{D}}$ over a Uniform distribution as $\Delta = R_U - R_S$.

Attack Analysis. Table 4 shows that using the skeleton key estimated defense distribution improves attack effectiveness over the uniform baseline. For the two evaluated defense configurations, robust accuracy drops from 76.50% to 65.80% and from 75.90% to 65.90%, corresponding to reductions of 10.70% and 10.00% percentage points, respectively. These results show that even an estimated defense distribution can provide useful information for improving evasion attacks, yielding about a 10 percentage-point gain in attack success in these settings.

5 Conclusion

Randomized model ensembles have demonstrated effective robustness to adversarial examples. However, to the best of our knowledge, our work is the first to investigate the open question of how the defense probability distribution can be learned. To start, we formulate how an input query can leak information. We then define the necessary properties an input should have to be leveraged by an attacker, which we term a skeleton key. From the algorithmic side, we develop a method for generating skeleton keys from benign inputs. From the theoretical side, we prove that the skeleton key negative log-likelihood is convex and can be minimized with simple gradient descent. We also

derive the number of skeleton keys necessary to estimate the defense probability distribution within a given confidence interval. Finally, we experimentally demonstrate the validity of our estimator in an attack setting using the CIFAR-10 dataset. We empirically show that our estimator can increase the attack success rate by up to 10% in some cases against state-of-the-art adversarially trained defense ensembles generated by the GaME framework.

REFERENCES

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004. URL https://web.stanford.edu/~boyd/cvxbook/.

George Casella and Roger L Berger. *Statistical inference*. Cengage Learning, 2021.

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, 2020.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*. PMLR, 2020.

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

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

Ali Hatamizadeh and Jan Kautz. Mambavision: A hybrid mamba-transformer vision backbone. In *2025 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 25261–25270. IEEE, 2025.

Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.

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.

Kaleel Mahmood, Deniz Gurevin, Marten van Dijk, and Phuoung Ha Nguyen. Beware the black-box: On the robustness of recent defenses to adversarial examples. *Entropy*, 23(10):1359, 2021a.

Kaleel Mahmood, Rigel Mahmood, and Marten Van Dijk. On the robustness of vision transformers to adversarial examples. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7818–7827. IEEE, 2021b.

Kaleel Mahmood, Ethan Rathbun, Ronak Sahu, Marten Van Dijk, Sohaib Ahmad, and Caiwen Ding. Game theoretic mixed experts for combinational adversarial machine learning. *IEEE Access*, 13: 158887–158905, 2025. doi: 10.1109/ACCESS.2025.3608117.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 32–42, 2021. doi: 10.1109/ICCV48922.2021.00010.

Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *Advances in neural information processing systems*, 33:1633–1645, 2020.

Zekai Wang, Tianyu Pang, Chao Du, Min Lin, Weiwei Liu, and Shuicheng Yan. Better diffusion models further improve adversarial training. In *International conference on machine learning*, pp. 36246–36263. PMLR, 2023.

Max A Woodbury. *Inverting modified matrices*. Department of Statistics, Princeton University, 1950.

Nuo Xu, Kaleel Mahmood, Haowen Fang, Ethan Rathbun, Caiwen Ding, and Wujie Wen. Attacking the spike: On the security of spiking neural networks to adversarial examples. *Neurocomputing*, 656:131506, 2025. doi: https://doi.org/10.1016/j.neucom.2025.131506.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith (eds.), *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 87.1–87.12. BMVA Press, September 2016. ISBN 1-901725-59-6.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7472–7482. PMLR, 09–15 Jun 2019.

Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11278–11287. PMLR, 13–18 Jul 2020.

A Adversarial Model Training

We maintain an exponential moving average (EMA) of the model weights with a decay of 0.995 and use the best-performing EMA checkpoint in all subsequent experiments.

