OUTPUT PROBABILITIES UNDER HOMOMORPHIC ENCRYPTION: DEPTH CHOSEN FOR ACCURACY IS NOT ENOUGH

Anonymous authors
Paper under double-blind review

ABSTRACT

Homomorphic encryption (HE) lets a server run a model on data it never sees, and HE inference is judged by task accuracy, approximation error, and multiplicative depth. Many deployments, however, consume the output probabilities themselves, through confidence thresholds, abstention rules and cost-weighted decisions, and none of those three quantities certifies them. Replacing the output softmax by a polynomial, the usual HE-compatible route, leaves the predicted label unchanged in every seed of ten backbone and dataset settings while inflating expected calibration error on the same logits by $1.1\times$ to $30\times$, and by roughly $10\times$ to $20\times$ over the full vocabularies of two language models. Under the repeated-squaring schedule used for high-precision homomorphic exponentials, accuracy reaches its ceiling one to four multiplicative levels before calibration comes within 0.005 of the exact softmax: the multiplicative depth sufficient to preserve predictions can be systematically insufficient to preserve output probabilities. We trace the effect to a non-affine residual in pairwise log-odds, which yields a label-free diagnostic, computable from the public polynomial and the operating range before deployment, that chooses depth where accuracy cannot. The repair depends on the interface. Where logits may leave the ciphertext, defer the softmax to the client; where an existing pipeline already returns polynomial numerators, invert them in plaintext as a compatibility layer, using a numerator monotone by construction; where the distribution itself must stay encrypted, fold the calibration temperature into the model and let the diagnostic set the depth, which in exact-arithmetic simulation lands within 0.004 ECE of the client-side result on 96 of 99 seeds. Real CKKS runs of the head and numerator reproduce the simulated effect.

1 INTRODUCTION

Homomorphic encryption lets a server evaluate a model on data it never sees in the clear. What the server must return, though, is not always a label. A triage queue, an abstention rule, a cost-weighted decision, and any downstream probabilistic model consume an output *probability distribution*, and they consume the numbers rather than the argmax. Forming that distribution is where encrypted inference becomes awkward: CKKS ciphertext arithmetic supports addition and multiplication (Cheon et al., 2017), so the exponential and the normalization that turn logits into probabilities are not native operations. A deployment has only two places to form the distribution, on ciphertext or on the client after decryption, and forming it under encryption typically means approximating the softmax with polynomials (Cho et al., 2024; Shaik et al., 2024; Lim et al., 2026). Which of the two is available is set by the deployment, not by the modeler. Our claim is simple: *in HE inference, the multiplicative depth sufficient to preserve predictions can be systematically insufficient to preserve output probabilities.*

Output-probability estimation under homomorphic encryption is the problem this paper is about, and it is not the same problem as accurate encrypted classification. The quantities the HE inference literature reports, task accuracy, relative approximation error, multiplicative depth and latency, certify that the encrypted model predicts the right class; none of them certifies what the probabilities attached to that prediction mean. We study that gap concretely through polynomial approximation of the output softmax, the most common HE-compatible route to probabilities, and we separate

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

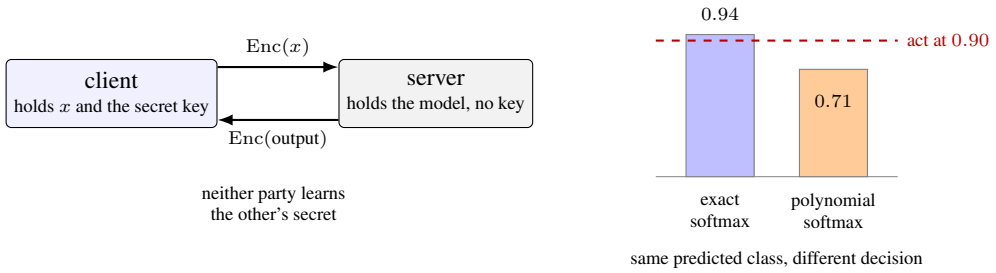


Figure 1: **Accuracy certifies the label, not the distribution.** *Left:* the client encrypts its input and the server evaluates the model on ciphertext, so neither party learns the other's secret. *Right:* forming the output distribution under encryption replaces the softmax with a polynomial. That leaves the predicted class, and so top-1 accuracy, unchanged while moving the confidence attached to it, and a policy that acts automatically above 0.90 therefore acts on one and defers on the other. The two values are illustrative; §3 reports the measured effect, and Figure 3 shows the remedies.

throughout what we claim for that mechanism from what we claim for the broader problem. Accuracy checks which class wins. Calibration checks whether the reported probability has the right meaning: whether predictions made at confidence 0.8 are correct about 80% of the time (Guo et al., 2017). It is a property of the numbers themselves, can fail in either direction, and matters wherever a probability rather than a label is consumed, at a triage threshold, an abstention rule or a cost-weighted decision, the healthcare and finance settings that motivate encrypted inference.

Is the output softmax ever really evaluated under encryption? Inside attention it is interior and cannot be deferred, which is why a dedicated literature approximates or replaces it homomorphically (Zimerman et al., 2026; Park et al., 2026; Xie et al., 2026); at the output, Cho et al. (2024) evaluate a vocabulary-wide distribution for token generation, a server that must *sample* evaluates it under encryption (Lim et al., 2026), and returning the logit vector is itself a disclosure, which is why Jovanović et al. (2022) evaluate the argmax under FHE. That literature measures approximation error, task accuracy, and cost. We ask a different question: *when a polynomial approximation preserves the predicted class, does it also preserve the meaning of the confidence?* The answer can be no, systematically, and it changes how a deployment should budget its multiplicative depth. We study the final-layer softmax, where the failure is invisible to accuracy; Appendix H shows the interior case behaves differently.

- **Prediction fidelity and probability fidelity are separate requirements (§3, §6).** Approximating only the output softmax leaves the prediction unchanged in every seed of ten backbone-and-dataset settings and of two language models at full vocabulary, while ECE on the same logits inflates by 1.1× to 30×, with NLL and Brier moving the same way. The hidden-activation approximation deployed alongside it *does* cost accuracy, up to 7.3 points, but that loss says nothing about the probabilities.
- **A depth-reliability separation, priced (§4, §6).** Under the repeated-squaring schedule of high-precision homomorphic exponentials, top-1 accuracy is at its ceiling at zero extra levels while calibration needs one to four more on five settings across two architecture families, and five or six at vocabulary scale. Stopping where accuracy stops ships 2.5–5.0× the exact softmax's ECE, and neither the relative approximation error the HE-softmax literature optimizes nor the output KL locates the right depth.
- **Mechanism and a label-free depth rule (§4).** The polynomial adds a non-affine residual to every pairwise log-odds, which is why the label survives and why no in-pipeline temperature undoes the damage. The residual's spread over the operating range is a diagnostic computable from the public polynomial before any encrypted inference without labels, and thresholding it chooses depth where accuracy cannot.
- **Remedies set by the interface (§5).** If logits may leave the ciphertext, defer the output softmax and temperature-scale on the client, a design rule whose contribution is the measurement that it suffices. If an existing pipeline already returns polynomial numerators, invert them in plaintext, a compatibility layer; a sum-of-squares numerator monotone by

certificate makes that inversion well defined. If the distribution itself must stay encrypted, fold the calibration temperature into the head and let the label-free rule set the depth, which in exact-arithmetic simulation lands within 0.004 ECE of the client-side result on 96 of 99 seeds. All three leave top-1 accuracy unchanged wherever the numerator itself preserves it, which is every head-only setting here and not the public-shift regime of Appendix A.

2 Background and Related Work

Probability reliability. Calibration is our primary lens on it. A classifier is calibrated if, among predictions of confidence p, a fraction p are correct (Guo et al., 2017); *confidence* is the maximum softmax probability. We report expected calibration error (ECE) with 15 equal-width bins (Naeini et al., 2015; Guo et al., 2017), $\mathrm{ECE} = \sum_m \frac{|B_m|}{n} |\mathrm{acc}(B_m) - \mathrm{conf}(B_m)|$, with NLL and Brier alongside; ECE is binning-sensitive (Zhang et al., 2020), so we re-estimate the headline results under an equal-mass binner (Appendix B). *Temperature scaling* (Guo et al., 2017) divides the logits by one scalar T and is the standard post-hoc remedy; the question is whether it inverts *this* distortion.

FHE inference and polynomial approximation. CKKS (Cheon et al., 2017) supports approximate arithmetic over encrypted reals and underlies much of the private neural inference that followed (Lee et al., 2023); earlier work used exact integer schemes (Gilad-Bachrach et al., 2016). Nonlinear operators are not native, so FHE pipelines replace activations and softmax-like computations with polynomial-friendly ones, tuned for accuracy, approximation error, depth or latency (Lee et al., 2023; Zimerman et al., 2026; Li et al., 2026). Cho et al. (2024) evaluate the softmax over wide input ranges by interleaving domain scaling and repeated squaring with normalization, and per-layer approximation settings can be searched against latency and accuracy (Xie et al., 2026).

Calibration under encryption, and what is new here. Calibration has been studied under encryption before (Shaik et al., 2024; Pathan et al., 2026), and we are *not* the first to observe accuracy-preserving miscalibration from numerical approximation (Gu et al., 2026), though reduced precision can also improve calibration (Bouguerra et al., 2025); Appendix G places the closest work, including Zirilli et al. (2026), whose learnable per-nonlinearity depths are regularised toward exactly the shallow regime we find dangerous and evaluated by agreement and perplexity rather than by any calibration metric. What is specific here is the error source and the interface: a fixed *public* polynomial on a bounded range rather than a rounding rule, which makes the distortion computable before deployment and exactly invertible by the client, and which the client cannot recompute at higher precision, since the computation ran under encryption.

