Cryptanalytic Extraction of Neural Networks for Privacy-Preserving Machine Learning

Anonymous Author(s)

Anonymous Institution

Abstract. Since Carlini et al. (CRYPTO'20) introduced a black-box neural network extraction method inspired by differential cryptanalysis, cryptanalytic model extraction has been extensively studied. Existing methods, however, still face three bottlenecks.

- First, these methods are largely confined to contractive architectures and generally do not extend to expansive networks because of rank deficiency. We classify the regimes for single-point and multi-point recovery and mitigate this limitation through projection-based aggregation in the multi-point case.
- Second, prior attacks fundamentally exploit the non-differentiability of activations such as ReLU. Asselineau et al. (CRYPTO'26) recently extended cryptanalytic extraction to a broad class of nonlinear activations, but their approach requires high numerical precision and may recover only a subset of the parameters. We show that privacy-preserving machine learning (PPML) unintentionally introduces a new attack surface: fixed-point arithmetic over finite rings induces silent modular wraparound, creating observable discontinuities even when the underlying activation is smooth. Exploiting these discontinuities, we recover all parameters of networks with smooth activations at much lower precision than prior work.
- Third, Neuron Wiggle, a widely used sign-recovery technique, is suboptimal. We recast it as a unified paradigm and analyze it from a signal-detection perspective. We prove that when the downstream coefficients are Gaussian, or the network is sufficiently wide, there exists an optimal probing direction, and the resulting method outperforms the original algorithm.

We evaluate our attack on eight CIFAR-10 DNNs and eight expansive networks with different activation functions. The attack extracts all parameters of expansive networks with smooth, unbounded activations. Our sign-recovery method further attains 100% accuracy on our benchmark while reducing the number of queries by 30%–74% relative to Neuron Wiggle.

Keywords: Model Extraction, Privacy-Preserving Machine Learning, Expansive Neural Network, Smooth Activation Function

1 Introduction

Deep neural networks (DNNs) have become a cornerstone of modern artificial intelligence and are widely deployed in domains such as computer vision and nat-

ural language processing. Training high-performance models typically requires large-scale datasets, substantial computational resources, and considerable research and engineering effort, making the resulting parameters valuable intellectual property. Understanding the extent to which an adversary can recover these parameters from black-box access has therefore become a central question in model security.

Model extraction has a long history, dating back to the seminal work of Fefferman [12] and to Lowd and Meek [23], who introduced the first adversarial reverse-engineering attack in this setting. A major milestone came in 2020, when Carlini et al. [6] recast model extraction in cryptanalytic terms and demonstrated parameter recovery for fully connected ReLU networks. However, their approach requires access to raw logits, is restricted to contractive networks, and relies on an exponential-time procedure for sign recovery. In 2024, Canales-Martínez et al. [3] introduced the Neuron Wiggle technique, enabling polynomial-time recovery of neuron signs. Building on these techniques, Foerster et al. [13] combined the methods of [6] and [3] and observed that, on many networks, Neuron Wiggle yields low-confidence sign estimates for a subset of neurons. More recently, Liu et al. [21] revisited the approach of [6], identified several technical limitations, and proposed a series of refinements to address them. In addition, they introduced a SOE+Neuron Wiggle method to mitigate the low-confidence issue of [3]. Building on these techniques, they successfully extended end-to-end parameter extraction from 3-layer to 8-layer networks. Xu et al. [42] later showed that this pipeline, previously evaluated in `float64` [6,21], becomes unstable in `float32`: the higher error floor can already disrupt critical-point search and signature merging. Shen et al. [31] studied the feasibility of recovering both the architecture and parameters when the architecture, i.e., the number of layers and layer dimensions, is unknown.

Subsequent work has proceeded along three parallel directions. The first line of work aims to broaden the class of supported activation functions. Most existing cryptanalytic extraction attacks target ReLU networks. Chen et al. [8] extended this line to PReLU via a neuron-splitting technique. Qi et al. [27] further recovered parameters for several activations with points of non-differentiability, including HardTanh, ELU, and the Step function, but their method does not apply to smooth activations. For smooth activations, Asselineau et al. [1] assumed that the activation approaches a linear or constant function exponentially fast in both tails and is substantially nonlinear only near the origin. This approximation, however, inevitably introduces errors. Their experiments show that even on CIFAR-10 networks, only 50%–62.5% of first-layer neurons can be recovered. More critically, their peeling-based recovery of deeper layers rapidly accumulates numerical errors: on a $16 \times 16^8 \times 10$ network, the recovery rate of fully denoised parameters drops from 58.6% in the second layer to 14.1% in the third layer, and further to 0% in the fourth layer. Practical cryptanalytic extraction of networks with smooth activations therefore remains an open problem.

The second line of work studies more general architectures. Sun et al. [33] extended parameter extraction to CNNs with average pooling, while Chen et

al. [9] and Liu et al. [20] considered CNNs with max pooling. Other works have further explored RNNs [39] and GNNs [18]. More recently, Wang et al. [37] extended cryptanalytic extraction to multi-head attention, a key component of modern Transformer architectures. Together, these results broaden the architectural scope of cryptanalytic extraction well beyond fully connected networks.

The third line of work considers more restrictive query models, where the attacker observes only partial output information. Chen et al. [8] showed that raw-output attacks extend to the top-k score setting. Chen et al. [7] introduced the first extraction attack in the hard-label setting, where only the top-1 label is revealed. However, their sign-recovery procedure has exponential complexity and is limited to binary classifiers. Carlini et al. [5] later introduced geometric techniques based on *dual points*, defined by intersections of decision and critical boundaries, enabling polynomial-time extraction in the hard-label setting. Canales-Martínez et al. [4] subsequently studied output-layer recovery. More recently, Ito et al. [15] showed that constantly active neurons can make dual-point search exponentially expensive and proposed a cross-layer strategy to address this limitation. Other recent works have extended hard-label extraction to PReLU [24] and CNN [10,17] models.

Despite significant progress in cryptanalytic extraction attacks, two challenges remain. We elaborate on the reasons for these challenges in detail below.
Challenge 1: Inability to Handle Expansive Networks. Chen et al. [8] pointed out that the extraction method of Carlini et al. [6] does not apply to expansive networks and left this as an open problem. Specifically, when recovering layer i, Carlini et al. compute differences in directional derivatives [1] across a critical point and require d_{i-1} independent directions, where d_{i-1} denotes the input dimension of layer i, to recover the neuron weight. However, the effective degrees of freedom near a critical point are bounded by the minimum width of the preceding layers, i.e., $\min_{k<i} d_k$, which may be insufficient in expansive networks. Existing methods for merging partial weight vectors are generally not applicable to expansive networks; see Subsection 4.3 for details.
Challenge 2: Difficulty in Handling Smooth Activation Functions. Previous extraction methods largely exploit *points of non-differentiability* in activation functions. For example, the derivative of ReLU is discontinuous at 0 ($\lim_{x\to 0^-} \sigma'(x) = 0$ and $\lim_{x\to 0^+} \sigma'(x) = 1$, where σ is the activation function). Consequently, the network gradient changes abruptly when a neuron's pre-activation crosses zero; we refer to such locations as *critical points*. These critical points lie on critical hyperplanes (or curved surfaces for ELU) that partition the input space $\mathbb{R}^{d_0}$. An attacker can recover the corresponding neuron weight either from the gradient difference across a critical point [6] or from the normal vector of the critical hyperplane in the i-th layer's input space [5]. Smooth activations such as Swish and GELU, however, are C^∞ and lack such derivative discontinuities, preventing direct localization of critical points using these methods. Asselineau et al. [1] address this by assuming that the activation is nonlinear only within a narrow

[1] We use the more precise notion of differences in derivatives rather than second-order derivatives, since the latter become unbounded near the critical point.

region, allowing the network near a critical hyperplane to be approximated as $F = \sigma(Ax + b) + L(x) + \beta$, where $L(x)$ is linear and β is a constant. Second-order derivatives then eliminate the contributions of $L(x)$ and β, enabling parameter recovery. However, this approximation introduces intrinsic errors, while estimating second- or higher-order derivatives using float64 arithmetic further amplifies numerical errors. Thus, practical parameter extraction from networks with smooth activations has not yet been fully resolved.

Silent Modular Wraparound in PPML. With the widespread deployment of Machine Learning as a Service (MLaaS), neural networks are increasingly used in sensitive applications such as medical diagnosis and financial risk assessment. To protect both client data and proprietary model parameters, privacy-preserving machine learning (PPML) based on Secure Multi-Party Computation (MPC) enables the data owner and the model owner to jointly perform inference without revealing their private inputs. Representative systems include SPDZ [11], SecureML [26], MiniONN [22], ABY3 [25], SecureNN [35], CrypTFlow2 [29], Falcon [36], CryptGPU [34], SIRNN [28], and Piranha [38]. To reduce the high cost of secure floating-point arithmetic, most MPC frameworks use fixed-point arithmetic over finite rings, e.g., $\mathbb{Z}_M$ with $M = 2^{64}$. A real value x is encoded as $\tilde{x} = \lfloor x \cdot 2^S \rceil \bmod M$. This design is efficient but introduces a subtle vulnerability: *silent modular wraparound*. When an intermediate value exceeds the representable range $[-M/2, M/2)$, it is silently reduced modulo M, causing a sudden discontinuous jump that does not occur in ordinary real-valued neural networks.

Remark. Prior work, MUSE [16], showed that a malicious client can extract model parameters during secure inference, but only by actively deviating from the protocol. In contrast, our adversary remains fully protocol-compliant and only adaptively submits valid inference queries and observes the corresponding outputs, as in ordinary PPML inference.

1.1 Our Contributions

Cryptanalytic Extraction in PPML. Although secure multi-party computation (MPC) is designed to protect both user data and model parameters, its arithmetic unintentionally introduces a new attack surface—silent modular wraparound. Exploiting this inherent behavior, we develop a polynomial-time cryptanalytic extraction attack against PPML systems. Table 1 compares our results with several relevant prior attacks. Silent modular wraparound creates output discontinuities, and we call the corresponding inputs *wraparound points*. We extend cryptanalytic extraction to PPML based on three observations:

- *Problem formulation.* PPML inference is typically realized by fixed-point arithmetic over a finite ring, which is not directly amenable to real-domain analysis. We construct an equivalent real-valued network (Definition 3) by replacing each activation function with its periodic counterpart.

Table 1: Comparison of our attack with several state-of-the-art cryptanalytic extraction methods.

Work	**Activation Functions**	**Expansive Network**		**Required API**
		Signature	**Sign**	
CRYPTO'20 [6]	ReLU	×	Exponential	Raw outputs*
EUROCRYPT'24 [3]	ReLU	×	Polynomial	Raw outputs
EUROCRYPT'25 [5]	ReLU	×	Polynomial	Top-1 label
ASIACRYPT'25 [8]	PReLU	×	×	Top-k scores
ToSC'26 [27]	HardTanh, ELU, Step	×	×	**
CRYPTO'26 [1]	Non-Linear†	×	×‡	Raw outputs
Ours	**Unbounded Activations**	✓	**Polynomial (Improved)**	**Top-1 score**

* *The raw-output setting readily extends to the top-k setting, since the log-ratio of any two softmax probabilities recovers the difference between their corresponding logits [8]. This does not extend to the top-1 setting, where no second probability is available as a reference.*

** *For piecewise-linear activation functions, the approach can be extended to the top-1 label setting, whereas for ELU it is applicable only under the raw-output setting.*

† *Because of errors in their approximation and in computing higher-order derivatives, they can generally recover only a subset of the parameters.*

‡ *[1] proposed an algebraic sign-recovery method claimed to apply to expansive networks with $m \leq n$. However, we find that the method is valid only when $m = n$. Moreover, Neuron Wiggle cannot recover signs for smooth activations in the standard setting, since their gradients are continuous and lack the discontinuities required for sign recovery.*

- *Signature Recovery.* In the ReLU setting, parameter recovery is possible because critical points can be localized and their corresponding pre-activation values are known to be zero. Wraparound points enjoy an analogous property: they can also be localized, while their pre-activation values are known constants, thereby exposing information about the underlying weights.
- *Sign Recovery.* Neuron Wiggle method for ReLU networks exploit the sharp divergence of gradients across critical points. We show that an analogous gradient divergence also arises across wraparound points.

The extraction techniques developed for the standard setting naturally carry over to PPML, while the arithmetic structure of PPML provides two additional advantages:

- *Smooth activations.* Although smooth activations lack locatable critical points, modular wraparound creates identifiable discontinuities, allowing us to recover networks with any unbounded activation function, regardless of whether it is smooth.
- *Top-1 score scenario.* While cryptanalytic extraction for ReLU networks has been extended to the top-1 label setting [5], existing attacks on smooth-

activation networks still require access to at least the top-k scores. Since our attack detects discontinuous changes in the observable output, it remains applicable when the oracle reveals only the top-1 label and its associated probability. Moreover, our attack is robust to output noise and therefore remains effective against defenses that add small perturbations to the returned probabilities.

Signature Recovery for Expansive Networks. We divide signature recovery into *single-point* and *multi-point* recovery according to the network architecture and activation function, as summarized in Table 3.

- *Single-point recovery.* This approach applies to contractive networks with non-ReLU activations and, in principle, to expansive networks with non-polynomial activations. In practice, however, limited numerical precision often leads to rank deficiency, as many activation functions rapidly approach linear or constant regimes—which is precisely the key property that enables the success of [1].
- *Multi-point recovery.* When a single point is insufficient, as in ReLU networks, expansive architectures, or numerically rank-deficient cases, we combine information from multiple points. The key challenge is determining whether multiple points correspond to the same neuron. Previous attacks [6,21] use only observations in the input space of layer i, ignoring the mapping induced by the preceding $i-1$ layers. We establish an algebraic relation between the boundary normal vector and the target neuron weight through this mapping, enabling us to identify points corresponding to the same neuron and aggregate their partial information to recover the complete weight.

Improved Sign Recovery Algorithm. We recast Neuron Wiggle into a unified paradigm based on selecting a probing direction and comparing the magnitudes of the resulting directional derivatives, and show that Neuron Wiggle is suboptimal within this framework. We prove that, when the downstream coefficients follow a Gaussian distribution or the network is sufficiently wide, the sign-recovery success probability increases monotonically with the signal-to-interference ratio (SIR), and derive the globally optimal probing direction that maximizes the SIR. Both theory and experiments show that our method outperforms Neuron Wiggle. Moreover, our approach is orthogonal to the SOE+Neuron Wiggle method proposed in [21].

Experiments. Under the MPC setting, we evaluate our attack on eight CIFAR-10 DNNs with different activation functions, including GELU, Swish, and Mish, and eight randomly generated expansive networks. Each CIFAR-10 network contains 986,634 parameters and four hidden layers of 256 neurons, while each expansive network takes 100-dimensional inputs and contains four hidden layers of 200 neurons. Our full code is available at

`https://anonymous.4open.science/r/PPML_Model_Extraction_Attack`.

2 Preliminaries

2.1 Basic Definitions and Notations

Definition 1. *Given a modulus M, the finite ring $\mathbb{Z}_M$ is defined as the set of integers in the interval $[-M/2, M/2)$. For any integer $x \in \mathbb{Z}$, the wraparound operator $\mathcal{W}_M : \mathbb{Z} \to \mathbb{Z}_M$ is defined as:*

$$\mathcal{W}_M(x) = x - k \cdot M$$

where $k = \lfloor \frac{x}{M} + \frac{1}{2} \rfloor$ is called the wraparound coefficient. This operator maps any integer back to the principal value interval of the ring $\mathbb{Z}_M$.

Definition 2. *Given a scaling factor S, for any real number $x \in \mathbb{R}$, its fixed-point representation $\tilde{x}$ on the ring $\mathbb{Z}_M$ is defined as:*

$$\tilde{x} = \mathcal{W}_M \left(\lfloor x \cdot 2^S \rceil \right)$$

where $\lfloor \cdot \rceil$ denotes rounding to the nearest integer.

For two ring elements $\tilde{x}, \tilde{y} \in \mathbb{Z}_M$, addition is defined as modular wraparound after ordinary addition: $\tilde{x} \oplus \tilde{y} = \mathcal{W}_M(\tilde{x} + \tilde{y})$. Fixed-point multiplication involves three steps: multiplication, truncation, and wraparound: $\tilde{x} \otimes \tilde{y} = \mathcal{W}_M \left(\left\lfloor \frac{\tilde{x} \cdot \tilde{y}}{2^S} \right\rceil \right)$. If $\tilde{x} \oplus \tilde{y}$ or $\tilde{x} \otimes \tilde{y}$ exceeds the valid range of $[-M/2, M/2)$, the wraparound phenomenon will occur, corresponding to a wraparound coefficient $k \neq 0$.

To extract parameters from a neural network operating over a finite ring, we first construct a universal attack framework that is equivalent to the finite-ring computation and amenable to rigorous analysis in the real domain.

Definition 3. *An r-deep neural network $f_\theta : \mathbb{R}^{d_0} \to \mathbb{R}^{d_{r+1}}$ equivalent to the computation on a finite ring $\mathbb{Z}_M$ with a scaling factor S is defined as:*

$$f = \sigma_{out} \circ f_{r+1} \circ \sigma \circ \cdots \circ \sigma \circ f_2 \circ \sigma \circ f_1 \circ \sigma_{in}$$

where f_i denotes the linear transformation, σ represents the equivalent activation function, and σ_{in} and σ_{out} denote the input and output activation functions, respectively.

Definition 4. *The i-th fully connected layer $f_i : \mathbb{R}^{d_{i-1}} \to \mathbb{R}^{d_i}$ is defined by*

$$f_i(x) = W^{(i)} x + b^{(i)},$$

where $W^{(i)} \in \mathbb{R}^{d_i \times d_{i-1}}$ is the weight matrix, and $b^{(i)} \in \mathbb{R}^{d_i}$ is the bias vector.

Definition 5. *The equivalent activation function σ is defined by*

$$\sigma(y) = \sigma_{original}\left(y - 2kT_{\max}\right),$$

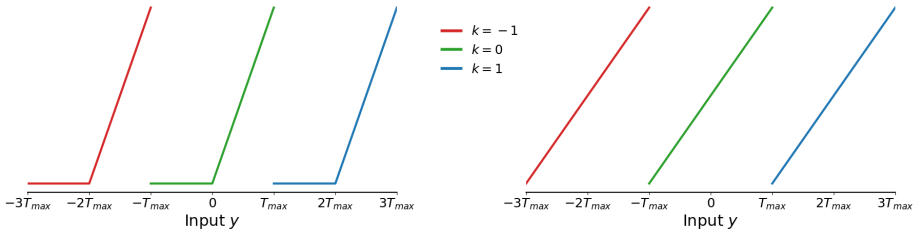


Fig. 1: The left panel shows the equivalent ReLU activation, while the right panel shows the input/output activations. The green segment ($k = 0$) denotes the principal interval without wraparound, whereas the red ($k = -1$) and blue ($k = 1$) segments denote the first wraparound intervals after negative underflow and positive overflow, respectively.

where $\sigma_{original}$ denotes the original activation function, as summarized in Table 2. The parameter $T_{\max} = \frac{M}{2^{S+1}}$ denotes the representation bound in the real domain, and $k \in \mathbb{Z}$ is the wraparound coefficient determined by $k = \left\lfloor \frac{y}{2T_{\max}} + \frac{1}{2} \right\rfloor$. The input and output activation functions are similarly defined as $\sigma_{in}(y) = \sigma_{out}(y) = y - 2kT_{\max}$. Figure 1 illustrates the resulting periodic equivalent function.