WRN-28-10, CaiT-XXS-12, and DM-R18 are trained from scratch, whereas ViT-L/16 and MambaVision-T are initialized from clean CIFAR-10 models and therefore use lower learning rates and fewer training epochs. For ViT-L/16 and MambaVision-T, a differentiable bilinear layer within the model upsamples the inputs to 224×224. Consequently, adversarial perturbations are still generated and bounded in the native 32×32 CIFAR-10 input space. DM-R18 uses five timesteps with parametric LIF neurons.

FAT-R18 uses the same SEW-ResNet-18 architecture as DM-R18 but follows Friendly Adversarial Training (Zhang et al., 2020) rather than diffusion-augmented training. Specifically, it uses early-stopped PGD-10 ($\tau = 5$) on CIFAR-10 without generated data, following Xu et al. (2025). Table 5 summarizes the training hyperparameters for all six models.

For the convolutional family, we use WRN-28-10 trained with the diffusion-augmented adversarial training approach of Wang et al. (2023). We extend the same training approach to ViT-L/16, CaiT-XXS-12, and MambaVision-T so that the transformer and state-space models are also adversarially trained rather than included as undefended baselines. For the spiking family, we use the SEW-ResNet-18 architecture under two different adversarial training approaches, DM-R18 and FAT-R18, following Xu et al. (2025). This results in a heterogeneous pool in which robustness is evaluated across both architectural families and adversarial training strategies.

B Experimental Results

Table 1 reports the clean and robust accuracy of the six adversarially trained models on the CIFAR-10 validation set. Robust accuracy is evaluated using a 40-step L_∞ PGD attack with a maximum perturbation of $\epsilon = 8/255$. Clean accuracy ranges from 66.76% to 93.03%, while robust accuracy ranges from 48.50% to 65.60%. WRN-28-10 achieves the highest robust accuracy (65.60%), while ViT-L/16 achieves the highest clean accuracy (93.03%).

The clean accuracy is outlined in 6 The same samples are used for all attacks. We apply APGD-CE independently to each defense model and ProtoSAGA to every pair of defense models. The attack parameters are summarized in Table 9, while the corresponding APGD-CE and ProtoSAGA results are reported in Tables 7 and 8, respectively.

For the six-model configuration, APGD-CE produces 1,000 adversarial examples for each of the six models, resulting in 6,000 single-model adversarial examples. ProtoSAGA is applied to all $\binom{6}{2} = 15$ model pairs and produces 1,000 adversarial examples per pair, resulting in an additional 15,000 pairwise adversarial examples. In total, the GaME payoff matrix is constructed from 21 attack strategies, corresponding to 21,000 adversarial examples.

Table 5: Adversarial training configurations for the six defense models. C10 denotes CIFAR-10, EDM denotes class-conditional diffusion-generated images, and R18 denotes ResNet-18. All models use a 10-step L_∞ PGD inner attack with $\epsilon = 8/255$ and step size $2/255$. For EDM-augmented training, 70% of each batch is drawn from generated images; FAT-R18 uses no generated data. [a]We use the checkpoint released by Xu et al. (2025) as is; – denotes values not reported there. [b]Epoch of the released checkpoint.

MODEL	DATA	INIT.	BATCH	EPOCHS	OPT.	LR	OBJECTIVE
WRN-28-10	C10+1M	Random	512	400	SGD-N	0.2	TRADES ($\beta = 5$)
ViT-L/16	C10+1M	Clean	64	200	AdamW	5×10^{-5}	TRADES ($\beta = 5$)
CaiT-XXS	C10+1M	Random	512	400	AdamW	10^{-3}	TRADES ($\beta = 5$)
MambaVision-T	C10+1M	Clean	64	200	AdamW	5×10^{-5}	TRADES ($\beta = 5$)
DM-R18	C10+10M	Random	1024	400	SGD-N	0.2	TRADES ($\beta = 5$)
FAT-R18	C10	Random	–	219[b]	SGD	0.1	FAT ($\tau = 5$)

Table 6: Clean accuracy of the six defense models on the full CIFAR-10 validation set and on 1,000 randomly selected class-wise balanced samples from the CIFAR-10 training set.