3 Problem: An Accuracy-Invisible Calibration Failure under Polynomial FHE

Setup. A classifier $f = h \circ g$ maps an input through a feature extractor g to a d-dimensional penultimate feature x; the head computes $z = Wx + b \in \mathbb{R}^K$ and $p = \mathrm{softmax}(z)$. Under FHE each nonlinearity becomes a Chebyshev polynomial P_k on a data-driven interval: hidden activations $\phi \approx P_k^{\phi}$ and/or the softmax exponential $\exp \approx P_k^{\exp}$. We study *isolation*, one nonlinearity at a time, and a *polynomialized pipeline* approximating hidden activations (degree 4) *and* softmax (degree 3) together. Pipeline scope differs by backbone. The substitution is whole-network on the MLP benchmarks, and head-only on the convolutional and ResNet-18 settings, where the backbone runs in plaintext and only the final ReLU, the head and the numerator are approximated; we therefore do not describe the latter as full FHE pipelines. Appendix A adds a convolutional network evaluated whole on ciphertext, without that caveat. We use d=64. A low-degree fit of $\exp$ is not positive across a wide interval, so the deployed numerator is floored, $\tilde{\eta}_i = \max(P(u_i), 10^{-8})$, before normalization: a negative numerator is not a probability, so the floor is part of the map we analyze. Every deployment parameter is fixed on the calibration split before the test split is touched, the *calibration-only* protocol (Appendix B): the hidden-ReLU fit covers the 99.5th percentile of the calibration split's absolute pre-activations, the numerator is fitted on $1.2\times$ its logit range with evaluated logits clipped into it, and every temperature minimizes calibration-set NLL. Table 6 in the appendix is a controlled isolation sweep, the softmax approximated alone at several degrees under the same protocol. The

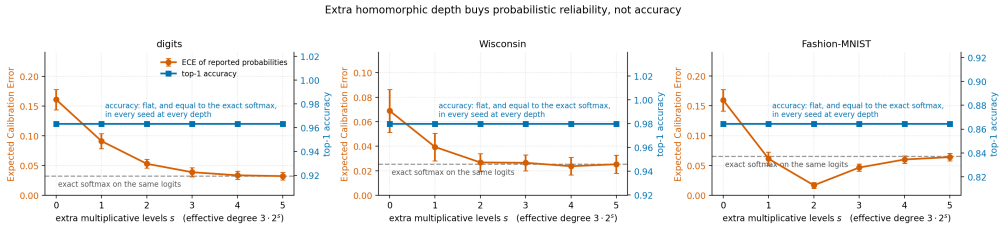


Figure 2: Extra homomorphic depth buys probabilistic reliability, not accuracy. The numerator is $(P_3(u/2^s))^{2^s}$ at s extra multiplicative levels: the domain scaling and repeated squaring of Cho et al. (2024) without their intermediate normalizations, which multiply every coordinate by one common factor and so leave the normalized output unchanged in exact arithmetic. Top-1 accuracy (blue) is at its ceiling at $s{=}0$ and equals the exact softmax in every seed at every depth; ECE (orange) does not, reaching the exact-softmax value (grey dashed) only later, or undershooting it. A deployment raising depth until accuracy stops improving stops at $s{=}0$. Whole-network MLP pipeline, 10 seeds, calibration-split interval; Table 4 gives per-depth accuracy, ECE and approximation error.

accuracy invariance holds cell by cell, and is exact by construction for the monotone numerator of §5. Throughout, the *recalibration target* is $\mathrm{softmax}(z/T^\star)$ on the pipeline's own logits with $T^\star$ minimizing calibration-set NLL. It is a reference point, not an optimum, since another scalar can score better on ECE. B2 is the same quantity for the exact plaintext model, differing only by the ReLU approximation's effect on calibration, at most 0.017 ECE.

Finding. Table 6 (isolation) shows that approximating the softmax leaves top-1 accuracy *identical* to the plaintext model at every tested degree, the argmax being preserved in these runs, while ECE rises from 0.070 to 0.679 on Fashion-MNIST and from 0.025 to 0.636 on digits at degree 2, decaying toward the baseline as the degree grows. Approximating the hidden ReLUs instead leaves ECE near baseline (0.065–0.068 Fashion; 0.026–0.044 digits) at a small accuracy cost, so the calibration damage is *dominated by the output nonlinearity*. It also has a direction: the polynomial numerator flattens the probability vector, so the encrypted model is under-confident at the top in every seed of all ten settings (Appendix C), and the client-side fix restores it.

4 Mechanism: An Unreliable Scale Fix, a Residual Shape Error

Softmax confidence is governed by the *scale* of the logits: $\mathrm{softmax}(z/T)$ sweeps from sharp to uniform. A polynomial approximation $P_k^{\exp}$ over- or under-shoots $\exp$ non-uniformly across the logit range, and after renormalization this reshapes the whole probability vector. The distortion is deterministic in the degree and fitting interval (k, I), and order-preserving for a reason weaker than monotonicity, which these fits do not have: with an exact max-shift the predicted class sits at $u{=}0$, so accuracy survives whenever the floored numerator is largest there, which is checkable before deployment (Proposition 9).

A scalar temperature is the exact inverse of a scale error and nothing more. It rescales every pairwise log-odds by one factor, whereas polynomial softmax applies $\log P(\cdot)$, non-affine for a non-exponential polynomial (Proposition 1); changing T changes the magnitude of the distortion but not its shape. Temperature scaling *inside* the pipeline (B1, Table 3) is accordingly *unreliable*: it sits at ECE 0.181 on the Fashion-MNIST CNN and 0.189 on PathMNIST against 0.019 and 0.025 once the softmax is deferred (R1), and the share of the excess over the exact softmax that it removes swings from 36% on CIFAR-100 to 76% on BloodMNIST. The claim is not that a temperature never reduces error, but that a scalar is not a general inverse for this shape and nothing in the deployment says which case one is in. Where the residual vanishes a scalar *is* an exact inverse, and at finite residual an inverse up to a bound we state (Proposition 6). That is why folding the calibration temperature into the head before encryption, free on ciphertext, fixes nothing at the shipped depth (Appendix C) but becomes the remedy for an encrypted output once the depth is set by the rule below (R3, §5.6).

A pre-deployment calibration diagnostic. The same algebra gives a check a practitioner can run *before* encrypted inference. Write $u_i = z_i - \max_j z_j \le 0$, let $P := P_k^{\exp}$ be the public polynomial,

and set $e(u) = \log P(u) - u$ wherever $P(u) > 0$. Before the floor, the polynomial softmax perturbs every pairwise log-odds by exactly $e(u_i) - e(u_j)$, so the worst-case distortion over the operating range is

$$D(P, \delta) \;=\; \max_{u \in [-\delta, 0]} e(u) \;-\; \min_{u \in [-\delta, 0]} e(u).$$

D needs no labels and no encrypted inference, though δ must come from a deployment bound or an *unlabeled* sample: label-free once the range is fixed, not assumption-free (Proposition 4). It is zero exactly when $P(u) = ce^u$ and is *undefined* wherever $P \leq 0$, which is not a corner case: at all ten operating ranges (δ from 22 to 94) the degree-3 fit is non-positive somewhere, so the shipped numerator stays positive only because of the floor. We use raw D only where $P > 0$ on the whole interval. The floored numerator $\max(P, \varepsilon)$ has a finite analogue D_ε (Proposition 4), which we do not use to choose anything, and neither quantity is a predicted ECE. The repeated-squaring schedules $(P_k(u/2^s))^{2^s}$, which omit the normalizations of Cho et al. (2024), satisfy $D_{\text{comp}} = 2^s D(P_k, \delta/2^s)$.

From diagnostic to depth rule. Because D is computable from public quantities alone, before any encrypted inference and without labels, it can be used to *choose* depth and not only to audit it. The rule takes the shallowest depth at which the numerator is positive on $[-\delta, 0]$ and $D(P, \delta) \leq \tau$; it is a calibrated heuristic, not a guarantee. It is worth having because the depth that preserves predictions need not preserve probabilities. At the shallowest depth whose numerator is positive across the whole operating interval without the floor, top-1 accuracy already equals the exact softmax's in every seed of every setting, while the calibration gap averages up to 0.020 and reaches 0.028 in the worst seed; reaching $\tau = 0.3$ costs one or two further squarings and brings the mean gap to at most 0.0022 across all seven backbones, within 0.005 on 67 of 69 seeds. The threshold τ is one global constant chosen with these benchmarks in view, so that hit rate is in-sample; a threshold chosen instead on nine of the ten settings meets the target on 98 of 99 seeds of the held-out one (Appendix E).

5 Method: Keep the Softmax Out of the Ciphertext

The mechanism points to one rule: do not approximate the *output* softmax under encryption. What that rule costs, and what to do when it cannot be followed, depends on the interface.

5.1 Deployment Setup

A client holds the secret key; a server evaluates the model on ciphertext, with z and u_i as in §3. Three interfaces matter. In the *logit-returning* interface, the server returns encrypted z and the client decrypts it. In the *numerator-returning* interface, which existing polynomial-softmax pipelines expose, the server evaluates the exponential's polynomial surrogate on ciphertext and returns encrypted $\eta_i = P(u_i)$, normalized after decryption. In the third, a server-side consumer such as an encrypted threshold decision or a sampler reads the distribution itself under encryption, so nothing can be undone on the client. Throughout, P is *public*: degree, coefficients, and fitting interval are part of the deployed pipeline. The client also holds a small labeled calibration set, disjoint from the test data: inverting P needs no labels, but fitting a temperature needs a few hundred labeled points, which is why the recalibration target of §3 is a stated reference rather than a ground truth.

5.2 Temperature Scaling, and What It Is Not

Temperature scaling (Guo et al., 2017) divides the logits by one scalar, $\hat{p} = \text{softmax}(z/T)$, with T fitted on the labeled calibration set by minimizing negative log-likelihood; we claim no novelty for it. What matters is *where* it is applied. Inside the pipeline, the result passes through the distorted map, the unreliable baseline B1 (§4); applied to logits that never met it, it is the first remedy.

5.3 Deferred Output Softmax (R1)

When the client may receive logits, keep only the linear head under encryption and move the softmax out of the ciphertext entirely: the server evaluates the encrypted model up to the logits and returns the encrypted logit vector; the client decrypts it, divides by the scalar T fitted on its calibration set, and applies the exact softmax. This adds no ciphertext operation beyond the linear head.

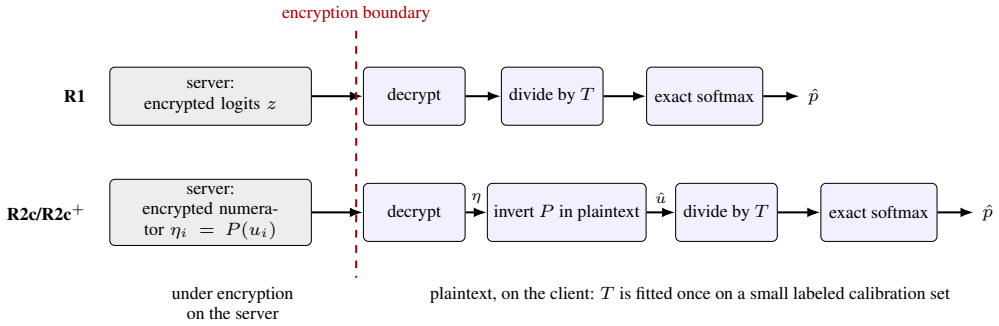


Figure 3: **The two client-side remedies.** R1 applies when the interface may return encrypted logits: the client decrypts them, divides by a temperature fitted on a labeled calibration set, and applies the exact softmax. R2c is a compatibility mechanism for a pipeline that already evaluates and returns the polynomial numerator under encryption: the client decrypts η, inverts the *public* polynomial in plaintext to recover shifted logits, and then temperature-scales; R2c$^+$ is the same path with a numerator built to be strictly monotone, so the inversion is well defined. Nothing is inverted, maximized or normalized under encryption, and the classifier weights never change. We contribute no encrypted-maximum primitive: both rows assume the deployment can already form the shifted logits.

