FITTING MULTILINEAR POLYNOMIALS FOR LOGIC GATE NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

We study learnable logic gate networks that stack layers of 2-input Boolean gates to build combinational circuits. Every 2-input gate has a unique multilinear polynomial with 4 coefficients, so the 16 Boolean gates form a codebook of prototypes in a 4-dimensional space, reducing training to a vector-quantization problem. The baseline method, Soft-Mix, learns a 16-dimensional softmax over gate identities, but the codebook has rank 4: 11 of 15 simplex directions carry nullspace gradient, and at uniform initialization the backward signal vanishes exactly. We prove that no affine product reparameterization fixes the resulting interaction-coefficient starvation under STE, and show that the covariance Jacobian of soft-VQ selection bypasses it by coupling the starved coefficient to the always-active constant channel. Working in the 4-dimensional polynomial space reduces each neuron from 16 to 4 parameters. On seven datasets, with every accuracy measured on the deployed strict-Boolean (zero-multiply-accumulate) circuit, our primary method, Multilinear-CovJac, matches or exceeds 16-parameter Soft-Mix on every dataset, and its advantage over STE grows monotonically with interaction demand. At depth, Soft-Mix collapses (-37.3pp on CIFAR-10 at 12 layers) while CovJac holds (-0.5pp on CIFAR-10, stable on MNIST).

1 INTRODUCTION

Learnable logic gate networks build combinational circuits by assigning a Boolean gate to every neuron (Petersen et al., 2022; 2024). These networks sit at the extreme efficiency end of the neural network spectrum: the deployed model is a combinational circuit requiring no floating-point arithmetic and no weight storage in the hidden layers (Andronic & Constantinides, 2023; Bacellar et al., 2024; Umuroglu et al., 2020; Wang et al., 2019), with inherent interpretability since each neuron computes a named logic function (AND, XOR, etc.) (Jukna, 2012). The challenge is training: selecting the right gate per neuron is a discrete optimization problem with structural gradient pathologies. For 2-input gates (the setting studied here), training must choose one of 16 possible gates per neuron, a discrete selection problem typically relaxed via continuous surrogates (§2).

Observation: the 16 gates live in a 4-dimensional space. Every 2-input Boolean function has a unique multilinear polynomial with 4 coefficients (§2.2) (Hammer & Rudeanu, 1968; Owen, 1972), so the 16 gates are 16 integer points in a 4-dimensional space. The softmax operates in 15 dimensions, but only 4 of those directions actually change the neuron's output; the remaining 11 carry gradient that does no work (§3). This suggests training directly in the 4-dimensional coefficient space instead, learning $\mathbf{c} \in \mathbb{R}^4$ and snapping to the nearest valid gate.

Two gradient pathologies in this space. Working in 4 dimensions eliminates the redundancy but exposes two problems. **(i)** The soft mixture used by the existing method suffers from exact gradient cancellation: at the standard uniform initialization, the backward signal through each neuron is provably zero for every input (Proposition 2). This worsens with depth. **(ii)** Under the straight-through estimator (STE), the interaction coefficient c_{ab}, which distinguishes AND from OR and XOR from XNOR, receives gradient on only 25% of training samples. We prove that no affine reparameterization of the polynomial basis can fix this (Proposition 7).

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.

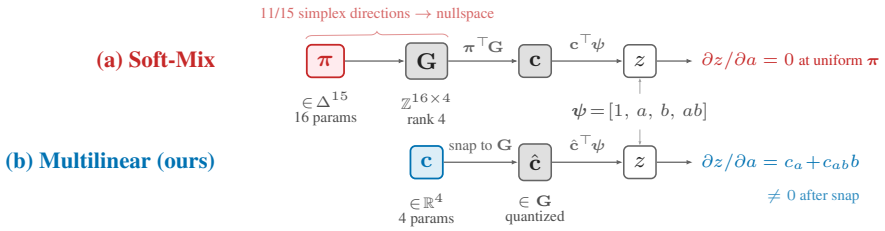


Figure 1: Both methods differ in how the coefficient vector $\mathbf{c}$ is obtained. Colored blocks are **learnable parameters** (red: Soft-Mix, blue: ours); gray blocks are fixed or derived. **(a)** Soft-Mix learns $\boldsymbol{\pi} \in \Delta^{15}$ (16 params) and projects through the rank-4 codebook $\mathbf{G}$: 11 of 15 simplex directions carry nullspace gradient, and at uniform $\boldsymbol{\pi}$ the input derivative vanishes exactly (Prop. 2). **(b)** Our method learns $\mathbf{c} \in \mathbb{R}^4$ (4 params) and quantizes to the nearest codebook entry $\hat{\mathbf{c}} \in \mathbf{G}$; the output is $z = \hat{\mathbf{c}}^\top \boldsymbol{\psi}$, and after snapping $|c_a + c_{ab} b| \geq 1$ always (Prop. 4).

Solution: soft vector quantization with a covariance-Jacobian gradient. Our primary method, Multilinear-CovJac, trains the four coefficients through a *soft* vector-quantization forward pass, $z = \sum_j \omega_j \, g_j(a, b)$ with $\boldsymbol{\omega} = \mathrm{softmax}(-\|\mathbf{c} - \mathbf{G}_j\|^2/\tau)$; its exact gradient is a codebook covariance Jacobian that couples c_{ab} to the always-active constant channel, routing gradient to it on every sample (Proposition 5). The soft-VQ forward and the covariance Jacobian are not two contributions: the Jacobian *is* the closed form of the soft-VQ backward, so Multilinear-CovJac is a single method. We use Multilinear-STE (a hard-snap forward with a straight-through gradient on the same four coefficients) only as an ablation anchor that isolates what the covariance Jacobian adds; it is not a competing proposal, and rows where this anchor trails Soft-Mix measure the cost of the straight-through gradient, not of soft vector quantization (Figure 1).

Contributions. Our contribution is not the multilinear representation of the 16 gates (which is textbook (Hammer & Rudeanu, 1968; O'Donnell, 2014)) but how to *train* in it:

- **An exact cancellation law.** The 16-gate softmax is rank-4, and its input gradient vanishes for *every* input at symmetric weights (Proposition 2), sharpening the approximate-at-initialization observation of Rüttgers et al. (2026) to an exact identity; moreover no affine basis removes the resulting c_{ab} starvation under STE (Proposition 7).
- **A gradient remedy (Multilinear-CovJac).** Training the four coefficients directly by soft vector quantization gives a closed-form covariance Jacobian that routes gradient to c_{ab} on every sample (Proposition 5), at $4\times$ fewer parameters than Soft-Mix.
- **A matched evaluation.** Under one harness with strict zero-multiply-accumulate deployment, our primary method Multilinear-CovJac matches or exceeds 16-parameter Soft-Mix on all seven datasets, the CovJac-over-STE gain grows with interaction demand, and at L=12 Soft-Mix collapses (-37.3pp on CIFAR-10) while CovJac holds (-0.5pp).

2 Background

2.1 Training logic gate networks

Selecting one of 16 known operations per neuron is a discrete optimization problem analogous to operator selection in neural architecture search (Liu et al., 2019), but over a small, fully enumerable set. The standard approach is continuous relaxation: replace the discrete gate choice with a differentiable surrogate so the network can be trained end-to-end with backpropagation, then discretize at deployment. Differentiable logic gate networks (DLGNs) (Petersen et al., 2022), the architecture we study, implement this with L hidden layers of width k. Each neuron in layer l receives two inputs from layer $l-1$ via fixed random wiring; the first layer draws from the input features. The final layer's k outputs are partitioned into C equal groups (one per class), summed within each group, and passed through softmax to produce class probabilities (GroupSum readout).

Soft mixture over all 16 gates. In the approach of Petersen et al. (2022), each neuron learns a distribution $\boldsymbol{\pi} \in \Delta^{15}$ over the 16 gates; the training output is the weighted mixture $z =$

$\sum_{j=0}^{15} \pi_j \, g_j(a,b)$, and the deployed gate is $g^* = \arg\max_g \pi_g$. Because training uses a soft mixture of all 16 gates while deployment uses a single selected gate, there is a potential train-to-deploy mismatch.

Single-gate selection via Gumbel-Softmax. Keeps 16 logits per neuron but samples a one-hot gate via Gumbel-Softmax (Jang et al., 2017; Maddison et al., 2017) during training (hard argmax at deployment), with straight-through gradients (Bengio et al., 2013). Only one gate is active per forward pass, avoiding the backward cancellation (§4.1). Yousefi et al. (2025) analyze the discretization gap in DLGNs and propose methods to reduce it.

Input-wise Parametrization (IWP). Rüttgers et al. (2026) reduce to 4 parameters per neuron in the corner basis $\phi_{ij}(a,b) = a^i(1-a)^{1-i}b^j(1-b)^{1-j}$ and identify that complementation symmetry ($g \mapsto 1-g$) causes pairwise gradient cancellation at symmetric initialization, stalling optimization at depth. This cancellation observation is theirs; we build on it and differ in two ways: we prove the cancellation is *exact for every input* at symmetric weights, not only approximate at initialization (Proposition 2), and we address it with a gradient remedy, the covariance Jacobian of a soft-VQ step (Proposition 5), rather than an initialization scheme. The multilinear representation of the gates (§2.2) is itself textbook (Hammer & Rudeanu, 1968; O'Donnell, 2014); our contribution is the training analysis and the soft-VQ method, not the representation.

Related but distinct: LUT-from-transform methods. A separate line of work targets FPGA inference by tabulating a learned per-neuron function into a lookup table. LogicNets (Umuroglu et al., 2020) sparsely wires low-bit neurons and enumerates each neuron's input combinations into a LUT; the PolyLUT family (Andronic & Constantinides, 2023; Lou et al., 2024; Andronic et al., 2025) raises each unit's expressiveness by evaluating a multivariate polynomial of the neuron's inputs, hidden inside the same LUT. In both, the polynomial (or affine map) acts on a neuron's *inputs* and its Boolean behavior is obtained by exhaustive tabulation. This is the opposite direction from the gate parametrizations above, which fit a polynomial to a gate's own truth table and deploy it as one of the 16 canonical two-input operations. The two constructions are complementary, not substitutable: they do not select among the 16 gates and we do not tabulate a feature transform, so neither is a drop-in baseline for the other.

2.2 The multilinear extension

Every 2-input Boolean function has a unique multilinear polynomial (Hammer & Rudeanu, 1968; Owen, 1972; O'Donnell, 2014; Jukna, 2012). In the canonical monomial basis this is

$$g(a,b) = c_0 + c_a\, a + c_b\, b + c_{ab}\, ab, \qquad \mathbf{c} = [c_0, c_a, c_b, c_{ab}]^\top \in \mathbb{R}^4, \tag{1}$$

where c_0 is the constant output, c_a and c_b are linear effects, and c_{ab} is the interaction term, zero for separable gates, nonzero for interactive gates. Since the inputs are Boolean, all 16 gates have integer coefficients, e.g. AND=$[0,0,0,1]$, OR=$[0,1,1,-1]$, XOR=$[0,1,1,-2]$, A=$[0,1,0,0]$, FALSE=$[0,0,0,0]$ (full table in Appendix A.1).

The same polynomial can be written in the corner basis $\phi_{ij}(a,b)$=$a^i(1-a)^{1-i}b^j(1-b)^{1-j}$, where each basis function isolates one input combination; this is the parameterization used by IWP (Rüttgers et al., 2026). The two bases are related by an invertible linear map (Appendix C) and span the same function space; the critical difference is the gradient structure under discrete training constraints, analyzed in §4. Since the canonical basis has the sparsest monomials under STE (§4), we train directly in this 4-dimensional coefficient space. Connections to fuzzy logic, probabilistic circuits, and degree as logical complexity are in Appendix B.