MODEL	CIFAR-10 VALIDATION	CIFAR-10 TRAINING
CaiT-XXS	86.22%	94.30%
DM-R18	66.71%	68.70%
FAT-R18	72.98%	76.50%
MambaVision-T	91.16%	95.90%
ViT-L/16	92.79%	97.90%
WRN-28-10	91.02%	98.60%

Table 7: APGD-CE evaluation on 1,000 randomly selected samples from the CIFAR-10 training set (100 samples per class), without filtering for initial correct classification. The same randomly selected samples are used for all six defense models. APGD-CE uses 100 attack steps, batch size 32, and a maximum L_∞ perturbation of $\epsilon = 0.031$.

DEFENSE	CLEAN ACCURACY	ROBUST ACCURACY
WRN-28-10-SiLU	98.60%	85.60%
CaiT-XXS-12	94.30%	70.80%
ViT-L/16	97.90%	79.40%
MambaVision-T	95.90%	73.40%
DM-R18	68.70%	25.60%
FAT-R18	76.50%	14.50%

Solving GaME for the six-model defense set assigns the entire defender probability to WRN-28-10, i.e., $\lambda^D_{\text{WRN}} = 1$, with zero probability assigned to the other five models. We therefore repeat the experiment after excluding WRN-28-10 to examine the mixed strategy over the remaining defenses. In this five-model setting, APGD-CE contributes five single-model attack strategies (5,000 adversarial examples), while ProtoSAGA contributes all $\binom{5}{2} = 10$ pairwise attack strategies (10,000 adversarial examples), for a total of 15 attack strategies and 15,000 adversarial examples.

The resulting defender mixed strategies are reported in Table 10. With WRN-28-10 excluded, GaME assigns probabilities of 0.4813 to ViT-L/16 and 0.5187 to MambaVision-T, with zero probability assigned to CaiT-XXS, DM-R18, and FAT-R18. The corresponding GaME robust accuracies are 93.80% for the six-model configuration and 80.28% for the five-model configuration.

B.1 Skeleton key generation

Skeleton keys and yield rate. Let $\mathcal{M} = \{f_1, \ldots, f_M\}$ denote the model pool and $f_t \in \mathcal{M}$ the target model. Given a clean input x with ground-truth label y, the Skeleton key attack generates a perturbed input x' subject to $\|x' - x\|_\infty \leq \epsilon$. We define x' as a *working skeleton key* for f_t when the target model retains the correct prediction while every non-target model is simultaneously misclassified:

$$f_t(x') = y \quad \text{and} \quad f_j(x') \neq y \;\; \forall f_j \in \mathcal{M} \setminus \{f_t\}. \tag{24}$$

An adversarial sample that does not satisfy both conditions is considered a *non-working skeleton key*. For a target model f_t, we measure the fraction of attacked samples that satisfy this condition using the *yield rate*:

$$\text{Yield}(f_t) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}[f_t(x'_i) = y_i \;\wedge\; f_j(x'_i) \neq y_i \;\; \forall f_j \in \mathcal{M} \setminus \{f_t\}]\,. \tag{25}$$

As shown in Table 11, the attack produces working skeleton keys for every target model in both configurations, although the yield varies considerably across models. In the six-model setting, WRN-28-10 has the highest yield rate at 5.15%, followed by ViT-L/16 at 2.15%, while the remaining models have yield rates below 2%.

Table 8: Pairwise ProtoSAGA evaluation on 1,000 randomly selected, class-balanced CIFAR-10 training samples (100 samples per class). ProtoSAGA uses 100 attack steps and a maximum L_∞ perturbation of $\epsilon = 0.031$. Clean and robust accuracies (%) are reported for both models in each pair.