What is and is not new. Returning logits is common in FHE classifiers and temperature scaling is standard: **R1 is a deployment design rule justified by our calibration analysis, not a new calibration algorithm**. What it adds is the reason to prefer that interface and what the alternative costs; Proposition 2 states what R1 computes and how it degrades under logit noise.

5.4 Recovering shifted logits from a polynomial numerator (R2c)

R2c is a compatibility mechanism, not a security argument: it serves an existing pipeline or API that already returns polynomial numerators, with the server unchanged. The client has a handle the literature does not usually exploit: P is public and deterministic, so its effect can be undone. The server evaluates $\eta_i = P(u_i)$ under encryption; the client decrypts η, inverts P **in plaintext** wherever it is monotone on the encountered range, $\hat{u}_i = P^{-1}(\eta_i)$, then divides by T and applies the exact softmax, $\mathrm{softmax}(\hat{u}/T)$. **Nothing is inverted under encryption.** Where the raw Chebyshev fit loses monotonicity on a wide range, the implementation inverts a clipped monotone rearrangement instead; that is a heuristic and every table marks it as one. The inversion is deterministic and needs no learned recalibration map: it stays within 0.003 of the target at every tested degree on digits but degrades with degree on Fashion-MNIST, to 0.057 at degree 7 against the target's 0.012, which the next construction removes. R2c removes a *known, deterministic* distortion rather than learning better probabilities; the calibration set then fixes the confidence scale.

5.5 Monotone-by-construction numerator (R2c$^+$)

R2c's only requirement is that the deployed numerator be strictly monotone; its quality as an $\exp$ approximation is irrelevant, since the client inverts P and applies the true softmax. Rather than check monotonicity and fall back when it fails, we construct it: parametrize the derivative as a strictly positive sum of squares, $P'(u) = \varepsilon + \mathbf{z}(u)^\top Q\,\mathbf{z}(u)$ with $Q \succeq 0$, $\mathbf{z}(u) = (1, u, \ldots, u^m)$ and $\varepsilon > 0$, certified algebraically rather than on a grid, and integrate with a positive constant so $P > 0$ (Proposition 5). Strict monotonicity makes P invertible, so $P^{-1}(P(u)) = u$ exactly in the noiseless limit, and it turns the accuracy invariance into a *theorem*, the softmax argmax being invariant under any increasing map of the numerators; the implemented inverse still carries interpolation and CKKS error. The price is a much worse *un-recalibrated* output (at degree 3 its B0$^+$ is 0.56/0.60 on Fashion-MNIST/digits against B0's 0.16 on both, and still 0.16/0.17 at degree 7), immaterial only when the client always inverts.

5.6 When the distribution itself must stay encrypted (R3)

R1 and R2c both finish on the client. When a server-side consumer reads the distribution under encryption, there is no client-side inverse, and two levers remain, the logit scale and the numerator depth. R3 uses both: the temperature $T^\star$, fitted once on the labeled calibration set, is folded into the head before encryption ($W/T^\star$ and $b/T^\star$, one public scalar), the numerator interval is refitted on the folded calibration logits, and the depth is the first s with $D \le \tau$ (§4). In exact-arithmetic simulation, across the three MLP benchmarks and the seven backbone settings, R3 lands within 0.004 ECE of R1's client-side output on 96 of 99 seeds and within 0.008 on all, at 3 to 7 extra levels (beyond four the ring dimension doubles at a 2^{40} scale, Appendix D), with top-1 identical to the exact softmax's at the selected depth; raising depth without folding instead converges to the exact softmax's own miscalibration, about 0.10 above R1 on CIFAR-100 (Appendix E). Folding moves the range either way: it shrinks δ where $T^\star > 1$, from 43 to 25 on Fashion-MNIST and 60 to 38 on CIFAR-100, saving a level, and widens it where $T^\star < 1$, as on the CNNs, costing up to one.

5.7 Assumptions, limits, and the relation to prior encrypted-calibration work

R2c and R2c$^+$ assume the deployment can already form $u_i = z_i - \max_j z_j$. **We contribute no encrypted-maximum, inversion or normalization primitive**; Appendix A runs an encrypted maximum on the binary Wisconsin stack, the one setting here that computes one on ciphertext. What they do not need is the un-normalized numerator: Proposition 8 shows the normalized vector already determines it once the max-shift is exact and P is public, monotone and positive, and recovering that way reaches the same ECE to the last digit on all three pipelines. Where logits may be returned R1 is simpler and equals R2c$^+$ everywhere we measure (Table 7); R2c exists for compatibility, not secrecy. Neither hides the logit vector, and no method can once the client receives the full distribution, which determines the logits up to a shift and the known temperature; a deployment that must keep them hidden has to consume the distribution under encryption, the case R3 serves. What R2c$^+$ changes is what the *ciphertext* must remain, a positive, exponential-like numerator a server-side consumer can keep computing on. The interface of Pathan et al. (2026) is superficially similar but serves adaptation under distribution shift, updating (W, b) from encrypted features, whereas R2 recovers shifted logits to repair the probability output with (W, b) unchanged.

6 Experiments

Setup. Fashion-MNIST (Xiao et al., 2017), the 8×8 handwritten-digit set of Alpaydin and Kaynak (1998) ("digits" throughout) and the Wisconsin diagnostic breast-cancer set of Street et al. (1993) ("Wisconsin"); an MLP with a 64-d penultimate layer; **10 seeds**, disjoint splits. Hidden activations are approximated at degree 4 and the softmax at degree 3, so all methods share one base network and differ only in output handling: B0 (no recalibration), B1 (in-pipeline temperature), R1 (defer-softmax + logit temperature) and B2 (plaintext temperature scaling on the exact model).

Main results. Under the calibration-only protocol (Table 1) the un-recalibrated pipeline ships 0.159, 0.161 and 0.069 while R1 and R2c$^+$ reach 0.012, 0.021 and 0.026, and the Chebyshev inversion R2c reaches 0.028, 0.021 and 0.028. The order-preserving methods share accuracy throughout (0.864 Fashion, 0.963 digits). The recommendation: *defer the softmax where the deployment permits it; where a pipeline already returns polynomial numerators, invert them deterministically*.

Additional backbones, datasets, and label spaces. We repeat the study on a trained CNN (conv32–conv64–conv64–GAP) on Fashion-MNIST, MNIST and three MedMNIST benchmarks (Yang et al., 2023) (PathMNIST, 9 classes; BloodMNIST, 8; DermaMNIST, 7), and on a ResNet-18 reaching 0.945/0.742 on CIFAR-10/100, whose penultimate features feed the approximated head. Table 7 in the appendix separates the two approximations across all seven settings. (i) On the same logits, the polynomial softmax preserves top-1 accuracy *exactly* in every seed of every row while inflating ECE by $1.1\times$ to $30\times$. (ii) The hidden ReLU alone leaves ECE within 0.019 of baseline but is what costs accuracy (up to 7.3 points on PathMNIST): the two approximations damage *different* properties. (iii) In-pipeline temperature scaling (B1, Table 3) reaches the target on none of the seven backbones: it removes 36% to 76% of the excess over the exact softmax and still ships $2.5\times$ to $30\times$ R1's ECE. Both remedies recover calibration everywhere: R2c$^+$ equals R1 throughout

Table 1: **The primary method comparison**, under the calibration-only protocol of §3: the hidden-ReLU fit covers the calibration split's pre-activation range, the numerator is fitted once on $1.2\times$ the *calibration* split's logit range (Fashion-MNIST 43.4 ± 2.1, digits 25.0 ± 1.6, Wisconsin 26.1 ± 5.2) and evaluated logits are clipped into that public interval, a clip that binds on 0.00% of them. Every number in the main text is recorded under this protocol. No test information selects the clip range, the interval, the degree or the temperature. ECE, mean±std over 10 seeds.

Method	Fashion-MNIST	digits	Wisconsin
B0	$0.159 \pm .018$	$0.161 \pm .017$	$0.069 \pm .018$
R2c	$0.028 \pm .007$	$0.021 \pm .006$	$0.028 \pm .011$
R2c$^+$	$0.012 \pm .003$	$0.021 \pm .006$	$0.026 \pm .009$
R1	$0.012 \pm .003$	$0.021 \pm .006$	$0.026 \pm .009$

because its inversion is exact, while Chebyshev-R2c matches them to within 0.001 everywhere except CIFAR-10 (0.019 against 0.012) and DermaMNIST (0.055 against 0.052), where the clipped monotone fallback leaves a gap.

What governs the size of the damage. On the same logits, these settings lose very different amounts ($+0.003$ ECE on the CIFAR-10 ResNet to $+0.47$ on the Fashion-MNIST CNN), and the diagnostic says why: the bound is *range*-conditioned, so the damage should track the operating logit range rather than architecture or task, and the ECE increase rises with δ at Spearman $\rho = 0.82$ over the seven backbone settings. That is an association over two architecture families, not seven independent draws; the causal evidence is a controlled sweep that rescales real logits with all else fixed and shows damage rising with δ and then saturating (Appendix C). At the extreme of that range the in-ciphertext numerator stops applying at all: a whole-network encrypted CNN leaves a calibration range of $\delta \approx 149$ and 254, and its floored numerator carries information on under six percent of test points at any degree and depth we can afford, so R1 is the remedy that fits such a deployment (Appendix A). The deployments at risk therefore have wide logit ranges and wide label spaces, both known before encrypted inference runs, and narrowing the range does not rescue them: folding the calibration temperature into the head before encryption, which is free, leaves the shipped ECE almost unchanged while the exact model's collapses, so the inflation ratio *rises*. The fit is a lever too, and not the one approximation theory suggests: holding degree, interval and depth fixed and swapping the deployed least-squares numerator for the minimax, Chebyshev-node or truncated-Maclaurin fit leaves top-1 accuracy identical in all 28 cells while ECE moves over a fortyfold range on CNN MNIST, and the minimax fit, which minimizes worst-case approximation error by construction, is not the best calibrated (Appendix F).