Definition 6. *The j-th neuron in the i-th layer of the network is defined by*

$$\eta_j^{(i)}(x) = \sigma\left(W_j^{(i)} x + b_j^{(i)}\right),$$

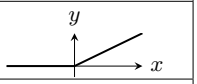
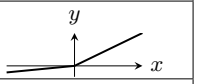
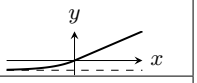
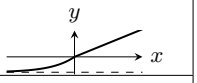
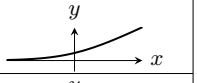
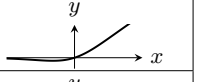
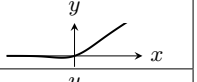
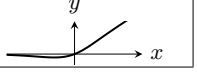
where $W_j^{(i)}$ is the j-th row of the weight matrix $W^{(i)}$, and $b_j^{(i)}$ is the j-th entry of the bias vector $b^{(i)}$.

The key idea of our equivalent model is to redefine the activation function σ, enabling analysis in the real domain. Under this representation, modular wraparound is no longer treated as a computational anomaly but as a periodic extension of the activation function. Although the input and output layers typically have no standard activation functions, they are still subject to modular wraparound; we therefore introduce dedicated functions σ_{in} and σ_{out}. Since our analysis restricts inputs to $(-T_{\max}, T_{\max})^{d_0}$, we omit σ_{in} from subsequent expressions.

Definition 7. *The signature of a neuron $\eta_j^{(i)}$ is $s_j^{(i)} \frac{W_j^{(i)}}{\|W_j^{(i)}\|_2}$, where $s_j^{(i)} \in \{-1, 1\}$, and its norm is $\|W_j^{(i)}\|_2$.*

Definition 8. *The architecture of a fully connected neural network is uniquely described by its number of layers r, the dimension of each layer $d_0, \ldots, d_{r+1}$ and the choice of activation function σ.*

Table 2: Several commonly used unbounded activation functions, along with their analytical expressions and function plots.

Activation Function	Analytical Expression	Graph
ReLU	$\mathrm{ReLU}(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$	
LeakyReLU	$\mathrm{LReLU}(x) = \begin{cases} x, & x \geq 0 \\ \alpha x, & x < 0 \end{cases} \ (\alpha \in (0,1))$	
ELU	$\mathrm{ELU}(x) = \begin{cases} x, & x \geq 0 \\ \alpha(e^x - 1), & x < 0 \end{cases} \ (\alpha > 0)$	
SELU	$\mathrm{SELU}(x) = \lambda \begin{cases} x, & x \geq 0 \\ \alpha(e^x - 1), & x < 0 \end{cases}$	
Softplus	$\mathrm{Softplus}(x) = \ln(1 + e^x)$	
Swish (SiLU)	$\mathrm{Swish}(x) = \dfrac{x}{1 + e^{-x}}$	
GELU	$\mathrm{GELU}(x) = \dfrac{1}{2}x\left(1 + \mathrm{erf}\left(\dfrac{x}{\sqrt{2}}\right)\right)$	
Mish	$\mathrm{Mish}(x) = x \tanh(\ln(1 + e^x))$	

Definition 9. *The parameters of an r-hidden-layer neural network f_θ consist of the weights and biases of all layers, i.e., $\theta = \{W^{(i)}, b^{(i)}\}_{i=1}^{r+1}$.*

Let $F_i = \sigma \circ f_i \circ \cdots \circ \sigma \circ f_1$ denote the mapping through the first i hidden layers, with F_0 being the identity map. Similarly, let $G_{i+1} = \sigma_{\text{out}} \circ f_{r+1} \circ \sigma \circ f_r \circ \cdots \circ \sigma \circ f_{i+1}$ denote the mapping from the output of layer i to the final network output. Then, for $0 \leq i \leq r$, $f = G_{i+1} \circ F_i$. We write $h^{(i)}(x) = F_i(x)$ for the output of layer i, where $h^{(0)}(x) = x$, and $z^{(i)}(x) = W^{(i)} h^{(i-1)}(x) + b^{(i)}$ for its pre-activation. Their j-th components are denoted by $h_j^{(i)}(x)$ and $z_j^{(i)}(x)$, respectively.

2.2 Adversarial Goals and Assumptions

Our security model assumes that the attacker can adaptively choose queries x to the oracle $\mathcal{O}_\theta$, which returns the corresponding predictions $\mathcal{O}_\theta(x)$. In the raw-output setting, the oracle returns the complete output vector $f(x)$; in the top-1 score setting, it returns the predicted class label together with its associated probability, i.e., $(\arg\max f(x), \max f(x))$. Given a bounded number of query pairs $(x, \mathcal{O}_\theta(x))$, the attacker aims to recover parameters $\hat{\theta}$ that minimize $\|\hat{\theta} - \theta\|$, up to neuron permutations. We use $\hat{\cdot}$ to denote recovered parameters.

Assumptions. We make the following assumptions about the oracle and the attacker's capabilities:

– **Architecture knowledge.** The attacker knows the model architecture and representation bound $T_{\max}$.
– **Full-domain inputs.** The attacker can adaptively choose any input $x \in (-T_{\max}, T_{\max})^{d_0}$.
– **Precise computation.** The DNN is evaluated using high-precision fixed-point arithmetic.
– **Unbounded activations.** The DNN employs an unbounded activation function.

2.3 Extraction strategy

Following prior work [6,5], we adopt a layer-by-layer strategy: when extracting layer i, we assume that the parameters of layers $1, \ldots, i-1$ have already been recovered. Parameter extraction for each layer consists of two main steps:

– *Signature and norm*[2] *recovery.* This step recovers the signature and norm of each neuron's weight vector, together with its bias, up to a common sign; that is, $W_j^{(i)} = \hat{W}_j^{(i)}$ or $W_j^{(i)} = -\hat{W}_j^{(i)}$.
– *Sign recovery.* This step determines the common sign of the neuron's weight vector and bias.

3 Wraparound Points Search

Since the equivalent activation function σ is discontinuous at $(2k+1)T_{\max}$ for $k \in \mathbb{Z}$, we refer to inputs that cause a jump in the neuron's output as wraparound points.

Definition 10. *If for an input x, there exist $i, j, k \in \mathbb{Z}$ such that $z_j^{(i)}(x) = (2k+1)T_{\max}$, then x is called a wraparound point of neuron $\eta_j^{(i)}$. Wraparound points originating from the same neuron are called* homologous[3].

Our method for searching for wraparound points relies on detecting changes in the network output. We therefore need to determine whether crossing a wraparound point necessarily induces an output discontinuity, unlike critical points, which almost always cause gradient discontinuities. As established in Theorem 1, the network output varies continuously as long as no wraparound point is crossed. In contrast, Theorem 2 shows that although crossing a wraparound point does not always induce a discontinuity, the probability of continuity is only $\sum_{k=i+1}^{r} p^{d_k}$. Among the activation functions considered, ReLU has the highest continuity probability because $p = 1/2$, making it more likely to enter the inactive region. Nevertheless, this probability remains extremely small. Thus,

[2] Conventional extraction attacks on piecewise-linear activations recover only a neuron's signature because of positive homogeneity. Modular wraparound breaks this homogeneity, allowing our attack to recover the norm as well.
[3] This corresponds to the notion of "consistent" in [5].

crossing a wraparound point is overwhelmingly likely to cause an abrupt output discontinuity, and searching for wraparound points does not incur exponential complexity. The detailed procedure is given in Algorithm 5.

Theorem 1. *Assume no wraparound points exist in the ϵ-neighborhood of x. Let the norm of each neuron's weight vector in layer i be bounded by K_i, i.e., $\|W_j^{(i)}\|_2 \le K_i$ for all j, and let σ be a λ-Lipschitz continuous activation function. Then, for any perturbation with $\|\Delta x\|_2 < \epsilon$, the network output satisfies $\|f(x+\Delta x) - f(x)\|_2 \le C\|\Delta x\|_2$, where $C = \lambda^r \prod_{i=1}^{r+1} \sqrt{d_i} K_i$.*

Theorem 2. *Suppose x is a wraparound point in the i-th layer. Let $\mathcal{S}_\sigma = \{z \in \mathbb{R} \mid \exists \epsilon > 0, \forall y \in (z-\epsilon, z+\epsilon), \sigma(y) = \sigma(z)\}$ denote the flat region of the activation function, and let $p = \Pr[z \in \mathcal{S}_\sigma]$ be the probability that an intermediate pre-activation falls into this region. When traversing across this wraparound point, the probability that the network's final output remains continuous is bounded by $\Pr[\lim_{\|\Delta x\|\to 0} f(x+\Delta x) = f(x-\Delta x)] < \sum_{k=i+1}^{r} p^{d_k}$.*

After locating a wraparound point x, we compute the normal vector at x. We follow the same approach as in [5]. Specifically, for any sufficiently small perturbation v, the point $x + \big(v - \mathrm{proj}_n(v)\big)$ remains on the tangent hyperplane at x. Equivalently, displacements orthogonal to n remain within the tangent hyperplane. Exploiting this geometric property, Algorithm 6 reconstructs n through a series of oracle queries.

Remark. In the top-1 score setting, the search for wraparound points remains largely unchanged. Rather than monitoring changes in the top-1 label, which may simply result from ordinary decision-boundary crossings, we instead detect discontinuous jumps in the associated confidence score.

4 Signature and Norm Recovery

We divide signature recovery into single-point recovery and multi-point recovery. Subsection 4.1 describes the scenarios to which each approach applies, subsection 4.2 and subsection 4.3 present the respective recovery methods, and subsection 4.4 details the recovery of the neuron norm.

4.1 Regimes of Single-Point and Multi-Point Recovery

Definition 11. *The wraparound boundary*[4] *associated with the k-th wrap of neuron $\eta_j^{(i)}$ is $B(\eta_j^{(i)}, k) = \left\{x \in \mathbb{R}^{d_0} \mid z_j^{(i)}(x) = (2k+1)T_{\max}\right\}$. The signature hyperplane associated with the k-th wrap of neuron in the input space of layer i is $S(\eta_j^{(i)}, k) = \left\{h \in \mathbb{R}^{d_{i-1}} \mid W_j^{(i)} h + b_j^{(i)} = (2k+1)T_{\max}\right\}$.*

[4] For nonlinear activations, the set of wraparound points may form a curved surface, whereas for piecewise-linear activations it forms a hyperplane; we therefore use the term "boundary" uniformly. In contrast, the signature hyperplane is always a hyperplane.

Table 3: Impact of activation functions and network architectures on neuron signature recovery. The symbols $=$ and $\neq$ indicate whether the affine-span condition $\mathrm{Aff}(B^{i-1}_{x,R}) = S(\eta^{(i)}_j, k)$ holds, corresponding to single-point and multi-point recovery, respectively.

Activation Function	**Contractive**	**Expansive**
ReLU	$\neq$	$\neq$
LeakyReLU	$=$	$\neq$
GELU, Swish, etc.	$=$	$=$

Since the normal vector of the signature hyperplane $S(\eta^{(i)}_j, k)$ is exactly the signature of neuron $\eta^{(i)}_j$, signature recovery reduces to recovering this normal vector. Given a wraparound point, the available information consists of the point itself and its local neighborhood. For $x \in B(\eta^{(i)}_j, k)$, we define the local wraparound boundary as $B_{x,R} = B(\eta^{(i)}_j, k) \cap N(x, R)$, where $N(x, R)$ denotes the R-neighborhood of x. Let $B^{i-1}_{x,R} = F_{i-1}(B_{x,R})$. Since $B^{i-1}_{x,R} \subseteq S(\eta^{(i)}_j, k)$, we have $\mathrm{Aff}(B^{i-1}_{x,R}) \subseteq S(\eta^{(i)}_j, k)$, where Aff denotes the affine span. Therefore, a single wraparound point suffices to recover the neuron's signature if and only if $\mathrm{Aff}(B^{i-1}_{x,R}) = S(\eta^{(i)}_j, k)$.

Both the lossiness of ReLU and network expansion can cause $\mathrm{Aff}(B^{i-1}_{x,R}) \neq S(\eta^{(i)}_j, k)$, meaning that the affine span fails to cover the entire signature hyperplane. For contractive networks with non-ReLU activations, the equality $\mathrm{Aff}(B^{i-1}_{x,R}) = S(\eta^{(i)}_j, k)$ holds, which allows the normal vector to be recovered directly from a single wraparound point. For expansive networks, since $\min_{k<i}\{d_k\} < d_i$, one would generally expect $\mathrm{Aff}(B^{i-1}_{x,R}) \neq S(\eta^{(i)}_j, k)$. Surprisingly, we identify a non-trivial property of non-polynomial activation functions such as GELU and Swish. Theorem 3 shows that, despite the dimensional deficiency $\dim(B^{i-1}_{x,R}) < d_{i-1}-1$, its affine span can still cover the entire signature hyperplane: $\mathrm{Aff}(B^{i-1}_{x,R}) = S(\eta^{(i)}_j, k)$, where the signature hyperplane has dimension $d_{i-1} - 1$. These properties are detailed in Table 3.

Theorem 3. *Let σ be a non-polynomial analytic activation function, and let $x \in B(\eta^{(i)}_j, k)$ be a known wraparound point. Provided that the network's forward mapping F_{i-1} is non-degenerate within the local R-neighborhood of x, the affine span of the mapped local boundary strictly encompasses the entire signature hyperplane. That is,* $\mathrm{Aff}(B^{i-1}_{x,R}) = S(\eta^{(i)}_j, k)$.

An example for Theorem 3. As shown in Figure 2, we consider a GELU-based neural network with a two-dimensional input space. Since the $(i-1)$-th

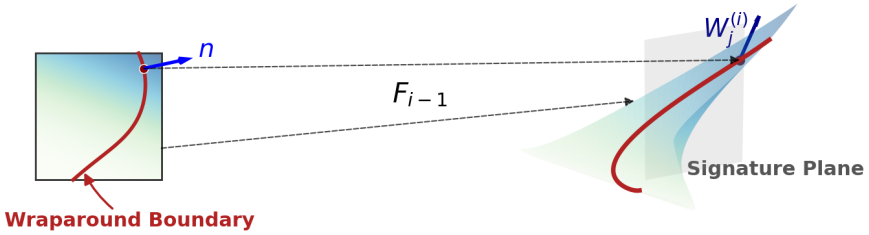


Fig. 2: This is a neural network based on the GELU activation function. F_{i-1} maps a local 2D green plane into a green curved surface in 3D space.

layer contains three neurons, the input space of layer i is three-dimensional. The local wraparound boundary $B_{x,R}$ is a one-dimensional curve segment in the two-dimensional input space, while $B_{x,R}^{i-1}$ is a one-dimensional curve segment in the three-dimensional space and lies within the two-dimensional signature hyperplane $S(\eta_j^{(i)}, k)$. Because three non-collinear points can be found on this curve, its affine span covers the entire signature hyperplane. Therefore, for networks with non-polynomial activations, $\mathrm{Aff}(B_{x,R}^{i-1}) = S(\eta_j^{(i)}, k)$, enabling recovery of the neuron's signature from a single wraparound point.

Remark. Note that ELU and SELU are two special cases in Table 2. Since they are linear in the positive regime but non-polynomial in the negative regime, the feasibility of single-point recovery depends on the activation states at the chosen point. For example, if all neurons are active, single-point recovery is infeasible; if all are inactive, it becomes feasible.

4.2 Single-Point Signature Recovery

We first introduce the Jacobian matrix. Although Theorem 4 is not strictly necessary for single-point recovery, it plays a crucial role in subsection 4.3.

Definition 12. *The Jacobian matrix of layer i is defined as the matrix of partial derivatives of the post-activation features $h^{(i)}(x)$ with respect to the network input x, denoted by $J^{(i)}(x) = \frac{\partial h^{(i)}(x)}{\partial x} \in \mathbb{R}^{d_i \times d_0}$.*

Theorem 4. *The Jacobian matrix of the i-th layer can be factorized as $J^{(i)}(x) = \prod_{k=i}^{1} \left(\Lambda^{(k)}(x) W^{(k)}\right)$, where the diagonal activation derivative matrix is given by:*

$$\Lambda^{(k)}(x) = \mathrm{diag}\left(\sigma'(z_1^{(k)}(x)), \sigma'(z_2^{(k)}(x)), \ldots, \sigma'(z_{d_k}^{(k)}(x))\right)$$

Furthermore, if x is a wraparound point of neuron $\eta_j^{(i)}$, its corresponding normal vector satisfies

$$n(x) \parallel J^{(i-1)}(x)^T (W_j^{(i)})^T, \tag{1}$$

where $\parallel$ denotes collinear.

For piecewise-linear activation functions such as ReLU, $\Lambda^{(k)}$ is a diagonal matrix with entries in $\{0, 1\}$, so $n(x)$ remains constant within each local linear region. For non-linear activation functions, however, $\sigma'(z)$ varies continuously, causing $\Lambda^{(k)}(x)$ and hence the normal vector $n(x)$ to vary continuously with the input x. Single-point recovery can recover all parameters of the first layer, regardless of the activation function or whether the network is expansive.

First Layer Extraction. For the first layer, since the input has not been transformed by any hidden layers, the normal vector n at a wraparound boundary is parallel to the neuron's weight vector $W_j^{(1)}$. To distinguish first-layer wraparound points from deep-layer ones, we analyze the geometric properties of their normal vectors: homologous wraparound points triggered by the same first-layer neuron yield parallel normal vectors, whereas deep-layer wraparound boundaries become highly distorted when mapped back to the input space, causing their normal vectors to exhibit strong irregularity. By clustering these vectors, we exploit this geometric contrast to identify d_1 dense clusters, while deep-layer points are excluded as discrete outliers. The centroids of these d_1 clusters correspond exactly to the signatures of the first-layer neurons. Additionally, we can adopt the methods from [1,21] to further improve the accuracy of filtering first-layer signatures.

Deep Layer Extraction. For layer i, let $r = \text{rank}(J^{(i-1)}(x))$. If $r = d_{i-1}$, i.e., $J^{(i-1)}(x)$ has full row rank, the target neuron's signature follows directly from the relation 1. According to Table 3, the lossiness of ReLU on negative inputs and the dimensional expansion of expansive networks typically make the Jacobian rank-deficient, i.e., $r < d_{i-1}$. A special case arises in contractive ReLU networks, where a wraparound point x may satisfy $r = d_{i-1}$. Such points, however, occur with negligible probability because, at a minimum, all neurons in layer $i-1$ must be active. In contrast, single-point recovery is impossible for deeper layers of expansive networks with piecewise-linear activations.