PAIR	CLEAN_1	CLEAN_2	ROBUST_1	ROBUST_2
WRN–CAIT	98.60	94.30	89.70	74.10
WRN–VIT	98.60	97.90	89.00	85.40
WRN–MAMBA	98.60	95.90	89.20	78.70
WRN–DM-R18	98.60	68.70	88.60	45.40
WRN–FAT-R18	98.60	76.50	90.00	49.00
CAIT–VIT	94.30	97.90	74.20	85.70
CAIT–MAMBA	94.30	95.90	74.20	78.90
CAIT–DM-R18	94.30	68.70	73.10	46.00
CAIT–FAT-R18	94.30	76.50	74.10	49.10
VIT–MAMBA	97.90	95.90	85.20	78.50
VIT–DM-R18	97.90	68.70	84.90	46.40
VIT–FAT-R18	97.90	76.50	86.80	48.90
MAMBA–DM-R18	95.90	68.70	78.20	46.00
MAMBA–FAT-R18	95.90	76.50	80.10	48.20
DM-R18–FAT-R18	68.70	76.50	46.50	49.50

Table 9: Attack parameters used to generate adversarial examples for the GaME experiments. Both APGD-CE and ProtoSAGA use the same 1,000 randomly selected class-balanced samples (100 samples per class) from the CIFAR-10 training set, without filtering for initial correct classification. APGD-CE is applied independently to each of the six defense models. ProtoSAGA is applied to all pairwise combinations of the six defense models, taking two models at a time, resulting in $\binom{6}{2} = 15$ model pairs.

PARAMETER	APGD-CE	ProtoSAGA
Dataset	CIFAR-10 training	CIFAR-10 training
Samples	1,000	1,000 per pair
Samples per class	100	100
Maximum perturbation	$\epsilon = 0.031$	$\epsilon = 0.031$
Attack steps	100	100
Coefficient learning rate	–	0.1
Fitting factor	–	10

We next evaluate whether Skeleton Key measurements can be used to inform attack-source selection against a randomized defense. For each experiment, we first fix a defender distribution P_D over the participating models and evaluate the resulting randomized defense on held-out adversarial examples. As a baseline, the source model for APGD is selected uniformly from the models participating in the defense. We compare this with a Skeleton-Key-informed setting, where the APGD source model is instead sampled according to an estimated defender distribution $\hat{P}_D$ derived from the Skeleton Key measurements. We repeat this evaluation across several defender distributions, including uniform, three-model, two-model, single-model, and normally distributed probability assignments. Table 12 reports the resulting held-out robust accuracy for each configuration.

For the experiments in Table 12, we vary P_D to study how well the Skeleton-Key approach works under different defense probability distributions. The experiment names simply describe how the probability mass is distributed across the models. For the *Single Defense Probability* setting, we use $P_D = [1, 0, 0, 0, 0, 0]$, where the probability of 1 corresponds to WRN based on the six-model GaME result. For *Two Defense Probability*, we use the distribution obtained from the five-model GaME experiment, $P_D = [0, 0.4813, 0.5187, 0, 0]$, where the nonzero probabilities corre-

Table 10: GaME-optimized defender mixed strategies λ^D for the six-model and five-model defense sets. The five-model configuration excludes WRN-28-10. The corresponding GaME robust accuracies are 93.80% and 80.28%, respectively.

MODEL	SIX-MODEL λ^D	FIVE-MODEL λ^D
CaiT-XXS	0.0000	0.0000
DM-R18	0.0000	0.0000
FAT-R18	0.0000	0.0000
MambaVision-T	0.0000	0.5187
ViT-L/16	0.0000	0.4813
WRN-28-10	1.0000	–
GaME Robust Accuracy	**93.80%**	**80.28%**

Table 11: Skeleton key evaluation for the six-model defense set using 2,000 randomly selected, class-balanced CIFAR-10 samples (200 samples per class). The same samples are used across all target models, without filtering for initial correct classification. A working Skeleton Key preserves the correct prediction of the target model while causing all non-target models to misclassify. All attacks use 100 iterations and a maximum L_∞ perturbation of $\epsilon_{\max} = 0.031$.