Depth buys reliability, not accuracy. Running domain scaling and repeated squaring without the intermediate normalizations of Cho et al. (2024), a low-degree polynomial on a scaled argument squared at one extra level per squaring, separates the two quantities a deployment might tune for (Figure 2, Table 4). Accuracy is at its ceiling immediately, equalling the exact softmax on the same logits at $s{=}0$ and at every depth on all three benchmarks, paired per-seed difference 0.0000 with bootstrap 95% CI $[0, 0]$. Calibration is not: it arrives at $s{=}4$ on digits and $s{=}2$ on Wisconsin, and on Fashion-MNIST it *undershoots* at intermediate depth, dipping to 0.017, below the exact softmax's own 0.065, before climbing back, because that model is over-confident and the polynomial's distortion is a flattening, so the two errors partly cancel. That cancellation is accidental, invisible from inside the deployment, and does not transfer; a client-side temperature reaches 0.012 and stays. Table 5 prices the stopping rules, and none of the label-free ones *locates* the calibration-tight depth: accuracy stops one to four levels early and ships 2.5–$5.0\times$ the exact softmax's ECE, while the one rule that is never early over-provisions by up to four levels, already a ring-dimension doubling and 214% more ciphertext (Fashion-MNIST, $s{=}5$ against the calibration-tight $s{=}1$: 3520 against 1120 KiB). Relative approximation error is a bound rather than a proxy, and output KL misses in both directions. The separation is not an artifact of the architecture: a small transformer with exact attention reproduces it in all 120 seed-depth cells (Appendix H). Enough depth does remove the failure, which is why we frame the contribution as locating the low-depth regime, and why §4 gives a label-free rule for leaving it.

Does it change decisions, and would a flexible calibrator do? The distortion is only mildly visible to *ranking*: against the deferred-softmax reference AURC changes by -4% to $+56\%$ on nine settings and rises $2.6\times$ on the CIFAR-10 ResNet (Appendix C). What changes is absolute confidence, and with it any threshold policy: a client acting automatically at confidence ≥ 0.9 decides 47% of digit cases against the calibrated references' 87–90%, and under 5% on six of seven backbone settings. Because the distortion is under-confidence, which Proposition 11 shows the floor can only produce, the cost is lost coverage rather than unsafe decisions, and a transplanted plaintext cut-off violates its error target in no seed. The probability *value* is a different matter: under an asymmetric cost matrix, whose threshold the cost ratio fixes and cannot be re-tuned, the shipped probabilities cost more than the plaintext reference in all nine benchmark-by-ratio cells, and refitting one scalar does not reliably repair it. A flexible post-decryption calibrator is no substitute: top-label isotonic regression (Zadrozny and Elkan, 2002) rivals R1 on top-label ECE only, with NLL and Brier worse in all ten settings, while Dirichlet calibration (Kull et al., 2019) changes predictions.

CKKS confirmation and what the levels cost. To rule out a numerical-simulation artifact, we rerun the pipeline under TenSEAL CKKS (Benaissa et al., 2021), evaluating the linear head and the degree-3 numerator on ciphertext for two MLPs, in one setting the maximum as well, as an arithmetic check for a linear head on ResNet-18 features, and for a small convolutional network whole (Table 2). The shipped output reproduces its plaintext simulation to within 2×10^{-4} ECE in every run, the decrypted logits give the simulation's label on every point, and R1 repairs calibration as it does in simulation, so the effect is not an artifact of simulating the arithmetic. Nor is it an artifact of small label sets: in exact-arithmetic simulation with an exact max-shift, GPT-2 and Llama-3.2-1B over their full vocabularies keep the predicted token at every depth while ECE at three levels is 0.36 to 0.50 (Appendix A). The encrypted CNN marks the limit of the numerator route: its logit range is too wide for a numerator under a public shift at any affordable depth, which then *does* lose the prediction, and R1 is what recovers calibration there. Appendix D prices the levels at 128-bit security (Albrecht et al., 2019): at a 2^{40} scaling factor the one to four extra levels share one ring dimension (N=16384) for 67% more ciphertext, while 2^{30} and 2^{50} factors each force the next N. R1 adds nothing beyond the linear head, so the recommended path is also the cheaper one.

7 Discussion, Limitations, and Conclusion

In HE inference, the multiplicative depth sufficient to preserve predictions can be systematically insufficient to preserve output probabilities, and accuracy certifies only the first. Buying the second costs multiplicative levels that accuracy never asks for, so a deployment whose consumers read probabilities must price that fidelity beside accuracy and depth. Seven limits (Appendix C). (i) The whole-network pipelines are the MLP benchmarks and the encrypted CNN of Appendix A; the ResNet-18 results approximate a plaintext backbone's head, and every polynomial-ReLU arm clips to the fitted interval, which a ciphertext cannot do. (ii) CIFAR-100 uses 9 seeds, one fewer than the rest. (iii) On Fashion-MNIST calibration is non-monotone in depth, so "more depth is safer" is false there. (iv) R2c needs a monotone numerator and an exact max-shift, though not the un-normalized one (Proposition 8), and we contribute no encrypted maximum. (v) The accuracy-invisibility is specific to the *output* softmax and to an exact max-shift: inside attention accuracy needs at least as much depth as calibration in 17 of 20 runs (Appendix H), inside the MLP blocks the prediction is never preserved, and under a public constant shift it fails on GPT-2 at every depth and on Llama-3.2-1B below six levels (Appendix A). (vi) R1 sits marginally above the plaintext baseline on three settings, within one standard deviation. (vii) R3 is measured only in exact-arithmetic simulation: our CKKS runs score the degree-3 numerator without squaring and only time the squared schedules (Appendix D); none covers an encrypted normalization or a server-side consumer, and the server must know $T^\star$.

Reproducibility Statement

All experiments use public datasets and fixed seeds. We release the code, configurations, and machine-readable per-seed outputs for every setting, with the scripts that regenerate each table from them; every deployment parameter behind a reported number is read from the calibration split, except in the top-k analysis that Appendix A flags. The backbone tables are recomputed from each

trained network's cached penultimate features, so checking them needs no retraining. Dependency and rerun provenance, including NumPy-sensitive Fashion-MNIST subsampling and the archived three-seed CIFAR-10 pilot, is documented in the repository README and the machine-readable logs.

AI-Use Statement

Large language models were used as a tool in preparing this work: implementation assistance, writing and debugging the experiment, analysis and plotting code; writing assistance, drafting and editing prose; literature search; and help checking the arguments in Appendix B. Experiments use fixed seeds and the released code. Figures 1 and 3 are TikZ drawings of steps the implementation performs, and no generated image appears in the paper. The authors take full responsibility for the work, including its experimental design, code, proofs and citations.

References

Martin Albrecht, Melissa Chase, Hao Chen, Jintai Ding, Shafi Goldwasser, Sergey Gorbunov, Shai Halevi, Jeffrey Hoffstein, Kim Laine, Kristin Lauter, Satya Lokam, Daniele Micciancio, Dustin Moody, Travis Morrison, Amit Sahai, and Vinod Vaikuntanathan. Homomorphic encryption standard. Cryptology ePrint Archive, Paper 2019/939, 2019.

Ethem Alpaydin and Cenk Kaynak. Optical recognition of handwritten digits. UCI Machine Learning Repository, 1998. DOI 10.24432/C50P49; distributed with scikit-learn as `load_digits`.

Ayoub Benaissa, Bilal Retiat, Bogdan Cebere, and Alaa Eddine Belfedhal. TenSEAL: A library for encrypted tensor operations using homomorphic encryption. In *ICLR Workshop on Distributed and Private Machine Learning (DPML)*, 2021. arXiv:2104.03152.

Aymen Bouguerra, Daniel Montoya, Alexandra Gomez-Villa, Chokri Mraidha, and Fabio Arnez. Less precise can be more reliable: A systematic evaluation of quantization's impact on VLMs beyond accuracy. *arXiv preprint arXiv:2509.21173*, 2025. Title and author order as of the current version (v6, May 2026).

Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers. In *Advances in Cryptology – ASIACRYPT*, pages 409–437, 2017. doi: 10.1007/978-3-319-70694-8_15.

Wonhee Cho, Guillaume Hanrot, Taeseong Kim, Minje Park, and Damien Stehlé. Fast and accurate homomorphic softmax evaluation. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 4391–4404, 2024. doi: 10.1145/3658644.3670369. arXiv:2410.11184.

Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy. In *International Conference on Machine Learning (ICML)*, pages 201–210, 2016.

Jiawei Gu, Fengyuan Nie, Hao Tang, and Yanpeng Sun. Precision-induced miscalibration: Understanding and correcting confidence distortion in quantized neural networks. In *International Conference on Machine Learning (ICML)*, 2026. ICML 2026 poster 60537.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, pages 1321–1330, 2017.

Nikola Jovanović, Marc Fischer, Samuel Steffen, and Martin Vechev. Private and reliable neural network inference. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1663–1677, 2022. doi: 10.1145/3548606.3560709.

Meelis Kull, Miquel Perelló-Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with Dirichlet calibration. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019. arXiv:1910.12656.

Junghyun Lee, Eunsang Lee, Joon-Woo Lee, Yongjune Kim, Young-Sik Kim, and Jong-Seon No. Precise approximation of convolutional neural networks for homomorphically encrypted data. *IEEE Access*, 11:62062–62076, 2023. doi: 10.1109/ACCESS.2023.3287564. arXiv:2105.10879.

Rui Li, Wenyuan Wu, and Weijie Miao. Decision-aware quadratic relu replacement for he-friendly inference. *arXiv preprint arXiv:2605.22237*, 2026.

Lawrence Lim, Jiaming Liu, Vikas Kalagi, Divyakant Agrawal, and Amr El Abbadi. Hyperion: Private token sampling with homomorphic encryption. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14150–14159, 2026. doi: 10.18653/v1/2026.acl-long.644. Cryptology ePrint Archive, Paper 2025/2318.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *AAAI Conference on Artificial Intelligence*, volume 29, 2015. doi: 10.1609/aaai.v29i1.9602.

Hanjun Park, Byeongseo Min, Jiheon Woo, Min-Wook Jeong, Jongho Shin, Yongwoo Lee, Young-Sik Kim, and Yongjune Kim. CGF-Softmax: A cumulant-based softmax reformulation for efficient inference under homomorphic encryption. *arXiv preprint arXiv:2602.01621*, 2026.