Non-Polynomial Activation Functions Extraction. From the Jacobian perspective, a single wraparound point suffices for signature recovery with non-polynomial activations because $J^{(i-1)}(x)$ varies continuously within the neighborhood of a wraparound boundary, allowing us to exploit multiple nearby points on the same boundary. Suppose x_0 is a wraparound point of neuron $\eta_j^{(i)}$ at the k-th wrap. We collect m nearby points $\{x_\ell\}_{\ell=1}^m$ on the same wraparound boundary. Their representations $h^{(i-1)}(x_\ell)$ lie on the same signature hyperplane $S(\eta_j^{(i)}, k)$, so $W_j^{(i)} h^{(i-1)}(x_\ell) + b_j^{(i)} = (2k+1)T_{\max}$. Let $\Delta h_\ell = h^{(i-1)}(x_\ell) - h^{(i-1)}(x_0)$ and $D = [\Delta h_1 \ \Delta h_2 \ \cdots \ \Delta h_m]^T$. Then $D(W_j^{(i)})^T = \mathbf{0}$. By Theorem 3, for non-polynomial analytic activations, $\text{rank}(D) = d_{i-1} - 1$. Hence, the null space of D is one-dimensional, and its direction gives the signature of $\eta_j^{(i)}$.

Under limited numerical precision, we can often recover only a subspace V containing the neuron signature, i.e., $W_j^{(i)} \in V$. This is mainly because

the second-order derivatives of common activation functions are close to zero, $\sigma''(x) \approx 0$, over large regions [1]. Consequently, the Jacobians at nearby points are nearly identical, so the difference matrix D has only about r significant singular values. The remaining $d_{i-1} - r - 1$ singular values are numerically indistinguishable. In our experiments, $\mathrm{rank}(D)$ is typically only one or two greater than r. Therefore, under limited numerical precision, we generally use multi-point recovery for all activation functions except LeakyReLU.

4.3 Multi-Point Signature Recovery

Prior Work. Since ReLU does not support single-point recovery, Carlini et al. [6] proposed a naive method for merging partial solutions. Specifically, let w_1 denote the partial weight vector recovered at input x_1, with valid entries indexed by $t_1 \subset \{1, \ldots, d_{i-1}\}$, and define w_2 analogously for x_2 with index set t_2. If x_1 and x_2 correspond to the same neuron and $t_1 \cap t_2 \neq \emptyset$, the two partial vectors can be merged to recover a larger weight vector over $t_1 \cup t_2$.

Extracting valid entries, however, requires the corresponding weight solution to be unique, i.e., $r = \mathrm{rank}(\Lambda^{(i-1)}(x))$. This condition does not always hold even in contractive networks, giving rise to the *insufficient rank* problem identified by Liu et al. [21]. This issue arises when the number of active neurons in layer $i-1$ exceeds the effective rank provided by the preceding-layer mapping. Expansive networks further exacerbate the problem, since $r < \min_{k<i}\{d_k\}$, imposing a stricter upper bound on the effective degrees of freedom. In particular, when $\mathrm{rank}(\Lambda^{(i-1)}(x)) > \min_{k<i}\{d_k\}$, the valid entries cannot be uniquely recovered. For expansive LeakyReLU networks, extracting valid entries is impossible because $r = \min_{k<i}\{d_k\} < d_{i-1} = \mathrm{rank}(\Lambda^{(i-1)}(x))$.

Liu et al.'s modification addresses the insufficient rank problem but does not resolve the difficulty caused by expansive architectures. We find that, for LeakyReLU, their merging condition cannot be satisfied when $d_{i-1} \geq 2\min_{k<i}\{d_k\}$, while for ReLU, it becomes extremely difficult to satisfy under the average-case condition when $d_{i-1} \geq \frac{8}{3}\min_{k<i}\{d_k\}$. See their subsection 3.1 for details.

Projection-based aggregation method. Instead of recovering the signature solely from the difference matrix in the input space of layer i, our method exploits the algebraic relation $J(x)^T W^T \parallel n$ established in the relation 1, which links the boundary normal vector n at x to the target neuron weight. Compared with the difference-matrix approach, this relation not only captures the projection of $W_j^{(i)}$ visible at x, but also incorporates the structure of the preceding mapping F_{i-1} into the recovery process. In fact, this relation captures all locally available projection information about the neuron signature at x. Although neighboring wraparound points may provide additional information, as discussed in subsection 4.2, such information is weak under finite numerical precision.

Consider two distinct wraparound points, x_a and x_b. For notational simplicity, let $J_a = J^{(i-1)}(x_a)$ and $J_b = J^{(i-1)}(x_b)$. We define the *visible subspaces* V_a

Algorithm 1 Homology Verification Algorithm

Input: Jacobian matrices J_a, J_b, normal vectors n_a, n_b, thresholds $t, \epsilon_{\text{svd}}, \tau_{\text{score}}$
Output: **true** if two wraparound points are homologous, otherwise **false**

$\hat{w}_a \leftarrow (J_a^T)^\dagger n_a, \hat{w}_b \leftarrow (J_b^T)^\dagger n_b$
$[U_a, \Sigma_a, V_a^T] \leftarrow \text{SVD}(J_a)$
$[U_b, \Sigma_b, V_b^T] \leftarrow \text{SVD}(J_b)$
$Q_a \leftarrow (U_a)_{:,\{\sigma_i > \epsilon_{\text{svd}}\}}$
$Q_b \leftarrow (U_b)_{:,\{\sigma_i > \epsilon_{\text{svd}}\}}$
$[U, \Sigma, V^T] \leftarrow \text{SVD}(Q_a^T Q_b)$ % Principal Angles
$K \leftarrow \{k \mid \sigma_k > 1 - \epsilon_{\text{svd}}\}$
if $|K| < t$ **then**
  return false % Common subspace is too small to determine
$U_{\text{int}} \leftarrow Q_a U_{:,K}$
$v_a \leftarrow U_{\text{int}}^T \hat{w}_a, v_b \leftarrow U_{\text{int}}^T \hat{w}_b$ % Project onto the common space
Score $\leftarrow \frac{|v_a \cdot v_b|}{\|v_a\| \|v_b\|}$
if Score $> \tau_{\text{score}}$: **return true**
else : **return false**

and V_b at x_a and x_b as

$$V_a = \text{Row}(J_a^T) \subseteq \mathbb{R}^{d_{i-1}}, \qquad \dim(V_a) = r_a,$$

$$V_b = \text{Row}(J_b^T) \subseteq \mathbb{R}^{d_{i-1}}, \qquad \dim(V_b) = r_b.$$

From the algebraic relations $J_a^T (W_a)^T \parallel n_a$ and $J_b^T (W_b)^T \parallel n_b$, we obtain the orthogonal projections of the weight vectors $W_a, W_b \in \mathbb{R}^{d_{i-1}}$ onto V_a and V_b, respectively:

$$\hat{w}_a = P_a((W_a)^T) \parallel (J_a^T)^\dagger n_a, \qquad \hat{w}_b = P_b((W_b)^T) \parallel (J_b^T)^\dagger n_b,$$

where P_a and P_b denote the orthogonal projection operators onto V_a and V_b, respectively, and $(\cdot)^\dagger$ denotes the Moore–Penrose pseudoinverse. The projection onto the visible subspace captures all information about the weight vector available at that point. Our projection-based aggregation method consists of two steps: homology verification and projection aggregation.

Step 1: Verify homology in the target layer. Homology verification requires two wraparound points to share sufficient common information, captured here by the intersection of their visible subspaces $V_{\text{int}} = V_a \cap V_b$, analogous to the common valid entries in [6]. Let P_{int} denote the orthogonal projection operator onto the common subspace V_{int}. Since $V_{\text{int}} \subseteq V_a$ and $V_{\text{int}} \subseteq V_b$, we have

$$P_{\text{int}}(\hat{w}_a) = P_{\text{int}}(P_a((W_a)^T)) = P_{\text{int}}((W_a)^T),$$

$$P_{\text{int}}(\hat{w}_b) = P_{\text{int}}(P_b((W_b)^T)) = P_{\text{int}}((W_b)^T).$$

If $W_a = W_b$, then

$$P_{\text{int}}(\hat{w}_a) = P_{\text{int}}(\hat{w}_b).$$

Table 4: Homology verification results evaluated on our CIFAR-10 network architecture with an intersection dimension threshold $t = 8$ and a collinearity constraint limit of 0.99.

Layer	Target Layer Homologous			Deeper Layer Homologous or Non-Homologous
	Correct Identification	Safe Rejection	Incorrect Judgment	Incorrect Judgment
1	100%	0%	0%	0%
2	100%	0%	0%	0%
3	100%	0%	0%	0%
4	99.994%	0.006%	0%	0%
5	100%	0%	0%	0%

If $W_a \neq W_b$, we assume that the projections of distinct weight vectors onto a random t-dimensional subspace are rarely collinear[5], analogous to requiring t common valid entries in [6]. The threshold t is chosen according to the network width and numerical precision. In the ideal error-free setting, $t = 2$ suffices to distinguish two weight directions; for our networks with widths 256 and 200, we set $t = 8$. Since the common subspace is independent of the weight vectors, we use it as the projection subspace. Let $k = \dim(V_{\text{int}})$. Under this assumption, when $k \geq t$,

$$P_{\text{int}}((W_a)^T) \nparallel P_{\text{int}}((W_b)^T),$$

and hence

$$P_{\text{int}}(\hat{w}_a) \nparallel P_{\text{int}}(\hat{w}_b).$$

The verification strategy is therefore straightforward. If $k \geq t$, the two wraparound points are considered homologous if $P_{\text{int}}(\hat{w}_a) \parallel P_{\text{int}}(\hat{w}_b)$, and non-homologous otherwise. If $k < t$, the shared information is insufficient for reliable verification. Verifying $P_{\text{int}}(\hat{w}_a) \parallel P_{\text{int}}(\hat{w}_b)$ is equivalent to checking

$$P_{\text{int}}((J_a^T)^\dagger n_a) \parallel P_{\text{int}}((J_b^T)^\dagger n_b).$$

To compute the intersection $V_{\text{int}} = V_a \cap V_b$, we use principal angles between subspaces [2]. Let Q_a and Q_b be orthonormal bases of V_a and V_b, respectively. The singular values of $Q_a^T Q_b$ satisfy $\sigma_j = \cos\theta_j$, where θ_j are the principal angles. Directions with $\theta_j = 0$ (equivalently, $\sigma_j = 1$) span the intersection V_{int}. We then construct P_{int} as the orthogonal projection onto this intersection. The detailed procedure is given in Algorithm 1.

[5] Since, for any two d-dimensional weight vectors, there always exists a $(d-1)$-dimensional subspace onto which their projections coincide, the chosen subspace must be independent of the weight vectors.

Algorithm 2 Homologous Projection Aggregation Algorithm

Input: Homologous signature projection $\hat{w}_a,\ \hat{w}_b$, visible subspaces Q_a, Q_b, intersection basis U_{int}, threshold ϵ_{svd}

Output: Aggregated neuron signature w_{agg} in the sum space

$v_a \leftarrow U_{\text{int}}^T \hat{w}_a, v_b \leftarrow U_{\text{int}}^T \hat{w}_b$

$s \leftarrow \frac{v_a^T v_b}{\|v_b\|_2^2}$

$Q_{\text{concat}} \leftarrow \begin{bmatrix} Q_a & Q_b \end{bmatrix}$

$[U, \Sigma, V^T] \leftarrow \text{SVD}(Q_{\text{concat}})$

$Q_{\text{union}} \leftarrow U_{:,\{\sigma_i > \epsilon_{\text{svd}}\}}$

$M_{\text{proj}} \leftarrow \begin{bmatrix} Q_a^T Q_{\text{union}} \\ Q_b^T Q_{\text{union}} \end{bmatrix}$

$b_{\text{proj}} \leftarrow \begin{bmatrix} Q_a^T \hat{w}_a \\ s \cdot Q_b^T \hat{w}_b \end{bmatrix}$

$y_{\text{proj}} \leftarrow M_{\text{proj}}^{\dagger} b_{\text{proj}}$ % Least squares solution in the union basis

$w_{\text{agg}} \leftarrow Q_{\text{union}} y_{\text{proj}}$

return w_{agg}

As shown in Table 4, our homology verification algorithm achieves high accuracy across all layers. It effectively prevents false positives, with a 0% misclassification rate across all tested layers and categories and no deep-layer homologous points falsely matched to homologous points in the target layer (see subsection 3.2 of [21]), mainly due to the large width of our networks. A small 0.006% safe-rejection rate occurs in Layer 4 due to the limited intersection dimension, but it does not affect the overall recovery process.

Step 2: Aggregate homologous projection. Upon confirming that two wraparound points are homologous, we aggregate their corresponding signature projections. We first align $\hat{w}_a$ and $\hat{w}_b$ using their components in the intersection subspace $V_a \cap V_b$, which determines their relative scaling factor. After alignment, we reconstruct the unified projection of the target weight W onto the sum space $V_a + V_b$. The detailed procedure is given in algorithm 2. We iteratively merge subspaces until their sum spans the entire feature space, at which point the exact neuron signature can be recovered.

4.4 Bias and Norm Recovery

Having obtained the signature $\hat{W}_j^{(i)} = c_j^{(i)} W_j^{(i)}$ of neuron $\eta_j^{(i)}$, we next recover its scaling factor and bias. For any wraparound point x, $W_j^{(i)} h^{(i-1)}(x) + b_j^{(i)} = (2k+1)T_{\max}$ for some integer k. Multiplying both sides by $c_j^{(i)}$, we observe that the values $\hat{W}_j^{(i)} h^{(i-1)}(x)$ corresponding to different k are equally spaced, with spacing $2|c_j^{(i)}|T_{\max}$. Thus, the spacing directly yields $|c_j^{(i)}|$. The bias $b_j^{(i)}$ can then be recovered and mapped back to $(-T_{\max}, T_{\max})$. Specifically,

$$b_j^{(i)} = T_{\max} - \left(\frac{\hat{W}_j^{(i)} h^{(i-1)}(x)}{|c_j^{(i)}|} \bmod 2T_{\max} \right).$$

Remark. Signature and norm recovery rely solely on the geometric properties of wraparound points in the input space and are independent of the specific output values. Since our method can localize wraparound points even in the restricted top-1 score setting, signature and norm recovery remain directly applicable.

5 Sign Recovery

Neuron Wiggle [3] is a heuristic neuron sign-recovery algorithm validated empirically through statistical testing. In Subsection 5.1, we reformulate it within an analyzable framework and extend it to the PPML setting with raw outputs. Under reasonable assumptions, we derive the probability of successful sign recovery and obtain the optimal probing direction. For simplicity, we first assume that the model has a one-dimensional output. We then demonstrate the feasibility of our algorithm in the more restrictive top-1 score setting, where only the top-1 prediction score is observable (subsection 5.2). Due to space constraints, the improved sign-recovery algorithm and its relationship to [21] are deferred to Appendix B.

5.1 Raw Outputs Sign Recovery

The Neuron Wiggle method exploits gradient divergence across critical points. We observe a similar phenomenon at wraparound boundaries: the derivatives of unbounded activation functions differ sharply across the two sides. In the interval $((2k+1)T_{\max} - \delta, (2k+1)T_{\max})$, the derivative approaches 1, whereas in $((2k+1)T_{\max}, (2k+1)T_{\max} + \delta)$, it drops sharply toward 0 (or $\alpha \ll 1$ for LeakyReLU). Below, we explain from the perspective of wraparound points why gradient divergence can be used to recover the sign.

Consider a wraparound point x^* of neuron $\eta_j^{(i)}$. The directional derivative of the network output f along an arbitrary unit vector v can be expanded using the chain rule as

$$\frac{\partial f}{\partial v} = \frac{\partial G_{i+1}}{\partial h^{(i)}} \frac{\partial h^{(i)}}{\partial z^{(i)}} \frac{\partial z^{(i)}}{\partial h^{(i-1)}} \frac{\partial h^{(i-1)}}{\partial v}.$$

Assume that the direction H is chosen such that $z_j^{(i)}(x^* + \epsilon H) < z_j^{(i)}(x^*)$. Then, the directional derivatives on the two sides of the wraparound boundary can be expressed as

$$\left.\frac{\partial f}{\partial v}\right|_{x^*+\epsilon H} = \frac{\partial G_{i+1}(x^* + \epsilon H)}{\partial h^{(i)}} \Lambda^{(i)}(x^* + \epsilon H) W^{(i)} J^{(i-1)}(x^* + \epsilon H) v,$$

$$\left.\frac{\partial f}{\partial v}\right|_{x^*-\epsilon H} = \frac{\partial G_{i+1}(x^* - \epsilon H)}{\partial h^{(i)}} \Lambda^{(i)}(x^* - \epsilon H) W^{(i)} J^{(i-1)}(x^* - \epsilon H) v.$$

Since x^* is a wraparound point only for the target neuron in layer i, the mapping through the first $i-1$ layers remains smooth in a neighborhood of x^*. Therefore, taking the limit as $\epsilon \to 0^+$, we obtain

$$\lim_{\epsilon\to 0^+} \left.\frac{\partial f}{\partial v}\right|_{x^*+\epsilon H} = y_v^+ = \begin{bmatrix} g_1^+ & g_2^+ & \cdots & g_{d_i}^+ \end{bmatrix} \begin{bmatrix} \tau_1^+ & 0 & \cdots & 0 \\ 0 & \tau_2^+ & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \tau_{d_i}^+ \end{bmatrix} W^{(i)} J^{(i-1)}(x^*) v,$$

$$\lim_{\epsilon\to 0^+} \left.\frac{\partial f}{\partial v}\right|_{x^*-\epsilon H} = y_v^- = \begin{bmatrix} g_1^- & g_2^- & \cdots & g_{d_i}^- \end{bmatrix} \begin{bmatrix} \tau_1^- & 0 & \cdots & 0 \\ 0 & \tau_2^- & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \tau_{d_i}^- \end{bmatrix} W^{(i)} J^{(i-1)}(x^*) v.$$

In the above equations, except for the target neuron j, the activation derivatives of the other neurons in the same layer remain unchanged within a sufficiently small neighborhood, i.e., $\tau_k^+ = \tau_k^-$ for $k \neq j$, which we denote by τ_k. For the target neuron, crossing the wraparound boundary causes a sharp change in its activation derivative, with $\tau_j^+ \approx 1$ and $\tau_j^- \approx 0$ (or $\alpha \ll 1$ for LeakyReLU). For simplicity, let $A = W^{(i)} J^{(i-1)}(x^*)$, where A_k denotes the k-th row of A. The directional derivatives on the two sides can then be written compactly as

$$y_v^+ = \sum_{k=1}^{d_i} g_k^+ \tau_k^+ A_k v, \qquad y_v^- = \sum_{k=1}^{d_i} g_k^- \tau_k^- A_k v.$$

Since $\tau_j^+ \gg \tau_j^-$, $|y_v^+|$ is more likely to exceed $|y_v^-|$. Thus, Neuron Wiggle can be viewed through a unified paradigm: choose a probing direction v and infer the neuron sign by comparing the magnitudes of the directional derivatives on the two sides, namely, $|y_v^+|$ and $|y_v^-|$.

