ACCELERATING END-TO-END CRYPTANALYTIC EXTRACTION VIA GEOMETRIC CRITICAL POINT SCREENING

Anonymous authors
Paper under double-blind review

ABSTRACT

Cryptanalytic model extraction recovers neural-network parameters from prediction queries, exposing information that model owners seek to protect in prediction systems. Layerwise methods often spend many queries on critical points from unrelated layers, while parameter errors propagate into subsequent recovery stages. To address these bottlenecks, we propose Geometric Critical Point Screening (GCPS) within an end-to-end framework for layerwise parameter recovery. GCPS checks geometric relations among critical points before measuring shallow-layer signatures, reducing the number of candidates requiring costly directional measurements. Numerical refinement and coordinate alignment prepare these shallow estimates for subsequent layers, where the full-prefix Jacobian identifies reachable measurement directions. Guarded low-dimensional measurements reduce directional queries, while native input-space repair, cross-region clustering, and completion handle unreliable checks and partial observations. Experiments compare independent true-prefix recovery and end-to-end extraction on a public FP64 784-$8^{(8)}$-1 ReLU checkpoint, supplemented by shallow MNIST and CIFAR comparisons. Both protocols use truth-assisted signs and native auxiliary truth, with end-to-end costs summed over selected completed or resumed stages. Relative to the reference implementation, our end-to-end procedure achieves $1.94\times$ query and $2.00\times$ recorded layer-time speedups under this accounting. Both end-to-end methods verify 6593 of 6643 effective weights. Independent true-prefix recovery achieves a $1.85\times$ query speedup and verifies 6607 effective weights, compared with 6603 for the reference under the same evaluation.

1 INTRODUCTION

Deep neural networks (DNNs) support computer vision, natural language processing, and other applications whose training requires substantial data and computation. Model owners keep their parameters private, but prediction interfaces can expose those parameters through carefully chosen queries (Tramèr et al., 2016). Research on model extraction has consequently progressed from theoretical identifiability to practical model stealing and cryptanalytic recovery of individual neurons.

Theoretical foundations and early exploration (1994–2015). Fefferman (1994) established neural-network identifiability from the output function under specific assumptions, with uniqueness understood up to permitted parameter equivalences. Lowd & Meek (2005) subsequently developed query-based reverse engineering of linear classifiers, linking functional observations to internal structure while deep ReLU extraction remained open.

The rise of practical attacks (2016–2019). Tramèr et al. (2016) demonstrated extraction through deployed prediction APIs, motivating subsequent work on kernel models (Reith et al., 2019) and substitute networks for transferable adversarial attacks (Papernot et al., 2017). High task accuracy, prediction fidelity, and parameter recovery are distinct objectives (Jagielski et al., 2020), and our evaluation focuses on parameter recovery and numerical error.

Cryptanalytic and geometric extraction (2020 onward). Carlini et al. (2020) decomposed ReLU extraction into signature extraction, which recovers neuron parameters up to scale, and sign recovery, which resolves their orientation. Related geometric work reconstructs networks from linear-region

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

boundaries (Rolnick & Kording, 2020), while Daniely & Granot (2023) give a polynomial-time membership-query algorithm for certain shallow ReLU networks. Canales-Martínez et al. (2024) introduced polynomial-time sign methods, including systems of equations and neuron wiggles, without removing the separate difficulty of deep-layer signature extraction. Foerster et al. (2024) analyze difficult signatures and signs, highlighting rank deficiency and numerical obstacles that prevent reliable recovery of deeper network layers.

Hard-label extraction uses decision boundaries instead of raw output contrasts (Chen et al., 2024, Carlini et al., 2025), with subsequent studies addressing layer-specific recovery (Canales-Martínez & Santos, 2026), convolutional networks (Li et al., 2026a), and graph networks (Li et al., 2026b). Our method uses raw outputs and therefore makes no claim about extraction cost under the more restrictive hard-label query interface. For deep raw-output extraction, Liu et al. (2026) develop a layerwise framework addressing rank deficiency, deep-layer interference, merging, precision improvement, and sign recovery. We adopt their framework and change candidate screening and local measurements, while retaining the clustering and completion procedures needed for deep layers.

Critical points from different layers are mixed in input space, so measuring signatures before identifying their layers spends queries on irrelevant candidates. Uneven coverage across neurons requires further collection and cross-region clustering, while probes near several activation boundaries can mix contributions and require remeasurement. These costs are especially high when the input dimension greatly exceeds the hidden width, as in the $784\text{-}8^{(8)}\text{-}1$ network studied here.

Geometric Critical Point Screening (GCPS) screens shallow candidates before signature measurement and refines accepted hyperplanes into consistent prefixes for subsequent layers. Later layers use the recovered-prefix Jacobian to restrict measurements to a local subspace, with native repair when consistency checks fail.

- **Geometric screening.** We derive midpoint and coplanarity tests for L1/L2 critical points. Applying these tests before signature measurement reduces queries spent on candidates from other layers, subject to visibility and validation conditions.
- **Layerwise integration.** We combine refined shallow prefixes with guarded Jacobian-based measurements in the end-to-end framework. We derive conditions for compact normal reconstruction, signature reuse, and net query savings.
- **Experimental comparison.** We report recovery, errors, queries, and runtime under true-prefix and assisted end-to-end protocols. The reported end-to-end procedure uses $1.94\times$ fewer queries and $2.00\times$ less recorded time at matched partial recovery of 6593/6643 effective weights.

Section 2 defines the protocols, Section 3 presents the method, and Section 4 reports results, with detailed derivations in Appendices B–C. Appendix D provides the complete layerwise records, while Appendix E retains the original manuscript's separate PReLU comparisons with Chen et al. (2026).

2 Problem and Evaluation Setting

We study parameter recovery for a fully connected network with k ReLU hidden layers, using chosen-input queries to its raw FP64 output. Its forward computation is

$$h_0(x) = x, \quad h_j(x) = \sigma\big(A^{(j)}h_{j-1}(x) + b^{(j)}\big), \quad f(x) = A^{(k+1)}h_k(x) + b^{(k+1)}. \tag{1}$$

Here $A^{(j)} \in \mathbb{R}^{d_j \times d_{j-1}}$ and $b^{(j)} \in \mathbb{R}^{d_j}$ denote layer weights and biases, while $\sigma(t) = \max(t, 0)$ acts coordinatewise on its input. Equation (1) fixes the matrix convention, with hats denoting recovered quantities. A neuron's critical boundary satisfies $a^\top h_{j-1}(x) + \beta = 0$, where $a^\top$ is its incoming weight row and β denotes its bias. An isolated visible boundary changes the output gradient, allowing measurement of its signature up to an unknown scale and orientation. The principal checkpoint is $784\text{-}8^{(8)}\text{-}1$, with layers labeled L1–L9.

True-prefix recovery supplies correct preceding layers independently, whereas the *end-to-end procedure* passes extracted parameters onward, including recovered L1 into L2. Both retain truth-assisted signs and native auxiliary truth, so neither establishes strict black-box sign recovery. Our geometric and compact checks avoid target-layer parameters when choosing measurement directions, accepting candidates, filling missing coordinates, or deciding when to stop. Evaluation aligns hidden-layer

permutations and positive scales, then tests represented weights and biases separately at $\mathrm{rtol} = 10^{-2}$ and $\mathrm{atol} = 10^{-10}$. The effective-weight denominator includes missing weights, and absence from the activation sample does not establish that a neuron is globally dead. Section 4 defines the mask and accounting, while Appendix A gives the oracle model.

3 End-to-End Framework

GCPS screens L1/L2 candidates before signature measurement, then supplies refined estimates as recovered prefixes. Later layers retain Liu's clustering, merging, difficult-point collection, and precision processing (Liu et al., 2026), with local measurements restricted to the recovered-prefix subspace. Appendices B–C give detailed derivations, while Appendix D.8 identifies the branches used by the selected experimental runs, including whole-layer native repairs.

3.1 L1 Screening before Signature Measurement

GCPS searches for kinks on $\ell_q(t) = x_0 + qu + tv$, $q \in \{-s, 0, s\}$, using independent directions u, v, a base point x_0, and spacing $s > 0$. Intersections of these lines with one L1 hyperplane provide a geometric screening test that requires no measurements of the candidate's signature.

Proposition 1 (GCPS midpoint identity)**.** *If $a^\top v \neq 0$, the intersections $p_q = \ell_q(t_q)$ with $a^\top x + \beta = 0$ satisfy*

$$t_q = -\frac{a^\top x_0 + \beta + q\, a^\top u}{a^\top v}, \qquad p_{-s} + p_s = 2p_0. \tag{2}$$

Here t_q is the intersection's line parameter.

Equation (2) provides a necessary L1 relation, but a deeper boundary can also satisfy it inside a single affine activation cell. For localized roots c_q satisfying $\|c_q - p_q\|_2 \leq \epsilon_c$, the numerical screen obeys

$$\|c_{-s} + c_s - 2c_0\|_2 \leq 4\epsilon_c \leq \tau_g. \tag{3}$$

Here ϵ_c bounds localization error and τ_g is the geometric tolerance. Equation (3) relates the tolerance to root accuracy when a localization bound is available. Appendix B.1 proves the relation and discusses degenerate intersections.

We match outer-line pairs against sorted middle-line roots and retain candidates repeated across independent rounds. At an isolated visible L1 kink p, with all other activation states fixed, the scalar output obeys

$$\nabla f_+(p) - \nabla f_-(p) = \lambda a, \qquad \lambda \neq 0. \tag{4}$$

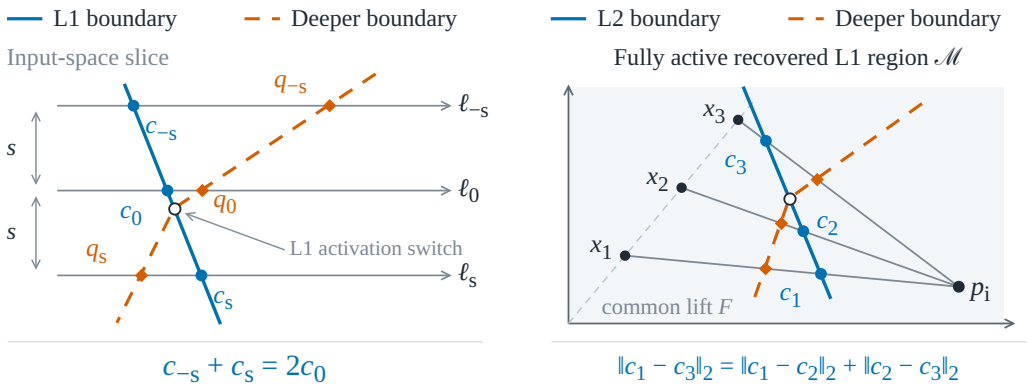
Here f_+, f_- are the affine output restrictions with the target neuron active or inactive, and λ denotes its downstream output coefficient. Equation (4) supports directional finite-difference measurement of a up to scale, whereas three coplanar roots alone leave that normal underdetermined. The resulting plane must fit additional paid roots, $|\widehat{a}^\top c_q + \widehat{\beta}| \leq \tau_r$, where τ_r is the residual tolerance. Figure 1(a) illustrates the geometry, and Appendix B.1 derives the gradient identity.

Corollary 1 (Conditional screening saving)**.** *For a shared pool of N candidates, retain $m \leq N$ after spending q_g screening queries. If each full normal uses $c_{\mathrm{fd}} d_0$ prediction rows, screening saves measurement queries exactly when*

$$Q_{\mathrm{all}} - Q_{\mathrm{screen}} = (N - m) c_{\mathrm{fd}} d_0 - q_g > 0. \tag{5}$$

Here $Q_{\mathrm{all}}, Q_{\mathrm{screen}}$ are costs without/with pre-screening and c_{fd} is the per-direction row cost.

Additional collection can offset the saving in Equation (5) when screening leaves insufficient coverage, as detailed in the comparison in Appendix B.1.



(a) L1 GCPS midpoint test.

(b) L2 GCPS test in one prefix cell.

Figure 1: Blue target-layer boundaries induce bends in the orange deeper-layer boundaries at their intersections, with hollow circles marking the corresponding activation switches. In (a), equally spaced probes retain the L1 midpoint relation, whereas the illustrated deeper boundary fails the same geometric test. The quadrant in (b) is a schematic affine chart of the fully active prefix region, with input-space point labels x_ℓ, p_i, c_j. The common affine lift preserves collinearity and the displayed distance identity, while locally straight deeper boundaries still require measurement and validation.

3.2 Refinement and Prefix Coordinates

For each candidate, we collect paid localized roots $\mathcal{C}$ and refit its hyperplane in recovered feature coordinates, with $\widehat{h}_0(x) = x$. The homogeneous least-squares objective is

$$(\widehat{a}, \widehat{\beta}) \in \operatorname*{arg\,min}_{\|(a^\top, \beta)\|_2 = 1} \sum_{x \in \mathcal{C}} \left(a^\top \widehat{h}_{j-1}(x) + \beta\right)^2. \tag{6}$$

Equation (6) selects the smallest right singular direction of rows $[\widehat{h}_{j-1}(x)^\top, 1]$, identifying a hyperplane when the nullspace is one-dimensional. Insufficient rank leaves partial constraints requiring further measurements. We check singular-value separation and root residuals, then compare the candidate against independent geometric rounds and additional paid held-out probes.

The selected shallow runs request 1024 roots per represented candidate and apply a direct SVD fit, derived in Appendix C.1. This fit serves the same purpose as Liu's precision improvement, but uses a different implementation for root collection and parameter fitting.

In an exact, completely represented prefix, the coordinate convention obeys

$$\widehat{h}_j = G_j h_j, \qquad \widehat{A}^{(j)} = G_j A^{(j)} G_{j-1}^{-1}, \qquad \widehat{b}^{(j)} = G_j b^{(j)}. \tag{7}$$

Here $G_j = D_j P_j$ combines positive diagonal scaling with permutation, with $G_0 = I$ fixing input coordinates. Equation (7) describes equivalent feature coordinates, rather than correcting errors in a measured hyperplane, as derived in Appendix C.2. Negative scales do not commute with ReLU, so we apply the declared truth-assisted orientation separately before exporting measured coordinates to subsequent layers.