3 Method

We collect the 16 integer coefficient vectors into a codebook $\mathbf{G} \in \mathbb{Z}^{16\times 4}$ of rank 4 ($\mathbf{G}_j$: row j, a 4-vector; G_{jk}: scalar entry), and observe that the Soft-Mix output can be written as $z = \mathbf{c}_{\text{eff}}^\top \boldsymbol{\psi}(a,b)$ with $\mathbf{c}_{\text{eff}} = \boldsymbol{\pi}^\top \mathbf{G}$ and $\boldsymbol{\psi} = [1, a, b, ab]^\top$. Since $\mathbf{G}$ has rank 4, the map $\boldsymbol{\pi} \mapsto \boldsymbol{\pi}^\top \mathbf{G}$ projects the 15-dimensional simplex onto a 4-dimensional space: **only 4 of 15 simplex directions change the neuron's output; the remaining 11 carry gradient that does no work.**

We therefore train directly in the 4-dimensional coefficient space. Each neuron stores $\mathbf{c} = [c_0, c_a, c_b, c_{ab}]^\top \in \mathbb{R}^4$ (4 learnable parameters). Because the deployed circuit requires a valid Boolean gate, $\mathbf{c}$ must be quantized to the nearest codebook entry:

$$\hat{\mathbf{c}} = \mathbf{G}_{j^*}, \qquad j^* = \operatorname{argmin}_j \|\mathbf{c} - \mathbf{G}_j\|_2^2. \tag{2}$$

By Eq. 1, $z = \hat{\mathbf{c}}^\top \boldsymbol{\psi}(a, b)$ agrees with the Boolean truth table on $\{0, 1\}^2$. The two methods below share this quantization and differ only in *when* it is applied and *how* gradients flow through it.

3.1 Multilinear–Straight-Through Estimator (STE)

Forward. Each neuron snaps $\mathbf{c}$ to the nearest gate $\hat{\mathbf{c}}$ via Eq. 2 and evaluates $z = \hat{\mathbf{c}}^\top \boldsymbol{\psi}(a, b) = \hat{c}_0 + \hat{c}_a a + \hat{c}_b b + \hat{c}_{ab}\, ab$. The same quantized gate is used in both training and deployment, so there is **no train-to-deploy mismatch** (Appendix A.1).

Backward. The quantization $\mathbf{c} \mapsto \hat{\mathbf{c}}$ (Eq. 2) maps every $\mathbf{c}$ in a region of $\mathbb{R}^4$ to the same codebook vector, so small perturbations of $\mathbf{c}$ leave $\hat{\mathbf{c}}$ unchanged and $\partial\hat{\mathbf{c}}/\partial\mathbf{c} = \mathbf{0}$ almost everywhere. The straight-through estimator (STE; Bengio et al. (2013)) bypasses this by approximating $\partial\hat{\mathbf{c}}/\partial\mathbf{c} \approx \mathbf{I}$ in the backward pass. Let $\mathcal{L}$ denote the network loss and $\delta = \partial\mathcal{L}/\partial z$ the upstream error signal backpropagated to this neuron's output $z = \hat{\mathbf{c}}^\top\boldsymbol{\psi}$. Then by the chain rule:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{c}} \approx \frac{\partial\mathcal{L}}{\partial\hat{\mathbf{c}}} = \frac{\partial\mathcal{L}}{\partial z} \cdot \frac{\partial z}{\partial\hat{\mathbf{c}}} = \delta \cdot \boldsymbol{\psi}(a, b) = \delta \cdot [1,\, a,\, b,\, ab]^\top, \tag{3}$$

The gradient for each coefficient c_k is scaled by the corresponding basis monomial $[1, a, b, ab]_k$, which is nonzero only when all its input arguments equal 1. This creates a **hierarchical** coverage pattern: under i.i.d. Bernoulli$(1/2)$ inputs (idealized; confirmed empirically in §5), c_0 receives gradient 100% of the time (its monomial is 1), c_a and c_b at 50% (active when a=1 or b=1), and c_{ab} at only 25% (active only when a=b=1). The expected active-component count is $1+\frac{1}{2}+\frac{1}{2}+\frac{1}{4} = 2.25$, versus $4\times\frac{1}{4} = 1.0$ for the corner basis, a $2.25\times$ ratio (Lemma 6; full table in Appendix A.2). The empirical consequence (a 17pp accuracy gap) is in §5.

However, this hierarchy has a cost. **The interaction coefficient c_{ab}, the interaction term (second-order Sobol index (Sobol', 1993; Owen, 2013)) that distinguishes AND from OR and XOR from XNOR, receives gradient on only 1 in 4 samples.** This is not an implementation artifact: it is intrinsic to the multilinear basis under STE, and no affine reparameterization of the basis can fix it without introducing bias or losing coherence (Proposition 7, §4.3). The c_{ab} starvation motivates the second variant below.

3.2 Multilinear–Covariance Jacobian (CovJac)

Under STE, the interaction coefficient c_{ab} learns from only 25% of training samples while c_0 learns from all of them. We seek a gradient mechanism that couples c_{ab} to the always-active dimensions, routing gradient to it on every sample.

Idea: soft vector quantization. Instead of hard-quantizing $\mathbf{c}$ to the nearest codebook entry (as in STE), we relax the quantization during training by computing a proximity-weighted average over all codebook entries (van den Oord et al., 2017; Agustsson et al., 2017):

$$\omega_j = \operatorname{softmax}\big(-\|\mathbf{c} - \mathbf{G}_j\|^2/\tau\big)_j, \qquad \mathbf{c}_{\text{soft}} = \sum_j \omega_j\, \mathbf{G}_j. \tag{4}$$

Here ω_j are proximity weights (not the Soft-Mix gate distribution $\boldsymbol{\pi}$): codebook entries closer to $\mathbf{c}$ receive higher weight. During training the neuron evaluates $z = \mathbf{c}_{\text{soft}}^\top\boldsymbol{\psi}(a, b)$; at deployment it uses the hard-quantized $\hat{\mathbf{c}}$ as before. We fix τ=1 throughout. The codebook is the fixed set of 16 Boolean gates, not learned.

Why this helps: the covariance Jacobian. The key property of soft-VQ is that gradients flow through the Jacobian $\partial\mathbf{c}_{\text{soft}}/\partial\mathbf{c}$. Since $\omega_j \propto \exp(-\|\mathbf{c} - \mathbf{G}_j\|^2/\tau)$, the derivative of each weight is $\partial\omega_j/\partial\mathbf{c} = (2/\tau)\,\omega_j(\mathbf{G}_j - \mathbf{c}_{\text{soft}})$; substituting into $\partial\mathbf{c}_{\text{soft}}/\partial\mathbf{c} = \sum_j(\partial\omega_j/\partial\mathbf{c})\,\mathbf{G}_j^\top$ gives

$$\mathbf{J} = \frac{\partial\mathbf{c}_{\text{soft}}}{\partial\mathbf{c}} = \frac{2}{\tau}\,\operatorname{Cov}_{\boldsymbol{\omega}}(\mathbf{G}), \tag{5}$$

the ω-weighted covariance of the codebook, $\mathrm{Cov}_\omega(\mathbf{G})_{ik} = \sum_j \omega_j (G_{ji} - (\mathbf{c}_{\text{soft}})_i)(G_{jk} - (\mathbf{c}_{\text{soft}})_k)$ (full proof in Appendix A.5; see also Huh et al. (2023) for STE failure modes motivating soft alternatives). This matrix is symmetric PSD, strictly positive on the diagonal, and, crucially, **independent of the input** (a, b) (Proposition 5, §4.3). Because $\mathbf{J}$ is a covariance matrix over the codebook, its off-diagonal entries couple different coefficient dimensions. Writing J_{ik} for the (i, k) entry of $\mathbf{J}$ (0-indexed: i=0 for c_0, i=3 for c_{ab}) and $\delta = \partial\mathcal{L}/\partial z$ for the upstream scalar gradient, the gradient reaching c_{ab} is:

$$\frac{\partial \mathcal{L}}{\partial c_{ab}} = \delta \cdot \big(\underbrace{1 \cdot J_{03}}_{\text{always active}} + a \cdot J_{13} + b \cdot J_{23} + ab \cdot J_{33} \big). \tag{6}$$

The leading term J_{03} multiplies the constant monomial 1, which is active on every sample. On the 16-gate codebook at the symmetric point (ω_j=1/16, τ=1) this coupling is exactly $J_{03} = (2/\tau)\,\mathrm{Cov}_\omega(G_{\cdot 0}, G_{\cdot 3}) = 0.5$, and the full c_{ab}-gradient prefactor $J_{03} + aJ_{13} + bJ_{23} + ab\,J_{33}$ (Eq. 6) has magnitude 0.5 on all four input corners; it is nonzero generically elsewhere. Thus c_{ab} **receives a non-zero gradient on every training sample**, not just the 25% where a=b=1. This is the exact analog, in the CovJac backward, of the symmetric point where the Soft-Mix gradient vanishes (§4, Proposition 2).

Comparison of the two mechanisms. Both methods reduce each neuron from 16 to 4 parameters. The two gradient mechanisms are complementary: STE is *support-weighted* (gradient fires when the relevant input is active) and quantizes in both phases (no train-deploy mismatch); CovJac is *codebook-coupling-weighted* (gradient fires via J_{03} regardless of input) and uses soft-VQ during training, quantizing only at deployment. The price of CovJac is a small train-deploy gap ($\leq 0.2\%$ on all main datasets; $\leq 1\%$ on MONK's-2 with k=136). The $4\times$ parameter reduction yields a $4\times$ reduction in Adam optimizer state. A full method comparison is in Appendix C.

4 Theoretical Analysis

Section 3 claimed that Soft-Mix gradient cancels at uniform $\boldsymbol{\pi}$, that Multilinear-STE avoids this cancellation, and that CovJac couples c_{ab} to an always-active channel. We now prove these claims. We also address whether a different polynomial basis could fix the c_{ab} starvation. We show that the answer is no: no affine product basis can simultaneously achieve full coverage, coherent direction, and unbiased updates for c_{ab} under STE. This is why CovJac (a different gradient mechanism, not a different basis) is needed.

4.1 Soft-Mix backward cancellation

We denote $c_{a,j}$, $c_{b,j}$, $c_{ab,j}$ for the coefficients of gate j (i.e. the entries of $\mathbf{G}_j$ from §3).

Lemma 1 (Zero-sum gate derivative symmetry; elementary)**.** *For the 16 Boolean coefficient vectors,*

$$\sum_{j=0}^{15} c_{a,j} = \sum_{j=0}^{15} c_{b,j} = \sum_{j=0}^{15} c_{ab,j} = 0. \tag{7}$$

Proof sketch. Complementation $g \mapsto 1 - g$ negates the non-constant coefficients ($\bar{c}_a = -c_a$, $\bar{c}_b = -c_b$, $\bar{c}_{ab} = -c_{ab}$) and pairs the 16 gates into 8 complementary pairs, so each pair cancels in those coefficients and the sum vanishes. (The complementation pairing is the observation of Rüttgers et al. (2026, Eq. 7); the zero-sum is standard (Crama & Hammer, 2011; O'Donnell, 2014).)

Proposition 2 (Soft-Mix backward vanishes at uniform $\boldsymbol{\pi}$)**.** *Rüttgers et al. (2026, Eq. 7) showed that this complementation symmetry causes approximate gradient cancellation at symmetric initialization. We sharpen this to an exact identity that holds for all inputs, not only at initialization: for* $z = \sum_{j=0}^{15} \pi_j g_j(a, b)$ *and* $\pi_j = 1/16$ *for all* j*,*

$$\frac{\partial z}{\partial a}(a, b) = \frac{\partial z}{\partial b}(a, b) = 0 \quad \textit{for every } (a, b) \in [0, 1]^2. \tag{8}$$

Proof sketch. Substituting the multilinear form and differentiating, $\partial z/\partial a = \sum_j \pi_j (c_{a,j} + c_{ab,j} b)$. At $\pi_j = 1/16$, this equals $(1/16)[\sum_j c_{a,j} + b \sum_j c_{ab,j}] = 0$ by Lemma 1. The same holds for $\partial z/\partial b$.