From this perspective, sign recovery can be reformulated as a *signal-detection* problem, where $A_j v$ is the target signal and $A_k v$ for $k \neq j$ represents interference from the other neurons in the same layer. Our goal is to choose a probing direction v that maximizes the separation between the target signal and the interference. To increase the target signal, Neuron Wiggle chooses a probing direction that maximizes $A_j v = W_j^{(i)} J^{(i-1)}(x^*) v$, namely, $v_1 = \left(J^{(i-1)}(x^*)\right)^\dagger (W_j^{(i)})^T$. In contrast, we propose the probing direction $v_2 = A^\dagger e_j$.

We take the first layer of a contractive ReLU network as an example. In the Neuron Wiggle approach, using the probing direction $v_1 = (W_j^{(1)})^T$, the interference terms are determined by the inner products between neuron weights: $A_k v_1 = \langle W_k^{(1)}, W_j^{(1)} \rangle$. The directional derivative is $|y_{v_1}^+| = |g_j^+ \tau_j^+ \|W_j^{(1)}\|^2 + \sum_{k\neq j} g_k^+ \tau_k \langle W_k^{(1)}, W_j^{(1)} \rangle|$. In deep networks, even when neurons k and j are nearly orthogonal, i.e., $\mathbb{E}[|\langle W_k^{(1)}, W_j^{(1)} \rangle|] = \Theta(\frac{\|W_k^{(1)}\| \|W_j^{(1)}\|}{\sqrt{d_0}})$, large downstream coefficients with $|g_k^+| \gg |g_j^+|$ can cause the interference $|\sum_{k\neq j} g_k^+ \tau_k \langle W_k^{(1)}, W_j^{(1)} \rangle|$

to overwhelm the target signal, leading to sign-recovery failure. In contrast, our method uses the probing direction $v_2 = A^\dagger e_j$. Since $AA^\dagger = I$, this gives $A_j v_2 = 1$ and $A_k v_2 = 0, \forall k \neq j$. Hence, the directional derivative reduces to $y^+_{v_2} = g^+_j \tau^+_j$. If the sign is correctly identified, $|y^+_{v_2}|$ is large, whereas the opposite side satisfies $y^-_{v_2} = 0$, enabling direct sign recovery. In what follows, we provide a rigorous probabilistic analysis showing that our method achieves a higher sign-recovery probability than Neuron Wiggle under the stated assumptions.

Definition 13. *For a probing direction v, the success probability of sign recovery is defined as* $\Pr\left[|y^+_v| > |y^-_v|\right]$. *The signal-to-interference ratio (SIR) associated with v is defined as* $\mathrm{SIR}(v) = \frac{(A_j v)^2}{\sum_{k \neq j} (A_k v)^2}$.

The distribution of downstream coefficients g^+_k and g^-_k in trained neural networks is generally complex and depends on various factors, including network width and depth, activation functions, initialization schemes, architectures, and training procedures [30,40]. Nevertheless, wide-network theory provides a principled motivation for a Gaussian approximation. In particular, mean-field analyses characterize backpropagated signals as aggregated contributions from many neurons [30], while wide-network scaling limits establish Gaussian behavior and justify gradient-independence approximations under suitable conditions [44,43]. Motivated by these results, we adopt an isotropic Gaussian model for analytical tractability. Furthermore, we consider the *worst case*, in which $\tau^+_j = 1$, $\tau^-_j = \alpha$, and $\tau^+_k = \tau^-_k = 1$ for all $k \neq j$ (the average-case discussion is provided in Appendix B.2). Under these assumptions, the sign-recovery success probability admits a closed-form expression, as formalized in Theorem 5.

Theorem 5. *Assume that the downstream coefficients satisfy $g^+_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2_+)$ and $g^-_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2_-)$, and that the two families $\{g^+_k\}_{k=1}^{d_i}$ and $\{g^-_k\}_{k=1}^{d_i}$ are mutually independent. Under the worst case, the sign-recovery success probability for a probing direction v is*

$$\Pr\left[|y^+_v| > |y^-_v|\right] = \frac{2}{\pi} \arctan\left(\frac{\sigma_+}{\sigma_-}\sqrt{\frac{\mathrm{SIR}(v)+1}{\alpha^2\,\mathrm{SIR}(v)+1}}\right).$$

Observing the above analytical solution, for unbounded activation functions (where the inactive side derivative ratio is $\alpha < 1$), the success probability p is a strictly monotonically increasing function of $\mathrm{SIR}(v)$. Specifically, as $\mathrm{SIR}(v) \to \infty$, the success rate $p \to 1$. In addition, some empirical studies suggest that gradient-related quantities may exhibit non-Gaussian and heavy-tailed behavior [32,41]. Nevertheless, Theorem 6 guarantees that, even when the downstream gradient coefficients deviate from an ideal Gaussian distribution, the sign-recovery success probability remains asymptotically increasing with $\mathrm{SIR}(v)$ as the network width becomes sufficiently large.

Theorem 6. *Assume that the downstream coefficients $\{g^+_k\}_{k\neq j}$ and $\{g^-_k\}_{k\neq j}$ form two mutually independent families of i.i.d. random variables with zero*

Algorithm 3 Sign Recovery Algorithm

Input: Recovered weights matrix $\hat{W}^{(i)}$, oracle $\mathcal{O}$, parameters ϵ, δ, threshold $\mathrm{CL}_{\mathrm{target}}$
Output: **true** if the sign of the target neuron is right, otherwise **false**

votes$_+ \leftarrow 0,\ $ votes$_- \leftarrow 0,\ N \leftarrow 0,\ \mathrm{CL} \leftarrow 0$
while $\mathrm{CL} < \mathrm{CL}_{\mathrm{target}}$ **do**
  $x \leftarrow \mathrm{FindWraparoundPoint}(\eta_j^{(i)})$
  $N \leftarrow N + 1$
  $A \leftarrow \hat{W}^{(i)} J^{(i-1)}(x)$
  $v \leftarrow A^{\dagger} e_j$
  $x_{pos} \leftarrow x - \epsilon v,\ x_{neg} \leftarrow x + \epsilon v$
  $y_v^+ \leftarrow \big(\mathcal{O}(x_{pos}) - \mathcal{O}(x_{pos} - \delta v)\big)/\delta$
  $y_v^- \leftarrow \big(\mathcal{O}(x_{neg} + \delta v) - \mathcal{O}(x_{neg})\big)/\delta$
  if $|y_v^+| > |y_v^-|$: votes$_+ \leftarrow$ votes$_+ + 1$
  else : votes$_- \leftarrow$ votes$_- + 1$
  $\mathrm{CL} \leftarrow \mathrm{ComputeConfidenceLevel}(\text{votes}_+, \text{votes}_-)$
if votes$_+ >$ votes$_-$: **return true**
else : **return false**

mean and finite variances σ_+^2 *and* σ_-^2*, respectively. Suppose further that the asymptotic stability condition* $\lim_{d_i \to \infty} \frac{\max_{k \neq j} (A_k v)^2}{\sum_{k \neq j} (A_k v)^2} = 0$ *holds. Under the worst-case activation-derivative configuration with* $\alpha = 0$*, and conditioning on a fixed* $g_j^+ \neq 0$*, the asymptotic sign-recovery success probability is*

$$\lim_{d_i \to \infty} \Pr\left[|y_v^+| > |y_v^-|\right] = \Phi_2\left(\Gamma, \Gamma; \rho\right) + \Phi_2\left(-\Gamma, -\Gamma; \rho\right),$$

where Φ_2 *denotes the cumulative distribution function of a standard bivariate normal distribution,*

$$\Gamma = |g_j^+| \sqrt{\frac{\mathrm{SIR}(v)}{\sigma_+^2 + \sigma_-^2}}, \qquad \rho = \frac{\sigma_+^2 - \sigma_-^2}{\sigma_+^2 + \sigma_-^2}.$$

Moreover, the asymptotic success probability is strictly increasing in $\mathrm{SIR}(v)$*.*

In summary, $\mathrm{SIR}(v)$ quantifies the strength of the target-neuron signal relative to the aggregate interference from the other neurons. Therefore, our goal is to choose a probing direction v that maximizes $\mathrm{SIR}(v)$. Theorem 7 shows that our probing direction $v = A^{\dagger} e_j$ achieves the optimal SIR.

Theorem 7. *The signal-to-interference ratio* $\mathrm{SIR}(v)$ *is maximized by the probing direction* $v = A^{\dagger} e_j$*.*

Based on the above analysis, the active side is more likely to yield a larger directional-derivative magnitude than the inactive side. Therefore, we adopt a statistical voting approach similar to those used in [3,5] to determine the neuron sign. Unlike prior approaches, our method terminates early once the *confidence level* for one sign, as introduced in Subsection 6.2, reaches a predefined threshold. The detailed procedure is given in Algorithm 3.

Compare with critical points. Because wraparound points introduce a discontinuous jump in the activation output, the activation states on the two sides are effectively decoupled, leading to uncorrelated downstream coefficients. Therefore, we assume that g_k^+ and g_k^- are independent. In contrast, critical points correspond to continuous transitions, where the downstream activation states remain identical and hence $g_k^+ = g_k^-$. According to Theorem 9, we obtain a similar closed-form expression, which remains monotonically increasing in $\mathrm{SIR}(v)$. Therefore, the same SIR-based conclusion holds; see Appendix A for details. Note that, since the downstream coefficients are identical, the optimization in Appendix B.1 is unnecessary, and only the optimization in Appendix B.2 is required.

5.2 Top-1 Score Sign Recovery

The sign-recovery phase relies heavily on the directional derivative of the output function along a probing direction v. When the oracle reveals only the top-1 label and its corresponding probability, we can observe only $\frac{\partial p_{\mathrm{top}}}{\partial v}$ rather than $\frac{\partial f}{\partial v}$, i.e., the directional derivative of the raw outputs. Therefore, it is necessary to analyze $\frac{\partial p_{\mathrm{top}}}{\partial v}$ on both sides of the wraparound boundary.

We define the limiting probability on the active and inactive sides of the wraparound boundary as $p^+ = [p_1^+, p_2^+, \ldots, p_{d_{r+1}}^+]^T$ and $p^- = [p_1^-, p_2^-, \ldots, p_{d_{r+1}}^-]^T$, respectively. The limiting directional derivatives of the top-1 probability on the active and inactive sides are denoted by

$$p_v^+ = \lim_{\epsilon \to 0^+} \left. \frac{\partial p_{\mathrm{top}^+}}{\partial v} \right|_{x^* + \epsilon H}, \qquad p_v^- = \lim_{\epsilon \to 0^+} \left. \frac{\partial p_{\mathrm{top}^-}}{\partial v} \right|_{x^* - \epsilon H}.$$

Definition 14. *For a probing direction v, the success probability of sign recovery in the top-1 score setting is defined as* $\Pr[|p_v^+| > |p_v^-|]$.

Theorem 8. *Under the assumptions of Theorem 5, the sign-recovery success probability in the top-1 score setting is*

$$\Pr[|p_v^+| > |p_v^-|] = \frac{2}{\pi} \arctan\left(\frac{\sigma_+ p_{\mathrm{top}^+}^+}{\sigma_- p_{\mathrm{top}^-}^-} \sqrt{\frac{\sum_{l \neq \mathrm{top}^+} (p_l^+)^2}{\sum_{l \neq \mathrm{top}^-} (p_l^-)^2} \cdot \frac{\mathrm{SIR}(v) + 1}{\alpha^2 \, \mathrm{SIR}(v) + 1}} \right).$$

Building on Theorem 8, we show that our sign-recovery method extends to the more restrictive top-1 score setting by comparing the directional derivatives of the corresponding top-1 probabilities.

Remark. The output probability vectors p^+ and p^- on the two sides of the wraparound boundary are generally different, which may introduce additional uncertainty into sign recovery. However, the improved method presented in Appendix B.1 effectively mitigates this issue.

6 Experiments

The experimental setup is designed to emulate an MPC environment over a finite ring with modulus $M = 2^{64}$ and a fixed-point scaling factor of $S = 26$ [38,45]. To accelerate the evaluation of the MPC protocols, we simulate all MPC computations in plaintext. Furthermore, we evaluate the layer-wise attack under the assumption that all preceding layers have already been perfectly extracted [6].

We evaluate our attack on two types of networks: CIFAR-10 networks and randomly generated expansive networks. The CIFAR-10 models are fully connected neural networks with an architecture of 3072-256 $\times$ 4-10, while the expansive networks use an architecture of 100-200 $\times$ 4-10. To comprehensively evaluate the generalizability of our attack, we consider the eight unbounded activation functions introduced previously for the hidden neurons: ReLU, LeakyReLU, ELU, SELU, Softplus, Swish, GELU, and Mish. The models are trained following a methodology consistent with established baselines [3,5].

6.1 Signature Recovery

Activation Functions. The eight activation functions evaluated fall into three categories: ReLU, LeakyReLU, and non-linear activations. As shown in Table 5, the number of wraparound points required for signature recovery depends on both the layer depth and the activation function. For the first layer, a single wraparound point is sufficient for all activation functions. In CIFAR-10 networks, LeakyReLU still requires only one point in shallow layers because the corresponding Jacobian maintains full rank. However, in deeper layers (layers 4 and 5), the rapidly growing condition number makes the Jacobian increasingly ill-conditioned, leading to numerical precision issues and requiring two wraparound points for signature recovery. In contrast, ReLU and nonlinear activations require an average of 9 to 20 wraparound points for successful deep-layer recovery.

Expansive Architectures. As shown in Table 5, for expansive networks, we successfully recover the signatures of all layers in networks using ReLU and other

Table 5: Wraparound points required for signature recovery across different activation functions on our CIFAR10 and expansive networks.

Layer	ReLU		LeakyReLU ($\alpha = 0.1$)		Non-Linear	
	CIFAR10	Expansive	CIFAR10	Expansive	CIFAR10	Expansive
1	1.00	1.00	1.00	1.00	1.00 ± 0.00	1.00 ± 0.00
2	9.48	10.17	1.00	3.25	9.45 ± 0.12	10.04 ± 0.15
3	19.43	28.06	1.00	-	20.52 ± 3.01	26.16 ± 3.38
4	9.57	14.76	2.00	-	11.25 ± 1.70	14.36 ± 1.07
5	9.80	11.00	2.00	-	11.98 ± 1.47	9.40 ± 0.40

nonlinear activation functions, with complete recovery of deep-layer parameters typically requiring 10 to 28 wraparound points. However, for LeakyReLU networks, our method is effective only for recovering the parameters of the first two layers. This behavior is consistent with the limitation analyzed in [21], except that our method can additionally recover the second layer. This limitation fundamentally stems from the fact that LeakyReLU has a nonzero derivative over its entire domain. In layers 3 to 5, after dense matrix multiplications, homologous wraparound points sampled at different locations in the input space induce substantially different sets of active background neurons. This strong position dependence introduces significant interference into their respective visible subspaces, causing the subspaces to become nearly unrelated. Since the intersection dimension of generic m- and n-dimensional subspaces in a d-dimensional ambient space is $\max\{0, m+n-d\}$, the intersection of two 100-dimensional visible subspaces in a 200-dimensional space can rapidly degenerate to the zero subspace under LeakyReLU, making signature recovery ineffective. However, this does not imply that our method fails for all expansive LeakyReLU networks. For networks satisfying $\forall i > 1,\ d_i < 2\min_{k<i}\{d_k\}$ i.e., those with an expansion factor below 2, our method remains applicable. For networks with an expansion factor of at least 2, a similar issue may arise even for ReLU in sufficiently deep layers, where the visible subspaces can also become nearly unrelated. Handling such cases requires multi-point aggregation. However, a naive multi-point aggregation strategy has complexity $O(N^m)$, where N is the number of points and m is the number of aggregated points, and we have not yet evaluated this approach experimentally.

Extraction Fidelity. As shown in Figure 3, extraction precision depends on the activation function, with consistent trends across both networks. ReLU and LeakyReLU achieve about 14 bits of precision in both settings. In contrast, nonlinear activations have nonzero curvature, which introduces systematic truncation errors in local Jacobian and normal-vector estimation, limiting extraction precision to about 7–10 bits.

It is worth noting that, in ReLU networks, both always-active and dead neurons can severely hinder parameter recovery [15,21]. However, in the PPML setting, both types remain recoverable as long as they wraparound. Conversely, non-wraparound neurons can hinder recovery, depending on the activation function and weight norm. For bounded activations such as sigmoid and tanh, all neurons beyond the first layer are non-wraparound and therefore unrecoverable. A neuron is also non-wraparound if its weight norm is below $\frac{1}{\sqrt{d}}$, where d is the network width. Nevertheless, we observed no non-wraparound neurons in either our normally trained networks or randomly initialized networks.

6.2 Sign Recovery

Confidence Level. To statistically validate the recovered signs, we compute a confidence level (CL) under a binomial model. Suppose a neuron is tested at

N wraparound points and the majority sign receives $M = \max(\text{votes}_+, \text{votes}_-)$ votes. Under the null hypothesis that each vote is an independent random guess, we define $\text{CL} = 1 - \Pr[X \geq M]$, where $X \sim \text{Binomial}(N, 0.5)$. For efficient evaluation we switch methods according to N. When $N < 10$, we obtain the exact CL from a precomputed table of binomial tail probabilities. When $N \geq 10$, Hoeffding's inequality [14] supplies a cheap and conservative lower bound $\text{CL} \geq 1 - \exp\left(-\frac{2(M-N/2)^2}{N}\right)$. A sign is declared once the CL exceeds a preset threshold (e.g., 95%). If every vote agrees, $N = 5$ already yields a 95% CL. If the empirical majority rate is close to chance (e.g., $M/N = 0.55$), the same threshold requires a much larger sample, at least $N = 600$.

Activation Functions. Our sign-recovery method achieves a 100% success rate across all recoverable layers of the evaluated CIFAR-10 and expansive networks. As shown in Table 6, sign recovery for CIFAR-10 networks with LeakyReLU generally requires only about 5 wraparound points. The only exception is the fourth layer, where an ill-conditioned Jacobian requires more points. For deeper layers with non-LeakyReLU activations, we observe an interesting trend: sign recovery becomes more efficient as the network depth increases. In expansive networks, however, an average of five wraparound points is insufficient for sign recovery.

Comparison with Neuron Wiggle. Table 7 compares our method with Neuron Wiggle on CIFAR-10 networks. For the hidden layers, our method correctly recovers the sign of every neuron, with all neurons reaching a 95% confidence level within 1,000 queries. Under the same query budget, Neuron Wiggle fails to reach this threshold for several neurons. More importantly, even among neurons that reach the confidence threshold, Neuron Wiggle still produces 4 to 29 incorrect signs, which can derail subsequent deep-layer parameter recovery. Moreover, our method requires fewer than 7 wraparound points per neuron on average, whereas Neuron Wiggle requires more than 24 for ReLU neurons. Overall, our method reduces the query complexity by 30%–74%.