Shadman Mahmood Khan Pathan, Qianlong Wang, Jonathan Takeshita, Sakan Binte Imran, and Sachin Shetty. Privacy-preserving post deployment model calibration with fully homomorphic encryption. In *International Conference on Computing, Networking and Communications (ICNC)*, pages 1–6. IEEE, 2026. doi: 10.1109/ICNC68183.2026.11416991.

Shahzeb Qamar, Lorenz Sparrenberg, Christian Bauckhage, Baha Rababah, Carson Leung, Murat Kantarcioglu, Cuneyt Gurcan Akcora, and Rafet Sifa. Accuracy is not enough: A divergence-based approach to evaluate fidelity loss in quantized LLMs. *arXiv preprint arXiv:2609.07664*, 2026.

Imtiyazuddin Shaik, Sitarama Brahmam Gunturi, Phani Sai Uppu, Rajan Mindigal Alasingara Bhattachar, and Kanaka Mahalakshmi Pathivada. Method and system for calibrating machine learning models in fully homomorphic encryption applications. U.S. Patent Application 20240054401; U.S. Patent 12,585,999, 2024. Priority filing July 26, 2022; U.S. patent granted March 24, 2026.

W. Nick Street, William H. Wolberg, and Olvi L. Mangasarian. Nuclear feature extraction for breast tumor diagnosis. In *Biomedical Image Processing and Biomedical Visualization*, volume 1905 of *Proc. SPIE*, pages 861–870, 1993. doi: 10.1117/12.148698.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Jianhang Xie, Sicheng Tan, Vishnu Naresh Boddeti, and Zhichao Lu. ATLAS: Automated approximation of transformers for efficient homomorphic inference in one hour. *arXiv preprint arXiv:2607.23478*, 2026.

Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. MedMNIST v2 – a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10:41, 2023. doi: 10.1038/s41597-022-01721-8.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 694–699, 2002. doi: 10.1145/775047.775151.

Jize Zhang, Bhavya Kailkhura, and T. Yong-Jin Han. Mix-n-Match: Ensemble and compositional methods for uncertainty calibration in deep learning. In *International Conference on Machine Learning (ICML)*, pages 11117–11128, 2020.

Lening Zhao, Qipeng Zhan, and Li Shen. Invertible logits transformation for accuracy-preserving post-hoc uncertainty calibration. *arXiv preprint arXiv:2608.10372*, 2026.

Itamar Zimerman, Allon Adir, Ehud Aharoni, Matan Avitan, Moran Baruch, Nir Drucker, Jenny Lerner, Ramy Masalha, Reut Moshe, and Omri Soceanu. PowerSoftmax: Towards secure LLM inference over encrypted data. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 300 of *Proceedings of Machine Learning Research*, pages 5203–5211. PMLR, 2026. arXiv:2410.09457.

Alessandro Zirilli, Davide Marincione, Evgenios M. Kornaropoulos, Giuseppe Ateniese, and Emanuele Rodolà. HEAT: Faster fully homomorphic inference via approximations-weights co-adaptation. Cryptology ePrint Archive, Paper 2026/1862, 2026. https://eprint.iacr.org/2026/1862.

A Encrypted Evidence and Scope

Six encrypted confirmations. Table 2 collects every run in this paper that uses real ciphertext. Two use the MLP pipeline under TenSEAL 0.3.16 at polynomial modulus degree 32768 with an 880-bit chain; two encrypt the cached 64-dimensional pre-activation features of the ResNet-18 of Table 7 under TenSEAL 0.3.18 at degree 16384, chain $(60, 40, 40, 40, 40, 60)$ and scale 2^{40}, and run a linear head and the numerator on ciphertext without the hidden ReLU, so they check CKKS arithmetic on a head that is not the one in Table 7; and two evaluate a convolutional network whole on ciphertext at degree 32768, chain $(60, 50^{10}, 60)$ and scale 2^{50}. Only the Wisconsin row evaluates the maximum on ciphertext, as an encrypted binary tournament; the multiclass rows supply the maximum or a public shift in plaintext, and normalization and temperature fitting always happen after decryption. The digits row also approximates the hidden activation and is where the accuracy cost of §3 appears under real encryption: 0.856 against the exact model's 0.972, a loss the plaintext simulation of the same polynomial network reproduces point for point, so it belongs to the ReLU approximation and not to CKKS. A separate n=1000 run recovers plaintext logits to 2.6×10^{-4} and evaluates the monotone numerator to within 3.4×10^{-4} of its plaintext value while satisfying its positivity and derivative certificates.

Three things follow. The shipped output B0 reproduces its plaintext simulation to between 4×10^{-8} and 2×10^{-4} ECE in every run, so the exact-arithmetic sweeps are a validated stand-in; R1 and R2c, which pass decrypted logits through a fitted temperature, differ from their simulations by up to 0.004 and 0.007 in the mean on the whole-network digits run, whose polynomial ReLUs carry CKKS error of about 0.05 into the logits. The label from the decrypted logits matches the simulation on every point of every seed, and on the ResNet rows the numerator's top-1 equals the exact softmax's on every point. The remedies land where the simulation puts them, except that on CIFAR-10 the Chebyshev inversion R2c is worse than the pipeline it repairs, because inverting a degree-3 numerator is ill-conditioned on that narrow a range, the case R2c$^+$ is built for.

A whole-network encrypted CNN, and the range that defeats the numerator. The Square-CNN rows run a CryptoNets-style network (Gilad-Bachrach et al., 2016) end to end on ciphertext, with square activations, the max-shift a public calibration constant and nothing in the clear until the client decrypts. The ciphertext is faithful: encrypted and plaintext top-1 agree on all 1200 points, the largest logit error is 4.4×10^{-5} against class spreads of 29 and 62, and the encrypted B0 ECE matches its simulation to 2.2×10^{-7}. The numerator itself is not usable, for a reason checkable before deployment. Two squarings leave a calibration range of $\delta \approx 149$ and 254, against 22 to 94 elsewhere in this paper, and a public shift cannot follow each example, so the numerator carries information only where the top shifted logit exceeds $\ln \varepsilon = -18.4$ (Proposition 10). The median top shifted logit is -29.5 to -86.3 and only 0.15 to 5.4% of test points clear that bound; sweeping degree 3 to 15 against $s = 0$ to 6, up to 11 levels, recovers neither accuracy nor calibration in any cell. Here accuracy *does* reveal the failure, and R1 is the requirement rather than the convenience: working from the decrypted logits, it recovers the exact softmax's ECE to within 0.002 at the accuracy the encrypted model already had.

Vocabulary scale. We ran the same simulation path, with an exact max-shift, on GPT-2 ($C = 50257$) and Llama-3.2-1B ($C = 128256$) over their full vocabularies, each on three disjoint stretches of the WikiText-2 test split. Argmax agreement with the exact softmax is 1.0000 at every depth and degree in both models, and calibration does not hold: at three levels, where accuracy is already

Table 2: Every real-ciphertext run in this paper. "Ciphertext" names the operations evaluated under encryption. ECE, mean over 3 seeds, with 114 test points per seed on Wisconsin, 60 on digits and 200 on every ResNet and CNN row. "exact" is the exact softmax on plaintext logits; on the two MLP rows those come from the exact-ReLU network, hence the higher accuracy on digits. "enc−sim" is the largest gap between the encrypted B0 ECE and its plaintext simulation. The ResNet rows use 200 points, where binned ECE is biased upward, most at 100 classes, and apply the linear head to the pre-activation features without the hidden ReLU, so their logits, accuracy and calibration-split interval (δ of 28 to 35 and 77 to 91, against 22 and 60 in Table 7) differ from Table 7: they validate the arithmetic against its own simulation and are not comparable cell by cell. On the Square-CNN rows B0 and R2c are omitted: under a public shift the degree-3 numerator does not keep the prediction (top-1 0.13 to 0.31 against 0.87 to 0.98 in the plaintext sweep of the same network), so its ECE would describe a different classifier.

Setting	ciphertext	acc. enc. / exact	exact	B0	R1	R2c	enc−sim (ECE)	s / point
MLP Wisconsin	head, numerator, max	0.977 / 0.977	0.023	0.062	0.032	0.049	1×10^{-4}	54.4
MLP digits	head, numerator, ReLU	0.856 / 0.972	0.040	0.354	0.090	0.095	2×10^{-4}	55.7
R18 CIFAR-10	head, numerator	0.958 / 0.958	0.031	0.050	0.031	0.088	7×10^{-7}	1.2
R18 CIFAR-100	head, numerator	0.750 / 0.750	0.145	0.535	0.081	0.081	2×10^{-6}	1.2
Square-CNN Fashion	conv, squares, head, numerator	0.903 / 0.903	0.042	–	0.041	–	4×10^{-8}	20.6
Square-CNN MNIST	conv, squares, head, numerator	0.968 / 0.968	0.025	–	0.025	–	2×10^{-7}	20.4

exact, polynomial ECE is 0.361, 0.428 and 0.396 on GPT-2 against exact values of 0.038, 0.032 and 0.029, and 0.479 to 0.497 on Llama-3.2-1B against 0.023 to 0.033, returning to the exact value only at nine and eight levels. The gap widens because δ grows with the class count: restricting each position to its top k logits, keeping the positions whose target is among them and refitting the interval on these evaluated logits, gives δ of about 14, 17, 21 to 22 and 26 at $k = 10$, 10^2, 10^3 and 10^4, with polynomial ECE rising alongside it while the exact ECE stays flat. D does not rank ECE within either model, but every setting with $D < 1$ matched the exact-softmax ECE to within 0.003.

Two regimes, and which recommendation applies to each. The bound above is about the operating range, not about encryption. It bites at $\delta \approx 149$ and 254, the calibration-split logit range of a whole network evaluated end to end in ciphertext. Every head-only pipeline here, which is what deployments ship when the feature extractor runs in the clear or on the client, has δ between 22 and 94, and at those widths the numerator is feasible: the clip binds on 0.00% of test entries in all seven backbone settings, the depth rule selects a depth in all sixty-nine seeds, R2c$^+$ equals the client-side result, and R3 lands within 0.004 of it on 96 of 99 seeds in simulation. The paper therefore says different things about the two. In the narrow-range regime the numerator works, accuracy hides the calibration error, and the rule and the remedies apply. In the wide-range regime the numerator carries usable information on under six percent of points at any degree and depth we can afford, and the recommendation is R1.