Dynamical interpretation (Observation 1; informal, not a theorem). A near-uniform bound (proved in Appendix A.3) gives $|\partial z/\partial a| \leq \|\boldsymbol{\pi} - \mathbf{u}\|_1$ where $\mathbf{u} = (1/16, \ldots, 1/16)$ is the uniform distribution, so if $\|\boldsymbol{\pi}^{(l)} - \mathbf{u}\|_1 \leq \epsilon$ at layer l, the per-layer signal attenuation is at most ϵ. At initialization with logit std σ=1.0, the expected $\|\boldsymbol{\pi} - \mathbf{u}\|_1 \approx 0.72$, giving per-layer signal ≤ 0.72; over L layers the product is $O(0.72^L)$, which at L=12 is ≈ 0.019. *What is rigorous*: the per-neuron bound $\|\boldsymbol{\pi} - \mathbf{u}\|_1$. *What is not*: that neurons remain near-uniform throughout training rather than committing quickly; this is confirmed empirically (§5.4, Figure 2c).

Single-gate-forward corollary. Proposition 2's cancellation requires summing over all 16 gates: only then do the Lemma 1 zero-sums cancel inside $\partial z/\partial a$. Any method committing to a *single* gate in the forward pass (whether by hard quantization or stochastic sampling (Jang et al., 2017; Maddison et al., 2017)) avoids the cancellation by construction, regardless of parameter count. This prediction is confirmed by the depth experiments (§5.2, Figure 2).

The soft mixture never reaches the cancellation point. During training, Multilinear-CovJac uses a *soft* forward, $z = \sum_j \omega_j g_j(a, b)$ with $\omega_j = \mathrm{softmax}(-\|\mathbf{c} - \mathbf{G}_j\|^2/\tau)_j$, so it does not commit to a single gate. It nonetheless avoids the *exact* cancellation of Proposition 2, which requires $\boldsymbol{\omega} = \mathbf{u}$, a weight vector no coefficient produces; the near-uniform regime, where the forward gradient is small but nonzero, and what the method actually rescues, are treated in the Scope paragraph below.

Lemma 3 (Uniform soft-VQ weights are unattainable)**.** *For the 16-Boolean codebook $\{\mathbf{G}_j\}$ there is no $\mathbf{c} \in \mathbb{R}^4$ with $\omega_j(\mathbf{c}) = 1/16$ for all j; the soft-VQ map $\mathbf{c} \mapsto \boldsymbol{\omega}(\mathbf{c})$ never attains the uniform $\mathbf{u}$ at which Proposition 2 forces $\partial z/\partial a = \partial z/\partial b = 0$.*

Proof sketch. Uniform $\boldsymbol{\omega}$ requires $\mathbf{c}$ equidistant from all 16 gates, a linear system $A\mathbf{c} = \mathbf{r}$ with rows $A_j = 2(\mathbf{G}_j - \mathbf{G}_0)$. For the codebook, $\mathrm{rank}(A) = 4 < \mathrm{rank}([A \mid \mathbf{r}]) = 5$, so the system is inconsistent (Rouché–Capelli) and no such $\mathbf{c}$ exists. Equivalently, the multilinear map is not a similarity ($M^\top M \neq \lambda I$), so the cospherical truth-table vertices map off any common sphere in coefficient space. Full proof in Appendix A.3.

Scope: the two initializations. The methods begin at a symmetric setting with opposite consequences, set by the parametrization. Soft-Mix initializes its *sixteen* gate weights uniformly ($\boldsymbol{\pi} = \mathbf{u}$), which is *exactly* the cancellation point of Proposition 2: its input gradient is zero at init, and because the weight update is rank-4 deficient (§3) $\boldsymbol{\pi}$ stays near $\mathbf{u}$ and collapses with depth. Our method learns *four* coefficients $\mathbf{c}$, whose induced soft-VQ weights $\boldsymbol{\omega}(\mathbf{c})$ can never equal $\mathbf{u}$ (Lemma 3); so from any coefficient initialization $\boldsymbol{\omega}$ is already concentrated and the input gradient is nonzero from the start ($s_l \approx 0.9$ at init versus 0.29 for Soft-Mix; §5.4). This is not a claim that the soft-VQ *forward* gradient could never cancel: at the unattainable uniform $\boldsymbol{\omega}$ it would, with $\mathbf{c}_{\text{soft}} = (\frac{1}{2}, 0, 0, 0)$, exactly as Soft-Mix. What keeps that point harmless is the *parameter* gradient, which feeds c_{ab} a nonzero signal ($J_{03} = 0.5$) even there, so the symmetric point is not a fixed point of training and the signal is preserved at depth (Appendix A.4, §5.2).

4.2 Multilinear-STE avoids the cancellation

Proposition 4 (Multilinear-STE backward non-cancellation)**.** *In Multilinear-STE with committed gate $\hat{\mathbf{c}} = (c_0, c_a, c_b, c_{ab})$, the STE backward derivative is $\partial z/\partial a = c_a + c_{ab}b$, and for every Boolean gate depending on input a, $\max_{b\in\{0,1\}} |c_a + c_{ab}b| \geq 1$.*

Proof sketch. STE differentiates the committed polynomial to $c_a + c_{ab}b$; for any a-dependent gate, evaluating at b=0 (if $c_a \neq 0$) or b=1 (if c_a=0, $c_{ab} \neq 0$) gives magnitude ≥ 1 (integer coefficients). Full proof and gate enumeration in Appendix A.5.

4.3 CovJac selection Jacobian

Proposition 4 shows that STE avoids cancellation but still starves c_{ab} (25% coverage). We now show that the CovJac Jacobian has three properties that together guarantee c_{ab} receives gradient on *every* sample.

Proposition 5 (CovJac input-independence, PSD, positive diagonal). *For* $\mathbf{c} \mapsto \mathbf{c}_{soft}(\mathbf{c}) = \sum_j \omega_j \mathbf{G}_j$, *the Jacobian is*

$$\mathbf{J}(\mathbf{c}) = \frac{2}{\tau} \operatorname{Cov}_{\boldsymbol{\omega}}(\mathbf{G}), \tag{9}$$

i.e. (i) input-independent (depends only on $\mathbf{c}$*, not* (a, b)*); (ii) symmetric PSD; (iii) strictly positive diagonal for every coordinate throughout training at* $\tau > 0$ *on the 16-Boolean codebook.*

Proof sketch. Differentiating $(\mathbf{c}_{\text{soft}})_i = \sum_j \omega_j G_{ji}$ with $\partial \omega_j / \partial c_m = (2/\tau)\omega_j(G_{jm} - (\mathbf{c}_{\text{soft}})_m)$ yields $J_{im} = (2/\tau)\operatorname{Cov}_\omega(G_{\cdot i}, G_{\cdot m})$, a covariance matrix; it is therefore PSD, with diagonal $J_{mm} = (2/\tau)\operatorname{Var}_\omega(G_{\cdot m}) > 0$ since each coordinate varies over the codebook. Full proof in Appendix A.5.

The cross-coefficient coupling consequence (c_{ab} receives gradient on every sample via J_{03}, Eq. 6) follows directly from Proposition 5's positive diagonal and input-independence.

Remark: first moment cancels, second moment survives. Proposition 2 and Proposition 5 are the same complementation symmetry acting on two different statistics of the codebook. The Soft-Mix input gradient is a *first moment*, $\partial z / \partial a = \sum_j \pi_j (c_{a,j} + c_{ab,j} b)$, whose non-constant part vanishes at uniform $\boldsymbol{\pi}$ because each complementary pair negates c_a and c_{ab} (Lemma 1). The signal that trains c_{ab} under CovJac is instead a *second (cross) moment*, $J_{03} = (2/\tau)\operatorname{Cov}_\omega(G_{\cdot 0}, G_{\cdot 3})$, which the same symmetry leaves intact: under complementation $c_{ab} \mapsto -c_{ab}$ but $c_0 \mapsto 1 - c_0$ (only the constant coefficient absorbs the $+1$ of $g \mapsto 1 - g$), so the product $c_0 c_{ab}$ is not antisymmetric across a complementary pair and the covariance does not cancel. At the symmetric point ($\omega_j = 1/16$, $\tau = 1$) the first moment is 0 while $J_{03} = 0.5$: exactly where the mixture's forward gradient vanishes, the CovJac backward still routes gradient to c_{ab}.

Basis futility. One might hope a different polynomial basis fixes the c_{ab} starvation without changing the gradient mechanism. It cannot: no affine product reparameterization simultaneously attains full interaction coverage ($\rho = 1$), gradient coherence ($\kappa = 1$), and zero STE bias ($B = 0$), so the canonical basis is optimal up to positive rescaling (Proposition 7, stated and proved in Appendix A.6).

5 Experiments

We validate the theoretical predictions on seven binary-input datasets: Adult, Splice (Dua & Graff, 2019), MNIST (LeCun et al., 1998), SVHN (Netzer et al., 2011), CIFAR-10/100 (Krizhevsky, 2009), and MONK's-2 (Thrun et al., 1991). All experiments use the DLGN architecture (Petersen et al., 2022) with $L = 6$ layers, GroupSum readout, Adam (Kingma & Ba, 2015) (lr$= 0.01$), batch 512, and 3 seeds. We report **last-10 accuracy** (mean of final 10 evaluation checkpoints). We compare *Soft-Mix* (16p), *Gumbel-ST* (16p), *Multilinear-STE* (ours, 4p), and *Multilinear-CovJac* (ours, 4p); CovJac uses $\tau = 1$ except Adult ($\tau = 0.1$; Appendix E.13). Full setup details in Appendix E.1; additional results including depth tables, width scaling, architecture baselines, and temperature sensitivity in Appendix E. Our Soft-Mix is a faithful reproduction of difflogic: at the identical 6-layer $\times$ 64,000-gate MNIST architecture it reaches 98.32% best-checkpoint, within 0.15pp of difflogic's reported 98.47% (Petersen et al., 2022, Table 4), and our primary last-10 (98.29%) is within 0.18pp.

5.1 Cross-dataset comparison

Table 1 shows that CovJac matches or exceeds all baselines on every dataset using $4\times$ fewer parameters. The CovJac advantage over Soft-Mix is small on easy tasks (+0.01pp on MNIST) but grows on harder ones (+0.84pp on CIFAR-10, +0.45pp on CIFAR-100 with the tightest variance). On Adult, the default $\tau = 1.0$ causes CovJac's c_{ab} coupling to add overhead on this low-interaction task; lowering τ to 0.1 reduces the coupling and recovers M-STE-level accuracy (84.93%).

The most informative comparison is CovJac vs STE, since both use 4 parameters and differ only in the gradient mechanism. This gap widens monotonically with task interaction demand: $+0.02$pp (Adult) $\rightarrow +0.13$pp (Splice) $\rightarrow +0.21$pp (MNIST) $\rightarrow +2.22$pp (SVHN) $\rightarrow +2.95$pp (CIFAR-10) $\rightarrow +4.35$pp (CIFAR-100) $\rightarrow +7.53$pp (MONK's-2; Table 15 and Figure 4 in Appendix E.9). On the most interaction-dense tasks (CIFAR-10/100), STE underperforms even Soft-Mix (56.02% / 24.02% vs 58.13% / 27.92%), confirming that c_{ab} starvation is the binding constraint. CovJac is

Table 1: Cross-dataset comparison. Every number is the last-10 accuracy (± std, 3 seeds) of the *deployed, strict-Boolean, zero-multiply-accumulate* circuit under one matched harness (identical wiring, width, budget, and seeds across methods); best-checkpoint accuracy in Appendix Table 9. **Bold** = best per column. L=6; k: MNIST 64k, Adult/Splice 4k, SVHN/CIFAR-10 128k, CIFAR-100 256k, MONK's-2 136 (100k iters for CIFAR-100; 10k for MONK's-2; rest 50k). Multilinear-CovJac is our soft-VQ method (the primary contribution); Multilinear-STE is a straight-through ablation anchor on the same four coefficients that isolates the gradient mechanism, not a separate proposal, so rows where it trails Soft-Mix measure the cost of the straight-through gradient, not of soft vector quantization.

Method	#p/n	Adult	Splice	MONK's-2	MNIST	SVHN	CIFAR-10	CIFAR-100
Soft-Mix	16	84.68±.07	97.01±.11	81.32±1.09	98.29±.04	68.21±.08	58.13±.12	27.92±.43
Gumbel-ST	16	84.69±.06	96.81±.30	81.46±.89	98.17±.07	66.12±.38	58.24±.25	27.72±.40
M-STE (ours, anchor)	4	84.91±.01	97.07±.00	78.53±1.25	98.09±.02	66.69±.11	56.02±.23	24.02±.17
M-CovJac (ours, primary)	4	**84.93**±.03†	**97.20**±.23	**86.06**±2.17	**98.30**±.04	**68.91**±.13	**58.97**±.26	**28.37**±.22

† Adult: τ=0.1 (all other datasets: τ=1.0; Appendix E.13).