3.3 L2 Screening in Recovered L1 Coordinates

Theorem 1 (Fully active recovered-prefix region). *If* $\operatorname{rank} \widehat{A}^{(1)} = d_1$, *the set* $\mathcal{M} = \{x : \widehat{A}^{(1)} x + \widehat{b}^{(1)} > 0\}$ *is nonempty, open, and convex. A common affine lift of positive features is*

$$F(z) = R(z - \widehat{b}^{(1)}) + v_0, \qquad \widehat{A}^{(1)} R = I, \qquad v_0 \in \ker \widehat{A}^{(1)}. \tag{8}$$

Here R is a fixed right inverse and v_0 a fixed nullspace offset.

Equation (8) preserves collinearity and coplanarity, with numerical use requiring full row rank, a well-conditioned lift, and positive activation margins. These conditions concern recovered L1, so

consistency with the target still requires output probes, as explained alongside the proof in Appendix B.2.

We choose collinear interior features and an axis-directed endpoint, then lift the three segments using

$$x_\ell = F(\ell t\mathbf{1}), \quad \ell = 1, 2, 3, \qquad p_i = F(\delta\mathbf{1} + \kappa e_i), \quad t, \delta, \kappa > 0. \tag{9}$$

Here $\mathbf{1}$ is the all-ones vector, e_i a coordinate vector, and t, δ, κ set positive margins and search extent. Equation (9) keeps every segment $[p_i, x_\ell]$ in $\mathcal{M}$.

Corollary 2 (L2 coplanar screen)**.** *For an exactly recovered affine prefix in the gauge of Equation* (7)*, isolated intersections of one L2 boundary with these coplanar segments are collinear. Spatially ordered exact roots satisfy*

$$\big|\|c_1 - c_3\|_2 - \|c_1 - c_2\|_2 - \|c_2 - c_3\|_2\big| = 0. \tag{10}$$

Here c_2 lies between the outer roots c_1, c_3.

We retain triplets satisfying the layer-specific tolerance τ_g in Equation (10), then measure their signatures and test additional points on the proposed planes. Candidates passing these paid probes undergo the refinement in Section 3.2. Full activity exposes incoming coordinates, but deeper boundaries may still pass locally and some L2 boundaries may lie outside the search region. Appendix B.2 gives the proof and validation procedure, while Figure 1(b) illustrates the configuration.

3.4 Measurements in the Full-Prefix Subspace

At each later-layer candidate, differentiating the recovered prefix within its local activation cell gives

$$\widehat{h}_{j-1}(x) = J_j x + c_j, \qquad J_j = M_{j-1}\widehat{A}^{(j-1)} \cdots M_1 \widehat{A}^{(1)}. \tag{11}$$

Here M_i is the diagonal activation mask of layer i, J_j the local Jacobian, and c_j its offset. Equation (11) uses all preceding layers, not only the first weight matrix. A feature-space boundary pulls back to

$$a^\top \widehat{h}_{j-1}(x) + \beta = n^\top x + \gamma, \qquad n = J_j^\top a, \quad \gamma = a^\top c_j + \beta. \tag{12}$$

Here n, γ are the input-space normal and intercept. Equation (12) identifies the subspace in which that normal can be measured.

Proposition 2 (Compact normal and local information)**.** *For a fixed exact affine prefix, take the thin SVD $J_j = U_j \Sigma_j V_j^\top$ of rank r_j. Every pulled-back normal satisfies*

$$n = V_j(V_j^\top n), \qquad r_j \le \min_{1 \le i < j} d_i. \tag{13}$$

Here U_j, V_j have orthonormal columns, and Σ_j contains the positive singular values of the local Jacobian. Feature normals $a + z$, with $z \in \ker J_j^\top$, share the same pulled-back normal, provided the intercept is adjusted by $\beta \mapsto \beta - z^\top c_j$.

At an isolated visible kink, the output-gradient jump is proportional to n, which Equation (13) measures using at most eight directions here. This replaces 784 input directions, although each retained direction still requires finite-difference queries and validation before entering cross-region completion. Rank-deficient observations contribute partial constraints, and Appendix C.3 proves the reconstruction result and examines errors introduced by numerical singular-value truncation.

Corollary 3 (Coverage for paid-signature reuse)**.** *After fixing local scale, let Ea denote the paid linear signature constraints. They uniquely determine the pullback in Equation* (12) *exactly when*

$$\ker E \subseteq \ker J_j^\top. \tag{14}$$

Here E is the measurement operator within the same prefix cell.

Equation (14) is the coverage requirement for signature reuse (A), not a numerical-accuracy certificate. Independent-direction measurement (B) instead lifts a reachable feature direction:

$$\Delta x_i = \epsilon V_j \Sigma_j^{-1} e_i, \qquad J_j \Delta x_i = \epsilon U_j e_i. \tag{15}$$

Here ϵ sets the feature-step scale and $e_i \in \mathbb{R}^{r_j}$ selects a coordinate, with Equation (15) mapping that direction into input space. Small singular values enlarge this displacement, requiring a smaller probe step or native measurement, while unresolved coordinates remain available for cross-region completion. Appendix C.4 derives the lift and proves the reuse condition.

3.5 Validation, Repair, and Query Cost

We check prefix-cell margins under probe displacements, rank and relative-sign stability, and agreement with independent output probes before accepting compact measurements. Failed checks trigger native input-space remeasurement, while small or incomplete components can receive additional native calibration before merging and refinement. All probes and failed attempts count toward query cost, and unresolved coordinates remain available for the inherited cross-region completion procedure.

For same-cell, same-gauge perturbations, the pulled-back error is bounded by

$$\|\Delta n\|_2 \le \|J_j\|_2\|\Delta a\|_2 + \|\Delta J_j\|_2(\|a\|_2 + \|\Delta a\|_2). \tag{16}$$

Here $\Delta J_j, \Delta a$ denote prefix and normal errors, with $\Delta n = (J_j + \Delta J_j)^\top (a + \Delta a) - J_j^\top a$ and matrix norms taken as spectral norms. Equation (16) separates both error sources within a fixed cell, while a mask switch requires a different local Jacobian and bound. Appendix C.3 gives the expansion under these shared-cell and coordinate assumptions.

For $r_j > 0$, with q_v validation rows per point and native-repair fraction ρ, the same per-direction cost c_{fd} gives the compact-to-native comparison

$$S_j = \frac{d_0}{r_j + q_v/c_{\mathrm{fd}} + \rho d_0} > 1 \quad \Longleftrightarrow \quad \rho < 1 - \frac{r_j}{d_0} - \frac{q_v}{c_{\mathrm{fd}} d_0}. \tag{17}$$

Here S_j is the native/compact measurement-cost ratio, and Equation (17) gives the largest fallback fraction compatible with savings in this simplified model. Search and completion add costs outside this comparison, whose assumptions and derivation are given separately in Appendix C.5. The layer returns supported components under the inherited stopping rule, independently of the post-hoc fraction of weights passing target-based verification. The reported end-to-end procedure additionally contains resumed native repairs, as documented in Appendix D.8.

3.6 Affine Output Recovery

On paid inputs $\mathcal{T}$, we fit the affine head to the recovered hidden representation by solving

$$(\widehat{A}^{(k+1)}, \widehat{b}^{(k+1)}) \in \arg\min_{W,b} \sum_{x \in \mathcal{T}} \|f(x) - W\widehat{h}_k(x) - b\|_2^2. \tag{18}$$

Here W, b are candidate output weights and bias, and Equation (18) fits them in the target's fixed output scale. Missing hidden features can bias these coefficients despite small training residuals, motivating separate weight and bias evaluation against fixed denominators. Appendix C.6 derives the linear system and its omitted-feature error.

4 Experiments

Table 1 compares shallow geometric screening across several architectures. The deep-network experiments measure individual layers with correct prefixes (Table 2) and end-to-end recovery (Table 3). Appendix D gives the complete L1–L9 records for both protocols.

4.1 Common Model, Metrics, and Cost Boundaries

The deep-layer comparisons use the public Keras $784\text{-}8^{(8)}\text{-}1$ checkpoint supplied with Liu's research code, evaluated in FP64 with seed 42 (Liu et al., 2026). Native later hidden layers begin with Count=3000, while the reused Liu L1 uses Count=500 and our shallow stages follow geometric collection criteria. Each result comes from one recorded run without multi-seed uncertainty estimates, and Section 2 defines the two protocols for supplying prefixes. The independent Liu reproduction includes later-layer index repairs, whereas the end-to-end comparison retains the native mathematical implementation for its recorded stages.

Recovery and precision. An aligned represented weight $\widehat{w}$ passes when $|\widehat{w} - w| \leq 10^{-10} + 10^{-2}|w|$. We fix a common effective mask using 200,000 inputs uniform on $[-1000, 1000]^{784}$, generated once with fixture seed 20260908. A hidden connection is effective when its incoming feature and outgoing neuron are observed active together in the sample. Input and affine-output coordinates use ungated conventions, giving a denominator of 6643 weights, including missing entries, and 65 separate bias slots. Recovery under this empirical mask measures parameter agreement on sampled activity, rather than establishing global neuron liveness or agreement between model predictions.

Errors compare the extracted parameters against the target after positive-scale and permutation alignment, while keeping the affine output scale fixed. E_{rel} denotes the largest relative ℓ_2 error of a recovered weight–bias signature, evaluated over its represented input coordinates. Unrepresented coordinates remain missing in the recovery count but are excluded from the signature norm, while E_b measures maximum absolute bias error. The appendix reports per-layer errors, and the main text gives their extrema.

Query cost and local runtime. Query count is our primary efficiency metric, with one query defined as one physical input row evaluated by the target. Formal totals charge search, measurement, validation, failed attempts, and repair within selected runs, including each input row submitted within a batch. Interface checks and discarded development trials are accounted for separately. Following Liu et al.'s reporting of queries and local extraction time (Liu et al., 2026), we retain elapsed layer time in seconds as a supplementary measure.

End-to-end totals count each inherited layer once, sum its recorded time, and include the costs of the selected native repair stages. Our method uses GCPS/refinement at L1/L2, compact measurements with A/B at L3–L6, and whole-layer native repairs at L7/L8 under the declared sign assistance. Appendix D.8 documents the L7 sign correction applied before fresh L8/L9 recovery. Speedups divide local Liu reproduction costs by our costs for the same layer and protocol, without estimating remote-service latency.

4.2 Multi-Model L1/L2 Comparison

Table 1 combines historical MNIST (LeCun et al., 1998) and CIFAR-10 (Krizhevsky, 2009) measurements with an earlier supplement (†) and six completed paired runs ($*$). L2 receives true L1, and the $*$ comparisons use repaired Liu code with a shared public CIFAR checkpoint or newly trained MNIST-32 checkpoint. New runs are not paired with historical values, and Appendices D.6–D.7 provide complete recovery metrics, training details, repairs, and earlier attempts.

On CIFAR L1, our method uses $21.45\times$ fewer queries and $23.71\times$ less time, while verifying more weights with lower normal error. MNIST-32 L1 gives $20.46\times$ / $21.07\times$ query/time speedups, although Liu has lower error and our result misses one weight coordinate. At L2, our method verifies 1024/1024 weights versus Liu's 992/1024, with $30.34\times$ / $13.17\times$ speedups and 30/32 verified biases for both methods. These comparisons cover shallow signatures only, with historical PReLU experiments retained in Appendix E.

4.3 Independent True-Prefix Recovery

With independent correct prefixes, our method uses 132,197,083 queries versus Liu's 244,026,401, verifying 6607 versus 6603 effective weights in Table 2. Query and time speedups are $1.85\times$ and $1.04\times$, with most savings occurring in the first four layers of the network. L5 costs more queries and time, while L6/L7 save queries but take longer because measurement savings do not eliminate other recovery costs. Our maximum signature error is 1.10×10^{-3} versus Liu's 9.34×10^{-4}, and L8 remains partial at 41/43 and 40/43 weights, respectively. Appendix D, Table 4, provides complete per-layer counts, errors, and costs.

4.4 End-to-End Extraction

Under the declared sign assistance, our end-to-end procedure uses 84,645,021 queries, compared with 164,584,812 queries for Liu, as shown in Table 3. Summed layer times are 9098.8 s and 18177.4 s, corresponding to $1.94\times$ query and $2.00\times$ local runtime speedups across the reported end-to-end procedure. Both methods verify 6593/6643 weights (99.25%) and 54/65 biases.

Table 1: Shallow-layer query/time speedups: historical results, earlier supplement (†), and completed repair campaign (∗). L2 uses true L1 and is evaluated independently of end-to-end recovery.

Model		L1				L2 (true prefix)			
		$\log_2 Q \downarrow$	Time (s)↓	Query speedup↑	Time speedup↑	$\log_2 Q \downarrow$	Time (s)↓	Query speedup↑	Time speedup↑
MNIST 784-$8^{(8)}$-1	Liu	21.59	204	30.48×	58.29×	24.21	425	849.22×	26.56×
	Our	16.66	3.5			14.48	16		
MNIST 784-$16^{(8)}$-1	Liu	21.60	377	9.99×	15.08×	24.22	1573	147.03×	36.58×
	Our	18.28	25			17.02	43		
CIFAR-10 3072-$8^{(8)}$-1	Liu	23.55	3408	17.75×	40.57×	$26.15^{\dagger}$	$4146.36^{\dagger}$	$3287.02\times^{\dagger}$	$666.66\times^{\dagger}$
	Our	19.40	84			$14.47^{\dagger}$	$6.22^{\dagger}$		
CIFAR-10 3072-$8^{(8)}$-10	Liu	23.55^{*}	7173.25^{*}	$21.45\times^{*}$	$23.71\times^{*}$	26.15	5580	2665.15×	242.61×
	Our	19.13^{*}	302.48^{*}			14.77	23		
MNIST 784-$32^{(8)}$-1	Liu	23.59^{*}	1867.47^{*}	$20.46\times^{*}$	$21.07\times^{*}$	24.28^{*}	1227.41^{*}	$30.34\times^{*}$	$13.17\times^{*}$
	Our	19.24^{*}	88.62^{*}			19.35^{*}	93.19^{*}		

Merged speedups are Liu/Our for the same model/layer and campaign. Historical query speedups are approximate ($2^{\log_2 Q_{\text{Liu}} - \log_2 Q}$). †, ∗ use exact Q. ∗: Liu uses documented candidate/bias repairs. MNIST-32 is a separately trained checkpoint. Ten-output CIFAR-10 L1 weights pass 24532/24576 (Liu) and 24575/24576 (Our), with 8/8 biases each. MNIST-32 L1 passes 25088/25088 and 25087/25088. L2 passes 992/1024 and 1024/1024. Bias counts are 31/32 (L1) and 30/32 (L2) for both. † L2 biases pass 1/8 and 0/8. Full recovery/error data and retained earlier attempts: Appendix D.6–D.7.