A public shift instead of an encrypted max. An exact per-example max is a circuit, so a deployment may instead subtract a public constant taken from the calibration split. Proposition 10 then predicts usability from one unlabeled pass: on GPT-2 the median top shifted logit is -250 to -270 and only 0.5 to 0.7% of positions clear $\ln \varepsilon$, while on Llama-3.2-1B, whose logits are far more tightly spread, the median is -9 to -11 and 99.5 to 99.8% clear it. Running that pipeline end to end at degree 3, with depth swept and an exact-exponential control, confirms the split, and it is where the accuracy invariance breaks. On GPT-2 agreement with the exact softmax never exceeds 0.75 and is 0.004 to 0.006 at nine levels, where over 99% of rows are exactly uniform; the exact-exponential control sits at 0.005 to 0.007, so the floor, not the polynomial, does the damage. On Llama-3.2-1B agreement is only 0.32 to 0.45 at three levels and 0.987 to 0.997 at six, while ECE is 0.23 to 0.39 at every depth against exact values of 0.023 to 0.033; the control reproduces the deep-end ECE to within 0.001, because at $C = 128256$ the floored tail adds about $C\varepsilon = 1.3 \times 10^{-3}$ to the denominator, which dwarfs the surviving top numerator. Clearing the floor is necessary for a usable numerator and certifies nothing about calibration.

The same approximation inside the MLP blocks. Replacing the activation in every MLP block by a Chebyshev fit of that activation, on GPT-2 (12 blocks, two samples) and Llama-3.2-1B (16 blocks), over four fitted ranges and degrees 2 to 32 (sixty settings), reverses both findings. The predicted token is never preserved: agreement with the exact model peaks at 0.930, 0.936 and 0.990 on the three runs. And calibration returns with accuracy: over the nineteen settings whose accuracy is at least 99% of the exact model's, ECE is 0.62 to 1.20 times the exact value, at a cost of at least 36 levels on GPT-2 and 64 on Llama-3.2-1B, against 3 to 9 for the output numerator. At both other positions we measured, accuracy falls when calibration does; the invisibility belongs to the final normalization.

Proposition 7 (client renormalization is exact). Let $\eta \in \mathbb{R}^K_{>0}$ and let $g > 0$ be any common approximate reciprocal. If the server returns $q_i = g\eta_i$, then $q_i / \sum_j q_j = \eta_i / \sum_j \eta_j$. *Proof.* The positive common factor cancels. □ Therefore ciphertext levels spent only on a common normalization factor are unnecessary whenever the client is allowed to renormalize the positive numerators. This statement does not apply after classwise clipping or to a consumer that requires an already normalized ciphertext vector.

B Protocol, Training and Estimators

The calibration-only protocol. We assume the standard client–server threat model: an honest-but-curious server that sees only ciphertext, and a client that holds the secret key. The model is trained on the training split. Every deployment parameter is then fixed on the calibration split alone, before the test split is touched: the clip range of the hidden degree-4 ReLU fit, the 99.5th percentile of the calibration split's absolute pre-activations; the numerator's fitting interval, 1.2 times the calibration split's logit range; its degree and depth; and every temperature. Evaluated logits outside the public interval are clipped into it, a clip that binds on 0.00% of test entries in every setting. The depth threshold τ is one global constant, and the real-ciphertext rows of Table 2 are arithmetic checks with their own recorded parameters. The backbone tables are computed from each run's cached penultimate features. AURC averages the risk over tied confidences, which the floored numerator produces. Table 6 is a controlled isolation sweep on the plaintext model's logits rather than a deployment measurement, and its interval too comes from the calibration split.

Table 3: **In-pipeline temperature scaling and the isolation arm, under the calibration-only protocol.** The test split selects no deployment parameter anywhere in this table. "exact" is the plaintext model's softmax, "B0" the shipped polynomial softmax on the pipeline's logits, "B1" the same pipeline with the logits divided by a temperature fitted on the calibration split *before* the numerator is evaluated on the same interval, and "iso" the isolation arm of Table 6: the degree-3 numerator on the plaintext model's logits, fitted to their own calibration range. ECE, mean over seeds (10 seeds, 9 on CIFAR-100). B1 reaches the recalibration target in no setting and removes between 36% and 76% of B0's excess over the exact softmax on the same logits.

setting	δ	exact	B0	B1	iso
CNN Fashion-MNIST	65.9	0.020	0.511	0.181	0.524
CNN MNIST	53.6	0.007	0.471	0.169	0.464
CNN PathMNIST	94.3	0.020	0.503	0.189	0.623
CNN BloodMNIST	58.5	0.045	0.373	0.129	0.415
CNN DermaMNIST	25.4	0.055	0.257	0.167	0.328
ResNet-18 CIFAR-10	21.9	0.033	0.037	0.035	0.024
ResNet-18 CIFAR-100	59.9	0.128	0.426	0.316	0.488

The MLP experiments use fixed 60/20/20 splits and ten seeds. The CNN is conv32–conv64–conv64–GAP trained with Adam for ten epochs; the ResNet-18 uses SGD with a cosine schedule for 60 epochs (nine seeds on CIFAR-100). Temperatures minimize calibration-set NLL. ECE uses 15 equal-width bins. Re-estimating the exact softmax and B0 on every backbone setting under an equal-mass binner moves no entry by more than 0.003, changes no ordering and leaves the depth separation intact, although Wisconsin's selected depth moves from s=2 to s=3; the per-setting table, with NLL and Brier for every entry, is in the released outputs.

Proposition 1 (shape, not scale). A scalar $T > 0$ exactly maps polynomial-softmax log-odds to temperature-scaled softmax log-odds on an interval iff $\log P(u) = u/T + d$. *Proof.* Fix one coordinate and equate every pairwise log-odds difference; the equality forces $\log P$ to be affine with positive slope. Conversely an exponential of that affine function gives the equality. No finite-degree polynomial equals a nonconstant exponential on an interval, so a residual shape error remains. □

Proposition 2 (R1 and perturbations). With exact returned logits, R1 is exactly $\mathrm{softmax}(z/T^\star)$, where $T^\star$ is fitted on the calibration split. For an ℓ_∞ logit perturbation at most ε, cross-entropy is Lipschitz in logits (its gradient has ℓ_1 norm at most $2/T^\star$), softmax is Lipschitz, and Brier score is Lipschitz on the simplex; hence their changes are $O(\varepsilon)$. NLL is *not* claimed to be globally Lipschitz in probabilities near zero. For binned ECE, if confidence changes by at most ε_c without changing the predicted label and π_{ε_c} is the fraction of samples within ε_c of a bin edge, a safe bound is $|\Delta\mathrm{ECE}| \le \varepsilon_c + 2\pi_{\varepsilon_c}$. Thus no deterministic $O(\varepsilon_c)$ ECE bound exists without a bin-margin or density assumption.

Proposition 3 (R2c recovery). If P is strictly increasing on $[-\delta, 0]$ and the client receives $\eta_i = P(u_i)$, then P is bijective and $P^{-1}(\eta_i) = u_i$ exactly. If $|\tilde{\eta}_i - P(u_i)| \le \epsilon$, the perturbed value stays in the image of P, and $m = \inf_{[-\delta,0]} P' > 0$, the mean-value theorem gives $|P^{-1}(\tilde{\eta}_i) - u_i| \le \epsilon/m$. Two hypotheses are easy to lose sight of. The client must receive the unfloored value, since where the floor binds it receives ε rather than a point in the image of P; and u_i must be the exact max-shift $z_i - \max_j z_j$, since under a public shift the recovered argument is offset and the statement does not apply. The inverse is public and deterministic; only the following temperature is learned from labels. When monotonicity fails, the paper's rearranged/clipped Chebyshev inverse is explicitly a heuristic rather than this proposition.

Proposition 4 (pre-deployment distortion). Where $P > 0$, define $e(u) = \log P(u) - u$. The pairwise log-odds error of the unfloored numerator is $e(u_i) - e(u_j)$, so

$$D(P, \delta) = \sup_{u,v \in [-\delta,0]} |e(u) - e(v)| = \max e - \min e.$$

This needs no labels once P and the operating bound are fixed. It is undefined where $P \le 0$; applying the same definition to the floored numerator $\max(P, \varepsilon)$ that is actually shipped gives a finite D_ε, which we do not use for any decision. Neither quantity predicts ECE.

Proposition 5 (R2c$^+$ certificate). Let $\mathbf{z}(u) = (1, u, \ldots, u^m)^\top$, $Q \succeq 0$, $\epsilon > 0$, and

$$P(u) = c + \int_{-\delta}^{u} \big(\epsilon + \mathbf{z}(t)^\top Q \mathbf{z}(t)\big) dt, \qquad c > 0.$$

Then $P'(u) \ge \epsilon$, $P > 0$ on $[-\delta, 0]$, and P is strictly increasing, so Proposition 3 applies and the argmax is preserved. The sum-of-squares construction is standard; its use here is a deployment certificate, not a new polynomial primitive. Fit quality changes the uncorrected forward probabilities but not exact noiseless invertibility.

Proposition 6 (conditional rescaling bound). Divide the head by $c > 0$ and fit P_c on $[-\delta/c, 0]$. If the fitting scheme is consistent on shrinking intervals so that $\rho_c = \sup_v |\log P_c(v) - v| \to 0$, then its distortion satisfies $D(P_c, \delta/c) \le 2\rho_c \to 0$. The shipped logits still have scale $1/c$, so this route moves the remaining error into a client-side scalar correction; it does not improve the shipped probabilities for free. The convergence statement requires the stated consistency assumption and is not a theorem about an arbitrary refit.

Proposition 8 (normalized interface). If P is public, positive and strictly increasing and the max-shift is exact, normalized values $q_i = P(u_i)/\sum_j P(u_j)$ already determine the unnormalized numerators. Any maximizer has $u_i = 0$, so $\sum_j P(u_j) = P(0)/\max_i q_i$ and $P(u_i) = q_i P(0)/\max_j q_j$. Applying P^{-1} recovers u. This shows that a public invertible numerator does not hide the shifted logits; it does not remove the exact-max assumption.