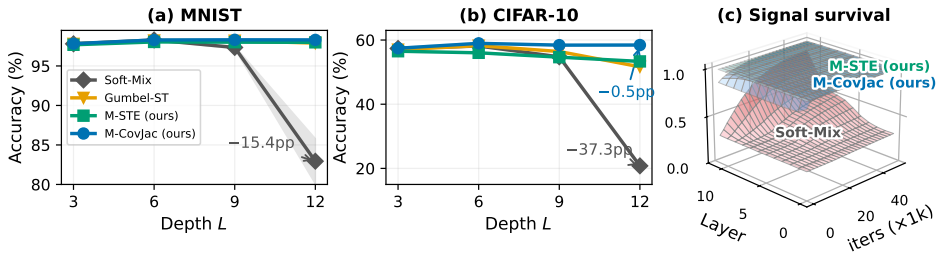


Figure 2: **(a,b)** Accuracy vs depth (3 seeds): Soft-Mix collapses on MNIST and CIFAR-10 while CovJac holds (−0.5pp on CIFAR-10 at L=12). **(c)** Per-layer signal survival s_l at initialization (MNIST, L=12): Soft-Mix stays near 0.30 (71% of the input gradient cancelled per layer, Proposition 2) while M-STE stays at 1 and M-CovJac near 0.9. Component ablation (Table 22) in Appendix E.11.

the best method on all seven datasets; under a null of equal performance, a one-sided sign test gives $p = 2^{-7} = 0.0078 < 0.01$ (one-sided because the theory predicts CovJac ≥ STE a priori; Appendix E.1).

5.2 Depth Stability

Depth (Figure 2). As predicted by Proposition 2 and Observation 1, Soft-Mix collapses at depth (−15.4pp on MNIST, −37.3pp on CIFAR-10 from L=6 to L=12) as its cancellation compounds across layers, while all single-gate-forward methods survive: CovJac is the most depth-stable (−0.5pp on CIFAR-10, essentially zero on MNIST and, at τ=0.1, on Adult), followed by STE and Gumbel-ST. Doubling training to 100k iterations only partially recovers Soft-Mix (a 6.0pp gap remains), confirming the penalty is structural, not a training-speed lag (Appendix E.10).

Ablation (Table 22). The +17.25pp IWP-STE → M-STE gap is purely from the basis change (both span the same function space; Appendix C); the effect is even larger on CIFAR-10 (+22.65pp) and present on Adult (+8.50pp). The 2×2 basis×constraint factorial (Appendix F) confirms: the gap appears only under STE, not under continuous training. At L=12, Multilinear-free drops to 92.95% while IWP-free holds at 97.80% (Appendix Table 25), reversing the ranking: the multilinear basis's degree hierarchy becomes a liability without the snap constraint.

5.3 Cost, Width Scaling, and Interaction-Demand Scaling

Cost. At deployment all methods evaluate a single hard gate (∼7 ops); at training M-STE is 25× cheaper than Soft-Mix and all 4-parameter methods use 4× less optimizer memory (Appendix Table 5).

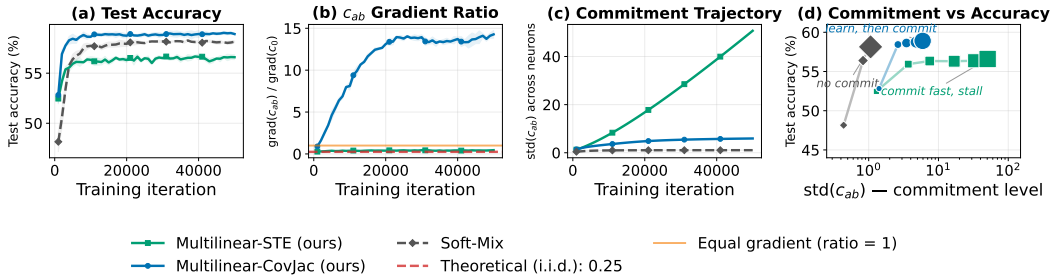


Figure 3: CIFAR-10 (k=128k, L=6, 3 seeds). **(a)** Accuracy over training. **(b)** c_{ab} gradient ratio: CovJac reaches $\sim$14×, STE only $\sim$0.42. **(c)** Commitment ($\mathrm{std}(c_{ab})$): STE commits fast, CovJac slow, Soft-Mix never. **(d)** Commitment vs accuracy: STE commits to wrong gates and stalls; CovJac learns first, then commits; Soft-Mix improves without committing.

Width and interaction-demand scaling. At L=1 on MNIST, CovJac's error decays faster with width (α=0.42 vs Soft-Mix's 0.26 in $A(k)$=100−$Bk^{-\alpha}$); at matched parameter budget (4× wider CovJac) the gap is +3.3pp at 1/6 the training-evaluation cost, both deploying to the same zero-multiply-accumulate circuit (Appendix E.4). The CovJac-over-STE advantage grows monotonically with task interaction demand, from +0.02pp (separable Adult) to +7.53pp (interaction-dense MONK's-2), correlating with a c_{ab} *starvation severity* at Pearson r=0.85 (Appendix E.9); it persists at larger scale (CIFAR-100 k=1280k, Tiny-ImageNet k=2560k; Appendix E.6).

5.4 Diagnostic experiments

Commitment dynamics (Figure 3). On CIFAR-10 over training, the c_{ab} gradient-norm ratio $\|\nabla c_{ab}\|/\|\nabla c_0\|$ reaches 14.27 for CovJac but only 0.42 for STE (a $\sim$34× amplification from the J_{03} coupling; STE sits above the $\mathbb{E}[ab]$=1/4 of Lemma 6 because CIFAR-10's threshold encoding is denser than Bernoulli(1/2)). Commitment, measured by $\mathrm{std}(c_{ab})$ across neurons, separates the methods: STE commits fast (std=50.7) but to suboptimal gates (56.02%), CovJac slowly (std=5.9) but accurately (58.97%), and Soft-Mix never (std=1.1); STE locks in before learning which gates are right, whereas CovJac explores first and commits after accuracy plateaus (Appendix E.7).

Signal survival, gate selection, and controls. A direct test of Proposition 2 confirms the cancellation: per-layer signal survival at initialization (MNIST, L=12) is $s_l \approx 0.29$ for Soft-Mix (71% cancelled per layer) versus 1.0 for M-STE and $\approx$0.9 for CovJac (Figure 2c). At convergence CovJac selects the most strong-interaction gates ($|c_{ab}|\geq$2: 26–36% vs 16–21% for others) and the fewest constant gates (<0.5% vs 4–10%), a direct effect of the J_{03} coupling (Figure 5). Control bases behave as predicted: the Walsh and smoothed alternatives fail exactly as Proposition 7 requires (Appendix D).

6 Conclusion

We recast logic-gate-network training as vector quantization in the 4-dimensional multilinear space of the 16 gates. We prove an *exact* cancellation law: the 16-way softmax is rank-4 and its input gradient vanishes for every input at symmetric weights (sharpening a prior approximate-at-initialization observation), and no affine basis removes the resulting c_{ab} starvation under a straight-through gradient. Our remedy, Multilinear-CovJac, trains the four coefficients through soft vector quantization whose exact gradient is a codebook covariance Jacobian feeding c_{ab} on every sample; it matches or exceeds 16-parameter Soft-Mix on all seven datasets with 4× fewer parameters, the advantage growing on interaction-dense tasks and holding at depth where Soft-Mix collapses. The mechanism is a property of the codebook, not of Boolean gates: the backward cancellation extends to any *fixed* dictionary closed under complementation, whereas methods that *learn* their elements (MoE (Shazeer et al., 2017), DARTS (Liu et al., 2019; Chu et al., 2020), VQ (van den Oord et al., 2017)) share the discretization gap but not the cancellation, since their elements are not complementation pairs. Extending these effects to k-input gates and learnable input wiring (Appendix E.1) is left to future work.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods, namely writing and debugging our experiment and analysis code (a task with required disclosure). We have reviewed all AI-assisted work and take responsibility for the final content of this paper.

ETHICS STATEMENT

This work studies the optimization of differentiable logic gate networks on standard, publicly available benchmarks (MNIST, SVHN, CIFAR-10/100, and UCI datasets). It involves no human subjects, no private or personally identifiable data, and no sensitive applications, and we foresee no direct ethical concerns. To the extent that the method improves the efficiency of deployed Boolean-circuit models, it may modestly reduce the energy cost of inference.

REPRODUCIBILITY STATEMENT

All results use a single matched training harness described in Section 5 and Appendix E.1 (architecture, optimizer, learning rate, batch size, per-dataset width, and the 3-seed protocol). The main text and appendix give the full equations, hyperparameters, and settings needed to reproduce every result: complete proofs of every proposition and the lemma are in Appendix A.3, A.5, and A.6, and dataset construction (thresholding/binarization) with all per-dataset hyperparameters is in Appendix E.1. Source code will be released upon acceptance.

REFERENCES

Eirikur Agustsson, Fabian Mentzer, Michael Tschannen, Lukas Cavigelli, Radu Timofte, Luca Benini, and Luc Van Gool. Soft-to-hard vector quantization for end-to-end learning compressible representations. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Marta Andronic and George A. Constantinides. PolyLUT: Learning piecewise polynomials for ultra-low latency FPGA LUT-based inference. In *International Conference on Field Programmable Technology (ICFPT)*, pp. 60–68, 2023.

Marta Andronic, Jiawen Li, and George A. Constantinides. PolyLUT: Ultra-low latency polynomial inference with hardware-aware structured pruning. *IEEE Transactions on Computers*, 74(9): 3181–3194, 2025.

Alan T. L. Bacellar, Zachary Susskind, Mauricio Breternitz Jr, Eugene John, Lizy Kurian John, Priscila Machado Vieira Lima, and Felipe M. G. França. Differentiable weightless neural networks. In *International Conference on Machine Learning (ICML)*, 2024. Differentiable training of k-input weightless lookup tables on FPGA-targeted classifiers.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation, 2013.

YooJung Choi, Antonio Vergari, and Guy Van den Broeck. Probabilistic circuits: A unifying framework for tractable probabilistic models. 2020.

Xiangxiang Chu, Tianbao Zhou, Bo Zhang, and Jixiang Li. Fair DARTS: Eliminating unfair advantages in differentiable architecture search. In *European Conference on Computer Vision (ECCV)*, 2020.

Yves Crama and Peter L. Hammer. *Boolean Functions: Theory, Algorithms, and Applications*, volume 142 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2011. doi: 10.1017/CBO9780511852008.

Luc De Raedt, Angelika Kimmig, and Hannu Toivonen. ProbLog: A probabilistic Prolog and its application in link discovery. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2462–2467, 2007.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2019.

Peter L. Hammer and Sergiu Rudeanu. *Boolean Methods in Operations Research and Related Areas*. Springer, Berlin, Heidelberg, 1968. doi: 10.1007/978-3-642-85823-9.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Minyoung Huh, Brian Cheung, Pulkit Agrawal, and Phillip Isola. Straightening out the straight-through estimator: Overcoming optimization challenges in vector quantized networks. In *International Conference on Machine Learning*, 2023.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*, 2017.

Stasys Jukna. *Boolean Function Complexity: Advances and Frontiers*, volume 27 of *Algorithms and Combinatorics*. Springer, 2012. doi: 10.1007/978-3-642-24508-4.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Erich Peter Klement, Radko Mesiar, and Endre Pap. *Triangular Norms*, volume 8 of *Trends in Logic*. Kluwer Academic Publishers, Dordrecht, 2000. doi: 10.1007/978-94-015-9540-7.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Ya Le and Xuan Yang. Tiny ImageNet visual recognition challenge. 2015. CS231N Course Report, Stanford University.