Table 2: Independent true-prefix results on the public 784-$8^{(8)}$-1 model, grouped into shallow and deeper hidden layers. Correct preceding layers and true signs are supplied independently. These are not end-to-end extraction results.

Stage		Verified weights ↑	Rate (%) ↑	Verified biases ↑	$\log_2 Q \downarrow$	Time (s) ↓	Query speedup ↑	Time speedup ↑
L1–L2	Liu	6328/6328	100.00	15/16	24.42	1,577.9	122.52×	54.10×
	Our	6328/6328	100.00	15/16	17.49	29.2		
L3–L8	Liu	269/309	87.06	40/48	27.72	22,358.7	1.68×	0.97×
	Our	273/309	88.35	40/48	26.98	22,942.8		
Total	Liu	6603/6643	99.40	56/65	27.86	23,947.8	1.85×	1.04×
	Our	6607/6643	99.46	56/65	26.98	22,983.8		

Total covers L1–L9, including the affine head omitted from stage rows. Recovery uses a fixed mask, $\text{rtol} = 10^{-2}$, and $\text{atol} = 10^{-10}$. Biases are separate and precision is not identical. Full per-layer counts, exact Q, time, errors, and interface costs are in Appendix D, Table 4. Merged speedups are Liu/Our cost ratios for each stage.

L1 contributes 94.42% of the weight denominator, so excluding it reduces recovery to 321/371 (86.52%) and exposes the concentration of deeper-layer omissions.

The maximum hidden-layer signature error is 8.56×10^{-10} for our method and 1.54×10^{-10} for Liu, evaluated only on represented coordinates. Both end-to-end methods retain inaccurate output biases, and the available held-out evaluation of the Liu reproduction falls short of the paper's reported output fidelity. The cost reduction therefore concerns matched *partial parameter recovery*, with complete layerwise comparisons and output-layer evaluation provided in Appendix D, Table 5.

Query costs by stage. L1–L2 require 2,356,633 queries for our method versus 27,285,225 for Liu, giving a 11.58× reduction that includes additional refinement roots (Table 3). Their summed time decreases from 2302.1 s to 226.9 s (10.15×), with both methods verifying all 6328 effective shallow weights and 15/16 biases.

Across L3–L8, our method uses 82,288,301 queries versus Liu's 137,299,500, giving a 1.67× query reduction and 1.79× local runtime speedup. Both methods verify 260/309 effective weights

Table 3: End-to-end extraction results on the public $784\text{-}8^{(8)}\text{-}1$ model, with sign/auxiliary-truth assistance. Hidden-layer stage costs and the end-to-end total are shown.

Stage		Verified weights ↑	Rate (%) ↑	Verified biases ↑	$\log_2 Q \downarrow$	Time (s) ↓	Query speedup ↑	Time speedup ↑
L1–L2	Liu	6328/6328	100.00	15/16	24.70	2,302.1	11.58×	10.15×
	Our	6328/6328	100.00	15/16	21.17	226.9		
L3–L8	Liu	260/309	84.14	39/48	27.03	15,866.6	1.67×	1.79×
	Our	260/309	84.14	39/48	26.29	8,863.3		
Total	Liu	6593/6643	99.25	54/65	27.29	18,177.4	1.94×	2.00×
	Our	6593/6643	99.25	54/65	26.33	9,098.8		

Total covers L1–L9, including the affine head omitted from stage rows. Costs include carried-forward layers and native repairs once, not all development retries. Recovery uses the same fixed mask and tolerances as Table 2. Equal counts do not imply equal precision or complete recovery. Full per-layer counts, exact Q, time, and errors are in Appendix D, Table 5. Merged speedups are Liu/Our cost ratios for each stage.

(84.14%), and these deeper layers account for 97.22% of our method's total query count. L7/L8 costs are nearly unchanged between methods, leaving later-layer search, repair, and completion as the principal remaining expenses in end-to-end extraction.

These costs reflect both the measurement method and recovered prefixes, so isolating compact filtering, A, or B requires matched prefixes and search schedules. Appendix D.8 documents module usage and fallback counts, without isolating individual savings.

4.5 Interpretation and Reproducibility

Fewer measurement directions do not guarantee lower runtime, because calibration, completion, and linear algebra contribute additional costs that vary across layers. The single-checkpoint, single-seed deep-network comparison leaves transfer to other networks untested, while archived hashes, CSV tables, and scripts support checking the reported records. The reproducibility statement identifies the additional saved parameters and checkpoints needed for offline audits and distinguishes these from a fresh extraction.

5 Conclusion

Geometric Critical Point Screening reduces shallow signature measurements, while refinement and consistent coordinates prepare the resulting estimates for use as recovered prefixes. For later layers, the prefix Jacobian reduces measurement dimension, while native repair and cross-region completion handle unreliable checks and incomplete observations.

On the public checkpoint, the reported end-to-end procedure uses $1.94\times$ fewer queries and $2.00\times$ less recorded time, with both methods verifying 6593/6643 effective weights. This sign-assisted comparison retains differences in numerical precision, and neither method recovers an accurate output bias despite matching parameter counts. Future work should address missing deeper-layer components and evaluate the combined procedure across additional models and seeds under comparable recovery protocols.

AI USE STATEMENT

We used generative AI tools, including OpenAI Codex, for research discussions, method development, literature searches, citation handling, and mathematical exposition. Assistance also covered code implementation and debugging, experiment scripting and orchestration, saved-artifact audits, table generation, and manuscript revision. Recorded validation includes synthetic and interface tests, query-counter and file-hash checks, offline parameter audits, citation checks against public sources, and LaTeX compilation and layout inspection. Experimental values come from archived local runs. These checks do not replace independent replication or proof review. The authors take responsibility for the final text, claims, code, citations, and experimental reporting, including all AI-assisted material.

ETHICS STATEMENT

Parameter extraction can compromise model confidentiality and intellectual property, including when applied without authorization. Our experiments use authorized local research models, public research checkpoints, and public MNIST and CIFAR data. They do not target third-party accounts, production services, or external infrastructure. We study extraction cost and numerical recovery to inform model-protection research, and report truth assistance, incomplete recovery, and failures alongside the costs.

REPRODUCIBILITY STATEMENT

Section 3 describes the method, with derivations in Appendices B–C. Appendix D specifies the models, seeds, accounting, tolerances, and layerwise results. Appendix D.8 links the selected results to recorded configurations, code-manifest hashes, and continuations. The accompanying source package includes frozen tabular records, model/source hashes, table-generation scripts, and saved-parameter audit scripts. Rebuilding the tables requires no target queries. Re-auditing parameters also requires the original saved parameters and checkpoints. This is not a self-contained distribution for fresh end-to-end extraction. Its end-to-end records comprise selected completed or resumed stages, as detailed in the experimental protocol.

REFERENCES

Isaac A. Canales-Martínez and David Santos. Extracting some layers of deep neural networks in the hard-label setting. In Daniel Escudero and Ivan Damgård (eds.), *Progress in Cryptology – LATINCRYPT 2025*, pp. 399–421, Cham, 2026. Springer Nature Switzerland.

Isaac A Canales-Martínez, Jorge Chávez-Saab, Anna Hambitzer, Francisco Rodríguez-Henríquez, Nitin Satpute, and Adi Shamir. Polynomial time cryptanalytic extraction of neural network models. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 3–33. Springer, 2024.

Nicholas Carlini, Matthew Jagielski, and Ilya Mironov. Cryptanalytic extraction of neural network models. In *Annual international cryptology conference*, pp. 189–218. Springer, 2020.

Nicholas Carlini, Jorge Chávez-Saab, Anna Hambitzer, Francisco Rodríguez-Henríquez, and Adi Shamir. Polynomial time cryptanalytic extraction of deep neural networks in the hard-label setting. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 364–396. Springer, 2025.

Yi Chen, Xiaoyang Dong, Jian Guo, Yantian Shen, Anyu Wang, and Xiaoyun Wang. Hard-label cryptanalytic extraction of neural network models. In *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 207–236. Springer, 2024.

Yi Chen, Xiaoyang Dong, Ruijie Ma, Yantian Shen, Anyu Wang, Hongbo Yu, and Xiaoyun Wang. Delving into cryptanalytic extraction of PReLU neural networks. In Goichiro Hanaoka and Bo-Yin Yang (eds.), *Advances in Cryptology – ASIACRYPT 2025*, pp. 576–607, Singapore, 2026. Springer Nature Singapore.

Amit Daniely and Elad Granot. An exact poly-time membership-queries algorithm for extracting a three-layer reLU network. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=-CoNloheTs.

Charles Fefferman. Reconstructing a neural net from its output. *Revista Matemática Iberoamericana*, 10(3):507–555, 1994. doi: 10.4171/RMI/160.

Hanna Foerster, Robert Mullins, Ilia Shumailov, and Jamie Hayes. Beyond slow signs in high-fidelity model extraction. *Advances in Neural Information Processing Systems*, 37:19496–19522, 2024.

Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. High accuracy and high fidelity extraction of neural networks. In *29th USENIX Security Symposium (USENIX Security 20)*, pp. 1345–1362. USENIX Association, August 2020. ISBN 978-1-939133-17-5. URL https://www.usenix.org/conference/usenixsecurity20/presentation/jagielski.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://cave.cs.toronto.edu/kriz/learning-features-2009-TR.pdf.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Chun Li, Zheng Gong, Di Li, Liping Zhuang, Yufeng Tang, Yin Lv, and Xingfu Yan. End-to-end polynomial-time cryptanalytic extraction of convolutional neural networks in the hard-label setting. Cryptology ePrint Archive, Paper 2026/902, 2026a. URL https://eprint.iacr.org/2026/902.

Chun Li, Liping Zhuang, Di Li, Yufeng Tang, and Zheng Gong. Polynomial-time cryptanalytic extraction of graph neural networks in the hard-label setting. Cryptology ePrint Archive, Paper 2026/719, 2026b. URL https://eprint.iacr.org/2026/719.

Haolin Liu, Adrien Siproudhis, Samuel Experton, Peter Lorenz, Christina Boura, and Thomas Peyrin. Navigating the deep: End-to-end extraction on deep neural networks. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 482–512. Springer, 2026.

Daniel Lowd and Christopher Meek. Adversarial learning. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pp. 641–647, 2005.

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

Robert Nikolai Reith, Thomas Schneider, and Oleksandr Tkachenko. Efficiently stealing your machine learning models. In *Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society*, pp. 198–210, 2019.

David Rolnick and Konrad Kording. Reverse-engineering deep ReLU networks. In *International conference on machine learning*, pp. 8178–8187. PMLR, 2020.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction APIs. In *25th USENIX security symposium (USENIX Security 16)*, pp. 601–618, 2016.

A Detailed Definitions and Oracle Background

A.1 Basic Definitions and Notation

We use the notation of Carlini et al. (2020) for the network, neuron states, and critical points. The definitions below support the geometric screening and recovered-prefix analysis. Signature extraction and sign recovery follow the protocols described by Carlini et al. (2020) and Canales-Martínez et al. (2024), respectively.

Definition 1 (k-Deep Neural Network (Carlini et al., 2020))**.** *A network with k hidden layers is a piecewise affine function $f : \mathcal{X} \to \mathcal{Y}$ obtained by composing affine maps f_i and activations σ,*

$$f = f_{k+1} \circ \sigma \circ f_k \circ \cdots \circ \sigma \circ f_1.$$

The main analysis uses $\mathcal{X} = \mathbb{R}^{d_0}$, $\mathcal{Y} = \mathbb{R}^{d_{k+1}}$, and ReLU activations $\sigma : x \mapsto \max(x, 0)$. The historical PReLU experiments are reported separately in Appendix E.

Definition 2 (Fully Connected Layer (Carlini et al., 2020))**.** *Layer j ($1 \le j \le k+1$) applies an affine map to the preceding activation vector,*

$$f_j(x) = A^{(j)}x + b^{(j)}.$$

Here $A^{(j)} \in \mathbb{R}^{d_j \times d_{j-1}}$ and $b^{(j)} \in \mathbb{R}^{d_j}$ are its weights and biases. The width is d_j, with d_0 and d_{k+1} denoting input and output dimensions.

Definition 3 (Total Number of Neurons (Carlini et al., 2020))**.** *The total number of hidden neurons is*

$$N = \sum_{j=1}^{k} d_j.$$

The output layer does not contain ReLU activation functions and thus is not counted in the total number of neurons.

Definition 4 (Neural Network Architecture (Carlini et al., 2020))**.** *We specify an architecture by*

1. *The total number of hidden layers k of the network;*
2. *The dimension sequence of each layer $\{d_0, d_1, \ldots, d_k, d_{k+1}\}$;*
3. *Any additional constraints imposed on the weights $A^{(j)}$ and biases $b^{(j)}$ of each layer (such as sparsity constraints for convolutional layers).*

We list layer widths in input-to-output order. Thus 784-32-1 *has one hidden layer of width 32. A parenthesized exponent denotes repeated hidden layers, so $784 - 8^{(8)} - 1$ has eight width-8 hidden layers.*

Definition 5 (Neuron State and Value (Carlini et al., 2020))**.** *For any neuron η in the j-th layer ($1 \le j \le k$):*

1. *$V(\eta; x)$ denotes the input value of this neuron before ReLU activation when the input is x;*
2. *$S(\eta; x)$ denotes the activation state of this neuron, defined as:*

$$S(\eta; x) = \sigma(V(\eta; x)) = \max(V(\eta; x), 0).$$

The neuron is active when $S(\eta; x) > 0$ and inactive when $S(\eta; x) = 0$.

Definition 6 (Critical points (Carlini et al., 2020))**.** *For any neuron η in the j-th layer, an input x is called a critical point of neuron η, denoted as $x \in W(\eta)$, if it satisfies*

$$V(\eta; x) = 0.$$

A neuron's activation state can change only across its critical set. Such a change produces an output kink only when its downstream contribution is visible.