Table 6: Wraparound points required for sign recovery across different activation functions on our CIFAR10 and expansive networks.

Layer	ReLU		LeakyReLU ($\alpha = 0.1$)		Non-Linear	
	CIFAR10	Expansive	CIFAR10	Expansive	CIFAR10	Expansive
1	5.00	6.97	5.00	6.93	5.00 ± 0.00	6.85 ± 0.36
2	11.92	6.54	5.00	5.83	5.27 ± 0.10	6.43 ± 0.17
3	9.51	6.13	5.02	5.53	5.65 ± 0.16	5.89 ± 0.12
4	5.42	6.53	9.51	5.43	5.12 ± 0.09	6.13 ± 0.32

Table 7: Comparison of sign-recovery results on CIFAR-10 networks with different activation functions. Each network contains 1,024 hidden neurons, with a maximum of 1,000 attempts per neuron and a target confidence level of 0.95.

Activation	**Method**	$\boldsymbol{\alpha \geq 0.95}$		$\boldsymbol{\alpha < 0.95}$		**Average Points**
		Correct	**Incorrect**	**Correct**	**Incorrect**	
ReLU	Neuron Wiggle	988/1024	26/1024	7/1024	3/1024	24.82
	Ours	**1024/1024**	**0/1024**	**0/1024**	**0/1024**	**6.61** (**0.26**×)
LeakyReLU	Neuron Wiggle	1020/1024	4/1024	0/1024	0/1024	7.25
	Ours	**1024/1024**	**0/1024**	**0/1024**	**0/1024**	**5.09** (**0.70**×)
Non-Linear	Neuron Wiggle	1016/1024	5/1024	1/1024	2/1024	14.58
	Ours	**1024/1024**	**0/1024**	**0/1024**	**0/1024**	**5.22** (**0.36**×)

Top-1 Score Scenario. In the restricted top-1 setting, observing only the maximum softmax probability substantially reduces the stability of sign recovery. To restore robustness, we require that at least 20 wraparound points be evaluated before a sign is decided. As shown in Table 8, this configuration recovers the correct signs of all 14,592 neurons across two architectures and eight activation functions. Five neurons initially fail to reach the target 95% confidence level within the allotted attempts; analyzing these neurons separately subsequently brings all of them above the threshold.

7 Conclusion and Remarks

In this paper, we address three key problems. First, we identify silent modular wraparound, an inherent phenomenon in MPC-based PPML, and exploit it to recover all parameters of networks with smooth activations, which prior methods cannot achieve. Second, we divide signature recovery into single-point and multi-point cases, where the projection-based aggregation in the latter eliminates the limitation of expansive architectures. Third, we recast Neuron Wiggle as an analyzable paradigm and, under mild assumptions, derive the optimal probing direction, improving sign-recovery accuracy and reducing the number of queries. However, our method depends on modular wraparound and therefore cannot extract networks with bounded activations such as Sigmoid and Tanh, whose outputs rarely leave the representable range. Because the recovery of smooth activations still relies on the PPML arithmetic model, developing a general attack that efficiently recovers all parameters of such networks in the standard real-valued setting remains an important direction for future work.

References

1. Asselineau, R., Derbez, P., Fouque, P., Minaud, B.: Cryptanalytic extraction of deep neural networks with non-linear activations. In: Heninger, N., Rosulek, M. (eds.) CRYPTO 2026, Part VII. LNCS, vol. 16806, pp. 36–66. Springer, Cham (Aug 2026). https://doi.org/10.1007/978-3-032-35415-0_2
2. Björck, Å., Golub, G.H.: Numerical methods for computing angles between linear subspaces. Math. Comp. **27**(123), 579–594 (Jul 1973). https://doi.org/10.1090/S0025-5718-1973-0348991-3
3. Canales-Martínez, I.A., Chávez-Saab, J., Hambitzer, A., Rodríguez-Henríquez, F., Satpute, N., Shamir, A.: Polynomial time cryptanalytic extraction of neural network models. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part III. LNCS, vol. 14653, pp. 3–33. Springer, Cham (May 2024). https://doi.org/10.1007/978-3-031-58734-4_1
4. Canales-Martínez, I.A., Santos, D.: Extracting some layers of deep neural networks in the hard-label setting. In: Escudero, D., Damgård, I. (eds.) LATINCRYPT 2025. LNCS, vol. 16129, pp. 399–421. Springer, Cham (Oct 2025). https://doi.org/10.1007/978-3-032-06754-8_15
5. Carlini, N., Chávez-Saab, J., Hambitzer, A., Rodríguez-Henríquez, F., Shamir, A.: Polynomial time cryptanalytic extraction of deep neural networks in the hard-label setting. In: Fehr, S., Fouque, P. (eds.) EUROCRYPT 2025, Part I. LNCS, vol. 15601, pp. 364–396. Springer, Cham (May 2025). https://doi.org/10.1007/978-3-031-91107-1_13
6. Carlini, N., Jagielski, M., Mironov, I.: Cryptanalytic extraction of neural network models. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part III. LNCS, vol. 12172, pp. 189–218. Springer, Cham (Aug 2020). https://doi.org/10.1007/978-3-030-56877-1_7
7. Chen, Y., Dong, X., Guo, J., Shen, Y., Wang, A., Wang, X.: Hard-label cryptanalytic extraction of neural network models. In: Chung, K., Sasaki, Y. (eds.) ASIACRYPT 2024, Part VIII. LNCS, vol. 15491, pp. 207–236. Springer, Singapore (Dec 2024). https://doi.org/10.1007/978-981-96-0944-4_7
8. Chen, Y., Dong, X., Ma, R., Shen, Y., Wang, A., Yu, H., Wang, X.: Delving into cryptanalytic extraction of PReLU neural networks. In: Hanaoka, G., Yang, B. (eds.) ASIACRYPT 2025, Part II. LNCS, vol. 16246, pp. 576–607. Springer, Singapore (Dec 2025). https://doi.org/10.1007/978-981-95-5096-8_18
9. Chen, Z., Tang, S., Gao, Z., Su, Y., Qin, L., Dong, X.: Algebraic attack on convolutional neural networks with max pooling. In: Heninger, N., Rosulek, M. (eds.) CRYPTO 2026, Part VII. LNCS, vol. 16806, pp. 3–35. Springer, Cham (Aug 2026). https://doi.org/10.1007/978-3-032-35415-0_1
10. Chen, Z., Tang, S., Gao, Z., Su, Y., Qin, L., Dong, X.: Algebraic cryptanalytic extraction on hard-label neural networks. Cryptology ePrint Archive, Paper 2026/1164 (2026), https://eprint.iacr.org/2026/1164
11. Damgård, I., Pastro, V., Smart, N., Zakarias, S.: Multiparty computation from somewhat homomorphic encryption. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 643–662. Springer, Heidelberg (Aug 2012). https://doi.org/10.1007/978-3-642-32009-5_38
12. Fefferman, C.: Reconstructing a neural net from its output. Rev. Mat. Iberoam. **10**(3), 507–555 (1994). https://doi.org/10.4171/rmi/160
13. Foerster, H., Mullins, R., Shumailov, I., Hayes, J.: Beyond slow signs in high-fidelity model extraction. In: Globerson, A., et al. (eds.) NeurIPS 2024. pp. 19496–19522. Curran Associates (Dec 2024). https://doi.org/10.52202/079017-0615

14. Hoeffding, W.: Probability inequalities for sums of bounded random variables. J. Am. Stat. Assoc. **58**(301), 13–30 (Mar 1963). https://doi.org/10.1080/01621459.1963.10500830
15. Ito, A., Miura, T., Todo, Y.: Is the hard-label cryptanalytic model extraction really polynomial? In: Heninger, N., Rosulek, M. (eds.) CRYPTO 2026, Part VII. LNCS, vol. 16806, pp. 67–98. Springer, Cham (Aug 2026). https://doi.org/10.1007/978-3-032-35415-0_3
16. Lehmkuhl, R., Mishra, P., Srinivasan, A., Popa, R.A.: Muse: Secure inference resilient to malicious clients. In: USENIX Security 2021. pp. 2201–2218. USENIX Association (Aug 2021), https://www.usenix.org/conference/usenixsecurity21/presentation/lehmkuhl
17. Li, C., Gong, Z., Li, D., Zhuang, L., Tang, Y., Lv, Y., Yan, X.: End-to-end polynomial-time cryptanalytic extraction of convolutional neural networks in the hard-label setting. Cryptology ePrint Archive, Paper 2026/902 (May 2026), https://eprint.iacr.org/2026/902
18. Li, C., Zhuang, L., Li, D., Tang, Y., Gong, Z.: Polynomial-time cryptanalytic extraction of graph neural networks in the hard-label setting. Cryptology ePrint Archive, Paper 2026/719 (Apr 2026), https://eprint.iacr.org/2026/719
19. Lin, V.Y., Pinkus, A.: Fundamentality of ridge functions. J. Approx. Theory **75**(3), 295–311 (Dec 1993). https://doi.org/10.1006/jath.1993.1106
20. Liu, H., Siproudhis, A., Boura, C., Peyrin, T.: Model extraction of convolutional neural networks with max-pooling. IACR Trans. Symmetric Cryptol. (2026), https://eprint.iacr.org/2026/464
21. Liu, H., Siproudhis, A., Experton, S., Lorenz, P., Boura, C., Peyrin, T.: Navigating the deep: End-to-end extraction on deep neural networks. In: Daemen, J., Thomé, E. (eds.) EUROCRYPT 2026, Part VI. LNCS, vol. 16546, pp. 482–512. Springer, Cham (May 2026). https://doi.org/10.1007/978-3-032-25333-0_17
22. Liu, J., Juuti, M., Lu, Y., Asokan, N.: Oblivious neural network predictions via MiniONN transformations. In: ACM CCS 2017. pp. 619–631. ACM (Oct 2017). https://doi.org/10.1145/3133956.3134056
23. Lowd, D., Meek, C.: Adversarial learning. In: KDD 2005. pp. 641–647. ACM (Aug 2005). https://doi.org/10.1145/1081870.1081950
24. Ma, R., Chen, Y., Zhang, J., Yu, H., Wang, X.: Breaking slope and structure restrictions: Broadening hard-label cryptanalytic extraction of PReLU neural networks. Cryptology ePrint Archive, Paper 2026/1066 (2026), https://eprint.iacr.org/2026/1066
25. Mohassel, P., Rindal, P.: ABY3: A mixed protocol framework for machine learning. In: ACM CCS 2018. pp. 35–52. ACM (Oct 2018). https://doi.org/10.1145/3243734.3243760
26. Mohassel, P., Zhang, Y.: SecureML: A system for scalable privacy-preserving machine learning. In: IEEE S&P 2017. pp. 19–38. IEEE (May 2017). https://doi.org/10.1109/SP.2017.12
27. Qi, X., Lei, H., Wei, L., Sun, X., Wang, M.: Cryptanalytic extraction of neural networks with various activation functions. IACR Trans. Symmetric Cryptol. **2026**(1), 468–505 (Mar 2026). https://doi.org/10.46586/tosc.v2026.i1.468-505
28. Rathee, D., Rathee, M., Goli, R.K.K., Gupta, D., Sharma, R., Chandran, N., Rastogi, A.: SiRNN: A math library for secure RNN inference. In: IEEE S&P 2021. pp. 1003–1020. IEEE (May 2021). https://doi.org/10.1109/SP40001.2021.00070
29. Rathee, D., Rathee, M., Kumar, N., Chandran, N., Gupta, D., Rastogi, A., Sharma, R.: CrypTFlow2: Practical 2-party secure inference. In: ACM CCS 2020. pp. 325–342. ACM (Nov 2020). https://doi.org/10.1145/3372297.3417274

30. Schoenholz, S.S., Gilmer, J., Ganguli, S., Sohl-Dickstein, J.: Deep information propagation. In: ICLR 2017. OpenReview.net (Apr 2017). https://doi.org/10.48550/arXiv.1611.01232
31. Shen, Y., Chen, Y., Wang, A., Yu, H., Wang, X.: Cryptanalytic extraction of neural networks without known architecture assumption. arXiv preprint arXiv:2609.14379 (Sep 2026), https://arxiv.org/abs/2609.14379
32. Simsekli, U., Sagun, L., Gurbuzbalaban, M.: A tail-index analysis of stochastic gradient noise in deep neural networks. In: Chaudhuri, K., Salakhutdinov, R. (eds.) ICML 2019. Proceedings of Machine Learning Research, vol. 97, pp. 5827–5837. PMLR (Jun 2019). https://doi.org/10.48550/arXiv.1901.06053
33. Sun, X., Lei, H., Wei, L., Qi, X., Hu, K., Wang, M., Wang, W.: Cryptanalytic extraction of convolutional neural networks. In: ACISP 2026. LNCS, Springer (Jul 2026), https://eprint.iacr.org/2026/139
34. Tan, S., Knott, B., Tian, Y., Wu, D.J.: CryptGPU: Fast privacy-preserving machine learning on the GPU. In: IEEE S&P 2021. pp. 1021–1038. IEEE (May 2021). https://doi.org/10.1109/SP40001.2021.00086
35. Wagh, S., Gupta, D., Chandran, N.: SecureNN: 3-party secure computation for neural network training. Proc. Priv. Enhancing Technol. **2019**(3), 26–49 (Jul 2019). https://doi.org/10.2478/popets-2019-0026
36. Wagh, S., Tople, S., Benhamouda, F., Kushilevitz, E., Mittal, P., Rabin, T.: Falcon: Honest-majority maliciously secure framework for private deep learning. Proc. Priv. Enhancing Technol. **2021**(1), 188–208 (Jan 2021). https://doi.org/10.2478/popets-2021-0009
37. Wang, S., Lei, H., Wei, L., Liu, Q., Hu, K., Wang, M.: Cryptanalytic extraction of multi-head softmax attention models. Cryptology ePrint Archive, Paper 2026/1678 (Aug 2026), https://eprint.iacr.org/2026/1678
38. Watson, J.L., Wagh, S., Popa, R.A.: Piranha: A GPU platform for secure computation. In: USENIX Security 2022. pp. 827–844. USENIX Association (Aug 2022), https://www.usenix.org/conference/usenixsecurity22/presentation/watson
39. Wei, L., Lei, H., Qi, X., Sun, X., Gao, L., Hu, K., Wang, W., Wang, M.: Cryptanalytic extraction of recurrent neural network models. Cryptology ePrint Archive, Paper 2026/168 (Feb 2026), https://eprint.iacr.org/2026/168
40. Xiao, L., Bahri, Y., Sohl-Dickstein, J., Schoenholz, S., Pennington, J.: Dynamical isometry and a mean field theory of CNNs: How to train 10,000-layer vanilla convolutional neural networks. In: Dy, J., Krause, A. (eds.) ICML 2018. Proceedings of Machine Learning Research, vol. 80, pp. 5393–5402. PMLR (Jul 2018). https://doi.org/10.48550/arXiv.1806.05393
41. Xie, Z., Tang, Q.Y., Sun, M., Li, P.: On the overlooked structure of stochastic gradients. In: NeurIPS 2023. pp. 66257–66276. Curran Associates (Dec 2023). https://doi.org/10.52202/075280-2892
42. Xu, D., Zhang, L., Wang, Z.: Finite-precision error analysis of cryptanalytic model extraction. Cryptology ePrint Archive, Paper 2026/1943 (2026), https://eprint.iacr.org/2026/1943
43. Yang, G.: Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. arXiv preprint arXiv:1902.04760 (2019). https://doi.org/10.48550/arXiv.1902.04760
44. Yang, G.: Wide feedforward or recurrent neural networks of any architecture are gaussian processes. In: NeurIPS 2019. pp. 9947–9960. Curran Associates (Dec 2019). https://doi.org/10.48550/arXiv.1910.12478

45. Zhou, L., Zhang, B., Wang, Z., Lu, T., Song, Q., Zhang, S., Cui, H., Yu, Y.: On probabilistic truncation in privacy-preserving machine learning. In: AAAI 2025. vol. 39, pp. 22955–22964. AAAI Press (Feb 2025). https://doi.org/10.1609/aaai.v39i21.34454

A Analysis of Sign Recovery for Critical Points

Unlike wraparound points, where the downstream coefficients g_k^+ and g_k^- on the two sides are modeled as independent, critical points yield identical downstream coefficients, i.e., $g_k^+ = g_k^-$. According to Theorem 9, under the Gaussian assumption, the success probability admits a similar closed-form expression.

Theorem 9. *Assume that the downstream coefficients satisfy* $g_k^+ = g_k^- \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$. *Under the worst case, the sign-recovery success probability for a probing direction* v *is*

$$\Pr[|y_v^+| > |y_v^-|] = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{1+\alpha}{2}\sqrt{\mathrm{SIR}(v)}\right).$$

This probability is also monotonically increasing in the SIR, implying that the optimal probing direction remains unchanged. The difference between the two cases lies in their success probabilities. For example, when $\alpha = 0$, $\mathrm{SIR} = 1$, and $\sigma_+ = \sigma_-$, the success probability is approximately 0.648 for critical points, compared with 0.608 for wraparound points. However, the two types of points exhibit the same asymptotic convergence rate with respect to $\mathrm{SIR}(v)$: as $\mathrm{SIR}(v) \to \infty$, the success probability approaches $1 - \Theta(\mathrm{SIR}(v)^{-1/2})$.

B Improved Sign Recovery Algorithm

The improved sign-recovery algorithm consists of two components: selecting a noise direction for normalization (subsection B.1) and exploiting additional available information to construct an improved probing direction (subsection B.2). The detailed algorithm is presented in Algorithm 4.

B.1 Improvement via Utilizing Noise Direction

In the previous analysis, directly comparing the derivatives $|y_v^+|$ and $|y_v^-|$ across the wraparound boundary can be unreliable. When the input crosses a neuron's wraparound boundary, the downstream coefficients on the two sides may become substantially different ($g_k^+ \neq g_k^-$). When the variance on the positive side is much smaller than that on the negative side, i.e., $\sigma_+^2 \ll \sigma_-^2$, the success probability decreases significantly according to Theorem 5. This issue becomes more pronounced in the top-1 score setting. Since the output probability vectors on the two sides are different ($p^+ \neq p^-$), they introduce additional imbalance into the effective gradient coefficients.

To mitigate these effects, we propose a more robust strategy that avoids directly comparing the two sides. Instead, we normalize the target signal using a noise-only measurement obtained from the same side. Specifically, we retain the probing direction v and introduce an additional noise direction u:

- **Probing direction v:** We use the optimal direction $v = A^\dagger e_j$, which isolates the target neuron response while eliminating interference from other neurons.
- **Noise direction u:** This direction eliminates the target-neuron contribution, i.e., $A_j u = 0$.