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.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *International Conference on Learning Representations*, 2019.

Binglei Lou, Richard Rademacher, David Boland, and Philip H. W. Leong. PolyLUT-add: FPGA-based LUT inference with wide inputs. In *International Conference on Field-Programmable Logic and Applications (FPL)*, 2024.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, Cambridge, UK, 2014. doi: 10.1017/CBO9781139814782.

Art B. Owen. Variance components and generalized Sobol' indices. *SIAM/ASA Journal on Uncertainty Quantification*, 1(1):19–41, 2013. doi: 10.1137/120876782.

Guillermo Owen. Multilinear extensions of games. *Management Science*, 18(5-Part-2):P64–P79, 1972. doi: 10.1287/mnsc.18.5.64.

Felix Petersen, Christian Borgelt, Hilde Kuehne, and Oliver Deussen. Deep differentiable logic gate networks. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Felix Petersen, Hilde Kuehne, Christian Borgelt, Julian Welzel, and Stefano Ermon. Convolutional differentiable logic gate networks. In *Advances in Neural Information Processing Systems*, 2024. arXiv:2411.04732.

Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. XNOR-Net: ImageNet classification using binary convolutional neural networks. In *European Conference on Computer Vision (ECCV)*, pp. 525–542, 2016.

Lukas Rüttgers, Till Aczel, Andreas Plesner, and Roger Wattenhofer. Light differentiable logic gate networks. In *International Conference on Learning Representations*, 2026. arXiv:2510.03250.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR)*, 2017.

Ilya M. Sobol'. Sensitivity estimates for nonlinear mathematical models. *Mathematical Modelling and Computational Experiments*, 1(4):407–414, 1993.

Sebastian B. Thrun et al. The MONK's problems: A performance comparison of different learning algorithms. Technical Report CMU-CS-91-197, Carnegie Mellon University, 1991.

Yaman Umuroglu, Yash Akhauri, Nicholas J. Fraser, and Michaela Blott. LogicNets: Co-designed neural networks and circuits for extreme-throughput applications. In *International Conference on Field-Programmable Logic and Applications (FPL)*, pp. 291–297, 2020.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Emile van Krieken, Erman Acar, and Frank van Harmelen. Analyzing differentiable fuzzy logic operators. *Artificial Intelligence*, 302:103602, 2022. doi: 10.1016/j.artint.2021.103602.

Erwei Wang, James J. Davis, Peter Y. K. Cheung, and George A. Constantinides. LUTNet: Rethinking inference in FPGA soft logic. In *IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2019.

Shakir Yousefi, Andreas Plesner, Till Aczel, and Roger Wattenhofer. Mind the gap: Removing the discretization gap in differentiable logic gate networks. In *Advances in Neural Information Processing Systems*, 2025. arXiv:2506.07500.

A Full Proofs

Notation. For each of the 16 Boolean functions $g_j : \{0,1\}^2 \to \{0,1\}$ (indexed $j = 0, \dots, 15$ by the truth-table bit pattern), we write its unique multilinear representation

$$g_j(a,b) = c_{0,j} + c_{a,j}\, a + c_{b,j}\, b + c_{ab,j}\, ab,$$

with coefficient vector $\mathbf{G}_j = (c_{0,j},\, c_{a,j},\, c_{b,j},\, c_{ab,j})^\top \in \mathbb{Z}^4$. Collect these as rows of the codebook $\mathbf{G} \in \mathbb{Z}^{16\times 4}$. The canonical polynomial basis is $\boldsymbol{\psi}(a,b) = (1, a, b, ab)^\top$. A continuous learned coefficient vector is denoted $\mathbf{c} \in \mathbb{R}^4$, and under Multilinear-STE the forward "hard" gate is $\mathbf{G}_{\text{hard}} = \mathbf{G}_{j^*}$ with $j^* = \operatorname{argmin}_j \|\mathbf{c} - \mathbf{G}_j\|^2$. Under Multilinear-CovJac the soft coefficient is $\mathbf{c}_{\text{soft}} = \sum_j \omega_j \mathbf{G}_j$ with $\omega_j = \operatorname{softmax}(-\|\mathbf{c} - \mathbf{G}_j\|^2/\tau)_j$ (we write ω_j for the CovJac proximity weights and reserve π_j for the Soft-Mix gate distribution, following §3).

A.1 Zero Discretization Gap (Identity 1)

Identity 1 (Polynomial collapse) Every function $g : \{0,1\}^2 \to \mathbb{R}$ admits a unique multilinear polynomial representation

$$g(a,b) = c_0 + c_a\, a + c_b\, b + c_{ab}\, ab,$$

with coefficients

$$c_0 = g(0,0),\ c_a = g(1,0) - g(0,0),\ c_b = g(0,1) - g(0,0),\ c_{ab} = g(1,1) - g(1,0) - g(0,1) + g(0,0).$$

For the 16 Boolean-valued functions $g : \{0,1\}^2 \to \{0,1\}$, the resulting coefficient vectors $\mathbf{G}_j = (c_0, c_a, c_b, c_{ab})^\top$ lie in $\mathbb{Z}^4$, with entries in the ranges $c_0 \in \{0,1\}$, $c_a, c_b \in \{-1,0,1\}$, $c_{ab} \in \{-2,-1,0,1,2\}$.

Proof. (Direct corner-value expansion / dimension argument.) Let V be the vector space of functions $\{0,1\}^2 \to \mathbb{R}$, which has dimension 4 since a function is determined by its four corner values

$g(0,0), g(1,0), g(0,1), g(1,1)$. Consider the four monomials $\{1,\ a,\ b,\ ab\}$ as elements of V. Their corner-value vectors are

$$1 \leftrightarrow (1,1,1,1),\ a \leftrightarrow (0,1,0,1),\ b \leftrightarrow (0,0,1,1),\ ab \leftrightarrow (0,0,0,1),$$

where entries are ordered $(a,b) = (0,0),(1,0),(0,1),(1,1)$. The 4×4 matrix formed by these vectors,

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix},$$

is lower triangular with unit diagonal, hence $\det M = 1 \neq 0$. The four monomials are therefore linearly independent in V, and since $\dim V = 4$, they form a basis. Uniqueness of the representation is immediate from basis uniqueness.

To obtain the explicit formulae, solve $M\mathbf{c} = (g(0,0), g(1,0), g(0,1), g(1,1))^\top$ by forward substitution:

- Row 1: $c_0 = g(0,0)$.
- Row 2: $c_0 + c_a = g(1,0) \ \Rightarrow\ c_a = g(1,0) - g(0,0)$.
- Row 3: $c_0 + c_b = g(0,1) \ \Rightarrow\ c_b = g(0,1) - g(0,0)$.
- Row 4: $c_0 + c_a + c_b + c_{ab} = g(1,1) \ \Rightarrow\ c_{ab} = g(1,1) - g(1,0) - g(0,1) + g(0,0)$.

For Boolean-valued g (i.e. $g \in \{0,1\}$), the four corner values are each in $\{0,1\}$, so $c_0 \in \{0,1\}$, $c_a, c_b \in \{-1,0,1\}$, and $c_{ab} \in \{-2,-1,0,1,2\}$. Enumerating the $2^4 = 16$ Boolean-valued corner assignments gives 16 distinct integer coefficient vectors, one per Boolean gate. □

Attribution: (Hammer & Rudeanu, 1968; Owen, 1972; Crama & Hammer, 2011). The $\{-1,+1\}$ Fourier reformulation is in O'Donnell (2014, Proposition 1.1). Elementary; proved here for self-containedness.

Zero discretization gap. For any Multilinear-STE layer with binary inputs $(a,b) \in \{0,1\}^2$,

$$z_{\text{train}}(a,b) = z_{\text{eval}}(a,b).$$

Proof. Let $\hat{\mathbf{c}} = \mathbf{G}_{j^*}$ be the committed Boolean gate (the codebook entry nearest to $\mathbf{c}$ in squared Euclidean distance, ties broken by lexicographic order). The training-time forward pass of Multilinear-STE emits

$$z_{\text{train}}(a,b) = \hat{\mathbf{c}}^\top \boldsymbol{\psi}(a,b) = \mathbf{G}_{j^*}^\top \boldsymbol{\psi}(a,b).$$

The evaluation-time forward pass emits the Boolean gate value itself, $z_{\text{eval}}(a,b) = g_{j^*}(a,b)$. By Identity 1, the multilinear polynomial with coefficient vector $\mathbf{G}_{j^*}$ equals g_{j^*} on $\{0,1\}^2$, i.e. precisely $g_{j^*}(a,b) = \mathbf{G}_{j^*}^\top \boldsymbol{\psi}(a,b)$ for all $(a,b) \in \{0,1\}^2$. Substituting, $z_{\text{eval}}(a,b) = \mathbf{G}_{j^*}^\top \boldsymbol{\psi}(a,b) = z_{\text{train}}(a,b)$. □

Remark. Multilinear-CovJac does not satisfy the zero-gap property as a pointwise identity (since $\mathbf{c}_{\text{soft}} \neq \mathbf{G}_{j^*}$ during training), yet empirically the discretization gap is $\leq 0.2\%$ on all main datasets because the trained $\boldsymbol{\pi}$ concentrates on $\mathbf{G}_{j^*}$ strongly enough that $\mathbf{c}_{\text{soft}}^\top \boldsymbol{\psi}(a,b)$ rounds to $g_{j^*}(a,b)$ at every Boolean input.

A.2 Gradient Utilization (Lemma 6)

Lemma 6 (Gradient coverage hierarchy)**.** *Under i.i.d. Bernoulli*$(1/2)$ *binary inputs* $(a,b) \in \{0,1\}^2$*,*

$$\mathbb{E}\big[\|\boldsymbol{\psi}(a,b)\|_0\big] = 2.25 \quad \textit{(Multilinear / polynomial basis)},$$

$$\mathbb{E}\big[\|\boldsymbol{\phi}(a,b)\|_0\big] = 1.0 \quad \textit{(Corner / IWP basis)}.$$

Proof. (Component-wise expectation.)

Polynomial basis. We have $\boldsymbol{\psi}(a,b) = (1, a, b, ab)^\top$. The ℓ_0 norm counts non-zero entries, so by linearity of expectation

$$\mathbb{E}\big[\|\boldsymbol{\psi}(a,b)\|_0\big] = \mathbb{E}[\mathbf{1}_{1\neq 0}] + \mathbb{E}[\mathbf{1}_{a\neq 0}] + \mathbb{E}[\mathbf{1}_{b\neq 0}] + \mathbb{E}[\mathbf{1}_{ab\neq 0}].$$

Compute each term: $\mathbb{E}[\mathbf{1}_{1\neq 0}] = 1$; $\mathbb{E}[\mathbf{1}_{a\neq 0}] = \Pr[a = 1] = 1/2$; $\mathbb{E}[\mathbf{1}_{b\neq 0}] = 1/2$; $\mathbb{E}[\mathbf{1}_{ab\neq 0}] = \Pr[a = 1 \wedge b = 1] = 1/4$ (independence). Sum: $1 + 1/2 + 1/2 + 1/4 = 9/4 = 2.25$.

Corner basis. The corner basis $\boldsymbol{\phi}(a,b)$ is the one-hot indicator of (a,b) over the four corners of $\{0,1\}^2$, i.e. $\phi_i(a,b) = \mathbf{1}_{(a,b)=v_i}$ where v_i ranges over $(0,0), (1,0), (0,1), (1,1)$. For any input (a,b), exactly one component of $\boldsymbol{\phi}$ equals 1 and the others equal 0, so $\|\boldsymbol{\phi}(a,b)\|_0 = 1$ deterministically. Hence $\mathbb{E}\|\boldsymbol{\phi}\|_0 = 1$. □

Interpretation. The polynomial basis distributes the learning load across degrees: the degree-d monomial has coverage $(1/2)^d$. The four components c_0, c_a, c_b, c_{ab} receive gradient on $100\%, 50\%, 50\%, 25\%$ of samples respectively, giving the natural degree-ordered curriculum discussed in Section 3. The corner basis, by contrast, has a flat 25% coverage per coordinate but only *one* non-zero coordinate per sample, yielding the same 1.0 expected ℓ_0 norm.