Definition 7 (Critical Hyperplane (Carlini et al., 2020))**.** *For a neuron $\eta_{j,i}$, its critical set $W(\eta_{j,i})$ satisfies an affine equation in the neuron's incoming feature coordinates,*

$$A_i^{(j)} X_{j-1} + b_i^{(j)} = 0$$

where X_{j-1} is the input to the j-th layer.

1. *When $j = 1$, $X_0 = x$, so $W(\eta_{1,i})$ is a global affine hyperplane in $\mathbb{R}^{d_0}$;*
2. *When $j \geq 2$, $X_{j-1} = \sigma(f_{j-1}(X_{j-2}))$ is piecewise affine in x. The critical set is therefore a union of affine pieces in input space, with geometry determined by the preceding activation masks.*

A.2 Carlini et al.'s Attack Method

The relevant parts of Carlini et al.'s method are critical-point search and signature grouping.

A.2.1 Random Search for Critical Points

A ReLU network is affine on a segment where no activation state changes. Its midpoint output then equals the mean of the endpoint outputs. A discrepancy indicates a boundary crossing, subject to floating-point tolerance; agreement alone may miss boundaries whose effects cancel. The search samples a starting point and direction, queries the segment endpoints and midpoint, and subdivides intervals with detectable nonlinearity. Repeated localization narrows an interval around a visible critical point to the required precision.

A.2.2 Signature Extraction and Compute First, Cluster Later Strategy

At an isolated visible critical point, one activation changes under a sufficiently small perturbation. The jump between the affine output gradients on its two sides is proportional to the neuron's input-space normal, with a proportionality factor determined by the downstream network. Finite differences estimate this jump and reveal weight ratios. ReLU has no classical second derivative at the kink; references to second-order measurements here mean finite-difference probes across that boundary. Carlini et al. measure candidate signatures before grouping points by neuron. The grouping uses two local geometric properties.

- Critical points of the same target-layer neuron satisfy one affine equation in that layer's input coordinates. Their measured signatures should agree on overlapping identifiable coordinates.
- Deeper-layer boundaries can change direction across activation cells, producing inconsistent apparent signatures in the current layer's coordinates. Within one cell, however, they may remain straight and yield repeated signatures.

Repeated compatible observations support a candidate neuron, but repetition by itself is insufficient to establish layer membership. Measuring candidates before grouping them incurs a directional-probe cost for every point. The two- and four-query coordinate stencils cost $2d_{j-1}$ and $4d_{j-1}$ rows, respectively, before search, validation, and retries. GCPS reduces the number of shallow candidates sent to this measurement stage. The combined method retains deeper-layer clustering.

A.3 Adversarial Objectives and Resources

A model extraction attack involves two parties: an oracle $\mathcal{O}$ that returns model outputs $f_\theta(x)$, and an attacker who generates queries x to the oracle.

Definition 8. *A model signature extraction attack obtains oracle access to a parameterized function f_θ (a k-layer deep neural network in this paper) and the architecture information of function f, and returns a set of parameters $\hat{\theta}$. Its goal is to achieve functional equivalence, i.e., $f_\theta(x) = f_{\hat{\theta}}(x)$ for all inputs x.*

Throughout this paper, we use the $\hat{\cdot}$ notation to denote extracted parameters. For example, $\hat{\theta}$ denotes the weight parameters extracted from the original model θ. We assess recovery quality together with the number of chosen-input queries and the recorded computation time. Cost alone does not determine whether extraction succeeded.

A.3.1 Attack Assumptions

We make the following assumptions about the oracle $\mathcal{O}$ and the attacker's knowledge:

- **Known Architecture**: The attacker knows the complete architecture information of the target neural network, including network depth, number of neurons per layer, and inter-layer connection patterns.
- **Full Domain Input**: The attacker can select any real vector from the complete input space $\mathcal{X}$ as a query input.
- **Full Output Access**: Each query returns the raw floating-point output of f, rather than only its predicted class.
- **Floating-Point Computation**: The model f is defined and computed using 64-bit double-precision floating-point arithmetic; this does not imply exact real arithmetic.
- **ReLU Activation Function**: All hidden layers use the ReLU activation function $\sigma(\cdot)$.

The oracle model describes the underlying extraction problem. Our experiments additionally declare their sign assistance, auxiliary truth use, and prefix protocol in Section 4; those experimental assumptions must not be inferred from oracle access alone.

B Detailed Derivations for Shallow Geometry

This appendix derives the L1 midpoint and L2 collinearity tests and states the conditions needed for their numerical use.

B.1 L1 Critical-Point Screening

From a visible kink to a signature. With all other activation states fixed, a scalar-output network near an L1 boundary can be written as an affine function plus $\lambda \max(a^\top x + \beta, 0)$. Differentiating its two affine restrictions gives Equation (4). The condition $\lambda \neq 0$ expresses local output visibility. If multiple boundaries switch, their gradient changes can mix, so this single-normal identity cannot be applied without isolating the kink.

Proof of Corollary 1. Let Q_0 denote the shared cost of collecting the candidate pool. Measuring every candidate costs $Q_{\text{all}} = Q_0 + N c_{\text{fd}} d_0$, while screening first costs $Q_{\text{screen}} = Q_0 + q_g + m c_{\text{fd}} d_0$. Subtracting gives Equation (5); the right-hand side is positive precisely when the saved full-normal measurements exceed the screening cost. This accounting statement neither bounds false rejections nor asserts that the two actual search schedules share the same pool. Additional search, validation or completion queries must be included when applying it to a complete run. □

Proof of Proposition 1 and the localization bound. Write an L1 hyperplane as $a^\top x + \beta = 0$ and the three lines as $x_0 + qu + tv$, with $q \in \{-s, 0, s\}$. For $a^\top v \neq 0$, their root parameter is

$$t(q) = -\frac{a^\top x_0 + \beta + q\, a^\top u}{a^\top v}. \tag{B.1}$$

Here q indexes the line offset and $t(q)$ is its intersection parameter. Equation (B.1) is affine in q, hence $t(-s) + t(s) = 2t(0)$. Substitution in the input points $p_q = x_0 + qu + t(q)v$ proves Equation (2). Write $c_q = p_q + e_q$ with $\|e_q\|_2 \leq \epsilon_c$. Then the exact midpoint terms cancel and the numerical residual is bounded by

$$\|c_{-s} + c_s - 2c_0\|_2 = \|e_{-s} + e_s - 2e_0\|_2 \leq \|e_{-s}\|_2 + \|e_s\|_2 + 2\|e_0\|_2 \leq 4\epsilon_c.$$

Choosing $\tau_g \geq 4\epsilon_c$ gives Equation (3). This conditional bound motivates a localization-dependent tolerance; it does not provide a known error bound for every measured root. □

If $a^\top v = 0$, a sampling line either never intersects the hyperplane or lies entirely in it, so it does not yield an isolated transverse root. Even when transverse, three points restricted to a two-dimensional plane cannot identify a full input-space hyperplane in a high-dimensional model. A deeper boundary confined to one affine cell also satisfies the same local identity. The screen is therefore a necessary geometric test for an L1 candidate, followed by directional measurement and validation, rather than a sufficient layer-membership classifier.

Random search in input space can encounter critical points from any layer. L1 screening exploits the fact that an L1 critical set is a global hyperplane, whereas a deeper critical set may change direction across prefix cells. The following notation expresses this difference in the coordinates used for extraction.

Definition 9. *Let $f_{1..j}$ denote the network prefix through the affine map f_j, excluding its final activation. In particular, $f = f_{1..k+1}$.*

At a critical point of neuron $\eta_{j,i}$, the preactivation vanishes,

$$V(\eta_{j,i}; x) = A_i^{(j)} \sigma(f_{1..j-1}(x)) + b_i^{(j)} = 0.$$

For a neuron $\eta_{l,m}$ in a later layer $l > j$, the corresponding condition is

$$V(\eta_{l,m}; x) = A_m^{(l)} f_{j..l-1}(\sigma(f_{1..j-1}(x))) + b_m^{(l)} = 0.$$

The target-layer equation is affine in X_{j-1}. For $l > j$, intervening ReLUs make the boundary piecewise affine, and its intersection with a sampling plane can bend when an intervening activation switches.

B.1.1 Linear Screening Algorithm

For L1, the prefix is the identity and critical points satisfy

$$A_j^{(1)} x + b_j^{(1)} = 0.$$

This affine hyperplane intersects a two-dimensional sampling plane in a line, except when the plane is parallel to or contained in the hyperplane. Figure 1 illustrates the construction. We sample a reference point x_0 near the origin and two linearly independent unit vectors $u, v \in \mathbb{R}^{d_0}$. A spacing s gives three equidistant reference points,

$$x_1 = x_0 - su, \quad x_2 = x_0, \quad x_3 = x_0 + su.$$

Searching in both directions along v from these points gives root sets C_1, C_2, C_3. Larger spacing may expose changes in deeper boundaries, but no finite choice guarantees this. If $c_1 \in C_1, c_2 \in C_2, c_3 \in C_3$ belong to one L1 neuron, the midpoint identity motivates the numerical test

$$\|\vec{c_1} + \vec{c_3} - 2\vec{c_2}\|_2 < \tau.$$

Here τ is the numerical tolerance. To reduce false positives from locally straight deeper boundaries, we require the same estimated parameters in at least two independent screening rounds before further validation.

B.1.2 Consistency Checks Using Paid Roots

Multiple activation boundaries can lie within the probe neighborhood of a critical point. A finite-difference measurement can then mix contributions from different neurons and produce an incorrect signature. Geometric checks on independently localized roots help detect this contamination before the candidate is propagated. Each retained triplet (c_1, c_2, c_3) provides a consistency check. After estimating $\hat{A}_j^{(1)}$ and $\hat{b}_j^{(1)}$ at c_1, we test the other two paid roots against the fitted plane,

$$|\hat{A}_j^{(1)} c_2 + \hat{b}_j^{(1)}| < \tau, \quad |\hat{A}_j^{(1)} c_3 + \hat{b}_j^{(1)}| < \tau.$$

An excessive residual can result from mixed boundaries, root-localization error, or signature-measurement error. This check reuses paid roots and therefore adds no queries, although it does not distinguish those causes.

B.1.3 Measurement and Matching Costs

Suppose S candidate roots have been collected. A coordinate finite-difference signature measurement costs $O(d_0)$ query rows per candidate, or $O(Sd_0)$ for a measure-then-cluster strategy. Uniform coupon-collector reasoning motivates a dependence on the number of neuron types, but actual critical-point sampling is neither necessarily uniform nor guaranteed to encounter every boundary. Our screening measures only the retained candidates. If there are $O(d_1)$ such candidates, this part of

the cost is $O(d_1 d_0)$. It excludes the cost of collecting the roots, validating false positives, and refining the final signatures; it is not an unconditional bound on the complete extraction. If each sampled line contains at most m roots, sorting the middle-line roots and using nearest-neighbor lookup for each pair from the outer lines gives $O(m^2 \log m)$ geometric matching time. The two-dimensional line parameterization avoids computing a full signature for every possible triplet. Repeated candidates and locally straight deeper boundaries still require validation.

Numerical Accuracy On isolated points, GCPS uses the underlying cryptanalytic signature equations. Their numerical error still depends on root localization, finite-difference scale, conditioning, and prefix perturbations. The refinement in Section 3.2 addresses these errors before chaining; all additional root and validation queries are charged.

B.2 L2 Critical-Point Screening

L2 screening uses a region where the recovered first layer is fully active. The construction requires full row rank, which implies $d_0 \geq d_1$.

B.2.1 Existence of a Fully Active Region

Theorem 1 applies to any full-row-rank affine first layer, including the recovered one used in the chain. We first prove the region claim for a general first layer, then specialize the lifting map to recovered coordinates.

Proof of Theorem 1. The first hidden layer linear transformation can be expressed as

$$x_1 = A^{(1)} x_0 + b^{(1)}$$

Here $x_0 \in \mathbb{R}^{d_0}$ is the input and $x_1 \in \mathbb{R}^{d_1}$ the first-layer preactivation. Full row rank makes the affine map surjective. The positive orthant $M_1 = (0, +\infty)^{d_1}$ therefore has a nonempty preimage $M_0 = \{x_0 : A^{(1)} x_0 + b^{(1)} \in M_1\}$. Continuity makes M_0 open, and affinity makes it convex because M_1 is convex. Every first-layer preactivation is strictly positive on M_0, where ReLU acts as the identity. For the recovered layer, replace $A^{(1)}, b^{(1)}$ throughout by $\widehat{A}^{(1)}, \widehat{b}^{(1)}$, obtaining $\mathcal{M}$ in the main statement. Full row rank supplies a right inverse R, and any fixed $v_0 \in \ker \widehat{A}^{(1)}$ gives

$$\widehat{A}^{(1)} F(z) + \widehat{b}^{(1)} = \widehat{A}^{(1)} R(z - \widehat{b}^{(1)}) + \widehat{A}^{(1)} v_0 + \widehat{b}^{(1)} = z.$$

This proves Equation (8); a positive feature margin is preserved in recovered feature coordinates. It is not a guarantee about the target cell when the recovered parameters are inaccurate. □

Within this fully activated space M_0, the first two preactivation maps compose to an affine function

$$f_{1..2}(x) = A^{(2)}(A^{(1)} x + b^{(1)}) + b^{(2)}.$$

Consequently, an L2 boundary intersects a two-dimensional sampling plane in a line when the intersection is nondegenerate. For later layers, L2 activation switches can bend the boundary even inside M_0, although a locally straight portion can still pass the same screen. Figure 1 illustrates how the affine region supports the L2 screen.