Table 4: Calibration changes with numerator depth while top-1 accuracy remains identical to exact softmax on the same logits in every seed. Calibration-only numerator interval, ten seeds.

extra levels	eff. degree	digits acc	digits ECE	digits rel. err	Wisconsin acc	Wisconsin ECE	Wisconsin rel. err	Fashion-MNIST acc	Fashion-MNIST ECE	Fashion-MNIST rel. err
0	3	0.9633	0.1607	2e+8	0.9798	0.0689	3e+10	0.8645	0.1595	3e+15
1	6	0.9633	0.0910	5e+6	0.9798	0.0394	1e+9	0.8645	0.0613	7e+13
2	12	0.9633	0.0529	1e+3	0.9798	0.0268	4e+5	0.8645	0.0167	5e+10
3	24	0.9633	0.0388	9e-1	0.9798	0.0265	3e+0	0.8645	0.0465	5e+3
4	48	0.9633	0.0336	1e-1	0.9798	0.0237	2e-1	0.8645	0.0600	9e-1
5	96	0.9633	0.0322	1e-2	0.9798	0.0253	2e-2	0.8645	0.0640	1e-1
exact softmax	–	0.9633	0.0320	–	0.9798	0.0253	–	0.8645	0.0649	–
+ client temp. (R1)	–	0.9633	0.0212	–	0.9798	0.0263	–	0.8645	0.0123	–

Proposition 9 (argmax, including ties). Suppose $P(0) > \epsilon$ and $P(0) > P(u)$ for all $u \in [-\delta, 0)$. For $\tilde{\eta}_i = \max(P(u_i), \epsilon)$, the maximizer set of $\tilde{\eta}$ equals the maximizer set of z. *Proof.* Every maximizing logit has $u_i = 0$ and hence numerator $P(0)$; every non-maximizer has $u_i < 0$ and numerator strictly below $P(0)$. Common normalization preserves this ordering and preserves all ties among the maximizing logits. □ This condition is weaker than monotonicity and is checked on every deployed numerator.

Proposition 10 (a shift too large collapses the point). Write $u_i = z_i - \mathrm{SHIFT}$, let the client floor the decrypted numerator at ε, and take the numerator to be exp. If $\max_i u_i < \ln \varepsilon$ then $\exp(u_i) < \varepsilon$ for every i, so every coordinate is clamped and the reported distribution is exactly uniform. The point then carries no class information. The statement is for the exact-exponential numerator; for a polynomial that approximates it on the same interval the same collapse is what we measure rather than prove, and Appendix A reports it over degrees 3 to 15 and depths $s = 0$ to 6. The usable condition is $\mathrm{SHIFT} - \max_i z_i < \ln(1/\varepsilon)$, which needs no labels and is computable on the calibration split before deployment. Appendix A measures it, and measures the pipeline it predicts about.

Proposition 11 (flooring is one-sided). Let $\hat{n} \in \mathbb{R}^C$ be any pre-floor numerator vector with $\sum_j \hat{n}_j > 0$, and let $n_i = \max(\hat{n}_i, \varepsilon)$. Then

$$\frac{\max_i n_i}{\sum_j n_j} \leq \frac{\max_i \hat{n}_i}{\sum_j \hat{n}_j},$$

so flooring never increases the reported top probability. Since $\max(\cdot, \varepsilon)$ dominates the identity, $\sum_j n_j \geq \sum_j \hat{n}_j$. If $\max_i \hat{n}_i \geq \varepsilon$ the maximum is unchanged while the sum can only grow, and the claim follows. Otherwise every coordinate is clamped and the left side is $1/C$; $\sum_j \hat{n}_j > 0$ forces $\max_i \hat{n}_i > 0$, so $\sum_j \hat{n}_j \leq C \max_i \hat{n}_i$ and the right side is at least $1/C$. The floor therefore produces under-confidence and never over-confidence, which is why the distortion we measure runs predominantly in that direction; the two-sided part comes from the polynomial's own error and not from the floor.

C Robustness Needed for the Main Claims

Depth–reliability separation. Table 4 gives the complete MLP sweep. Paired bootstrap intervals for the accuracy difference are exactly $[0, 0]$ at every depth, and equal-mass ECE and tolerance sweeps preserve the ordering. Fashion-MNIST's non-monotone ECE is the strongest warning against the slogan "more depth is safer".

Controlled range intervention. Rescaling real logits to prescribed ranges removes the architecture/dataset confound. At fixed depth calibration damage grows with δ and then saturates, while accuracy is identical in every cell-seed. Required depth has no clean law in δ, so this is evidence rather than a fitted formula.

Table 5: **What each stopping rule buys, and what it costs.** Levels s selected / ECE then shipped; 10 seeds, degree-3 numerator, exact-softmax references 0.065, 0.032, 0.025; "vs. calibration" is levels early ($-$) or late ($+$); last column, the worst benchmark's depth priced at $\Delta = 2^{40}$ (Appendix D). A relative-error tolerance of 2^{-10} is unreached on all 3 at every tested depth.

stopping rule	Fashion-MNIST	digits	Wisconsin	vs. calibration	worst N / KiB
top-1 accuracy plateau	0 / 0.159	0 / 0.161	0 / 0.069	$-1, -4, -2$	16384 / 960
relative error < 1	4 / 0.060	3 / 0.039	4 / 0.024	$+3, -1, +2$	16384 / 1600
output KL $< 10^{-2}$	3 / 0.046	2 / 0.053	2 / 0.027	$+2, -2, \ 0$	16384 / 1440
output KL $< 10^{-4}$	5 / 0.064	4 / 0.034	4 / 0.024	$+4, \ 0, +2$	32768 / 3520
ECE within 0.005 of exact	1 / 0.061	4 / 0.034	2 / 0.027	**tight**	16384 / 1600

Folding the temperature at the shipped depth. Folding $T^\star$ into the head before encryption is free on ciphertext but fixes nothing at degree 3 with no extra levels: the shipped ECE moves from 0.159 to 0.185 on Fashion-MNIST and from 0.161 to 0.151 on digits, and stays at 0.069 on Wisconsin, while the exact model's falls to R1's level, so the inflation ratio rises (Fashion-MNIST 2.5× to 16.0×). With the depth rule it becomes R3 (Table 9).

Operational checks. The polynomial map is under-confident at the top in every seed of all ten settings (mean top-label confidence below accuracy), which costs selective coverage (at τ=0.9 it auto-decides under 5% of cases on six of the seven backbone settings, where R1 spans 22–88%); the transplanted plaintext risk-control threshold adds no violations in the tested seeds, cost-sensitive decisions worsen in all nine benchmark-by-ratio cells, and top-label isotonic regression has higher NLL and Brier than R1 in all ten settings. Against R1, AURC changes by -4% (digits), 0% (Wisconsin) and $+6\%$ (Fashion-MNIST) on the MLPs, by $+7\%$ to $+56\%$ on the CNNs and CIFAR-100, and by 2.6× on CIFAR-10.

Range-control limitation. The hidden-ReLU simulations clip out-of-range pre-activations, which a ciphertext cannot; removing the clip damages accuracy on digits and Fashion-MNIST, so the whole-network MLP rows stand for a deployment with activation range control. The output-softmax isolation claim does not depend on this clip.

D CKKS Level Model

On the output path the linear head costs one level, TenSEAL's degree-3 numerator has multiplicative depth two, and repeated squaring adds s, hence $L = 3 + s$. At scale 2^{40}, $s = 0, \ldots, 4$ fit at $N = 16384$ while ciphertext size increases from 960 to 1600 KiB; a fifth level exceeds the stated modulus budget and moves to $N = 32768$. At scales 2^{30} and 2^{50} the calibration-tight choice crosses a ring-dimension boundary in the reported settings. These are security-table-based parameter counts, not measured end-to-end CKKS timings. Full-stack max/inverse depths are therefore kept as a conservative arithmetic model and are not used to claim an optimized TenSEAL latency.

Measured, not only counted. The figures above are parameter-table arithmetic, so we also timed the encrypted numerator itself under TenSEAL at scale 2^{40} on a packed 10-class logit vector, allocating $L = 3 + s$ levels. It takes 0.063 s per point at $s = 0$, then 0.090, 0.129, 0.171 and 0.207 s as s rises to 4, all at $N = 16384$, and 0.550 s at $s = 5$, where the chain no longer fits and N doubles. Each extra level therefore costs 1.2 to 1.4× the previous one's time while the ring dimension holds and 2.7× at the level that doubles it; the one label-free rule of Table 5 that is never early costs 8.7× the $s = 0$ time. The timed interval runs from encrypting the packed logits to decrypting the numerator, so it includes encryption and decryption at each depth's parameters but not the linear head. The logits are synthetic, so these runs time the squared schedules without scoring their calibration; a repeat run agrees within 7%.

Table 6: Isolation: approximating the *softmax* alone preserves accuracy while degrading calibration by up to 25.4× here; approximating *ReLU* is comparatively benign. ECE (mean±std, 10 seeds), calibration-only protocol: the numerator is fitted on 1.2 times the calibration split's logit range and evaluated logits are clipped into it. Accuracy is constant across softmax degrees.

Dataset	baseline ECE (acc)	softmax→poly ECE by degree k=2	k=3	k=5	k=7
Fashion-MNIST	$0.070 \pm .005$ (.869)	$0.679 \pm .055$	$0.538 \pm .133$	$0.248 \pm .159$	$0.121 \pm .070$
digits	$0.025 \pm .003$ (.967)	$0.636 \pm .163$	$0.415 \pm .231$	$0.185 \pm .145$	$0.103 \pm .057$
ReLU→poly ECE (benign)		Fashion 0.065–0.068; digits 0.026–0.044 (acc drops $\leq$ 2.1pts, at k=2)			

Table 7: Additional backbones and datasets, ECE (mean over seeds). CNN and ResNet-18 backbones, 10 seeds except ‡9; training details in Appendix B. Head-only: the backbone runs in plaintext. "ReLU only" approximates the final hidden activation alone: ECE stays within 0.019 of the plaintext baseline, and it is where the accuracy the pipeline loses comes from. B0 adds the polynomial softmax on those same logits: top-1 accuracy is *exactly* unchanged in every seed of every row while ECE inflates by 1.1× to 30× (ratios of unrounded values). δ is the operating range; B2 is the recalibrated plaintext reference. Every column follows the calibration-only protocol of Appendix B; the clip into the public interval binds on 0.00% of test entries in all seven settings. The in-pipeline temperature-scaling and isolation columns are in Table 3.