Under the worst case, the directional derivatives on the positive side are

$$y_v^+ = g_j^+(A_j v) + \sum_{k \neq j} g_k^+(A_k v),$$

$$y_u^+ = g_j^+(A_j u) + \sum_{k \neq j} g_k^+(A_k u) = \sum_{k \neq j} g_k^+(A_k u).$$

Similarly, we obtain y_v^- and y_u^- on the negative side. Since the noise direction eliminates the target-neuron contribution, y_u^+ and y_u^- only capture background interference, providing a reliable estimate of the noise level on each side. We then define the signal-to-noise ratios

$$R^+ = \frac{|y_v^+|}{|y_u^+|}, \qquad R^- = \frac{|y_v^-|}{|y_u^-|}.$$

If our sign guess is correct, the target neuron is active. Since the probing direction v focuses on the target neuron ($A_j v \approx 1$) and suppresses interference, the target contribution dominates the numerator, making R^+ significantly larger than one. If our sign guess is incorrect, the target neuron is inactive. For small

α, the target contribution is suppressed, and both numerator and denominator are dominated by background interference. Therefore, $R^- \approx 1$.

Definition 15. *For a probing direction v with an auxiliary noise direction u, the success probability of noise-normalized sign recovery is defined as* $\Pr[R^+ > R^-]$.

Theorem 10. *Under the assumptions of Theorem 5, the sign-recovery success probability with noise normalization is*

$$\Pr[R^+ > R^-] = 1 - \frac{4}{\pi^2} \int_0^\infty \frac{\arctan(kt)}{t^2+1} dt,$$

where $k = \sqrt{\frac{\alpha^2 \operatorname{SIR}(v)+1}{\operatorname{SIR}(v)+1}}$ *is a constant determined by α and* $\operatorname{SIR}(v)$.

Theorem 10 shows that the success probability is independent of the unbalanced variances σ_+ and σ_-. Therefore, noise normalization effectively eliminates the bias caused by side-dependent scaling. This mitigates the errors caused by both downstream coefficient shifts ($\sigma_+ \ll \sigma_-$) and asymmetric top-1 probabilities ($p^+ \neq p^-$), enabling reliable sign recovery under restrictive output settings.

B.2 Improvement via Refined Probe Direction

In the previous analysis, we considered the worst-case scenario where all background neurons are fully active ($\tau_k = 1$ for $k \neq j$). However, in practice, we have additional information about the statistical behavior of τ_k. Therefore, we further consider the average-case setting.

During layer-wise recovery, some neuron signs in the target layer have already been recovered. Let I denote the set of neurons with known signs, and let I' denote the set of neurons with unknown signs (excluding the target neuron j). The distribution of τ_k is modeled as

$$\tau_k = \begin{cases} \sigma'(z_k^{(i)}(x^*)), & k \in I, \\ \sigma'(\epsilon_k z_k^{(i)}(x^*)), & k \in I', \end{cases}$$

where $\epsilon_k \in \{-1, 1\}$ with $\Pr(\epsilon_k = 1) = \Pr(\epsilon_k = -1) = 1/2$.

Based on this distribution, the expected value $w_k = \mathbb{E}[\tau_k^2]$ can be computed as:

$$w_k = \mathbb{E}[\tau_k^2] = \begin{cases} (\sigma'(z_k^{(i)}(x^*)))^2, & k \in I, \\ \frac{1}{2}(\sigma'(z_k^{(i)}(x^*)))^2 + \frac{1}{2}(\sigma'(-z_k^{(i)}(x^*)))^2, & k \in I'. \end{cases}$$

Theorem 11. *Assume that the conditions of Theorem 5 hold, and that the target neuron satisfies $\tau_j^+ = 1$ and $\tau_j^- = \alpha$. Furthermore, suppose that the following asymptotic stability condition holds:* $\lim_{d_i \to \infty} \frac{\sum_{k \neq j} (A_k v)^4 \operatorname{Var}(\tau_k^2)}{\left(\sum_{k \neq j} (A_k v)^2 \mathbb{E}[\tau_k^2]\right)^2} = 0$. *Then the asymptotic sign-recovery success probability is*

$$\lim_{d_i \to \infty} \Pr\left[|y_v^+| > |y_v^-|\right] = \frac{2}{\pi} \arctan\left(\frac{\sigma_+}{\sigma_-} \sqrt{\frac{(A_j v)^2 + \sum_{k \neq j} w_k (A_k v)^2}{\alpha^2 (A_j v)^2 + \sum_{k \neq j} w_k (A_k v)^2}} \right).$$

Furthermore, the success probability is maximized by the probing direction $v = \tilde{A}^{\dagger} e_j$, *where the rows of* $\tilde{A}$ *are defined as* $\tilde{A}_j = A_j$ *and* $\tilde{A}_k = \sqrt{w_k} A_k, \quad k \neq j$.

Theorem 11 derives an optimized probing direction $v = (A')^{\dagger} e_j$. The use of the refined direction shares a similar motivation with the SOE+Neuron Wiggle method in [21]. The SOE+Neuron Wiggle method uses Neuron Wiggle to estimate confidence and selects high-confidence wraparound points until the remaining signs can be recovered by solving a linear system. In contrast, although our refined method does not explicitly select I by ranking confidence scores, it can also directly recover the signs when I contains as many points as those selected by SOE+Neuron Wiggle. In this case, the inactive neurons in I provide sufficient constraints to eliminate all interference, resulting in $\mathrm{SIR}(v) = \infty$. Specifically, this condition depends on the number of inactive neurons in I, denoted by m; it suffices that $m \geq d_i - \mathrm{rank}(A)$.

Algorithm 4 Improved Sign Recovery Algorithm

Input: Recovered weights matrix $\hat{W}^{(i)}$, oracle $\mathcal{O}$, recovered neurons I and unrecovered neurons I', parameters ϵ, δ, threshold $\mathrm{CL}_{\mathrm{target}}$
Output: **true** if the sign of target neuron is right, otherwise **false**

$\mathrm{votes}_+ \leftarrow 0,\ \mathrm{votes}_- \leftarrow 0,\ N \leftarrow 0,\ \mathrm{CL} \leftarrow 0$
while $\mathrm{CL} < \mathrm{CL}_{\mathrm{target}}$ **do**
  $x \leftarrow \mathrm{FindWraparoundPoint}(\eta_j^{(i)})$
  $N \leftarrow N + 1$
  $A \leftarrow \hat{W}^{(i)} J^{(i-1)}(x)$
  for all $k \in I \cup I'$ **do**
    if $k \in I : w_k \leftarrow (\sigma'(z_k^{(i)}(x)))^2$
    else : $w_k \leftarrow ((\sigma'(z_k^{(i)}(x)))^2 + (\sigma'(-z_k^{(i)}(x)))^2)/2$
    $A'_k \leftarrow \sqrt{w_k} A_k$
  $A'_j \leftarrow A_j$
  $v \leftarrow (A')^{\dagger} e_j$
  $u \xleftarrow{\$} \{u \mid A_j u = 0\}$
  $x_{pos} \leftarrow x - \epsilon v, \quad x_{neg} \leftarrow x + \epsilon v$
  $y_v^+ \leftarrow (\mathcal{O}(x_{pos} + \delta v) - \mathcal{O}(x_{pos}))/\delta, \quad y_u^+ \leftarrow (\mathcal{O}(x_{pos} + \delta u) - \mathcal{O}(x_{pos}))/\delta$
  $y_v^- \leftarrow (\mathcal{O}(x_{neg} + \delta v) - \mathcal{O}(x_{neg}))/\delta, \quad y_u^- \leftarrow (\mathcal{O}(x_{neg} + \delta u) - \mathcal{O}(x_{neg}))/\delta$
  $\tau^+ \leftarrow |y_v^+|/|y_u^+|, \quad \tau^- \leftarrow |y_v^-|/|y_u^-|$
  if $\tau^+ > \tau^-$: $\mathrm{votes}_+ \leftarrow \mathrm{votes}_+ + 1$
  else : $\mathrm{votes}_- \leftarrow \mathrm{votes}_- + 1$
  $\mathrm{CL} \leftarrow \mathrm{ComputeConfidenceLevel}(\mathrm{votes}_+, \mathrm{votes}_-)$
if $\mathrm{votes}_+ > \mathrm{votes}_-$: **return true**
else : **return false**

C Experiment Results

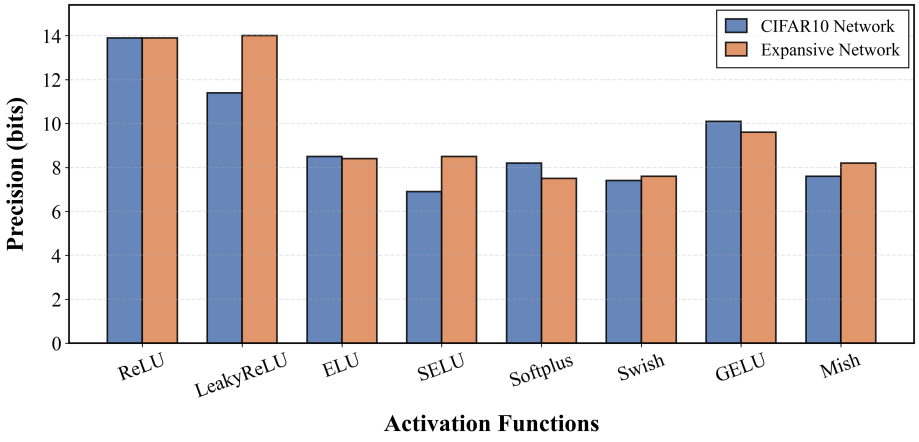


Fig. 3: Bits of precision of the extracted weights on the CIFAR-10 and expansive networks (first two layers only for the LeakyReLU expansive network).

Table 8: Sign recovery analysis in the top-1 score scenario.

Network		**ReLU**	**LeakyReLU**	**ELU**	**SELU**
CIFAR-10	Average Points	55.3	34.4	36.3	38.5
	$\alpha < 0.95$	2/1024	1/1024	0/1024	0/1024
Random	Average Points	55.3	34.4	36.3	38.5
	$\alpha < 0.95$	0/800	0/800	0/800	1/800
Network		**Softplus**	**Swish**	**GELU**	**Mish**
CIFAR-10	Average Points	61.9	56.0	62.1	60.5
	$\alpha < 0.95$	0/1024	0/1024	0/1024	0/1024
Random	Average Points	64.1	62.1	66.2	60.6
	$\alpha < 0.95$	1/800	0/800	0/800	0/800

D Proofs of Theorems

Theorem 1. *Assume no wraparound points exist in the ϵ-neighborhood of x. Let the norm of each neuron's weight vector in layer i be bounded by K_i, i.e., $\|W_j^{(i)}\|_2 \leq K_i$ for all j, and let σ be a λ-Lipschitz continuous activation function. Then, for any perturbation with $\|\Delta x\|_2 < \epsilon$, the network output satisfies $\|f(x + \Delta x) - f(x)\|_2 \leq C\|\Delta x\|_2$, where $C = \lambda^r \prod_{i=1}^{r+1} \sqrt{d_i} K_i$.*

Proof. Because there are no wraparound points in the ϵ-neighborhood of x, we can bound the output change for a single neuron j in the i-th linear layer as:

$$\begin{aligned} |z_j^{(i)}(x + \Delta x) - z_j^{(i)}(x)| &= |W_j^{(i)}(h^{(i-1)}(x + \Delta x) - h^{(i-1)}(x))| \\ &\leq \|W_j^{(i)}\|_2 \|h^{(i-1)}(x + \Delta x) - h^{(i-1)}(x)\|_2 \\ &\leq K_i \|h^{(i-1)}(x + \Delta x) - h^{(i-1)}(x)\|_2 \end{aligned}$$

For the entire vector of the i-th layer, the total change is:

$$\begin{aligned} \|z^{(i)}(x + \Delta x) - z^{(i)}(x)\|_2 &= \sqrt{\sum_{j=1}^{d_i} (z_j^{(i)}(x + \Delta x) - z_j^{(i)}(x))^2} \\ &\leq \sqrt{\sum_{j=1}^{d_i} \left(K_i \|h^{(i-1)}(x + \Delta x) - h^{(i-1)}(x)\|_2\right)^2} \\ &= \sqrt{d_i} K_i \|h^{(i-1)}(x + \Delta x) - h^{(i-1)}(x)\|_2 \end{aligned}$$

Because the activation function $\sigma(\cdot)$ is λ-Lipschitz continuous, the change after the activation layer is bounded by:

$$\|h^{(i)}(x + \Delta x) - h^{(i)}(x)\|_2 \leq \lambda \|z^{(i)}(x + \Delta x) - z^{(i)}(x)\|_2$$

The network f has $r + 1$ linear layers and r activation layers. By chaining these bounds from the input to the output, we get:

$$\begin{aligned} \|f(x + \Delta x) - f(x)\|_2 &\leq \left(\lambda^r \prod_{i=1}^{r+1} \sqrt{d_i} K_i\right) \|\Delta x\|_2 \\ &= C\|\Delta x\|_2 \end{aligned}$$

□

Theorem 2. *Suppose x is a wraparound point in the i-th layer. Let $\mathcal{S}_\sigma = \{z \in \mathbb{R} \mid \exists \epsilon > 0, \forall y \in (z - \epsilon, z + \epsilon), \sigma(y) = \sigma(z)\}$ denote the flat region of the activation function, and let $p = \Pr[z \in \mathcal{S}_\sigma]$ be the probability that an intermediate pre-activation falls into this region. When traversing across this wraparound point, the probability that the network's final output remains continuous is bounded by $\Pr[\lim_{\|\Delta x\| \to 0} f(x + \Delta x) = f(x - \Delta x)] < \sum_{k=i+1}^{r} p^{d_k}$.*

Proof. Assume x is a wraparound point in the i-th layer. When the input crosses this point, the output of the i-th layer has a sudden jump. Let this jump be:

$$\Delta h^{(i)} = \lim_{\|\Delta x\| \to 0} (h^{(i)}(x + \Delta x) - h^{(i)}(x - \Delta x)) \neq 0$$

Because the network weights are continuous, this sudden jump will not perfectly cancel out in the next layer. Therefore, the input to the $(i+1)$-th layer will also have a jump with a probability of 1:

$$\Pr\left(\lim_{\|\Delta x\| \to 0} W_j^{(i+1)}(h^{(i)}(x + \Delta x) - h^{(i)}(x - \Delta x)) \neq 0\right) = 1$$

For the network to hide this jump (so the output becomes continuous again), the value before the activation function must fall into the flat region $\mathcal{S}_\sigma$. The chance of this happening for a single neuron is bounded by p:

$$\begin{aligned}\Pr\left(\lim_{\|\Delta x\| \to 0} h_j^{(i+1)}(x + \Delta x) = h_j^{(i+1)}(x - \Delta x)\right) &\leq \Pr\left(z_j^{(i+1)} \in \mathcal{S}_\sigma\right) \\ &= p\end{aligned}$$

For the entire $(i+1)$-th layer to hide the jump, all d_{i+1} neurons in this layer must independently fall into the flat region. The probability of this is:

$$\Pr\left(\lim_{\|\Delta x\| \to 0} h^{(i+1)}(x + \Delta x) = h^{(i+1)}(x - \Delta x)\right) \leq p^{d_{i+1}}$$

If the final output of the network is continuous, at least one hidden layer from $i+1$ to r must completely hide the jump. Using the union bound, the total probability of a continuous final output is at most the sum of the probabilities for each layer:

$$\begin{aligned}&\Pr\left(\lim_{\|\Delta x\| \to 0} f(x + \Delta x) = f(x - \Delta x)\right) \\ &\quad\leq \sum_{k=i+1}^{r} \Pr\left(\lim_{\|\Delta x\| \to 0} h^{(k)}(x + \Delta x) = h^{(k)}(x - \Delta x)\right) \\ &\quad\leq \sum_{k=i+1}^{r} p^{d_k}\end{aligned}$$

□

Theorem 3. *Let σ be a non-polynomial analytic activation function, and let $x \in B(\eta_j^{(i)}, k)$ be a known wraparound point. Provided that the network's forward mapping F_{i-1} is non-degenerate within the local R-neighborhood of x, the affine span of the mapped local boundary strictly encompasses the entire signature hyperplane. That is,* $\mathrm{Aff}(B_{x,R}^{i-1}) = S(\eta_j^{(i)}, k)$.

Proof. We use a proof by contradiction. Assume the mapped local boundary $B_{x,R}^{i-1}$ does not span the entire signature hyperplane $S(\eta_j^{(i)}, k)$. This means $B_{x,R}^{i-1}$ must lie inside a lower-dimensional subspace of $S(\eta_j^{(i)}, k)$. Therefore, there must exist a non-zero vector $v \in \mathbb{R}^{d_{i-1}}$ and a constant c such that:

1. v is parallel to the hyperplane, meaning it is orthogonal to the normal vector: $v \cdot W_j^{(i)} = 0$.
2. For all $x \in B(\eta_j^{(i)}, k) \cap \mathcal{N}(x^*, R)$, the equation $v \cdot F_{i-1}(x) = c$ holds.

By the definition of the wraparound boundary, for any point x in this neighborhood, we have:

$$\Phi(x) = \sum_{k=1}^{d_{i-1}} W_{j,k}^{(i)} F_{i-1,k}(x) - C = 0$$

At the same time, from our assumption, we have:

$$g(x) = \sum_{k=1}^{d_{i-1}} v_k F_{i-1,k}(x) - c = 0$$

Because the activation function is analytic, the identity theorem states that if $g(x) = 0$ wherever $\Phi(x) = 0$, they must be proportional. This means there is an analytic function $\lambda(x)$ such that for all x in the neighborhood:

$$\sum_{k=1}^{d_{i-1}} v_k F_{i-1,k}(x) - c = \lambda(x) \left(\sum_{k=1}^{d_{i-1}} W_{j,k}^{(i)} F_{i-1,k}(x) - C \right)$$

To make the proof easier to follow, consider the case where $i = 2$. The equation above becomes:

$$\begin{aligned} &\sum_{k=1}^{d_{i-1}} v_k \sigma(W_k^{(1)} x + b_k^{(1)}) - c \\ &= \lambda(x) \left(\sum_{k=1}^{d_{i-1}} W_{j,k}^{(i)} \sigma(W_k^{(1)} x + b_k^{(1)}) - C \right) \end{aligned}$$

If σ is not a polynomial, its ridge functions are linearly independent[19]. If $\lambda(x)$ is not a constant, its Taylor expansion has at least one non-zero term like $c_m x_m$. Multiplying this term with the right side creates mixed terms like $x_m \sigma(W_k^{(1)} x + b_k^{(1)})$.

However, the left side only contains simple ridge functions $\sigma(W_k^{(1)} x + b_k^{(1)})$. Because non-polynomial ridge functions are linearly independent, a simple ridge function can never equal a mixed term. To keep both sides strictly equal, the

right side cannot have these mixed terms. This forces all non-constant terms in $\lambda(x)$ to be exactly 0.