B.2.2 Screening in Recovered Feature Coordinates

Normalization in the Recovered First-Layer Coordinates The extracted hyperplane determines its normal and bias only up to scale. We choose a positive normalization $\alpha_i = 1/\|\widehat{A}_i^{(1)}\|_2$ for each nonzero row and multiply both its normal and bias by α_i. This gives a signed-distance convention once the orientation is fixed; it does not recover the target's original scale. All following lifting equations use this recovered and consistently oriented first layer. Coordinate changes are propagated as described in Section 3. We use a fixed affine lifting map $x = R(z - \widehat{b}^{(1)}) + v_0$, where R is a right inverse of the recovered first-layer matrix and $v_0 \in \ker(\widehat{A}^{(1)})$ is fixed throughout the construction. This common lift preserves collinearity and coplanarity. Independently chosen nullspace offsets would lose that property. If the right inverse is poorly conditioned, the method needs a different

region or further refinement of the recovered layer. Writing $\mathbf{1}_d$ for the all-ones vector and $\mathbf{e}_i$ for the ith coordinate vector, the common lift gives three points $x_1, x_2, x_3 \in \mathbb{R}^{d_0}$ satisfying

$$\widehat{A}^{(1)}x_1 + \widehat{b}^{(1)} = t\mathbf{1}_{d_1}, \quad \widehat{A}^{(1)}x_2 + \widehat{b}^{(1)} = 2t\mathbf{1}_{d_1}, \quad \widehat{A}^{(1)}x_3 + \widehat{b}^{(1)} = 3t\mathbf{1}_{d_1}.$$

Here $t > 0$ sets the interior feature scale. The endpoint p_i satisfies $\widehat{A}^{(1)}p_i + \widehat{b}^{(1)} = \delta\mathbf{1}_{d_1} + \kappa\mathbf{e}_i$, where $\delta > 0$ sets the activation margin and $\kappa > 0$ the search extent. The common lift and positivity keep the segments $\overline{p_i x_1}, \overline{p_i x_2}, \overline{p_i x_3}$ coplanar and inside the recovered fully active region. Searching them gives root sets C_1, C_2, C_3. Roots from one L2 neuron are collinear under an exact prefix. We order each triplet spatially, with c_2 between c_1, c_3, and apply the distance test

$$|\|\overrightarrow{c_1c_3}\| - (\|\overrightarrow{c_1c_2}\| + \|\overrightarrow{c_2c_3}\|)| < \tau.$$

Proof of Corollary 2. The feature endpoints in Equation (9) belong to the plane spanned by $\mathbf{1}$ and e_i. Their convex combinations remain strictly positive, and the common affine map F preserves their plane and segments. With an exactly recovered prefix in a consistent gauge, a target L2 boundary is an affine equation $a^\top z + \beta = 0$ in these coordinates. Its nondegenerate intersection with the feature plane is a line, and F maps that line to a line. Therefore its three segment intersections are collinear in input space. For points in spatial order, write $c_2 = (1-\lambda)c_1 + \lambda c_3$ with $0 \le \lambda \le 1$. Then

$$\|c_1 - c_2\|_2 + \|c_2 - c_3\|_2 = \lambda\|c_1 - c_3\|_2 + (1-\lambda)\|c_1 - c_3\|_2 = \|c_1 - c_3\|_2,$$

which gives Equation (10) with zero residual in exact arithmetic. If each root has error at most ϵ_c, the perturbation of the three-distance residual is at most $6\epsilon_c$, by the reverse triangle inequality applied to each distance. A tolerance for L2 must account for this bound as well as prefix error; it need not equal the numerical value used for L1. A plane contained in a boundary does not provide isolated transverse intersections, and a deeper boundary can obey the same identity inside one affine cell. □

Additional Output Probes Deeper-layer roots may also satisfy the collinearity test. For each retained triplet (c_1, c_2, c_3), we estimate $\hat{w}, \hat{b}$, sample $z_1 \in M_1$ satisfying $\hat{w}z_1 + \hat{b} = 0$, and lift it to x_0 with $\sigma(\widehat{A}^{(1)}x_0 + \widehat{b}^{(1)}) = z_1$. Output probes then test whether the proposed plane contains a visible target boundary at this additional point. We solve $\widehat{A}^{(1)}\hat{w}' = \hat{w}$ for $\hat{w}' \in \mathbb{R}^{d_0}$ and choose a step $\epsilon > 0$ that preserves the recovered prefix cell. If the second finite difference satisfies

$$|f(x_0 + \epsilon\hat{w}') - 2f(x_0) + f(x_0 - \epsilon\hat{w}')| < \tau,$$

the kink signal is too weak at the chosen scale. Numerical error, weak downstream influence, or cancellation can cause this even at a critical point. The original geometric implementation tests 20 constructed points and requires at least 15 positive checks before the refinement in Section 3.2.

B.2.3 L2 Measurement Costs

For the $784 - 8^{(8)} - 1$ network, a four-query stencil over the eight incoming coordinates costs $4 \times 8 = 32$ rows per candidate, before search and validation. A random point may leave some L1 features inactive, so recovering the full L2 signature can require complementary regions. In a correctly identified fully active region, one isolated, visible, and accurately measured point can determine an L2 neuron's incoming signature. GCPS exploits this coverage to reduce the need for partial-signature merging. Its actual cost still includes repeated candidates, false positives, additional search, and refinement. Boundaries outside the region remain unobserved, and errors or poor conditioning in the recovered L1 can invalidate the lift. Section 4 reports the independent and recovered-prefix costs separately.

B.2.4 Piecewise Linear Activations

The shallow geometric argument can also be formulated for activations such as Leaky ReLU and PReLU (Chen et al., 2026): a fixed prefix cell is affine and each neuron boundary is affine in its incoming features. Changing an activation's slopes generally changes the deeper input-space boundaries and the numerical contrast at a kink, so unchanged thresholds and identical behavior are not guaranteed. The new chain comparison in Section 4 is restricted to ReLU; legacy PReLU results are retained in Appendix E.

C Detailed Refinement and Recovered-Prefix Analysis

This appendix derives the refinement, coordinate transformations, and local measurement rules in the end-to-end framework. The analysis is local to a prefix activation cell. To recover coordinates invisible in one cell, the framework retains the cross-region clustering, merging, difficult-point collection, and precision processing of Liu et al. (2026).

C.1 Shallow-Layer Geometry and Numerical Refinement

The screens in Sections 3.1 and 3.3 select candidate points; refinement then improves their hyperplane estimates. In the recovered-prefix chain, L2 uses the recovered L1 for both lifting and feature evaluation. With $\widehat{h}_0(x) = x$, let $\widehat{h}_j(x)$ denote the recovered first j layers. We search where recovered L1 activation margins are positive and check local output consistency because the target cell may differ.

For a layer-j candidate, the refinement objective from Equation (6) is

$$(\widehat{a}, \widehat{\beta}) \in \underset{\|(a^\top,\beta)\|_2=1}{\arg\min} \sum_{x\in\mathcal{C}} \big(a^\top \widehat{h}_{j-1}(x) + \beta\big)^2. \tag{C.1}$$

Here $\mathcal{C}$ contains the paid, localized roots assigned to the candidate, a is its normal in recovered feature coordinates, and β its intercept. Stacking the rows $[\widehat{h}_{j-1}(x)^\top, 1]$ into H and writing $\theta = (a^\top, \beta)^\top$ makes Equation (C.1) equal to minimizing $\|H\theta\|_2^2 = \theta^\top H^\top H\theta$ under $\|\theta\|_2 = 1$. A smallest right singular vector solves this problem. A one-dimensional nullspace identifies the plane up to scale and orientation; several small singular values leave its coefficients poorly determined even at a small residual.

Our implementation collects roots by locally guided search and fits them directly by SVD, rather than invoking Liu's precision solver unchanged. Acceptance requires singular-value separation, small root residuals, agreement across independent geometric rounds, and paid held-out probes. Rank-deficient fits retain unresolved directions.

Refinement changes the estimated hyperplane, whereas permutation and positive scaling change only its representation. An accurate fit can still have the wrong orientation. We therefore refine the geometry before applying the declared sign protocol and exporting the prefix in a consistent coordinate convention. All additional localization and validation queries are charged.

C.2 Maintaining the Recovered-Prefix Gauge

Positive monomial transformations commute with ReLU. For a permutation matrix P and positive diagonal D, coordinatewise ReLU satisfies $\sigma(DPz) = DP\sigma(z)$. Assume inductively that $\widehat{h}_{j-1} = G_{j-1}h_{j-1}$. Substituting $\widehat{A}^{(j)} = G_j A^{(j)} G_{j-1}^{-1}$ and $\widehat{b}^{(j)} = G_j b^{(j)}$ in the recovered preactivation gives $G_j(A^{(j)}h_{j-1} + b^{(j)})$. Commutation then gives $\widehat{h}_j = G_j h_j$, proving the ideal relation in Equation (7). This argument assumes represented features and consistent orientation; it cannot compensate for a missing neuron or an inaccurate normal. Negative diagonal entries do not satisfy the same commutation identity, which is why sign recovery is a separate operation.

For an exactly recovered hidden representation, permutation and positive scaling can be written as

$$\widehat{h}_j(x) = G_j h_j(x), \qquad G_j = D_j P_j. \tag{C.2}$$

Here $h_j(x)$ is the target hidden activation vector, P_j is a permutation matrix, and D_j is a positive diagonal matrix. Equation (C.2) describes a representation equivalence, not an operation that requires access to target weights. With $G_0 = I$, compatible adjacent-layer parameters obey

$$\widehat{A}^{(j)} = G_j A^{(j)} G_{j-1}^{-1}, \qquad \widehat{b}^{(j)} = G_j b^{(j)}. \tag{C.3}$$

Equation (C.3) explains why a change in one layer's ordering or scale must be propagated to the next layer's input coordinates. In practice we store the recovered ordering and normalization and measure subsequent layers directly in that coordinate system. Missing neurons or unresolved coordinates are recorded explicitly; the equivalence does not hold on an unobserved feature merely because its stored value is zero.

The experiments resolve orientation using the declared Liu-compatible truth-assisted sign protocol and retain native auxiliary truth. Post-hoc matching to the target is used for evaluation; it is separate from the geometric and compact-measurement checks. This assisted protocol does not independently solve black-box sign recovery.

C.3 Full-Prefix Jacobian Measurements

Proof of Proposition 2. Inside a fixed prefix cell, each ReLU derivative is a fixed diagonal zero–one matrix. Applying the chain rule gives the masked product below. The rank of a matrix product is no larger than the rank of any factor, hence no larger than the width of the narrowest preceding hidden layer. For a thin SVD $J_j = U_j \Sigma_j V_j^\top$, $J_j^\top a = V_j \Sigma_j U_j^\top a$ lies in the span of V_j. Since $V_j^\top V_j = I$, projection onto that span loses no information about this pulled-back normal. This proves Equation (13), using the full masked product in Equation (11).

In contrast, $J_j^\top (a + z) = J_j^\top a$ for every $z \in \ker(J_j^\top)$. Thus local normal measurements cannot distinguish feature-space normals along that nullspace. To preserve the complete affine boundary, including its intercept, pair $a + z$ with $\beta - z^\top c_j$: substituting $\widehat{h}_{j-1}(x) = J_j x + c_j$ cancels the added $z^\top c_j$. Leaving the bias fixed need not preserve the boundary. Additional cells can contribute different Jacobians and constrain previously invisible components. The argument requires a stable local prefix cell; it is not a global assertion that every target normal is captured by an approximate recovered Jacobian. □

The proof uses the exact nonzero singular subspace. If numerical truncation retains only columns $\widetilde{V}_j$, the discarded normal is $(I - \widetilde{V}_j \widetilde{V}_j^\top) J_j^\top a$ and need not vanish. Small discarded singular values alone do not certify a small relative error when the retained normal is also weak. Cell, rank, and independent-probe checks therefore accompany the practical compact measurement.

Effect of prefix and signature error. After fixing the same coordinate gauge, let a local Jacobian estimate be $J + \Delta J$ and a feature-normal estimate be $a + \Delta a$. Their pulled-back error satisfies

$$\|(J + \Delta J)^\top (a + \Delta a) - J^\top a\|_2 \le \|J\|_2 \|\Delta a\|_2 + \|\Delta J\|_2 \big(\|a\|_2 + \|\Delta a\|_2\big). \tag{C.4}$$

Here ΔJ and Δa denote perturbations of the local map and feature normal, respectively, and the matrix norms are spectral norms. To obtain Equation (C.4), and hence Equation (16), expand the difference as $J^\top \Delta a + \Delta J^\top a + \Delta J^\top \Delta a$, bound each term using the spectral norm, and apply the triangle inequality. It separates refinement error from prefix error, but assumes that the compared maps describe the same local cell. Crossing an activation boundary can change a mask discontinuously, which motivates checking cell margins rather than relying on this bound alone.

Within a region with a fixed recovered-prefix activation pattern, the prefix is affine:

$$\widehat{h}_{j-1}(x) = J_j x + c_j, \qquad J_j = M_{j-1} \widehat{A}^{(j-1)} \cdots M_1 \widehat{A}^{(1)}. \tag{C.5}$$

Here $J_j \in \mathbb{R}^{d_{j-1} \times d_0}$ is the full recovered-prefix Jacobian, c_j is its local offset, and M_i is the diagonal ReLU activation mask of recovered layer i. Equation (C.5) uses every preceding layer and mask, rather than only the first weight matrix. A candidate hyperplane $a^\top \widehat{h}_{j-1}(x) + \beta = 0$ consequently has the input-space form

$$n^\top x + \gamma = 0, \qquad n = J_j^\top a, \qquad \gamma = a^\top c_j + \beta. \tag{C.6}$$

Here $n \in \mathbb{R}^{d_0}$ and γ are the pulled-back normal and intercept. Equation (C.6) is exact for the recovered affine prefix and is an approximation to the target geometry when that prefix has numerical error.

For the exact identity, take a thin singular value decomposition $J_j = U_j \Sigma_j V_j^\top$ containing all r_j positive singular values. Numerical truncation requires the additional checks described above. The columns of $V_j \in \mathbb{R}^{d_0 \times r_j}$ span the input directions relevant to this prefix. Since n lies in that span, its directional measurements determine it through

$$n = V_j \big(V_j^\top n\big), \qquad r_j \le \min_{1 \le i < j} d_i. \tag{C.7}$$

Equation (C.7) gives the dimensional reduction: for the evaluated 784-$8^{(8)}$-1 network, $r_j \leq 8$ after L1. A local scalar output-gradient jump at a single visible ReLU boundary is proportional to n. Measuring that jump along the columns of V_j therefore needs $O(r_j)$ directional probes instead of $O(d_0)$, up to validation and finite-difference constants. L1 itself has no such recovered bottleneck; its savings come from screening candidates before measuring them.