A.3 Backward Gradient Cancellation in Soft-Mix (Lemma 1, Proposition 2, Observation 1)

Lemma 1 (Zero-sum gate derivative symmetry)

$$\sum_{j=0}^{15} c_{a,j} \;=\; \sum_{j=0}^{15} c_{b,j} \;=\; \sum_{j=0}^{15} c_{ab,j} \;=\; 0.$$

Consequently, for every $(a,b) \in [0,1]^2$,

$$\sum_{j=0}^{15} \frac{\partial g_j}{\partial a}(a,b) \;=\; \sum_{j=0}^{15} (c_{a,j} + c_{ab,j}\, b) \;=\; 0,$$

and symmetrically for ∂_b.

Proof. (Complementation pairing.) The set $\mathcal{B}$ of Boolean functions $\{0,1\}^2 \to \{0,1\}$ is closed under pointwise complementation $g \mapsto \bar{g} := 1 - g$. This map is an involution on $\mathcal{B}$ ($\bar{\bar{g}} = g$) and has no fixed points: $g = \bar{g}$ would imply $g \equiv 1/2$, which is not Boolean-valued. Hence $\mathcal{B}$ partitions into $16/2 = 8$ complementary pairs $(g_j, g_{\bar{j}})$.

Compute the multilinear coefficients of $\bar{g}_j$. Using the identity $1 - g_j(a,b) = 1 - c_{0,j} - c_{a,j}a - c_{b,j}b - c_{ab,j}ab$:

$$\bar{g}_j(a,b) = (1 - c_{0,j}) + (-c_{a,j})\, a + (-c_{b,j})\, b + (-c_{ab,j})\, ab.$$

By uniqueness of the multilinear representation (Identity 1) applied to $\bar{g}_j$, the coefficient vector of $\bar{g}_j$ is $(1 - c_{0,j}, -c_{a,j}, -c_{b,j}, -c_{ab,j})$. In particular, the three *non-constant* coefficients satisfy $c_{a,\bar{j}} = -c_{a,j}$, $c_{b,\bar{j}} = -c_{b,j}$, $c_{ab,\bar{j}} = -c_{ab,j}$, so that $c_{a,j} + c_{a,\bar{j}} = 0$, and likewise for c_b and c_{ab}.

Summing over all 16 gates by grouping them into the 8 complementary pairs,

$$\sum_{j=0}^{15} c_{a,j} \;=\; \sum_{\text{pairs } (j,\bar{j})} (c_{a,j} + c_{a,\bar{j}}) \;=\; 0.$$

The same argument gives $\sum_j c_{b,j} = 0$ and $\sum_j c_{ab,j} = 0$. The partial derivative identity is then immediate:

$$\sum_{j=0}^{15} \frac{\partial g_j}{\partial a}(a,b) = \Big(\sum_j c_{a,j}\Big) + b\Big(\sum_j c_{ab,j}\Big) = 0.$$

□

Attribution. Elementary from the complementation symmetry of Boolean functions; the closely related Fourier-basis zero-sum identity is in O'Donnell (2014). We do not claim it as a contribution. The novelty is its application in Proposition 2.

Proposition 2 (Soft-Mix backward gradient vanishes at uniform $\boldsymbol{\pi}$) Let $z(a,b) = \sum_{j=0}^{15} \pi_j\, g_j(a,b)$ for $\boldsymbol{\pi} \in \Delta^{15}$. If $\pi_j = 1/16$ for all j, then $\partial z/\partial a(a,b) = \partial z/\partial b(a,b) = 0$ for every $(a,b) \in [0,1]^2$. Hence for any scalar loss $\mathcal{L}$, $\partial \mathcal{L}/\partial a|_{\text{through } z} = 0$.

Proof. Substituting the multilinear form into z and differentiating,

$$\frac{\partial z}{\partial a}(a,b) = \sum_{j=0}^{15} \pi_j (c_{a,j} + c_{ab,j} b).$$

At $\pi_j = 1/16$, this equals $(1/16)[\sum_j c_{a,j} + b \sum_j c_{ab,j}] = 0$ by Lemma 1. This holds for every $(a,b) \in [0,1]^2$ because the algebraic cancellation is independent of a and holds for every fixed b. The $\partial z/\partial b$ statement follows by the identical computation with a, b swapped. The loss-gradient statement is the chain rule. □

Near-uniform regime. If $\boldsymbol{\pi}$ deviates from uniform $\mathbf{u} = (1/16, \ldots, 1/16)$, write $\pi_j = 1/16 + \delta_j$ with $\sum_j \delta_j = 0$. Then

$$\frac{\partial z}{\partial a} = \sum_j \delta_j (c_{a,j} + c_{ab,j} b),$$

since the $1/16$-term vanishes by Lemma 1. On Boolean inputs $b \in \{0,1\}$, every gate satisfies $|c_{a,j} + c_{ab,j} b| \le 1$ (verified by enumeration, Table 2). Hölder's inequality gives

$$\left|\frac{\partial z}{\partial a}\right| \;\le\; \|\boldsymbol{\pi} - \mathbf{u}\|_1,$$

so the backward signal grows only as $\boldsymbol{\pi}$ departs from uniformity. This will be important for Observation 1.

Lemma 3 (Uniform soft-VQ weights are unattainable). For the 16-Boolean codebook $\{\mathbf{G}_j\} \subset \mathbb{R}^4$ there is no $\mathbf{c} \in \mathbb{R}^4$ with $\omega_j(\mathbf{c}) = \mathrm{softmax}(-\|\mathbf{c} - \mathbf{G}_j\|^2/\tau)_j = 1/16$ for all j. Hence the soft-VQ weight map $\mathbf{c} \mapsto \boldsymbol{\omega}(\mathbf{c})$ never attains the uniform distribution $\mathbf{u}$ at which Proposition 2 forces $\partial z/\partial a = \partial z/\partial b = 0$.

Proof. The softmax is uniform iff all its logits are equal, i.e. $\|\mathbf{c} - \mathbf{G}_i\|^2 = \|\mathbf{c} - \mathbf{G}_j\|^2$ for all i, j. Expanding and cancelling the common $\|\mathbf{c}\|^2$ term, each equality is *linear* in $\mathbf{c}$: $2\,\mathbf{c}^\top(\mathbf{G}_j - \mathbf{G}_i) = \|\mathbf{G}_j\|^2 - \|\mathbf{G}_i\|^2$. Fixing $\mathbf{G}_0$ as reference gives the system $A\mathbf{c} = \mathbf{r}$ with rows $A_j = 2(\mathbf{G}_j - \mathbf{G}_0)$ and $r_j = \|\mathbf{G}_j\|^2 - \|\mathbf{G}_0\|^2$, $j = 1, \ldots, 15$. For the 16-Boolean codebook one computes $\mathrm{rank}(A) = 4$ while $\mathrm{rank}([A \mid \mathbf{r}]) = 5$; since the augmented rank exceeds the coefficient rank, by the Rouché–Capelli theorem the system is inconsistent and admits no solution $\mathbf{c}$.

Equivalently, a common equidistant center exists iff the 16 points $\{\mathbf{G}_j\}$ lie on a common sphere. The coefficients are the image $\mathbf{G}_j = M\,\mathbf{t}_j$ of the truth-table vertices $\mathbf{t}_j \in \{0,1\}^4$ under the multilinear map M with rows $(1,0,0,0)$, $(-1,0,1,0)$, $(-1,1,0,0)$, $(1,-1,-1,1)$ (rows give c_0, c_a, c_b, c_{ab}; columns index $f_{00}, f_{01}, f_{10}, f_{11}$). The vertices $\{0,1\}^4$ are cospherical, but $M^\top M \neq \lambda I$ (its diagonal is $(4,2,2,1)$ and its off-diagonal entries are nonzero), so M is not a similarity and does not preserve cosphericity; the images lie off any common sphere. □

Scope. The lemma removes the *exact* cancellation point from the soft-VQ parametrization, a structural contrast with Soft-Mix, whose free weights $\boldsymbol{\pi}$ can equal (and are initialized near) $\mathbf{u}$. It does not on its own bound $\partial z/\partial a$ away from zero: at large τ the weights are near-uniform (the softmax logits scale as $1/\tau$), so the input gradient is small but nonzero. That the signal is preserved, and stays preserved at depth, is the empirical content of §5.4 and §5.2.

Connection to Petersen et al. (2022). The forward-pass analogue of this identity is $z(a,b) = \frac{1}{16}\sum_j g_j(a,b) = 1/2$ for all (a,b), because exactly 8 of the 16 gates output 1 at each corner. Both cancellations share the same root: complementation symmetry.

Observation 1 (Dynamical depth attenuation). In a DLGN without residual connections, trained with Soft-Mix from near-uniform gate-logit initialization:

1. At initialization, every neuron has $\boldsymbol{\pi}^{(l)}$ near uniform, so by the near-uniform bound after Proposition 2, its input Jacobian $\partial z^{(l)}/\partial(a^{(l)}, b^{(l)})$ has small norm.
2. A layer l can commit (move its $\boldsymbol{\pi}^{(l)}$ away from uniform) only if it receives a non-trivial $\boldsymbol{\pi}$-gradient, which requires $\partial\mathcal{L}/\partial z^{(l)}$ to be non-trivial, which in turn requires layers $l+1, \ldots, L$ to already have non-trivial input Jacobians.
3. Hence commitment proceeds greedily from the output layer toward the input layer. At sufficient depth, early layers may never commit within a finite training budget.

Assumptions used. (a) No residual/skip connections; (b) near-uniform initialization of gate logits; (c) the analysis is dynamical, not a static worst-case bound; (d) the gradient path of interest is the loss-to-input path through layer input Jacobians.

What we do NOT claim. We do *not* claim a bound of the form $|\partial\mathcal{L}/\partial x| \leq C\varepsilon^{L-l}$ as a theorem, because the hypothesis "all intermediate layers are simultaneously near-uniform throughout training" is effectively the conclusion restated. Proposition 2 is the per-layer pointwise fact; Observation 1 is the dynamical interpretation; Section 5.2 is the empirical confirmation. Together they explain the depth penalty; none is a worst-case depth-decay theorem.

Single-gate-forward corollary (remark). The cancellation argument in Proposition 2 is specific to the forward expression $z = \sum_{j=0}^{15} \pi_j g_j(a,b)$: it is only because $\partial z/\partial a = \sum_j \pi_j(c_{a,j} + c_{ab,j}b)$ contains the Lemma 1 zero-sums inside a 16-term sum that those zero-sums can collapse the derivative to 0 at uniform $\boldsymbol{\pi}$. Any forward pass that commits to a *single* gate $j^\star$, writing $z = g_{j^\star}(a,b) = c_{0,j^\star} + c_{a,j^\star}a + c_{b,j^\star}b + c_{ab,j^\star}ab$, gives $\partial z/\partial a = c_{a,j^\star} + c_{ab,j^\star}b$, a derivative of a single committed polynomial with no sum in which to cancel; Proposition 4 then guarantees magnitude ≥ 1 for every a-dependent gate. This corollary applies to the methods whose forward commits to an *actual gate with integer coefficients*: (i) Multilinear-STE (hard L_2 snap; Proposition 4 guarantees magnitude ≥ 1) and (ii) Gumbel-ST (Gumbel noise + argmax samples a one-hot $\boldsymbol{\pi} = e_{j^\star}$ per forward, reducing to case (i) with a stochastic $j^\star$). It does *not* cover Multilinear-CovJac. Although the soft-VQ forward can be written as a single polynomial $z = \mathbf{c}_{\text{soft}}^\top \boldsymbol{\psi}(a,b)$ with $\mathbf{c}_{\text{soft}} = \sum_j \omega_j \mathbf{G}_j$, this equals the 16-term mixture $\sum_j \omega_j g_j(a,b)$, so Proposition 2 applies to it as well: at uniform $\boldsymbol{\omega}$ one has $\mathbf{c}_{\text{soft}} = (\frac{1}{2}, 0, 0, 0)$ and $\partial z/\partial a = \partial z/\partial b = 0$, exactly as for Soft-Mix. Multilinear-CovJac therefore does *not* escape the cancellation through its forward pass. It escapes through its *parameter* gradient (§A.4), aided by Lemma 3 (its $\boldsymbol{\omega}$ is never exactly uniform). The distinction from Soft-Mix is not the forward but the weights: Soft-Mix's $\boldsymbol{\pi}$ are *free* and stay near their uniform initialization, whereas Multilinear-CovJac's $\boldsymbol{\omega}$ are a function of the trained coefficients and are driven off uniformity by the covariance-Jacobian. Empirically the depth sweep (Appendix E.10) bears this out: at L=12, k=64k, Multilinear-STE (98.03%), Multilinear-CovJac (98.23%), and Gumbel-ST (97.81%) all remain above 97.8%, while Soft-Mix stalls at 84.92%. Gumbel-ST has the same 16-parameter count as Soft-Mix, so the dichotomy is not about parameter count.