Therefore, $\lambda(x)$ is just a constant λ. Matching the coefficients of the ridge functions on both sides, we get:

$$v_k = \lambda W_{j,k}^{(i)}, \quad \forall k \implies v = \lambda W_j^{(i)}$$

Substituting this into our first condition ($v \cdot W_j^{(i)} = 0$), we get:

$$(\lambda W_j^{(i)}) \cdot W_j^{(i)} = \lambda \|W_j^{(i)}\|_2^2 = 0$$

Because the weight vector $W_j^{(i)}$ is not zero ($\|W_j^{(i)}\|_2^2 > 0$), we must have $\lambda = 0$. This means $v = \mathbf{0}$, which directly contradicts our starting assumption that v is a non-zero vector.

This contradiction proves that our assumption is wrong. The mapped boundary must fully span the entire signature hyperplane $S(\eta_j^{(i)}, k)$. □

Theorem 4. *The Jacobian matrix of the i-th layer can be factorized as $J^{(i)}(x) = \prod_{k=i}^{1} \left(\Lambda^{(k)}(x) W^{(k)}\right)$, where the diagonal activation derivative matrix is given by:*

$$\Lambda^{(k)}(x) = \operatorname{diag}\left(\sigma'(z_1^{(k)}(x)), \sigma'(z_2^{(k)}(x)), \ldots, \sigma'(z_{d_k}^{(k)}(x))\right)$$

Furthermore, if x is a wraparound point of neuron $\eta_j^{(i)}$, its corresponding normal vector satisfies

$$n(x) \parallel J^{(i-1)}(x)^T (W_j^{(i)})^T,$$

where $\parallel$ denotes collinear.

Proof. First, we find the derivative of the k-th layer's output $h^{(k)}$ with respect to its input $h^{(k-1)}$ using the chain rule:

$$\frac{\partial h^{(k)}}{\partial h^{(k-1)}} = \frac{\partial h^{(k)}}{\partial z^{(k)}} \frac{\partial z^{(k)}}{\partial h^{(k-1)}}$$

Because $z^{(k)} = W^{(k)} h^{(k-1)} + b^{(k)}$, its derivative with respect to $h^{(k-1)}$ is simply the weight matrix $W^{(k)}$. The term $\frac{\partial h^{(k)}}{\partial z^{(k)}}$ is the derivative of the activation function. Because the activation function is applied element by element, this forms a diagonal matrix $\Lambda^{(k)}(x)$:

$$\frac{\partial h^{(k)}}{\partial z^{(k)}} = \Lambda^{(k)}(x) = \operatorname{diag}\left(\sigma'(z_1^{(k)}(x)), \ldots, \sigma'(z_{d_k}^{(k)}(x))\right)$$

So, the derivative for a single layer is $\frac{\partial h^{(k)}}{\partial h^{(k-1)}} = \Lambda^{(k)}(x) W^{(k)}$. To find the full Jacobian matrix $J^{(i)}(x)$ from the network input x to the i-th layer, we multiply these single-layer matrices together:

$$J^{(i)}(x) = \frac{\partial h^{(i)}}{\partial x} = \prod_{k=i}^{1} \frac{\partial h^{(k)}}{\partial h^{(k-1)}} = \prod_{k=i}^{1} \left(\Lambda^{(k)}(x) W^{(k)}\right)$$

For the target neuron j in the i-th layer, its value before activation is $z_j^{(i)}(x) = W_j^{(i)} h^{(i-1)} + b_j^{(i)}$. Taking the gradient with respect to the input x gives the normal vector $n(x)$:

$$\begin{aligned} n(x) \parallel \nabla_x z_j^{(i)} \\ &= \left(\frac{\partial z_j^{(i)}}{\partial x} \right)^T \\ &= \left(\frac{\partial z_j^{(i)}}{\partial h^{(i-1)}} \frac{\partial h^{(i-1)}}{\partial x} \right)^T \\ &= \left(W_j^{(i)} J^{(i-1)}(x) \right)^T \\ &= J^{(i-1)}(x)^T (W_j^{(i)})^T \end{aligned}$$

□

Theorem 5. *Assume that the downstream coefficients satisfy* $g_k^+ \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_+^2)$ *and* $g_k^- \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_-^2)$*, and that the two families* $\{g_k^+\}_{k=1}^{d_i}$ *and* $\{g_k^-\}_{k=1}^{d_i}$ *are mutually independent. Under the worst case, the sign-recovery success probability for a probing direction* v *is*

$$\Pr\left[|y_v^+| > |y_v^-|\right] = \frac{2}{\pi} \arctan\left(\frac{\sigma_+}{\sigma_-} \sqrt{\frac{\mathrm{SIR}(v) + 1}{\alpha^2 \, \mathrm{SIR}(v) + 1}} \right).$$

Proof. Because g_k^+ are independent zero-mean Gaussian variables, their linear combination y_v^+ is also a zero-mean Gaussian variable. Its variance is:

$$y_v^+ = \sum_{k=1}^{d_i} g_k^+ A_k v \sim \mathcal{N}\left(0, \sigma_{v+}^2\right), \quad \sigma_{v+}^2 = \sigma_+^2 \left((A_j v)^2 + \sum_{k \neq j} (A_k v)^2 \right)$$

Similarly, y_v^- is a zero-mean Gaussian variable with variance:

$$y_v^- = \sum_{k=1}^{d_i} g_k^- A_k v \sim \mathcal{N}\left(0, \sigma_{v-}^2\right), \quad \sigma_{v-}^2 = \sigma_-^2 \left(\alpha^2 (A_j v)^2 + \sum_{k \neq j} (A_k v)^2 \right)$$

Because y_v^+ and y_v^- are independent, the probability of a successful attack is the probability that $|y_v^+| > |y_v^-|$. We can write this as:

$$\begin{aligned} \Pr(|y_v^+| > |y_v^-|) &= \Pr\left(\frac{|y_v^+|}{|y_v^-|} > 1 \right) \\ &= \Pr\left(\frac{|y_v^+| / \sigma_{v+}}{|y_v^-| / \sigma_{v-}} > \frac{\sigma_{v-}}{\sigma_{v+}} \right) \end{aligned}$$

The terms $|y_v^+|/\sigma_{v+}$ and $|y_v^-|/\sigma_{v-}$ are the absolute values of independent standard normal variables. Their ratio follows a standard half-Cauchy distribution, which has the cumulative distribution function $F(y) = \frac{2}{\pi}\arctan(y)$. Using this, the probability becomes:

$$\Pr(|y_v^+| > |y_v^-|) = 1 - \frac{2}{\pi}\arctan\left(\frac{\sigma_{v-}}{\sigma_{v+}}\right) = \frac{2}{\pi}\arctan\left(\frac{\sigma_{v+}}{\sigma_{v-}}\right)$$

Next, we can rewrite the ratio σ_{v+}/σ_{v-} using the signal-to-interference ratio, $\mathrm{SIR}(v) = \frac{(A_j v)^2}{\sum_{k\neq j}(A_k v)^2}$:

$$\begin{aligned}\frac{\sigma_{v+}}{\sigma_{v-}} &= \sqrt{\frac{\sigma_+^2\left((A_j v)^2 + \sum_{k\neq j}(A_k v)^2\right)}{\sigma_-^2\left(\alpha^2(A_j v)^2 + \sum_{k\neq j}(A_k v)^2\right)}} \\ &= \frac{\sigma_+}{\sigma_-}\sqrt{\frac{\mathrm{SIR}(v)+1}{\alpha^2\mathrm{SIR}(v)+1}}\end{aligned}$$

Substituting this ratio back into the probability equation gives the final result:

$$\Pr(|y_v^+| > |y_v^-|) = \frac{2}{\pi}\arctan\left(\frac{\sigma_+}{\sigma_-}\sqrt{\frac{\mathrm{SIR}(v)+1}{\alpha^2\mathrm{SIR}(v)+1}}\right)$$

□

Theorem 6. *Assume that the downstream coefficients $\{g_k^+\}_{k\neq j}$ and $\{g_k^-\}_{k\neq j}$ form two mutually independent families of i.i.d. random variables with zero mean and finite variances σ_+^2 and σ_-^2, respectively. Suppose further that the asymptotic stability condition $\lim_{d_i\to\infty}\frac{\max_{k\neq j}(A_k v)^2}{\sum_{k\neq j}(A_k v)^2} = 0$ holds. Under the worst-case activation-derivative configuration with $\alpha = 0$, and conditioning on a fixed $g_j^+ \neq 0$, the asymptotic sign-recovery success probability is*

$$\lim_{d_i\to\infty}\Pr\left[|y_v^+| > |y_v^-|\right] = \Phi_2\left(\Gamma, \Gamma; \rho\right) + \Phi_2\left(-\Gamma, -\Gamma; \rho\right),$$

where Φ_2 denotes the cumulative distribution function of a standard bivariate normal distribution,

$$\Gamma = |g_j^+|\sqrt{\frac{\mathrm{SIR}(v)}{\sigma_+^2+\sigma_-^2}}, \qquad \rho = \frac{\sigma_+^2-\sigma_-^2}{\sigma_+^2+\sigma_-^2}.$$

Moreover, the asymptotic success probability is strictly increasing in $\mathrm{SIR}(v)$.

Proof. Because $\tau_j^+ = 1$, the positive derivative y_v^+ contains both the target signal and the background interference:

$$y_v^+ = g_j^+ A_j v + \sum_{k\neq j} g_k^+ A_k v$$

Because $\tau_j^- = 0$, the target neuron is inactive on the negative side. Therefore, the negative derivative y_v^- only contains the interference, and its signal term is exactly zero:

$$y_v^- = 0 + \sum_{k \neq j} g_k^- A_k v$$

By the Central Limit Theorem, as the layer width grows, the sums of the interference terms converge to normal distributions:

$$\sum_{k \neq j} g_k^+ A_k v \xrightarrow{d} \mathcal{N}\left(0, \sigma_{v+}^2\right), \quad \sigma_{v+}^2 = \sigma_+^2 \sum_{k \neq j} (A_k v)^2$$

$$\sum_{k \neq j} g_k^- A_k v \xrightarrow{d} \mathcal{N}\left(0, \sigma_{v-}^2\right), \quad \sigma_{v-}^2 = \sigma_-^2 \sum_{k \neq j} (A_k v)^2$$

So, the full directional derivatives follow these distributions:

$$y_v^+ \xrightarrow{d} \mathcal{N}\left(g_j^+ A_j v, \sigma_{v+}^2\right), \quad y_v^- \xrightarrow{d} \mathcal{N}\left(0, \sigma_{v-}^2\right)$$

To easily compare their absolute values, we use the difference of squares:

$$|y_v^+| > |y_v^-| \iff (y_v^+ + y_v^-)(y_v^+ - y_v^-) > 0$$

Let $U = y_v^+ + y_v^-$ and $V = y_v^+ - y_v^-$. Their expected values are:

$$\mathrm{E}(U) = \mathrm{E}(V) = g_j^+ A_j v$$

Their variances and covariance are:

$$\mathrm{Var}(U) = \mathrm{Var}(V) = \sigma_{v+}^2 + \sigma_{v-}^2$$
$$\mathrm{Cov}(U, V) = \mathrm{Var}(y_v^+) - \mathrm{Var}(y_v^-) = \sigma_{v+}^2 - \sigma_{v-}^2$$

Because U and V are linear combinations of independent Gaussian variables, they form a bivariate normal distribution. Their correlation coefficient ρ is:

$$\rho = \frac{\mathrm{Cov}(U, V)}{\sqrt{\mathrm{Var}(U)\mathrm{Var}(V)}} = \frac{\sigma_{v+}^2 - \sigma_{v-}^2}{\sigma_{v+}^2 + \sigma_{v-}^2} = \frac{\sigma_+^2 - \sigma_-^2}{\sigma_+^2 + \sigma_-^2}$$

The probability of a successful attack is the chance that U and V share the same sign (both positive or both negative):

$$\Pr(|y_v^+| > |y_v^-|) = \Pr(U > 0, V > 0) + \Pr(U < 0, V < 0)$$

Using the standard bivariate normal cumulative distribution function Φ_2, we divide the variables by their standard deviation $\sqrt{\sigma_{v+}^2 + \sigma_{v-}^2}$. Notice how the integration boundary Γ is formed using $\mathrm{SIR}(v)$:

$$\begin{aligned} \frac{\mathrm{E}(U)}{\sqrt{\mathrm{Var}(U)}} &= \frac{g_j^+ A_j v}{\sqrt{(\sigma_+^2 + \sigma_-^2) \sum_{k \neq j} (A_k v)^2}} \\ &= g_j^+ \sqrt{\frac{\mathrm{SIR}(v)}{\sigma_+^2 + \sigma_-^2}} \\ &= \Gamma \end{aligned}$$

Using this boundary Γ, the asymptotic probability is:

$$\lim_{d_i \to \infty} \Pr(|y_v^+| > |y_v^-|) = \Phi_2(\Gamma, \Gamma; \rho) + \Phi_2(-\Gamma, -\Gamma; \rho)$$

To show this probability increases with $\mathrm{SIR}(v)$, let $g(x) = \Phi_2(x, x; \rho) + \Phi_2(-x, -x; \rho)$. Its derivative is:

$$g'(x) = 2\phi(x) \left[\Phi\left(x\frac{\sigma_-}{\sigma_+} \right) - \Phi\left(-x\frac{\sigma_-}{\sigma_+} \right) \right]$$

Because $x > 0$ and the standard normal cumulative distribution $\Phi(z)$ is strictly increasing, we know $\Phi\left(x\frac{\sigma_-}{\sigma_+} \right) > \Phi\left(-x\frac{\sigma_-}{\sigma_+} \right)$. Also, the normal density $\phi(x)$ is always positive. Therefore, $g'(x) > 0$, meaning $g(x)$ is strictly increasing. Since the input Γ strictly increases with $\mathrm{SIR}(v)$, the final success probability also strictly increases with $\mathrm{SIR}(v)$. □

Theorem 7. *The signal-to-interference ratio* $\mathrm{SIR}(v)$ *is maximized by the probing direction* $v = A^\dagger e_j$.

Proof. We can write the signal-to-interference ratio $\mathrm{SIR}(v)$ in matrix form:

$$\mathrm{SIR}(v) = \frac{v^T A_j^T A_j v}{v^T A_{-j}^T A_{-j} v}$$

where A_{-j} is the matrix A without its j-th row. We want to find the vector v that maximizes this ratio.

Let $R = A_{-j}^T A_{-j}$. Assuming R is positive definite, it has an inverse R^{-1} and a square root $R^{1/2}$. We introduce a new variable $u = R^{1/2} v$, which means $v = R^{-1/2} u$. Substituting this into our ratio gives:

$$\max_u \frac{u^T R^{-1/2} A_j^T A_j R^{-1/2} u}{u^T u} = \max_u \frac{(A_j R^{-1/2} u)^2}{u^T u}$$

By the Cauchy-Schwarz inequality, the maximum is reached when u is proportional to the vector $R^{-1/2} A_j^T$. Changing back to v, the optimal direction is:

$$v_{opt} \propto R^{-1/2}(R^{-1/2} A_j^T) = R^{-1} A_j^T = (A_{-j}^T A_{-j})^{-1} A_j^T$$

Now, recall the definition of the pseudoinverse $A^\dagger = (A^T A)^{-1} A^T$. Multiplying by e_j gives the j-th column:

$$A^\dagger e_j = (A^T A)^{-1} A^T e_j = (A^T A)^{-1} A_j^T$$

We can split $A^T A$ into $A_{-j}^T A_{-j} + A_j^T A_j = R + A_j^T A_j$. Substituting this in:

$$A^\dagger e_j = (R + A_j^T A_j)^{-1} A_j^T$$

Using the Sherman-Morrison formula, we can invert this matrix:

$$(R + A_j^T A_j)^{-1} = R^{-1} - \frac{R^{-1} A_j^T A_j R^{-1}}{1 + A_j R^{-1} A_j^T}$$

Substituting this back into the expression for $A^\dagger e_j$, we get:

$$\begin{aligned} A^\dagger e_j &= \left(R^{-1} - \frac{R^{-1} A_j^T A_j R^{-1}}{1 + A_j R^{-1} A_j^T} \right) A_j^T \\ &= R^{-1} A_j^T - \frac{R^{-1} A_j^T (A_j R^{-1} A_j^T)}{1 + A_j R^{-1} A_j^T} \\ &= \left(1 - \frac{A_j R^{-1} A_j^T}{1 + A_j R^{-1} A_j^T} \right) R^{-1} A_j^T \\ &= \left(\frac{1}{1 + A_j R^{-1} A_j^T} \right) R^{-1} A_j^T \end{aligned}$$

Because $A_j R^{-1} A_j^T$ is a positive number, the fraction $\frac{1}{1+A_j R^{-1} A_j^T}$ is just a positive constant. This means $A^\dagger e_j$ is strictly proportional to $R^{-1} A_j^T$:

$$A^\dagger e_j \propto v_{opt}$$

Because $\mathrm{SIR}(v)$ is a ratio, its value does not change if we scale the vector v. Therefore, choosing the direction $v = A^\dagger e_j$ will perfectly reach the maximum $\mathrm{SIR}(v)$. □

Theorem 8. *Under the assumptions of Theorem 5, the sign-recovery success probability in the top-1 score setting is*

$$\Pr[|p_v^+| > |p_v^-|] = \frac{2}{\pi} \arctan \left(\frac{\sigma_+ p^+_{\text{top}^+}}{\sigma_- p^-_{\text{top}^-}} \sqrt{\frac{\sum_{l \neq \text{top}^+} (p_l^+)^2}{\sum_{l \neq \text{top}^-} (p_l^-)^2} \cdot \frac{\mathrm{SIR}(v) + 1}{\alpha^2 \, \mathrm{SIR}(v) + 1}} \right).$$

Proof. Using the derivative rule of the softmax function, the derivatives of the top probability p_{top} with respect to each output y_i are:

$$\frac{\partial p_{\text{top}}}{\partial y_{\text{top}}} = p_{\text{top}}(1 - p_{\text{top}}), \quad \frac{\partial p_{\text{top}}}{\partial y_l} = -p_{\text{top}} p_l \quad \forall l \neq \text{top}$$

Let p_v^+ be the directional derivative of the top probability on the positive side. By the chain rule, it can be written as:

$$\begin{aligned} p_v^+ &= \frac{\partial p^+_{\text{top}^+}}{\partial v} = \frac{\partial p^+_{\text{top}^+}}{\partial y} \frac{\partial f}{\partial v} \\ &= p^+_{\text{top}^+} (e_{\text{top}^+} - p^+)^T G^+ (\Lambda^{(i)})^+ A v \end{aligned}$$

Because $\sum_{l=1}^{n} p_l^+ = 1$, we can simplify this derivative to:

$$p_v^+ = p_{\text{top}^+}^+ \sum_{l \neq \text{top}^+} p_l^+ \left(G_{\text{top}^+}^+ (\Lambda^{(i)})^+ Av - G_l^+ (\Lambda^{(i)})^+ Av \right)$$