The columns of V_j span the input row space $\mathrm{range}(J_j^\top)$, whereas the columns of U_j span the reachable hidden space $\mathrm{range}(J_j)$. Rank-deficient observations constrain only the projection of a onto the latter. We retain these partial constraints and seek complementary regions; target parameters never fill the nullspace.

C.4 Reuse, Independent Directions, and Guarded Repair

We use the same local geometry for two optional optimizations. Signature reuse (A) pulls back an already paid signature through Equation (C.6) when its constraints cover the required directions within the same cell. Independent-direction measurement (B) probes a basis of the reachable hidden subspace and retains redundant observations for validation. Both feed their constraints to Liu's cross-region completion process. The tables identify the combined configuration as *Our*.

Proof of Corollary 3. Fix the local normalization and let E encode the paid constraints Ea. Two feasible normals differ by a vector in $\ker E$. If $\ker E \subseteq \ker J_j^\top$, the pullback annihilates every such difference, proving sufficiency. Conversely, if some $z \in \ker E$ satisfies $J_j^\top z \neq 0$, then a and $a + z$ give identical observations but different pullbacks. This proves necessity and hence Equation (14). For a coordinate mask, unresolved coordinates must contribute nothing to the required input normal; otherwise additional measurements are needed. The condition establishes coverage, while noisy constraints and uncertain rank still require validation. □

Lifting independent feature directions. For the retained thin SVD $J_j = U_j \Sigma_j V_j^\top$, let e_i be the ith coordinate vector in the rank-r_j space. The input displacement $v_i = V_j \Sigma_j^{-1} e_i$ satisfies $J_j v_i = U_j e_i$. It realizes one independent reachable feature direction while the affine cell remains fixed. Multiplying by a step ϵ yields Equation (15); the actual input norm is $|\epsilon|$ divided by the corresponding singular value. Small singular values therefore magnify the input displacement needed for a fixed feature-space step, which can violate the cell assumption or amplify finite-difference error. A numerical implementation must truncate unstable directions, adapt probe scale, or use native measurement. Components orthogonal to $\mathrm{range}(J_j)$ remain unidentifiable in this cell and need complementary regions.

In compact mode, native d_0-dimensional measurement is triggered by failed local checks. These examine prefix-cell margins under the actual displacements, rank stability under nearby step sizes, relative directional signs, and agreement with independent paid probes. They can reveal inconsistencies before the method measures a full 784D normal, although they do not certify the target prefix. Failed native measurements are charged as well. Small components can receive additional calibration points before merging and refinement, and unresolved coordinates remain marked as such. The selected chain also includes inherited checkpoints and whole-layer native repair runs; Appendix D.8 distinguishes these from pointwise fallback.

C.5 Query Accounting and the Net-Saving Condition

For m measured points, let q_v be the validation rows per point and ρ the fraction requiring native remeasurement. The local measurement cost is approximated by

$$Q_{\mathrm{meas}} \approx m\big(c_{\mathrm{fd}} r_j + q_v + \rho c_{\mathrm{fd}} d_0\big). \tag{C.8}$$

Here c_{fd} is the number of oracle rows per finite-difference direction and $\rho \in [0, 1]$ is the fallback fraction. Equation (C.8) covers measurement and validation; a full run also pays for search, completion, refinement, and unsuccessful attempts. Linear algebra adds time without adding target queries.

Under this simplified constant-rank measurement model, a native measurement of the same m points costs $mc_{\mathrm{fd}}d_0$. Dividing this cost by the compact cost in Equation (C.8) cancels $m > 0$ and gives S_j in Equation (17). For $c_{\mathrm{fd}} > 0$ and positive measurement cost, requiring $S_j > 1$ gives the equivalent fallback threshold

$$\rho < 1 - \frac{r_j}{d_0} - \frac{q_v}{c_{\mathrm{fd}}d_0}. \tag{C.9}$$

Equation (C.9) has a feasible nonnegative fallback fraction only when the right side is positive. If ranks or validation costs vary by point, their average costs replace the corresponding fixed terms. This inequality compares measurement work under matched conditions, not the cost of different searches or a guarantee of end-to-end recovery.

C.6 Chaining and the Output Layer

GCPS and numerical refinement recover L1, followed by L2 in the recovered L1 coordinates. Later hidden layers use the configured compact or native-repair branch within the end-to-end framework, retaining its clustering and precision routines. Each signed result becomes the next prefix. The native return condition depends on the recovery procedure and available components, independently of the post-hoc true-weight pass rate. A returned layer can therefore leave effective coordinates unresolved.

For the final affine layer, form a design matrix whose rows are $[\widehat{h}_k(x)^\top, 1]$ and solve a linear least-squares system against queried outputs. A missing hidden feature can make that system misspecified even if its residual is small on selected inputs. The output layer has a fixed output gauge and is evaluated without an arbitrary output rescaling. Producing all layer files therefore establishes completion of a chain, not global functional equivalence or complete parameter recovery.

Linear-system derivation of Equation (18). Enumerate the paid inputs as $\mathcal{T} = \{x_p\}_{p=1}^{|\mathcal{T}|}$ and stack $Z_{p,:} = [\widehat{h}_k(x_p)^\top, 1]$ and $Y_{p,:} = f(x_p)^\top$. With coefficient matrix $\Theta = [W, b]^\top$, the objective is $\|Z\Theta - Y\|_F^2$. Differentiation gives the normal equations $Z^\top Z\Theta = Z^\top Y$. A minimum-norm least-squares solution is

$$\widehat{\Theta} = Z^\dagger Y, \qquad \widehat{\Theta} = (Z^\top Z)^{-1} Z^\top Y \text{ if } \operatorname{rank} Z = d_k + 1. \tag{C.10}$$

Here $Z^\dagger$ is the Moore–Penrose pseudoinverse. Equation (C.10) supplies the output weights and bias in Equation (18); the inverse expression is an identity, not a recommendation to form normal equations numerically instead of a stable QR/SVD solve. With deficient rank, unobserved feature combinations remain unresolved. If an omitted hidden feature contributes a residual vector r on these inputs, the fit can project r into the represented columns of Z, including the constant column. Thus a biased head can have small training residual without reproducing the target function.

D Full Experimental Protocols and Layerwise Results

D.1 Model, Protocols, and Accounting

Tables 4 and 5 use the public Keras checkpoint supplied with Liu et al.'s research code (Liu et al., 2026), with architecture $784 \to 8 \to \cdots \to 8 \to 1$, eight ReLU hidden layers, and an affine output layer evaluated in FP64. The checkpoint hash accompanies the tables' source manifest. Each row reports one recorded layer result with seed 42; repeated-seed uncertainty was not measured.

Independent true-prefix protocol. Table 4 corresponds to the layerwise setting of Liu's Table 4: every layer receives the correct preceding layers and true signs. Our L1/L2 results use geometric screening and the archived later-layer configuration combines compact filtering, signature reuse, and independent-direction measurement. The native later-layer collection starts with Count=3000. The Liu reproduction preserves original code at L1–L4 and uses an indexing repair at L5–L9. This experiment isolates layer recovery and does not expose later layers to errors in earlier recovered layers.

Recovered-prefix protocol. Table 5 uses the layerwise presentation of Liu's Table 6. L1 is recovered first and the recovered prefix is carried into subsequent layers. Our L1/L2 use geometric screening followed by numerical root refinement. L3–L6 use full-prefix Jacobian compact measurements with A/B enabled; the selected L7/L8 runs instead use a whole-layer native 784D repair configuration with A/B disabled. They do not establish compact-first execution at every hidden layer. Appendix D.8 gives the exact correspondence and continuation history. Native hidden-layer collection uses Count=3000; the reused Liu L1 used Count=500, while geometric shallow-layer collection is controlled by its own screening and refinement settings. The selected shallow refinement records request 1024 roots per represented candidate and accept 8 L1 and 7 L2 candidates. Although the run flags include `real_attack=1` and `cheat_sign=0`, the recorded sign mode explicitly applies Liu-compatible truth-assisted signs. Native auxiliary truth use is also retained. These flags therefore do not establish a strict black-box protocol.

The chain includes selected completed layers and continuations. Our chain applies a sign-orientation correction before continuation, and the Liu chain reuses a native L1 result from the same model. Each inherited layer and selected native repair is counted once. Discarded experiments, development retries, and interface tests are excluded, so the totals describe the assembled chain rather than a fresh uninterrupted run.

Query cost and local runtime. Query count is the primary efficiency metric. One query is one physical input row evaluated by the target, including search, measurement, validation, failures, and fallback in the selected run's ledger. A batch of b rows costs b. Tables report exact Q and $\log_2 Q$. Following Liu et al.'s reporting of query counts alongside local extraction time (Liu et al., 2026), we also record elapsed layer time in seconds and sum it over the selected stages. This supplementary measure includes local prediction and recovery computation under the declared protocol. Speedups compare our local Liu reproduction with our method for the same layer and protocol. They do not divide by the runtimes published by Liu et al. Local runtimes depend on hardware and implementation and do not estimate remote-service latency.

D.2 Recovery and Error Metrics

The effective-weight mask is fixed before comparison using the archived activation fixture: 200,000 inputs sampled uniformly from $[-1000, 1000]^{784}$ with seed 20260908. A hidden-to-hidden weight is included when its incoming feature and outgoing neuron are observed active together; input and affine-output layers use their corresponding ungated convention. This finite sample defines an empirical denominator, not a proof that an unobserved neuron is globally dead. The denominators at L1–L9 are 6272, 56, 56, 62, 50, 46, 52, 43, and 6, summing to 6643. Biases are counted separately against the architecture's 65 bias slots.

A represented weight passes when its aligned absolute error is at most $10^{-10} + 10^{-2}|w|$. Missing effective coordinates remain in the denominator. The recovery rate is the verified count divided by that fixed denominator; the total rate divides the summed verified counts by 6643. L1 contributes 94.42% of this denominator, so a total near 99% can coexist with substantially incomplete deeper layers.

All numerical errors in the two tables compare against the target checkpoint. The chain table is recomputed from saved parameters with positive-scale and permutation alignment propagated through the recovered prefix, and with a fixed output gauge. No new target predictions are needed. For each represented neuron, the aligned weight–bias signature is compared on its represented input support. E_{rel} is the maximum relative ℓ_2 error of these augmented signatures; E_b is the maximum absolute bias error. Unknown input coordinates are not imputed for the error norm, but remain missing in the recovery count. Thus a small signature error measures precision on recovered support, not completeness. Table-wide error maxima include the output layer and are not averaged across layers.

At L7, 43 of the 44 stored nonzero entries belong to the effective mask and pass for each chain. At L9, five weights pass but the bias fails for both chains. The totals are 6593/6643 weights and 54/65 biases. An earlier dashboard comparison used the extracted Liu network as its reference; that comparison remains archived and is superseded here by the target-checkpoint audit.

D.3 Independent Layer Recovery

Table 4 locates the largest query savings in L1–L4. The reduction in measurement cost does not consistently improve precision or runtime. Our maximum signature error is larger at L4, L5 needs more queries and time than Liu, and L6/L7 save queries but take longer. Validation, completion, and linear algebra contribute costs beyond the directional measurements in Equation (C.8).

L8 remains incomplete at 41/43 weights for our method and 40/43 for Liu. The final linear layer reaches 6/6 with a supplied correct prefix, unlike the output fit from a partially recovered chain.

D.4 Recovered-Prefix Chain

Table 5 gives the per-layer counts and errors underlying the chain totals in Section 4. Both methods recover all effective weights through L3; later layers remain partial. Excluding L1, the verified count is 321/371 (86.52%), compared with the overall 99.25% dominated by L1's input weights.

The maximum hidden-layer augmented-signature error is approximately 8.56×10^{-10} for our method and 1.54×10^{-10} for Liu. Thus the methods have different numerical errors despite matching recovery counts at the declared tolerance. The cost comparison includes native repairs as well as compact measurements, so it does not isolate the saving from either optional optimization.

Both output layers represent only 5 of the 6 empirically effective input weights. Their bias errors against the target are approximately 0.1381, and neither bias passes verification. The extracted heads are close to each other but remain inaccurate relative to the target. The available held-out evaluation of this Liu reproduction also falls short of the paper's reported output fidelity. The measured saving therefore concerns partial parameter recovery; full-model fidelity has not been reproduced.

D.5 Reproducibility and Scope

The accompanying source includes table CSV files, source hashes, the common-target parameter-audit script, and a deterministic table generator. It retains the original frozen files and evaluation history. Only local authorized research artifacts are used. The two tables are single-checkpoint, single-seed results from different protocols and implementation revisions, so they are not a controlled ablation of refinement alone. Additional architecture and PReLU results from the supplied manuscript remain in Appendix E; they were not independently re-audited in this revision and are not used to establish the new chain's recovery or precision claims.

Tables 4 and 5 provide the full L1–L9 counts, costs, errors, and speedups.

Table 4: Independent true-prefix recovery on the public $784\text{-}8^{(8)}\text{-}1$ ReLU model. Each layer receives the correct preceding layers and true signs. All Liu and Our entries come from local runs under this protocol.