A.4 Does the cancellation persist under soft vector quantization?

A natural objection, in the style of Rüttgers et al. (2026), is that soft vector quantization inherits the Soft-Mix cancellation: since $z = \sum_j \omega_j g_j(a,b)$ is a softmax-weighted mixture, expanding the softmax reproduces the same zero-sum structure and the input derivative $\partial z/\partial a$ vanishes at uniform weights. This is correct and we do not dispute it. As shown above, at $\boldsymbol{\omega} = \mathbf{u}$ the effective coefficient is $\mathbf{c}_{\text{soft}} = (\frac{1}{2}, 0, 0, 0)$ and $\partial z/\partial a = \partial z/\partial b = 0$: the *forward-path (input) gradient* of soft-VQ cancels at the symmetric point exactly as Soft-Mix's does. Multilinear-CovJac does not claim otherwise.

The contribution is about a different gradient. Training updates the coefficients $\mathbf{c}$, and the signal reaching them is the *parameter* gradient $\partial\mathcal{L}/\partial\mathbf{c} = \delta\,\mathbf{J}^\top\boldsymbol{\psi}(a,b)$, where $\mathbf{J} = \partial\mathbf{c}_{\text{soft}}/\partial\mathbf{c} = (2/\tau)\,\mathrm{Cov}_{\boldsymbol{\omega}}(\mathbf{G})$ is the covariance Jacobian (Proposition 5). The component feeding the interaction coefficient is $\partial\mathcal{L}/\partial c_{ab} = \delta\,(J_{03} + aJ_{13} + bJ_{23} + ab\,J_{33})$, and at the symmetric point $J_{03} = (2/\tau)\,\mathrm{Cov}_{\boldsymbol{\omega}}(G_{\cdot 0}, G_{\cdot 3}) = 0.5$ (at $\tau = 1$), with the full prefactor equal to 0.5 on all four Boolean corners. Thus *exactly where the forward gradient vanishes, the parameter gradient to c_{ab} does not*: the two are different statistics of the codebook (a first moment versus a second moment), and complementation symmetry annihilates the first while leaving the second intact (§4). A nonzero

Table 2: STE backward derivative $\partial z/\partial a = c_a + c_{ab}b$ for all 12 a-dependent Boolean gates. Every row satisfies $\max_{b\in\{0,1\}} |c_a + c_{ab}b| \geq 1$.

Gate	c_a	c_{ab}	at b=0	at b=1
a	1	0	1	1
$\neg a$	-1	0	-1	-1
AND ab	0	1	0	1
NAND $1-ab$	0	-1	0	-1
OR $a+b-ab$	1	-1	1	0
NOR $1-a-b+ab$	-1	1	-1	0
XOR $a+b-2ab$	1	-2	1	-1
XNOR $1-a-b+2ab$	-1	2	-1	1
$a \wedge \neg b$	1	-1	1	0
$\neg a \wedge b$	0	-1	0	-1
$a \vee \neg b$	0	1	0	1
$\neg a \vee b$	-1	1	-1	0

$\partial\mathcal{L}/\partial c_{ab}$ moves $\mathbf{c}$, which moves $\boldsymbol{\omega}$ off uniformity, which by the near-uniform bound restores a nonzero *forward* gradient. The symmetric point is a fixed point of the forward map, not of training.

Soft-Mix cannot use this escape. Its free weights $\boldsymbol{\pi}$ are updated by a gradient confined to the rank-4 image of the codebook (11 of 15 simplex directions carry no output gradient; §3), so the useful directions do not drive $\boldsymbol{\pi}$ to concentrate; empirically $\boldsymbol{\pi}$ stays near uniform and the forward gradient stays cancelled, which is the depth collapse. Figure 3 measures this on CIFAR-10: over training, $\mathrm{std}(c_{ab})$ across neurons reaches 5.9 for Multilinear-CovJac (it commits) but only 1.1 for Soft-Mix (it never commits), and the c_{ab} gradient-norm ratio reaches 14.27 for CovJac versus 0.42 for STE. At depth the consequence is unambiguous (Appendix E.10): at L=12 on CIFAR-10, Multilinear-CovJac holds at 58.44% while Soft-Mix falls to 20.79%. The cancellation is thus real and conceded at the level of the forward pass; what removes it during training is the parameter gradient, and the depth behavior is the measurement that decides the question.

A.5 Multilinear-STE and CovJac Eliminate Cancellation (Propositions 4, 5)

Proposition 4 (Multilinear-STE backward non-cancellation) In Multilinear-STE, the forward pass at a neuron with committed gate $\hat{\mathbf{c}} = (c_0, c_a, c_b, c_{ab})$ is $z = \hat{\mathbf{c}}^\top \boldsymbol{\psi}(a, b)$, and the STE backward derivative is $\partial z/\partial a = c_a + c_{ab}\, b$. For every non-trivial Boolean gate depending on input a (i.e. every Boolean gate with $(c_a, c_{ab}) \neq (0, 0)$), $\max_{b\in\{0,1\}} |c_a + c_{ab}\, b| \;\geq\; 1$.

Proof. The STE rule treats $\hat{\mathbf{c}}$ as if it were the free parameter for the backward pass, so $\partial z/\partial a = c_a + c_{ab}\, b$. Since $\hat{\mathbf{c}}$ is a Boolean-gate coefficient vector, Identity 1 gives $c_a \in \{-1, 0, 1\}$ and $c_{ab} \in \{-2, -1, 0, 1, 2\}$. A Boolean gate depends on a iff $c_a \neq 0$ or $c_{ab} \neq 0$.

Case 1: $c_a \neq 0$**.** At $b = 0$, $|c_a + c_{ab} \cdot 0| = |c_a| \geq 1$.

Case 2: $c_a = 0$ **and** $c_{ab} \neq 0$**.** At $b = 1$, $|c_a + c_{ab} \cdot 1| = |c_{ab}| \geq 1$.

These two cases exhaust all gates depending on a. □

Enumeration. For completeness, here are (c_a, c_{ab}) and $(\partial z/\partial a|_{b=0},\ \partial z/\partial a|_{b=1})$ for the 12 Boolean gates that depend on a:

Every row has $\max_b |\cdot| \geq 1$, confirming the bound.

Key structural contrast with Soft-Mix. The Multilinear-STE backward derivative is the derivative of *one* committed polynomial, not a convex combination of 16 gate derivatives. The zero-sum of Lemma 1 therefore has no avenue to manifest. The per-layer signal magnitude is $O(1)$, and a depth-L product of such factors is $O(1)^L = O(1)$, not exponentially decaying.

Pre-commitment remark. During training, the learned $\mathbf{c}$ is continuous and may lie near a Voronoi boundary, in which case the STE derivative $c_a + c_{ab}b$ (using the *continuous* c_a, c_{ab}) can be small. The content of Proposition 4 is structural: unlike Soft-Mix, *structural cancellation across gates*

cannot occur, so any smallness of the backward signal is a property of a single neuron's continuous parameter state, not a systematic symmetry that affects all neurons at initialization.

Proposition 5 (CovJac selection Jacobian: input-independence, PSD, positive diagonal) For the soft-selection map $\mathbf{c} \mapsto \mathbf{c}_{\text{soft}}(\mathbf{c}) = \sum_j \omega_j(\mathbf{c})\mathbf{G}_j$, with $\omega_j(\mathbf{c}) = \text{softmax}(-\|\mathbf{c} - \mathbf{G}_j\|^2/\tau)_j$, the Jacobian $J_{ik}(\mathbf{c}) = \partial(\mathbf{c}_{\text{soft}})_i/\partial c_k$ equals

$$J_{ik} = \frac{2}{\tau}\,\text{Cov}_\omega(G_{\cdot i},\, G_{\cdot k}) = \frac{2}{\tau}\Big(\sum_j \omega_j G_{ji} G_{jk} - \big(\sum_j \omega_j G_{ji}\big)\big(\sum_j \omega_j G_{jk}\big)\Big),$$

and satisfies (i) input-independence, (ii) symmetric PSD, (iii) strictly positive diagonal for every coordinate at $\tau{>}0$ on the 16-Boolean codebook.

Proof. Throughout this proof, ω_j denotes the CovJac proximity weights from Eq. 4.

(a) Derivation of J. Let $d_j(\mathbf{c}) = \|\mathbf{c} - \mathbf{G}_j\|^2$. Then $\omega_j = e^{-d_j/\tau}/Z$ where $Z = \sum_\ell e^{-d_\ell/\tau}$. Compute $\partial\omega_j/\partial c_k$ via the softmax chain rule. First, $\partial d_j/\partial c_k = 2(c_k - G_{jk})$. By the standard softmax derivative ($\partial\omega_j/\partial l_m = \omega_j(\delta_{jm} - \omega_m)$ applied to logits $l_j = -d_j/\tau$ via the chain rule),

$$\frac{\partial \omega_j}{\partial c_k} = -\frac{1}{\tau}\omega_j\Big(\frac{\partial d_j}{\partial c_k} - \sum_\ell \omega_\ell \frac{\partial d_\ell}{\partial c_k}\Big).$$

Substituting $\partial d_j/\partial c_k = 2(c_k - G_{jk})$:

$$\begin{aligned}
\frac{\partial \omega_j}{\partial c_k} &= -\frac{2}{\tau}\omega_j\Big[(c_k - G_{jk}) - \sum_\ell \omega_\ell (c_k - G_{\ell k})\Big] \\
&= -\frac{2}{\tau}\omega_j\Big[\cancel{c_k} - G_{jk} - \cancel{c_k}\underbrace{\textstyle\sum_\ell \omega_\ell}_{=1} + \underbrace{\textstyle\sum_\ell \omega_\ell G_{\ell k}}_{=(\mathbf{c}_{\text{soft}})_k}\Big] \\
&= \frac{2}{\tau}\omega_j\big(G_{jk} - (\mathbf{c}_{\text{soft}})_k\big). \qquad (*)
\end{aligned}$$

Now differentiate $(\mathbf{c}_{\text{soft}})_i = \sum_j \omega_j G_{ji}$ by c_k:

$$J_{ik} = \sum_j \frac{\partial \omega_j}{\partial c_k} G_{ji} \overset{(*)}{=} \frac{2}{\tau}\sum_j \omega_j\big(G_{jk} - (\mathbf{c}_{\text{soft}})_k\big)G_{ji}.$$

Expanding $G_{jk} - (\mathbf{c}_{\text{soft}})_k = G_{jk} - \sum_\ell \omega_\ell G_{\ell k}$ and distributing:

$$\begin{aligned}
J_{ik} &= \frac{2}{\tau}\Big(\sum_j \omega_j G_{ji} G_{jk} - \underbrace{(\mathbf{c}_{\text{soft}})_k}_{\sum_\ell \omega_\ell G_{\ell k}} \sum_j \omega_j G_{ji}\Big) \\
&= \frac{2}{\tau}\Big(\mathbb{E}_\omega[G_{\cdot i}G_{\cdot k}] - \mathbb{E}_\omega[G_{\cdot i}]\,\mathbb{E}_\omega[G_{\cdot k}]\Big) = \frac{2}{\tau}\text{Cov}_\omega(G_{\cdot i}, G_{\cdot k}).
\end{aligned}$$

(b) Input-independence. Both $\mathbf{G}$ (a fixed codebook) and $\boldsymbol{\omega}(\mathbf{c})$ (a function of $\mathbf{c}$ alone) are independent of (a, b), so $\mathbf{J}(\mathbf{c})$ is input-independent.