MODEL	VALID	NON-VALID	YIELD RATE
CaiT-XXS	15	1985	0.75%
DM-R18	37	1963	1.85%
FAT-R18	28	1972	1.40%
MambaVision-T	16	1984	0.80%
ViT-L/16	43	1957	2.15%
WRN-28-10	103	1897	5.15%

spond to ViT and MambaVision-T. We manually set the *Three Defense Probability* case to $P_D = [0.5, 0, 0.25, 0.25, 0, 0]$, giving probability only to WRN, ViT, and MambaVision-T. For *Uniform Defense Probability*, all six models are equally likely to be selected, with $P_D = [1/6, \ldots, 1/6]$. Finally, for the *Normal Dist. Probability* setting, we manually use $P_D = [0.26, 0.07, 0.26, 0.17, 0.17, 0.07]$ to evaluate a more distributed, non-uniform defense configuration.

We further study how the perturbation limit affects Skeleton key generation for WRN-28-10, ViT-L/16, and CaiT-XXS. Figure 1 shows the number of working keys, target-correct samples, and samples that fool all five non-target models as ϵ varies from $4/255$ to $255/255$. The number of working keys does not increase in one direction with the perturbation. For WRN-28-10, the number

Table 12: Held-out robust accuracy of the randomized defense under uniform and Skeleton-Key-informed APGD attack-source selection. For each experiment, the defense distribution P_D is fixed. In the uniform baseline, the APGD attack source is selected uniformly among the participating defense models. In the Skeleton-Key-informed setting, the attack source is selected according to the estimated defense distribution $\hat{P}_D$ obtained from Skeleton-Key measurements.

EXPERIMENT	MODELS	GROUND-TRUTH ROBUST ACC.	SKELETON-KEY-INFORMED ROBUST ACC.
Uniform Defense Probability	6	66.90%	65.00%
Three Defense Probability	6	76.30%	76.60%
Two Defense probability	5	75.90%	65.90%
Single Defense Probability	6	76.50%	75.70%
Normal Dist. Probability	6	70.10%	66.80%

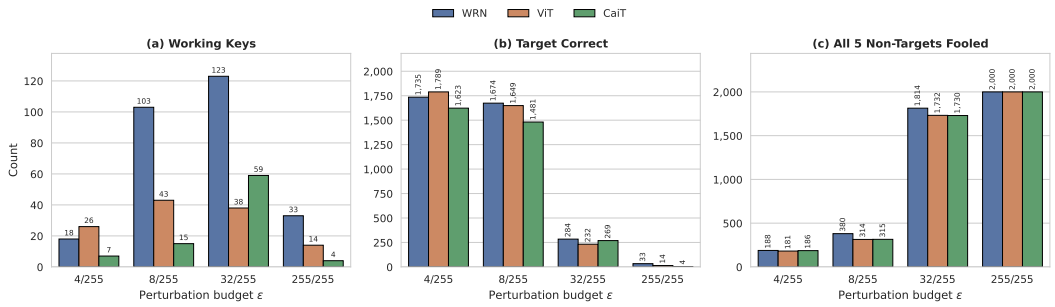


Figure 1: Robust accuracy across architectures.

of working keys increases from 18 at $\epsilon = 4/255$ to 103 at $8/255$ and peaks at 123 at $32/255$, before decreasing to 33 at $255/255$. ViT-L/16 and CaiT-XXS show a similar trade-off, although their peak yields occur at different perturbation. As ϵ increases, the attack becomes increasingly effective at fooling the non-target models, but it also becomes more likely to change the prediction of the target model. At $\epsilon = 255/255$, all five non-target models are fooled for all 2,000 samples, while the number of target-correct samples drops to 33, 14, and 4 for WRN-28-10, ViT-L/16, and CaiT-XXS, respectively. These results illustrate the competing objectives of Skeleton key generation: increasing the perturbation budget makes it easier to fool the non-target models, but an excessively large perturbation also makes it difficult to preserve the target model's correct prediction.