Layer		Verified weights ↑	Rate (%) ↑	Verified biases ↑	$Q\downarrow$	$\log_2 Q\downarrow$	Time (s) ↓	Query speedup ↑	Time speedup ↑	$E_{\mathrm{rel}}\downarrow$	$E_b\downarrow$
L1	Liu	6272/6272	100.00	8/8	3,163,517	21.59	438.5	26.70×	26.37×	3.122e-08	5.000e-07
	Our	6272/6272	100.00	8/8	118,493	16.85	16.6			4.396e-08	3.602e-07
L2	Liu	56/56	100.00	7/8	19,331,460	24.20	1,139.4	296.89×	90.86×	1.461e-07	1.998e-07
	Our	56/56	100.00	7/8	65,113	15.99	12.5			6.718e-10	3.627e-10
L3	Liu	56/56	100.00	8/8	19,542,954	24.22	1,311.5	28.09×	4.82×	7.795e-08	1.365e-07
	Our	56/56	100.00	8/8	695,756	19.41	272.1			1.166e-08	2.042e-08
L4	Liu	56/62	90.32	7/8	19,749,939	24.24	1,960.3	21.75×	2.15×	8.946e-04	2.054e-04
	Our	56/62	90.32	7/8	908,114	19.79	910.3			1.100e-03	2.526e-04
L5	Liu	35/50	70.00	5/8	20,029,210	24.26	2,043.4	0.95×	0.49×	2.140e-05	3.592e-05
	Our	35/50	70.00	5/8	21,071,777	24.33	4,184.4			1.942e-05	3.330e-05
L6	Liu	38/46	82.61	7/8	46,879,208	25.48	4,828.4	1.18×	0.76×	3.136e-08	3.047e-08
	Our	41/46	89.13	7/8	39,698,964	25.24	6,319.4			6.387e-09	1.781e-09
L7	Liu	44/52	84.62	7/8	47,666,662	25.51	4,517.9	1.33×	0.77×	3.326e-07	3.102e-07
	Our	44/52	84.62	7/8	35,873,415	25.10	5,875.5			9.570e-08	6.861e-08
L8	Liu	40/43	93.02	6/8	67,663,351	26.01	7,697.3	2.00×	1.43×	9.343e-04	5.051e-04
	Our	41/43	95.35	6/8	33,765,351	25.01	5,381.1			6.520e-06	1.026e-05
L9	Liu	6/6	100.00	1/1	100	6.64	11.1	1.00×	0.94×	4.871e-14	7.228e-14
	Our	6/6	100.00	1/1	100	6.64	11.8			4.871e-14	7.228e-14
Total	Liu	6603/6643	99.40	56/65	244,026,401	27.86	23,947.8	1.85×	1.04×	9.343e-04	5.051e-04
	Our	6607/6643	99.46	56/65	132,197,083	26.98	22,983.8			1.100e-03	2.526e-04

Q counts formal physical prediction rows, including failed attempts and individual rows within batches. Separate interface checks use 3,817,464 rows for Liu and 189,814 for Our. Recovery uses the fixed empirical effective-weight mask, $\mathrm{rtol} = 10^{-2}$ and $\mathrm{atol} = 10^{-10}$. E_{rel} is the largest aligned relative ℓ_2 error of a represented weight–bias signature; E_b is the largest absolute bias error. Missing coordinates remain in the recovery denominator and are excluded from signature errors. Total rates divide summed counts; total errors are maxima. Speedups divide Liu cost by Our cost for the same layer, with precision reported separately. Liu uses original code for L1–L4 and an index repair for L5–L9. Our is the archived combined implementation.

Table 5: Recovered-prefix chain on the public 784-$8^{(8)}$-1 ReLU model. Both methods propagate recovered layers and retain truth-assisted signs and native auxiliary truth. Parameters are verified against the target checkpoint after consistent permutation and positive-scale alignment.

Layer		Verified weights ↑	Rate (%) ↑	Verified biases ↑	$Q\downarrow$	$\log_2 Q\downarrow$	Time (s) ↓	Query speedup ↑	Time speedup ↑	$E_{\mathrm{rel}}\downarrow$	$E_b\downarrow$
L1	Liu	6272/6272	100.00	8/8	7,755,927	22.89	898.2	6.84×	8.18×	2.152e-13	6.539e-14
	Our	6272/6272	100.00	8/8	1,133,480	20.11	109.8			1.082e-12	8.471e-12
L2	Liu	56/56	100.00	7/8	19,529,298	24.22	1,403.9	15.97×	11.99×	1.080e-12	1.579e-12
	Our	56/56	100.00	7/8	1,223,153	20.22	117.1			4.767e-12	6.381e-12
L3	Liu	56/56	100.00	8/8	20,872,891	24.32	1,688.3	3.98×	2.82×	1.453e-12	2.452e-12
	Our	56/56	100.00	8/8	5,242,684	22.32	597.7			8.949e-12	7.409e-12
L4	Liu	56/62	90.32	7/8	19,772,511	24.24	2,293.0	5.24×	2.15×	4.523e-12	4.405e-12
	Our	56/62	90.32	7/8	3,775,071	21.85	1,068.4			2.028e-11	2.247e-11
L5	Liu	35/50	70.00	5/8	20,026,209	24.26	2,550.2	4.35×	1.84×	7.137e-12	1.139e-11
	Our	35/50	70.00	5/8	4,606,506	22.14	1,382.9			1.964e-11	2.099e-11
L6	Liu	35/46	76.09	7/8	21,892,392	24.38	2,368.6	1.58×	1.39×	8.168e-12	6.303e-12
	Our	35/46	76.09	7/8	13,877,067	23.73	1,705.5			1.077e-10	6.494e-11
L7	Liu	43/52	82.69	7/8	33,554,927	25.00	4,122.6	1.00×	1.51×	1.542e-10	1.684e-10
	Our	43/52	82.69	7/8	33,627,889	25.00	2,725.9			1.542e-10	1.633e-10
L8	Liu	35/43	81.40	5/8	21,180,570	24.34	2,843.9	1.00×	2.06×	1.168e-10	1.582e-10
	Our	35/43	81.40	5/8	21,159,084	24.33	1,382.9			8.556e-10	3.934e-10
L9	Liu	5/6	83.33	0/1	87	6.44	8.8	1.00×	1.02×	9.295e-02	1.381e-01
	Our	5/6	83.33	0/1	87	6.44	8.6			9.295e-02	1.381e-01
Total	Liu	6593/6643	99.25	54/65	164,584,812	27.29	18,177.4	1.94×	2.00×	9.295e-02	1.381e-01
	Our	6593/6643	99.25	54/65	84,645,021	26.33	9,098.8			9.295e-02	1.381e-01

The effective-weight denominators and tolerances match Table 4. E_{rel} includes bias, and L9 is evaluated without free output rescaling. Both output biases fail verification. Liu uses the native mathematical implementation, including the cost of its reused L1 once. Our uses GCPS/refinement at L1/L2, compact measurements with A/B at L3–L6, and whole-layer native repairs with A/B disabled at L7/L8 (Appendix D.8). The chain includes resumed checkpoints and a truth-assisted sign correction. Queries and times sum the selected layers and repairs; separate interface tests, discarded development runs, and post-hoc evaluation are excluded. These totals describe the selected chain, rather than a fresh uninterrupted trial or the entire development budget. Recovery counts and support-restricted errors are parameter metrics, not a test of full-model output equivalence.

D.6 Earlier Shallow-Layer Measurements

Table 6 reports three returned measurements and one failed comparison attempt from the earlier signature-stage campaign. The scalar-output L2 pair appears as † in Table 1; the ten-output L1 result is retained here and replaced in the main table by the paired rerun in Table 7. L2 uses the public GCPS scalar-output CIFAR checkpoint, and L1 uses Liu's ten-output CIFAR checkpoint. Both have input dimension 3072 and eight width-8 hidden layers. Although the latter directory ends in `_1`, the model has ten outputs. A preregistered raw-logit-0 adapter preserves the hidden arrays and bypasses softmax. The registration snapshot records model and container hashes. Comparisons use matched runs within a campaign.

Paired protocol and timing. All jobs use seed 42 and FP64, with eight disjoint seed-917 interface rows per job. Liu's original signature callable uses Count=500 for L1 and Count=3000 for L2, `real_attack=0`, and `cheat_sign=1`. L2 receives true L1. Our calls the original shallow GCPS routines without the later high-precision chain refinement; Count does not control its geometric loops. Native auxiliary truth and original diagnostics are retained. No stopping rule was changed. Returned runs time the signature callable, including search/measurement/completion and excluding imports, model loading, interface checks, and offline evaluation. Jobs run serially on CPU 63 with single-threaded numerical libraries; concurrent unrelated work can affect time. Each result is one run, with no retry or multi-seed uncertainty estimate.

Frozen returns and offline counter correction. Both Our routines returned and saved their signatures before a wrapper row-counter assertion failed. The wrapper missed default-bound calls to `run`; the unchanged native function counts every input row. Its frozen totals give 22,656 rows for scalar L2 and 572,968 for ten-output L1, rather than the wrapper's incomplete 11,116 and 565,524. Independent offline audits verify the source/model/selection hashes and retain the original failures and frozen files. These audits introduce no predictions and do not rerun extraction; a full per-call transcript is unavailable. The successful Liu L2 wrapper and native counters agree at 74,470,730 rows.

Unsigned recovery and precision. Post-hoc one-to-one matching permits either orientation and a scalar factor, so the supplement does not measure sign recovery. Its relative-normal error excludes bias; all layer weights are tested, not an empirical activation-effective subset. Although queries use FP64, the original GCPS routines return normals stored in FP32; evaluation promotes these saved values to FP64 without restoring lost precision. Scalar L2 verifies 64/64 weights for both methods, but only 1/8 and 0/8 biases. Ten-output L1 Our verifies 24,575/24,576 weights and 8/8 biases. The scalar L2 recorded query/time ratios are 3287.02×/666.66×; these are signature-stage cost ratios with unequal bias recovery, not a same-quality speedup or an end-to-end claim.

Failed and unstarted cases. Ten-output L1 Liu retained 20 normal columns while allocating eight bias entries; the native `compute_layer_values` operation on shapes $(8, 1)$ and $(20, 1)$ raised a broadcasting error before returning signatures. Its 12,289,358 spent formal rows remain reported without a recovery rate or completed-run speedup. MNIST 784-$32^{(8)}$-1 was unstarted in this campaign because its original manuscript checkpoint was unavailable. The historical Our 90-second L2 measurement remains separate from the new model below. Old failures and frozen returns are unchanged.

Table 6: Earlier shallow-layer measurements on CIFAR checkpoints, including three returned results and one failed Liu attempt. Unsigned signatures use seed 42 and FP64 raw logit 0; L2 receives true L1.

Case		Q	$\log_2 Q$	Time (s)	Unsigned normals	Verified weights	Verified biases	E_w	E_b	Interface rows
CIFAR-1 L2	Liu	74,470,730	26.15	4146.36	8/8	64/64	1/8	2.198e-08	1.043e-07	8
CIFAR-1 L2	Our	22,656	14.47	6.22	8/8	64/64	0/8	2.583e-07	1.377e-06	8
CIFAR-10 L1	Our	572,968	19.13	313.93	8/8	24575/24576	8/8	2.033e-06	8.954e-05	8
CIFAR-10 L1	Liu (failed)	12,289,358	23.55	7641.49‡	—	—	—	—	—	8

CIFAR-1 and CIFAR-10 denote 3072-$8^{(8)}$-1 and 3072-$8^{(8)}$-10, respectively. We count all layer weights, not an activation-effective mask. Verification uses $\mathrm{rtol} = 10^{-2}$ and $\mathrm{atol} = 10^{-10}$. E_w is the maximum relative ℓ_2 error of a matched incoming weight vector after unrestricted-sign scalar alignment; unlike E_{rel} in Tables 4/5, it excludes the bias. E_b is the maximum aligned absolute bias error. Eight matched normals do not imply that every coordinate or bias passes. Completed time covers the signature callable through return. ‡ is failed-worker elapsed time including setup, not a completed signature time. Its spent queries remain reported, but no speedup is computed from the failed attempt. Each disjoint interface check costs eight rows and is excluded from formal Q. Our returned before a wrapper counter assertion; the original upstream row counter was audited offline, including default-bound calls. Original failures and frozen records are retained. See Appendix D.6 for source/quality limits.

D.7 Completed Repair and New-Checkpoint Comparisons

Table 7 reports six completed runs from the repair campaign, marked $*$ in Table 1. CIFAR L1 uses the same public ten-output checkpoint and raw-logit-0 interface. Both methods recollect all critical points and measurements from scratch. Liu's successful query count equals the earlier failed count because the export repair adds no measurements to the repeated computation. Speedups use these completed runs; Table 6 retains the earlier failure cost.

Repaired baseline. The repaired Liu implementation aligns each bias with its returned normal column in all three export branches and identifies padded slots by zero normals. For L1, when the candidate count exceeds the known layer width, it retains the width-many components with the largest number of distinct witnesses. A tie at the cutoff is left unresolved. This selection uses neither target parameters nor additional queries. All normals, component indices, and decisions are archived. In CIFAR, it reduces 20 candidates to eight, each supported by at least eight witnesses; discarded candidates have at most four. MNIST L1 returns 32 components and needs no cutoff. Support ranking is a selection heuristic, so these rows are labelled repaired Liu. Native sign assistance, auxiliary truth, and diagnostics remain as specified above.

MNIST-32 checkpoint. Both methods use the same newly trained $784\text{-}32^{(8)}\text{-}1$ network with eight ReLU hidden layers and a linear scalar head. Training uses MNIST digits 0/1, 12,665 official training examples and 2,115 test examples, with inputs normalized once as $\mathrm{pixel}/127.5 - 1$. Seed 42, Kaiming-normal initialization, zero biases, Adam (learning rate 10^{-3}, batch size 256), and the final epoch 20 are fixed; neither test accuracy nor extraction quality selects the checkpoint. Training stores FP32 parameters; extraction uses their unchanged values with FP64 arithmetic. Final test accuracy is 99.9527%, a classification metric, not parameter recovery. The checkpoint hash and training/data provenance accompany the source. The MNIST-32 pair in Table 1 uses this checkpoint; Appendix E keeps the unmatched historical 90-second result separate.

Protocol, accounting, and outcome. All six runs use seed 42, `real_attack=0`, `cheat_sign=1`, and true L1 for L2. Liu uses Count=500 for width-8 L1, Count=2000 for width-32 L1 (the same width-proportional allocation), and Count=3000 for L2. Our retains its native geometric stopping rules; Count does not control those loops. Each job runs once, serially on CPU 63 with single-threaded prediction libraries. Eight disjoint interface rows are charged separately per job. The repaired wrapper counts default-bound calls as well as module aliases; wrapper and native physical-row counters agree in every run. Returned parameter files and costs are frozen before offline evaluation. Training, setup, interface and evaluation time are excluded from the signature timer.

All six calls return normally, but completion is not full parameter recovery. MNIST L2 Liu has 31 nonzero columns and one zero placeholder; Our has 32 matched normals. Both pass 30/32 biases. MNIST L1 Liu passes every weight, while Our misses one and has higher normal error; both pass 31/32 biases. Our stores returned normals as FP32, Liu as FP64. The table retains these differences alongside query/time speedups, using the all-weight denominator and unsigned matching from Appendix D.6. No missing neuron is declared globally dead from these observations.

Table 7: Completed shallow-layer comparisons of repaired Liu and Our on CIFAR-10 and MNIST-32. Each pair uses the same checkpoint; MNIST-32 is newly trained. L2 receives true L1.