Because G^+ follows a zero-mean normal distribution, this linear combination p_v^+ is also a zero-mean normal variable. Its variance is:

$$p_v^+ \sim \mathcal{N}(0, \sigma_{v+}^2), \quad \sigma_{v+}^2 = 2(p_{\text{top}^+}^+)^2 \sum_{l \neq \text{top}^+} (p_l^+)^2 \sigma_+^2 \left((A_j v)^2 + \sum_{k \neq j} (A_k v)^2 \right)$$

Similarly, the directional derivative on the negative side, p_v^- is a zero-mean normal variable with variance:

$$\begin{aligned} p_v^- &\sim \mathcal{N}(0, \sigma_{v-}^2) \\ \sigma_{v-}^2 &= 2(p_{\text{top}^-}^-)^2 \sum_{l \neq \text{top}^-} (p_l^-)^2 \sigma_-^2 \left(\alpha^2 (A_j v)^2 + \sum_{k \neq j} (A_k v)^2 \right) \end{aligned}$$

Because p_v^+ and p_v^- are independent zero-mean normal variables, the ratio of their absolute values follows a Half-Cauchy distribution. We can find the success probability p' using the arctangent function:

$$\Pr[|p_v^+| > |p_v^-|] = \frac{2}{\pi} \arctan\left(\frac{\sigma_{v+}}{\sigma_{v-}} \right)$$

We can rewrite the terms containing $(A_j v)^2$ using $\text{SIR}(v) = \frac{(A_j v)^2}{\sum_{k \neq j} (A_k v)^2}$:

$$\begin{aligned} \frac{\sigma_{v+}}{\sigma_{v-}} &= \frac{\sigma_+ p_{\text{top}^+}^+}{\sigma_- p_{\text{top}^-}^-} \sqrt{\frac{\sum_{l \neq \text{top}^+} (p_l^+)^2}{\sum_{l \neq \text{top}^-} (p_l^-)^2}} \sqrt{\frac{(A_j v)^2 + \sum_{k \neq j} (A_k v)^2}{\alpha^2 (A_j v)^2 + \sum_{k \neq j} (A_k v)^2}} \\ &= \frac{\sigma_+ p_{\text{top}^+}^+}{\sigma_- p_{\text{top}^-}^-} \sqrt{\frac{\sum_{l \neq \text{top}^+} (p_l^+)^2}{\sum_{l \neq \text{top}^-} (p_l^-)^2} \cdot \frac{\text{SIR}(v) + 1}{\alpha^2 \, \text{SIR}(v) + 1}} \end{aligned}$$

Substituting this ratio back into the probability equation gives the final result:

$$\Pr[|p_v^+| > |p_v^-|] = \frac{2}{\pi} \arctan\left(\frac{\sigma_+ p_{\text{top}^+}^+}{\sigma_- p_{\text{top}^-}^-} \sqrt{\frac{\sum_{l \neq \text{top}^+} (p_l^+)^2}{\sum_{l \neq \text{top}^-} (p_l^-)^2} \cdot \frac{\text{SIR}(v) + 1}{\alpha^2 \, \text{SIR}(v) + 1}} \right)$$

□

Theorem 9. *Assume that the downstream coefficients satisfy* $g_k^+ = g_k^- \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$. *Under the worst case, the sign-recovery success probability for a probing direction* v *is*

$$\Pr[|y_v^+| > |y_v^-|] = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{1+\alpha}{2} \sqrt{\text{SIR}(v)} \right).$$

Proof. Under the worst-case assumption, the directional derivatives on the two sides of the critical point can be written as

$$y_v^+ = g_j(A_j v) + \sum_{k \neq j} g_k(A_k v),$$

$$y_v^- = \alpha g_j(A_j v) + \sum_{k \neq j} g_k(A_k v),$$

where $g_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$.

Let $U = g_j(A_j v)$ and $W = \sum_{k \neq j} g_k(A_k v)$. Then U and W are independent Gaussian variables with variances

$$\mathrm{Var}(U) = \sigma^2 (A_j v)^2, \qquad \mathrm{Var}(W) = \sigma^2 \sum_{k \neq j} (A_k v)^2.$$

The success event

$$|y_v^+| > |y_v^-|$$

is equivalent to

$$(U + W)^2 > (\alpha U + W)^2,$$

which can be rewritten as

$$(1 - \alpha) U ((1 + \alpha) U + 2W) > 0.$$

Since $0 \leq \alpha < 1$, the above event is equivalent to

$$U((1 + \alpha) U + 2W) > 0.$$

Define $V = (1 + \alpha) U + 2W$. The pair (U, V) follows a zero-mean bivariate Gaussian distribution. Its correlation coefficient is

$$\rho = \frac{\mathrm{Cov}(U, V)}{\sqrt{\mathrm{Var}(U)\,\mathrm{Var}(V)}}.$$

Because U and W are independent,

$$\mathrm{Cov}(U, V) = (1 + \alpha)\,\mathrm{Var}(U),$$
$$\mathrm{Var}(V) = (1 + \alpha)^2\,\mathrm{Var}(U) + 4\,\mathrm{Var}(W).$$

Thus,

$$\rho = \frac{(1 + \alpha)(A_j v)}{\sqrt{(1 + \alpha)^2 (A_j v)^2 + 4 \sum_{k \neq j} (A_k v)^2}}.$$

Using the definition of $\mathrm{SIR}(v)$, we have

$$\rho = \frac{(1+\alpha)\sqrt{\mathrm{SIR}(v)}}{\sqrt{(1+\alpha)^2\,\mathrm{SIR}(v)+4}}.$$

For two zero-mean jointly Gaussian random variables, the probability that they have the same sign is

$$\Pr[UV > 0] = \frac{1}{2} + \frac{1}{\pi}\arcsin(\rho).$$

Therefore,

$$\Pr[|y_v^+| > |y_v^-|] = \frac{1}{2} + \frac{1}{\pi}\arcsin\left(\frac{(1+\alpha)\sqrt{\mathrm{SIR}(v)}}{\sqrt{(1+\alpha)^2\,\mathrm{SIR}(v)+4}}\right).$$

Finally, applying the identity

$$\arcsin\left(\frac{x}{\sqrt{x^2+4}}\right) = \arctan\left(\frac{x}{2}\right),$$

where

$$x = (1+\alpha)\sqrt{\mathrm{SIR}(v)},$$

we obtain

$$\Pr[|y_v^+| > |y_v^-|] = \frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{1+\alpha}{2}\sqrt{\mathrm{SIR}(v)}\right).$$

□

Theorem 10. *Under the assumptions of Theorem 5, the sign-recovery success probability with noise normalization is*

$$\Pr[R^+ > R^-] = 1 - \frac{4}{\pi^2}\int_0^\infty \frac{\arctan(kt)}{t^2+1}dt,$$

where $k = \sqrt{\frac{\alpha^2\,\mathrm{SIR}(v)+1}{\mathrm{SIR}(v)+1}}$ *is a constant determined by* α *and* $\mathrm{SIR}(v)$.

Proof. From our previous derivations, the directional derivatives along the probe direction v follow the normal distributions:

$$y_v^+ \sim \mathcal{N}\left(0, \sigma_{v+}^2\right), \quad \sigma_{v+}^2 = \sigma_+^2\left((A_j v)^2 + \sum_{k\neq j}(A_k v)^2\right)$$

$$y_v^- \sim \mathcal{N}\left(0, \sigma_{v-}^2\right), \quad \sigma_{v-}^2 = \sigma_-^2\left(\alpha^2(A_j v)^2 + \sum_{k\neq j}(A_k v)^2\right)$$

For the noise direction u, which is designed not to excite the target neuron ($A_j u = 0$), the distributions are:

$$y_u^+ \sim \mathcal{N}\left(0, \sigma_{u+}^2\right), \quad \sigma_{u+}^2 = \sigma_+^2 \sum_{k \neq j} (A_k u)^2$$

$$y_u^- \sim \mathcal{N}\left(0, \sigma_{u-}^2\right), \quad \sigma_{u-}^2 = \sigma_-^2 \sum_{k \neq j} (A_k u)^2$$

We define the ratios $R^+ = \frac{|y_v^+|}{|y_u^+|}$ and $R^- = \frac{|y_v^-|}{|y_u^-|}$. Because the unbalanced variances (σ_+^2 and σ_-^2) cancel out during the division on each side, R^+ and R^- strictly follow Half-Cauchy distributions:

$$R^+ \sim \text{Half-Cauchy}(0, \gamma_+), \quad R^- \sim \text{Half-Cauchy}(0, \gamma_-)$$

where the scaling parameters γ_+ and γ_- are:

$$\gamma_+ = \frac{\sigma_{v+}}{\sigma_{u+}} = \sqrt{\frac{(A_j v)^2 + \sum_{k \neq j} (A_k v)^2}{\sum_{k \neq j} (A_k u)^2}}$$

$$\gamma_- = \frac{\sigma_{v-}}{\sigma_{u-}} = \sqrt{\frac{\alpha^2 (A_j v)^2 + \sum_{k \neq j} (A_k v)^2}{\sum_{k \neq j} (A_k u)^2}}$$

Let $X = R^+$ and $Y = R^-$. The probability density function (PDF) of Y and the cumulative distribution function (CDF) of X are:

$$f_Y(y) = \frac{2\gamma_-}{\pi(y^2 + \gamma_-^2)}, \quad F_X(x) = \frac{2}{\pi} \arctan\left(\frac{x}{\gamma_+}\right)$$

Because the measurements on the positive and negative sides are independent, X and Y are independent random variables. The success probability p'' is:

$$\begin{aligned} \Pr[R^+ > R^-] &= \Pr(X > Y) \\ &= \int_0^\infty f_Y(y) \Pr(X > y) dy \\ &= \int_0^\infty f_Y(y) \left[1 - F_X(y)\right] dy \end{aligned}$$

Substituting the PDF and CDF into the integral gives:

$$\Pr[R^+ > R^-] = \int_0^\infty \frac{2\gamma_-}{\pi(y^2 + \gamma_-^2)} \left[1 - \frac{2}{\pi} \arctan\left(\frac{y}{\gamma_+}\right)\right] dy$$

We define a constant k using the signal-to-interference ratio $\mathrm{SIR}(v)$:

$$\begin{aligned} k &= \frac{\gamma_-}{\gamma_+} \\ &= \sqrt{\frac{\alpha^2 (A_j v)^2 + \sum_{k \neq j} (A_k v)^2}{(A_j v)^2 + \sum_{k \neq j} (A_k v)^2}} \\ &= \sqrt{\frac{\alpha^2 \mathrm{SIR}(v) + 1}{\mathrm{SIR}(v) + 1}} \end{aligned}$$

By substituting $t = \frac{y}{\gamma_-}$, we have $dy = \gamma_- dt$ and $\frac{y}{\gamma_+} = \frac{\gamma_- t}{\gamma_+} = kt$. This simplifies the integral to the final expression:

$$\begin{aligned} \Pr[R^+ > R^-] &= \int_0^\infty \frac{2}{\pi(t^2+1)} \left[1 - \frac{2}{\pi} \arctan(kt) \right] dt \\ &= 1 - \frac{4}{\pi^2} \int_0^\infty \frac{\arctan(kt)}{t^2+1} dt \end{aligned}$$

□

Theorem 11. *Assume that the conditions of Theorem 5 hold, and that the target neuron satisfies $\tau_j^+ = 1$ and $\tau_j^- = \alpha$. Furthermore, suppose that the following asymptotic stability condition holds: $\lim_{d_i \to \infty} \frac{\sum_{k \neq j} (A_k v)^4 \operatorname{Var}(\tau_k^2)}{\left(\sum_{k \neq j} (A_k v)^2 \mathbb{E}[\tau_k^2] \right)^2} = 0$. Then the asymptotic sign-recovery success probability is*

$$\lim_{d_i \to \infty} \Pr\left[|y_v^+| > |y_v^-| \right] = \frac{2}{\pi} \arctan\left(\frac{\sigma_+}{\sigma_-} \sqrt{\frac{(A_j v)^2 + \sum_{k \neq j} w_k (A_k v)^2}{\alpha^2 (A_j v)^2 + \sum_{k \neq j} w_k (A_k v)^2}} \right).$$

Furthermore, the success probability is maximized by the probing direction $v = \tilde{A}^\dagger e_j$, where the rows of $\tilde{A}$ are defined as $\tilde{A}_j = A_j$ and $\tilde{A}_k = \sqrt{w_k} A_k, \quad k \neq j$.

Proof. First, for a specific set of activation states $\boldsymbol{\tau}$, we can write the directional derivatives as:

$$y_v^+ = g_j^+ A_j v + \sum_{k \neq j} g_k^+ \tau_k A_k v, \quad y_v^- = g_j^- \alpha A_j v + \sum_{k \neq j} g_k^- \tau_k A_k v$$

Because g_k^+ and g_k^- are independent zero-mean Gaussian variables, y_v^+ and y_v^- are also independent Gaussian variables. Their conditional variances are:

$$\operatorname{Var}(y_v^+ \mid \boldsymbol{\tau}) = \sigma_+^2 \left((A_j v)^2 + \sum_{k \neq j} \tau_k^2 (A_k v)^2 \right)$$

$$\operatorname{Var}(y_v^- \mid \boldsymbol{\tau}) = \sigma_-^2 \left(\alpha^2 (A_j v)^2 + \sum_{k \neq j} \tau_k^2 (A_k v)^2 \right)$$

Let $S = \sum_{k \neq j} \tau_k^2 (A_k v)^2$ be the sum of the background interference. Its expected value is:

$$\mathbb{E}[S] = \sum_{k \neq j} \mathbb{E}[\tau_k^2](A_k v)^2 = \sum_{k \neq j} w_k (A_k v)^2$$

The stability condition in the theorem guarantees that:

$$\lim_{d_i \to \infty} \frac{\mathrm{Var}(S)}{(\mathbb{E}[S])^2} = \lim_{d_i \to \infty} \frac{\sum_{k \neq j} (A_k v)^4 \mathrm{Var}(\tau_k^2)}{\left(\sum_{k \neq j} (A_k v)^2 \mathbb{E}[\tau_k^2] \right)^2} = 0$$

By the Law of Large Numbers, this means that as the network width grows ($d_i \to \infty$), the random variable S converges to its exact expected value $\mathbb{E}[S]$. Therefore, for large networks, the variances become fixed values, which we denote as σ_{v+}^2 and σ_{v-}^2:

$$\sigma_{v+}^2 = \sigma_+^2 \left((A_j v)^2 + \sum_{k \neq j} w_k (A_k v)^2 \right), \quad \sigma_{v-}^2 = \sigma_-^2 \left(\alpha^2 (A_j v)^2 + \sum_{k \neq j} w_k (A_k v)^2 \right)$$

Because the measurements are independent, the ratio $|y_v^+|/|y_v^-|$ follows a Half-Cauchy distribution. As shown in our earlier derivations, the success probability $\Pr(|y_v^+| > |y_v^-|)$ can be calculated using the arctangent function:

$$\begin{aligned} \lim_{d_i \to \infty} p &= \Pr(|y_v^+| > |y_v^-|) \\ &= \frac{2}{\pi} \arctan\left(\frac{\sigma_{v+}}{\sigma_{v-}} \right) \\ &= \frac{2}{\pi} \arctan\left(\frac{\sigma_+}{\sigma_-} \sqrt{\frac{(A_j v)^2 + \sum_{k \neq j} w_k (A_k v)^2}{\alpha^2 (A_j v)^2 + \sum_{k \neq j} w_k (A_k v)^2}} \right) \end{aligned}$$

We can define a refined signal-to-interference ratio using the modified matrix $\tilde{A}$, where $\tilde{A}_j = A_j$ and $\tilde{A}_k = \sqrt{w_k} A_k$ for all $k \neq j$:

$$\mathrm{SIR}_{\mathrm{refined}}(v) = \frac{(\tilde{A}_j v)^2}{\sum_{k \neq j} (\tilde{A}_k v)^2}$$

Substituting this into our probability formula gives:

$$\lim_{d_i \to \infty} \Pr\left[|y_v^+| > |y_v^-| \right] = \frac{2}{\pi} \arctan\left(\frac{\sigma_+}{\sigma_-} \sqrt{\frac{\mathrm{SIR}_{\mathrm{refined}}(v) + 1}{\alpha^2 \mathrm{SIR}_{\mathrm{refined}}(v) + 1}} \right)$$

The optimal direction $v = (\tilde{A})^\dagger e_j$ maximizes this probability, which (as in the standard scenario) strictly increases with $\mathrm{SIR}_{\mathrm{refined}}(v)$. □

E Algorithm

Algorithm 5 Wraparound Point Search Algorithm

Input: Model oracle $\mathcal{O}$, step size s, jump threshold $\mathcal{M}$, tolerance ϵ
Output: Wraparound point x^*

$x_0 \xleftarrow{\$} \mathbb{R}^{d_0}$, $v \xleftarrow{\$} \{v \in \mathbb{R}^{d_0} \mid \|v\|_2 = s\}$, $n \leftarrow 0$
loop
  $y_n \leftarrow \mathcal{O}(x_0 + nv)$
  $y_{n+1} \leftarrow \mathcal{O}(x_0 + (n+1)v)$
  if $\|y_{n+1} - y_n\|_2 > \mathcal{M}$ **then**
    $\alpha_L \leftarrow n$, $\alpha_H \leftarrow n+1$
    while $\alpha_H - \alpha_L \geq \epsilon$ **do**
      $\alpha_M \leftarrow (\alpha_H + \alpha_L)/2$
      $\Delta_H \leftarrow \|\mathcal{O}(x_0 + \alpha_H v) - \mathcal{O}(x_0 + \alpha_M v)\|_2$
      $\Delta_L \leftarrow \|\mathcal{O}(x_0 + \alpha_M v) - \mathcal{O}(x_0 + \alpha_L v)\|_2$
      if $\Delta_H > \mathcal{M}$ **then**
        $\alpha_L \leftarrow \alpha_M$
      else if $\Delta_L > \mathcal{M}$ **then**
        $\alpha_H \leftarrow \alpha_M$
      else
        break
    if $\alpha_H - \alpha_L < \epsilon$ **then**
      return $x^* \leftarrow x_0 + \alpha_M v$
  $n \leftarrow n+1$

Algorithm 6 Normal Vector Calculation Algorithm

Input: Wraparound point x, step size α, orthogonal basis $\{e_i\}_{i=0}^{d_0-1}$
Output: Normal vector $\mathbf{n}$

$n_0 \leftarrow 1$
$x' \leftarrow x + \alpha e_0$
for $i = 1$ **to** $d_0 - 1$ **do**
  $\beta_i \leftarrow \arg\min_\beta \{|\beta| \mid x' + \beta e_i \in \mathcal{W}\}$ % via binary search
  $n_i \leftarrow -\beta_i / \alpha$
$\mathbf{n}_{\text{raw}} \leftarrow (n_0, n_1, \ldots, n_{d_0-1})^T$
return $\mathbf{n} \leftarrow \mathbf{n}_{\text{raw}} / \|\mathbf{n}_{\text{raw}}\|_2$