Setting	δ	acc. pl. / FHE	ReLU base	ReLU only	B0	R2c	**R2c⁺/R1**	B2
CNN Fashion	66	0.83/0.79	0.020	0.037	0.511	0.019	**0.019**	0.012
CNN MNIST	54	0.96/0.96	0.007	0.016	0.471	0.005	**0.006**	0.005
CNN Path	94	0.78/0.71	0.020	0.039	0.503	0.025	**0.025**	0.011
CNN Blood	58	0.68/0.65	0.045	0.053	0.373	0.050	**0.051**	0.034
CNN Derma	25	0.67/0.67	0.055	0.061	0.257	0.055	**0.052**	0.048
R18 C-10	22	0.94/0.94	0.033	0.034	0.037	0.019	**0.012**	0.010
R18 C-100‡	60	0.74/0.74	0.128	0.120	0.426	0.023	**0.023**	0.025

E A Depth-Selection Rule

The numerator is built by domain-scaled repeated squaring, $P_s(u) = \left(P_3(u/2^s)\right)^{2^s}$ with P_3 the degree-3 Chebyshev fit of $\exp$ on $[-\delta/2^s, 0]$, so s counts extra multiplicative levels and the effective degree is $3 \cdot 2^s$. For each setting and each s we compute $D(P_s, \delta)$, which needs only the public polynomial and the public interval, and the gap between the polynomial softmax's ECE and the exact softmax's ECE on the same logits. The interval is the calibration-only one of §3; seeds as in Table 7.

Two depths matter. s_0 is the shallowest depth at which P_s is positive on the whole interval, below which D is undefined and only the floor keeps the numerator positive; $s^\star$ is the shallowest depth with $D \leq \tau$, $\tau = 0.3$. Table 8 gives both. Top-1 accuracy equals the exact softmax's in every seed at *every* depth in the sweep, s_0 included, so accuracy cannot distinguish the two columns; the mean calibration gap falls from as much as 0.020 to at most 0.0022.

How sensitive the threshold is. No single threshold separates misses from hits perfectly: over all settings, seeds and depths the smallest D that misses a 0.005 ECE target is 0.13 and the largest that meets it is 32.8. At $\tau = 0.3$ the selected depth meets the target on 67 of 69 seeds; the two misses are DermaMNIST, the narrowest range, at 0.0053 and 0.0056. A stricter $\tau = 0.1$ meets it on all 69 at about a quarter of a level more on average, and a looser $\tau = 1$ on 59. The threshold was set once, for all settings, with these benchmarks in view, so these counts are in-sample. A leave-one-setting-out check removes that: choosing τ on nine of the ten settings (the seven backbones and the three MLP pipelines) as the largest value that meets the target on all of their seeds gives $\tau = 0.12$ in nine folds and 0.18 with DermaMNIST held out, at most one level deeper than $\tau = 0.3$, and applied unchanged to the held-out setting it meets the target on 98 of 99 seeds, the one miss a DermaMNIST seed. We report the rule as a calibrated heuristic, not a guarantee.

Table 8: Choosing depth from a label-free quantity, on each backbone's own pipeline logits (degree-4 polynomial ReLU on the calibration split's range) and the calibration-only interval. s_0 is the shallowest depth at which the numerator stays positive on $[-\delta, 0]$ without the floor; $s^\star$ is the shallowest depth with $D \le 0.3$; both are ranges over ten seeds (9 on CIFAR-100). D is the mean over seeds; "gap" is $|\mathrm{ECE}_{\mathrm{poly}} - \mathrm{ECE}_{\mathrm{exact}}|$ on the same logits, mean / max over seeds. Top-1 accuracy equals the exact softmax's in every seed at every depth.

Setting	δ	K	shallowest positive s_0	shallowest positive D	shallowest positive gap	rule, $D \le 0.3$ $s^\star$	rule, $D \le 0.3$ D	rule, $D \le 0.3$ gap	extra levels
CNN Fashion	65.9	10	4–5	13.3	0.0129 / 0.0282	6	0.102	0.0011 / 0.0016	+1–2
CNN MNIST	53.6	10	4	11.9	0.0089 / 0.0146	5–6	0.090	0.0006 / 0.0016	+1–2
CNN Path	94.3	9	5	13.7	0.0128 / 0.0273	6–7	0.125	0.0012 / 0.0028	+1–2
CNN Blood	58.5	8	4–5	9.93	0.0101 / 0.0229	5–6	0.085	0.0012 / 0.0030	+1–2
CNN Derma	25.4	7	3–4	3.14	0.0079 / 0.0208	4–5	0.100	0.0022 / 0.0056	+1–2
R18 C-10	21.9	10	3	1.66	0.0013 / 0.0020	4	0.095	0.0003 / 0.0007	+1
R18 C-100	59.9	100	4	30.4	0.0201 / 0.0226	6	0.062	0.0008 / 0.0009	+2

When the distribution must stay encrypted (R3). Table 9 applies the rule after folding the calibration temperature into the head, on the MLP pipelines of Table 4 and the seven backbone settings (the calibration-only interval is refitted on the folded calibration logits). The target is R1's client-side output on the same logits. Without folding, the depth the rule selects brings the polynomial to the exact softmax, whose own miscalibration then remains; with folding it brings it to the target. Top-1 accuracy equals the exact softmax's at the selected depth in both arms.

Table 9: R3: fold $T^\star$ into the head, then choose depth by $D \le 0.3$ on the folded interval; ten seeds (9 on CIFAR-100). $T^\star$ is the mean over seeds, s the selected number of extra levels (range over seeds), and "gap" is $|\mathrm{ECE} - \mathrm{ECE}_{\mathrm{R1}}|$ at that depth, mean / max over seeds. Top-1 accuracy equals the exact softmax's at the selected depth in every row.

Setting	$T^\star$	$\delta \to \delta_{\mathrm{fold}}$	without folding s	without folding gap	R3 (folded) s	R3 (folded) gap
MLP Fashion-MNIST	1.75	43.4 → 24.7	5	0.052 / 0.063	4	0.0009 / 0.0014
MLP digits	0.80	25.0 → 31.9	4	0.012 / 0.020	4–5	0.0011 / 0.0038
MLP Wisconsin	1.16	26.1 → 25.7	4–5	0.008 / 0.024	3–5	0.0010 / 0.0079
CNN Fashion	0.88	65.9 → 75.5	6	0.018 / 0.051	6–7	0.0006 / 0.0011
CNN MNIST	0.87	53.6 → 62.0	5–6	0.011 / 0.018	6	0.0004 / 0.0008
CNN Path	0.97	94.3 → 97.5	6–7	0.014 / 0.025	6–7	0.0006 / 0.0018
CNN Blood	0.93	58.5 → 63.1	5–6	0.005 / 0.029	5–6	0.0010 / 0.0026
CNN Derma	0.99	25.4 → 26.0	4–5	0.012 / 0.041	4–5	0.0020 / 0.0063
R18 CIFAR-10	1.56	21.9 → 14.1	4	0.022 / 0.025	3–4	0.0005 / 0.0010
R18 CIFAR-100	1.59	59.9 → 37.6	6	0.097 / 0.103	5	0.0012 / 0.0033

F The Numerator Family Is a Design Choice

Holding the shipped pipeline fixed and changing only the family used to approximate $\exp$ on $[-\delta, 0]$, at the same degree 3 and the same depth, two things hold across that sweep. Top-1 accuracy is identical to the exact softmax in every family on every setting, all 28 cells, so the accuracy-invisibility of §3 is a property of the interface and not of a particular fit. And the damage is not ordered by approximation quality: minimax minimizes worst-case error over the interval by construction and is better calibrated than least squares on six of seven settings, yet still ships 1.1 to 12× the exact softmax's ECE, and on CIFAR-10 it is four times worse than least squares. On CNN MNIST four numerators of the same degree and the same depth, all with identical top-1 accuracy, give ECE from 0.015 to 0.607.

The truncated Maclaurin column is best calibrated on five of seven settings and on MNIST indistinguishable from the exact softmax, but we report it without recommending it: one domain-scaled squaring collapses its top-1 on Fashion-MNIST from 0.794 to 0.000, and it leans hardest on the client-side floor, clipping 68 to 99% of its entries. What it does show is the design rule this paper

argues for: choose a numerator for its log-odds distortion near the top of the operating range, which $D(P, \delta)$ measures, and not for its uniform approximation error.

A second construction. The separation is not an artifact of the repeated-squaring schedule. Replacing it with a direct Chebyshev fit of $\exp$ on the same calibration-only interval, with no domain scaling and no squaring, so that the multiplicative depth is $\lceil \log_2 d \rceil$ at degree d, top-1 accuracy is bit-identical to the exact softmax at *one* multiplicative level in all 65 seeds, while a 0.005 calibration tolerance needs one to five: a separation of up to four levels and three on average, wider than the schedule the main text studies. At that one level the shipped ECE runs 0.7 to 56.5 times the exact softmax's.

G Novelty Boundary

Calibration has been studied under encryption before: a patent family computes ECE inside FHE for a Maclaurin numerator (Shaik et al., 2024), and Pathan et al. (2026) adapt a deployed model's final layer from encrypted features under shift. Accuracy-preserving confidence distortion also appears outside HE under reduced precision (Gu et al., 2026), where quantization can also improve calibration (Bouguerra et al., 2025), and concurrent quantized-LLM work shows task accuracy masking full-distribution drift without establishing calibration (Qamar et al., 2026). We claim neither observation first. Monotone, accuracy-preserving transformations of logits are used for plaintext post-hoc calibration as well: Zhao et al. (2026) *learn* an element-wise map of the logits whose monotonicity, and so the argmax, is encouraged by a reconstruction regularizer. Our monotone numerator serves a different purpose and is monotone by certificate rather than by training. It is not learned and does not recalibrate; it makes the client-side inverse of an already-public deployed polynomial well defined, which is what lets the client recover shifted logits it never received. Existing encrypted-softmax work optimizes approximation error, agreement, KL, perplexity or cost (Cho et al., 2024; Zirilli et al., 2026; Park et al., 2026). What is new here is the fixed public polynomial setting in which accuracy can saturate before calibration across depth, the distortion is computable from the polynomial and its operating range, and the client interface admits exact recovery under stated max-shift and monotonicity assumptions. R1 is a design rule, not a new primitive; R2c inverts the polynomial in public rather than homomorphically; R2c$^+$ uses a standard sum-of-squares construction as a deployment certificate.

H Transformer Replication and the Attention Boundary

A two-block, four-head, $d = 64$ transformer repeats the output-depth separation (released outputs). With attention exact, approximating the output softmax leaves top-1 accuracy *identical* to the exact softmax in all 120 seed-depth cells (ten seeds on digits and on Fashion-MNIST, six depths), while calibration first lands within 0.005 of the exact softmax at $s = 1$ on digits and $s = 3$ on Fashion-MNIST, undershooting on the way as in the MLP study. Inside attention the conclusion changes: accuracy falls to 0.891 on digits and 0.236 on Fashion-MNIST at zero extra levels and needs at least as much depth as calibration in 17 of the 20 runs, which bounds the headline to an output softmax.