Case		Unsigned normals	Verified weights	Rate (%)	Verified biases	Q	$\log_2 Q$	Time (s)	Query speedup	Time speedup	E_w	E_b
CIFAR-10 L1	Liu	8/8	24532/24576	99.8210	8/8	12,289,358	23.55	7173.25	21.45×	23.71×	2.240e-04	1.819e-03
	Our	8/8	24575/24576	99.9959	8/8	572,968	19.13	302.48			2.033e-06	8.954e-05
MNIST-32 L1	Liu	32/32	25088/25088	100.0000	31/32	12,641,354	23.59	1867.47	20.46×	21.07×	6.407e-09	5.519e-08
	Our	32/32	25087/25088	99.9960	31/32	617,789	19.24	88.62			1.179e-06	7.598e-06
MNIST-32 L2	Liu	31/32	992/1024	96.8750	30/32	20,331,584	24.28	1227.41	30.34×	13.17×	5.775e-07	6.083e-07
	Our	32/32	1024/1024	100.0000	30/32	670,051	19.35	93.19			3.399e-07	5.675e-07

CIFAR-10: $3072\text{-}8^{(8)}\text{-}10$; MNIST-32: $784\text{-}32^{(8)}\text{-}1$. Q counts physical prediction rows; eight disjoint interface rows per job are excluded (48 total). Time covers the signature callable through return, excluding training, imports, interface checks and offline evaluation. Merged speedups divide the two recorded costs within each pair. All layer weights remain in the denominator, including the missing Liu L2 column. Verification uses $\mathrm{rtol} = 10^{-2}$ and $\mathrm{atol} = 10^{-10}$ after one-to-one unsigned scaling; this does not evaluate sign recovery. E_w is the maximum matched incoming-weight relative ℓ_2 error, excluding bias; E_b is the maximum matched aligned bias absolute error. Both maxima cover matched columns only, so the missing column is represented by the recovery counts, not by its error. Rates use four decimals to distinguish one failed coordinate from 100%. Our saves normals as FP32 and Liu as FP64; both query in FP64. See Appendix D.7 for repairs, model training, native assistance and partial-bias recovery.

D.8 ALGORITHM–RESULT CORRESPONDENCE

Tables 8 and 9 identify the branches of Section 3 used by each layer of the selected chain. The source package contains nine frozen configurations, recorded code-manifest hashes, stage query counts, and a script that generates the mapping. Recovery counts and errors come from the common-target audit in Table 5; the older snapshot is retained only as provenance.

Procedure and branch selection. The selected runs use the following procedure, with queries recorded by branch.

1. Recover L1 using the GCPS midpoint screen, measure retained signatures, collect additional localized roots, and apply the direct SVD refinement in Equation (6). Export the represented coordinates and the declared truth-assisted orientation.
2. Recover L2 using the recovered L1 and the common lift in Equation (8). Apply the collinearity screen, paid signature checks, refinement, and the same sign protocol. Do not substitute the target L1 to remove prefix errors.
3. For each later hidden layer, collect the native initial critical-point pool. In compact mode, B measures independent reachable directions subject to validation. A attempts to reuse the paid signature for the pulled-back normal when coverage, cell, and numerical checks pass. If A fails, measure a compact normal using the full recovered-prefix Jacobian. A failed compact check sends that point to the native input-space measurement. Retain unresolved coordinates and partial constraints.
4. Perform the configured native calibration checks and charge their predictions. In whole-layer repair mode, send every filtering call through the native input-space path with A/B disabled. Record this mode separately from pointwise compact fallback.
5. Feed accepted constraints to Liu's cross-region clustering, merging, difficult-point collection, and precision processing. Return represented parameters under the recorded native-return behavior, apply the declared orientation, and export the next prefix. The post-hoc true-weight pass fraction is not an acquisition stopping rule.
6. Fit the affine output head using Equation (18). Audit saved parameters with the fixed empirical denominator and propagated gauge. Sum each selected layer's query and time record once; keep discarded development work outside this selected-chain total and disclose that boundary.

Table 8: Layerwise execution modes of Our for the selected recovered-prefix chain. A denotes paid-signature reuse and B independent-direction measurement. R1–R3 identify recorded code-manifest versions. Inherited layers retain their original query and time costs, each counted once.

Layer	Recorded operation	A/B	Version	Final continuation
L1	GCPS + root refinement	n/a	R1	Inherited
L2	GCPS + root refinement	n/a	R1	Inherited
L3	Full-prefix compact	on/on	R1	Inherited
L4	Full-prefix compact	on/on	R1	Inherited
L5	Full-prefix compact	on/on	R1	Inherited
L6	Full-prefix compact	on/on	R1	Inherited
L7	Whole-layer native 784D	off/off	R2	Inherited; sign corrected
L8	Whole-layer native 784D	off/off	R3	Fresh continuation
L9	Affine head fit	n/a	R3	Fresh continuation

Version and continuation provenance. R1 is the recorded code-manifest prefix `9c60242f0eaf` for L1–L6, R2 is `7a1ff153edce` for L7, and R3 is `adb4455685b1` for fresh L8/L9. Full SHA256 values and recorded per-file hashes are in `data/algorithm_result_map.json`; the source archive does not contain the complete extraction implementation.

The final continuation inherits completed L1–L7. A failed L7 sign hook had left an orientation unapplied. Before fresh L8/L9, the saved L7 column 0 was multiplied by -1 using the declared

truth-compatible sign information. The before/after hashes record this target-assisted change to the inherited parameters, whose measurement cost remains included. L9 is a linear fit; its inherited `native784` label does not indicate an output ReLU normal measurement.

Recorded settings and counter interpretation. All selected layers use seed 42 and FP64 target evaluation. The shallow refinement requests 1024 roots per represented candidate, with a 100,000-candidate scan cap, 24 quality probes, local search span 0.01, and local margin 0.02. These are recorded settings, not universal accuracy guarantees; the final counts need not equal the requested root count. Native hidden-layer collection starts at Count=3000. The selected compact layers use risk-based native validation rather than validating every point, with configured random sampling rate 0.02, warm-up budgets of 4 accepted and 16 rejected decisions, and mismatch escalation over a 24-check window. The complete configuration fields are preserved verbatim alongside the mapping.

Table 9: Module calls in layers configured for compact measurement. Used/tried counts successful and attempted calls; fallback counts failed calls that invoke another path. Compact normal measurement follows A fallback. Native validation can overlap these events. The counts describe execution and cannot be summed to obtain query costs or recovery rates.

	A reuse		B measurement		Compact normal		Native
Layer	Used/tried	Fallback	Used/tried	Fallback	Used/tried	Fallback	validation
L3	2927/3000	73	2949/3047	98	52/73	21	203/3000
L4	2863/3000	137	2901/3060	159	111/137	26	321/3000
L5	2792/3000	208	2825/3067	242	177/208	31	427/3000
L6	2731/3000	269	2761/3074	313	228/269	41	532/3000

Table 9 records successful A/B calls as well as fallbacks in L3–L6. B is invoked beyond the initial filtering pool; the A and native-validation counters shown here correspond to that pool. These counts describe module use, while the query totals come from physical prediction ledgers. They cannot by themselves isolate either module's saving. A/B are disabled throughout the selected L7/L8 runs.

Scope of the comparisons. The three main tables support different claims. Table 1 measures shallow signature costs, including repaired Liu export in the latest pairs. Table 2 measures independent layer recovery with supplied prefixes and signs. Table 3 measures the assisted chain whose configurations and continuations are recorded here. Their distinct denominators, precision measures, and cost boundaries prevent pooling the results or attributing the chain's saving to A or B alone.

E Additional Experiments from the Original Manuscript

The following tables are retained from the supplied manuscript and belong to separate experimental campaigns. They have not been independently re-audited in this revision. Their timing and query figures are not substituted for the newly audited comparisons in Section 4. In particular, their MNIST shallow-layer numbers differ from the archived campaign in Table 4, and their PReLU results do not establish the new ReLU chain's accuracy.

E.1 First Hidden Layer Signature Extraction

Table 10 compares the recorded L1 signature-extraction costs of GCPS and the method of Liu et al. (2026). GCPS uses fewer queries and less time in all three reported pairs. The models differ in input dimension and hidden width, as discussed below.

E.1.1 Comparison across Input Dimensions

Coordinate-based normal measurement costs a number of queries proportional to the input dimension. Screening before measurement can therefore avoid expensive probes of deeper-layer candidates. In the width-8 models, the recorded L1 costs rise from MNIST to CIFAR for both methods. These are different trained networks, however, so the comparison does not isolate the effect of input dimension or establish how the speedup scales with it.

E.1.2 Comparison across Hidden Widths

For the two MNIST models, Liu's recorded query counts are close, while both methods take longer at width 16 than at width 8. On MNIST $784 - 16^{(8)} - 1$, GCPS takes 25 seconds compared with Liu's 6 minutes 17 seconds and uses $2^{18.28}$ queries compared with $2^{21.60}$. The table reports total signature-stage costs; it does not separately measure the contribution of clustering to these time differences.

Table 10: First Hidden Layer Signature Extraction On Multiple Models

Model	Approach	Signature Extraction Time	Signature Query Cost
MNIST $784 - 8^{(8)} - 1$	Liu et al. (2026)	3m24s	$2^{21.59}$
	Ours	3.5s	$2^{16.66}$
MNIST $784 - 16^{(8)} - 1$	Liu et al. (2026)	6m17s	$2^{21.60}$
	Ours	25s	$2^{18.28}$
CIFAR-10 $3072 - 8^{(8)} - 1$	Liu et al. (2026)	56m48s	$2^{23.55}$
	Ours	1m24s	$2^{19.40}$

Table 11: Second Hidden Layer Signature Extraction On Multiple Models

Model	Approach	Signature Extraction Time	Signature Query Cost
MNIST $784 - 8^{(8)} - 1$	Liu et al. (2026)	7m5s	$2^{24.21}$
	Ours	16s	$2^{14.48}$
MNIST $784 - 16^{(8)} - 1$	Liu et al. (2026)	26m13s	$2^{24.22}$
	Ours	43s	$2^{17.02}$
CIFAR-10 $3072 - 8^{(8)} - 10$	Liu et al. (2026)	1h33m	$2^{26.15}$
	Ours	23s	$2^{14.77}$
MNIST $784 - 32^{(8)} - 1$	Ours	90s	$2^{19.11}$

E.2 Second Hidden Layer Signature Extraction

This historical L2 experiment supplies true first-layer parameters. Table 11 compares L2 signature-extraction costs with Liu et al. (2026). Randomly collected points can leave L1 features inactive, yielding partial L2 signatures that require observations from other regions. GCPS instead searches within a fully active L1 region. With a full-rank prefix and an isolated visible boundary, one point in this region can expose all incoming signature coordinates. Coverage and numerical accuracy still require validation. All three paired historical results have lower query and time costs for GCPS. The width-32 MNIST row has no Liu comparison and is reported without a speedup. These cost records alone do not establish a scaling law in hidden width or equal recovery precision across methods.

E.3 Extension to PReLU Networks

Within a fixed activation cell, PReLU also gives an affine prefix, so the construction in Section 3.3 applies subject to rank and visibility conditions. The original manuscript compares against Chen et al. (2026) on networks with two hidden layers and reports an unpaired shallow result on MNIST $784\text{-}8^{(8)}\text{-}1$.

E.3.1 First Hidden Layer Signature Extraction

Table 12 reports lower L1 costs for GCPS in all three paired PReLU cases. On $64\text{-}32\text{-}32\text{-}1$, it uses 21.4% of the recorded time and approximately 28.3% of the queries of Chen et al.'s method. The unpaired MNIST $784\text{-}8^{(8)}\text{-}1$ result takes 1 minute 23 seconds and $2^{16.47}$ queries. The query percentages are approximate because the table stores rounded logarithms.

Table 12: First Hidden Layer Signature Extraction on PReLU Networks

Model	Approach	Signature Extraction Time	Signature Query Cost
20-10-10-1	Chen et al. (2026)	7.4s	$2^{13.15}$
	Ours	3.8s	$2^{12.34}$
32-16-16-1	Chen et al. (2026)	16s	$2^{15.46}$
	Ours	9.6s	$2^{14.12}$
64-32-32-1	Chen et al. (2026)	117s	$2^{17.30}$
	Ours	25s	$2^{15.48}$
MNIST $784\text{-}8^{(8)}\text{-}1$	Ours	1m23s	$2^{16.47}$

E.3.2 Second Hidden Layer Signature Extraction

Table 13 shows lower query counts but longer execution times for GCPS in all three paired L2 cases. The largest time difference is 79 seconds versus 49 seconds on 64-32-32-1. A nonzero PReLU negative slope can preserve directions that ReLU would suppress, reducing one reason to seek a fully active prefix region. The remaining query and runtime differences depend on the search and implementation; these historical totals do not isolate their causes.

Table 13: Second Hidden Layer Signature Extraction on PReLU Networks

Model	Approach	Signature Extraction Time	Signature Query Cost
20-10-10-1	Chen et al. (2026)	5.2s	$2^{13.05}$
	Ours	5.6s	$2^{12.71}$
32-16-16-1	Chen et al. (2026)	11.8s	$2^{14.42}$
	Ours	12.6s	$2^{14.32}$
64-32-32-1	Chen et al. (2026)	49s	$2^{16.58}$
	Ours	79s	$2^{16.09}$

E.3.3 End-to-End Extraction

Table 14 compares historical end-to-end costs with Method 1 of Chen et al. (2026). GCPS has lower costs in both pairs. On 32-16-16-1, it takes 54.0% of the recorded time and approximately 8.4% of the queries. This table reports costs without parameter-recovery or output-fidelity measurements.

Table 14: End-to-End Extraction on PReLU Networks

Model	Approach	Extraction Time	Total Query Cost
20-10-10-1	Method 1 of Chen et al. (2026)	30.2s	$2^{18.17}$
	Ours	14.1s	$2^{14.79}$
32-16-16-1	Method 1 of Chen et al. (2026)	50.4s	$2^{19.38}$
	Ours	27.2s	$2^{15.81}$

The historical PReLU records show lower query counts across the reported pairs. Time decreases in the L1 and end-to-end comparisons but increases at L2. Assessing whether these savings persist at matched recovery quality requires the same parameter and output audits used for the ReLU experiments.