(c) PSD. For any $\mathbf{v} \in \mathbb{R}^4$, $\mathbf{v}^\top \mathbf{J}\mathbf{v} = (2/\tau)\text{Var}_\omega(\sum_i v_i G_{\cdot i}) \geq 0$. Symmetry $J_{ik} = J_{ki}$ is immediate.

(d) Positive diagonal. $J_{kk} = (2/\tau)\text{Var}_\omega(G_{\cdot k}) > 0$ iff $G_{\cdot k}$ is not $\boldsymbol{\omega}$-almost-surely constant. For each coordinate $k \in \{0, 1, 2, 3\}$ on the 16-Boolean codebook, $G_{\cdot k}$ takes at least two distinct values, and softmax at finite $\tau{>}0$ assigns $\omega_j > 0$ for every j. Therefore $J_{kk} > 0$ throughout training. □

Gradient to c_{ab} on every sample. The parameter gradient through the soft coefficient is $\partial\mathcal{L}/\partial c_k = (\partial\mathcal{L}/\partial z) \cdot \sum_i \psi_i(a, b)\, J_{ik}$. For $k = 3$:

$$\frac{\partial \mathcal{L}}{\partial c_{ab}} = \frac{\partial \mathcal{L}}{\partial z} \cdot \big(1 \cdot J_{03} + a \cdot J_{13} + b \cdot J_{23} + ab \cdot J_{33}\big).$$

The leading term involves the constant basis element $\psi_0 = 1$, which is non-zero on every input. Since J_{03} is input-independent and generically non-zero, c_{ab} receives a non-zero gradient on *every* training sample, in contrast to Multilinear-STE where $\partial\mathcal{L}/\partial c_{ab} = \delta \cdot ab$ vanishes on 75% of Boolean inputs.

A.6 Basis Optimality (Proposition 7)

Basis optimality: full statement and proof. *Goal.* An ideal STE basis for the interaction coefficient should satisfy three properties simultaneously: **(a)** full *coverage*: the gradient fires on every training sample (ρ=1); **(b)** *coherence*: the gradient consistently points in the correct direction (κ=1); **(c)** zero *STE bias*: the basis is equivalent to the canonical one up to positive rescaling (B=0). We show no affine product basis achieves all three.

Proposition 7 (No affine product basis simultaneously fixes STE)**.** *Let $\phi(x) = \alpha + \beta x$ with $\beta \neq 0$, and let $\tilde{\boldsymbol{\psi}}(a,b) = (1,\ \phi(a),\ \phi(b),\ \phi(a)\phi(b))^\top$. Under uniform $(a,b) \in \{0,1\}^2$, define the following three properties of the interaction component $\tilde{\psi}_3(a,b) = \phi(a)\phi(b)$ (the 4th entry of $\tilde{\boldsymbol{\psi}}$, corresponding to c_{ab}):* coverage $\rho := \Pr[\tilde{\psi}_3(a,b) \neq 0]$*;* coherence $\kappa := |\mathbb{E}[\tilde{\psi}_3]|/\mathbb{E}[|\tilde{\psi}_3|] \in [0,1]$*;* STE bias $B := \inf_D \mathbb{E}[\|\tilde{\boldsymbol{\psi}} - D\boldsymbol{\psi}_{\text{can}}\|^2]$ *where D ranges over positive diagonal matrices. Then: (i) If $\phi(0) = 0$ or $\phi(1) = 0$: $\rho = 1/4$, $\kappa = 1$, $B = 0$. (ii) If $\phi(0) \neq 0$ and $\phi(1) \neq 0$: $\rho = 1$, but either $\kappa < 1$ or $B > 0$ (or both). (iii) No affine product basis simultaneously attains $\rho = 1$, $\kappa = 1$, and $B = 0$.*

Proof. Under STE with Boolean inputs $(a,b) \in \{0,1\}^2$, the gradient $\partial\mathcal{L}/\partial c_k$ is scaled by $\tilde{\psi}_k(a,b)$, so the properties ρ, κ, B depend only on the four corner values of $\tilde{\psi}_3$. Since $\phi(x) = \alpha + \beta x$, define $p := \phi(0) = \alpha$ and $q := \phi(1) = \alpha + \beta$ (so $p \neq q$ since $\beta \neq 0$). The interaction basis function is $\tilde{\psi}_3(a,b) = \phi(a)\phi(b)$, which at the four corners evaluates to: $\tilde{\psi}_3(0,0) = \phi(0)\phi(0) = p^2$, $\tilde{\psi}_3(0,1) = \phi(0)\phi(1) = pq$, $\tilde{\psi}_3(1,0) = \phi(1)\phi(0) = pq$, $\tilde{\psi}_3(1,1) = \phi(1)\phi(1) = q^2$.

Part (i): $p = 0$ **(i.e.** $\alpha = 0$**,** $\phi(x) = \beta x$**,** $\beta > 0$**).** Substituting $p = 0$: the corner values become $(0,0,0,q^2) = (0,0,0,\beta^2)$. Three of four corners are zero, so $\rho = \Pr[\tilde{\psi}_3 \neq 0] = 1/4$. The one non-zero value is $\beta^2 > 0$, so $|\mathbb{E}[\tilde{\psi}_3]| = \beta^2/4$ and $\mathbb{E}[|\tilde{\psi}_3|] = \beta^2/4$, giving $\kappa = 1$. The full basis evaluates to $\tilde{\boldsymbol{\psi}}(a,b) = (1,\ \phi(a),\ \phi(b),\ \phi(a)\phi(b)) = (1,\ \beta a,\ \beta b,\ \beta^2 ab) = D\boldsymbol{\psi}_{\text{can}}$ with $D = \text{diag}(1,\beta,\beta,\beta^2)$, a positive diagonal matrix (since $\beta > 0$). Hence $B = 0$.

Part (i'): $q = 0$ **(i.e.** $\phi(1) = 0$**, so** $\alpha + \beta = 0$**,** $\phi(x) = \alpha(1-x)$**).** The corner values become $(p^2,0,0,0) = (\alpha^2,0,0,0)$, so $\rho = 1/4$ and $\kappa = 1$ (same reasoning as Part (i)). However, $\tilde{\psi}_1 = \phi(a) = \alpha - \alpha a = \alpha(1-a)$. For $B = 0$ we need $\tilde{\psi}_1 = d_1 \cdot a$ for some $d_1 > 0$, but $\alpha(1-a)$ has value α at $a = 0$ and 0 at $a = 1$, which is not proportional to $a = (0,1)$. Hence $B > 0$.

Part (ii): $p \neq 0$ **and** $q \neq 0$**.** All four corner values (p^2, pq, pq, q^2) are non-zero (since neither p nor q is zero), so $\rho = \Pr[\tilde{\psi}_3 \neq 0] = 1$.

Sub-case (ii-a): $pq > 0$ *(same sign).* All four values $p^2, pq, pq, q^2 > 0$, so $|\mathbb{E}[\tilde{\psi}_3]| = \mathbb{E}[|\tilde{\psi}_3|]$ and $\kappa = 1$. However, expanding $\phi(a)\phi(b) = (\alpha + \beta a)(\alpha + \beta b) = \alpha^2 + \alpha\beta(a+b) + \beta^2 ab = p^2 + p\beta(a+b) + \beta^2 ab$, the interaction component $\tilde{\psi}_3$ contains a constant term p^2 and linear terms $p\beta \cdot a$, $p\beta \cdot b$. For $\tilde{\boldsymbol{\psi}} = D\boldsymbol{\psi}_{\text{can}}$ we need $\tilde{\psi}_1 = d_1 \cdot a$; but $\tilde{\psi}_1 = \phi(a) = p + \beta a$ evaluates to p at $a = 0$ and $p + \beta$ at $a = 1$. Since $p \neq 0$, this is not proportional to $a = (0,1)$ for any $d_1 > 0$. Hence $B > 0$.

Sub-case (ii-b): $pq < 0$ *(opposite signs).* The four corner values (p^2, pq, pq, q^2) have mixed signs: $p^2, q^2 > 0$ but $pq < 0$. Under uniform $(a,b) \in \{0,1\}^2$:

$$\mathbb{E}[\tilde{\psi}_3] = \tfrac{1}{4}(p^2 + pq + pq + q^2) = \tfrac{1}{4}(p^2 + 2pq + q^2) = \tfrac{(p+q)^2}{4},$$
$$\mathbb{E}[|\tilde{\psi}_3|] = \tfrac{1}{4}(p^2 + |pq| + |pq| + q^2) = \tfrac{1}{4}(p^2 - 2pq + q^2) = \tfrac{(p-q)^2}{4} = \tfrac{\beta^2}{4},$$

where the last step uses $p - q = \alpha - (\alpha + \beta) = -\beta$. Therefore $\kappa = |\mathbb{E}[\tilde{\psi}_3]|/\mathbb{E}[|\tilde{\psi}_3|] = (p+q)^2/(p-q)^2 = (2\alpha + \beta)^2/\beta^2$. The Walsh/Hadamard basis maps $\{0,1\}$ to symmetric $\pm$ values: $\phi(0) = -\phi(1)$, i.e. $p = -q$. In our parameterization this requires $\alpha = -(\alpha + \beta)$, giving $\alpha = -\beta/2$. At this choice, $p + q = 0$, so $\kappa = 0$: exact cancellation. Since $p \neq 0$ (because $pq < 0$), the same diagonal-rescaling argument as (ii-a) gives $B > 0$: $\tilde{\psi}_1 = p + \beta a$ cannot equal $d_1 a$ for any $d_1 > 0$.

Part (iii). Combining all cases: $B = 0$ requires $p = 0$ (Part (i)), which gives $\rho = 1/4$. Every basis with $\rho = 1$ (Parts (ii-a) and (ii-b)) has $B > 0$. Therefore no affine product basis achieves $(\rho, \kappa, B) = (1,1,0)$ simultaneously. □

Table 3: Monomial degree as logical complexity for $n = 2$ inputs.

Degree	Term	Logical meaning	Example gates
0	c_0	Unconditional output	FALSE, TRUE
1	$c_a a$	Independent input-a effect	A, ¬A
1	$c_b b$	Independent input-b effect	B, ¬B
2	$c_{ab} ab$	Pairwise interaction	AND, XOR, NAND

Empirical confirmation. The Walsh basis ($\alpha = -\beta/2$, sub-case (ii-b) with $\kappa = 0$) achieves 86.86% vs canonical 96.65%; smoothed $\epsilon = 0.2$ (sub-case (ii-a), $B > 0$) achieves 76.95%; expected-value and uniform bases (degenerate cases with $\beta = 0$) achieve 13.30% and 12.42%. See Appendix D.

Interpretation. The 25% interaction-coverage "starvation" of Multilinear-STE is a fundamental property of every affine product basis, not a deficiency fixable by reparameterization. The correct response is to bypass basis-dependent gradients via Multilinear-CovJac (§4.3).

Scope. Proposition 7 restricts to affine product bases. Non-affine and non-product bases are outside our scope; we flag them as an open question.

B Background: Fuzzy Logic, WMC, and SPN View

B.1 Degree as logical complexity

The table below relates each monomial degree to its logical meaning. For $n = 2$, maximum degree 2 forces exactly 4 terms and exactly 4 parameters; the 4-parameter design of the Multilinear layer is *determined* by the MLE structure, not chosen (Table 3).

B.2 Fuzzy logic and the product t-norm

Under the product t-norm (Klement et al., 2000) (probabilistic fuzzy logic), fuzzy connective evaluation is *identical* to MLE evaluation:

$$a \wedge b = ab, \qquad a \vee b = a + b - ab, \qquad \neg a = 1 - a, \qquad a \oplus b = a + b - 2ab.$$

These are not approximations. The MLE *is* the fuzzy truth-degree function under the product t-norm (van Krieken et al., 2022). Łukasiewicz ($\min$) and Gödel ($\max$) fuzzy logics yield piecewise-linear extensions instead.

B.3 Probabilistic circuit and WMC interpretation

If inputs are independent, $X \sim \mathrm{Ber}(a)$ and $Y \sim \mathrm{Ber}(b)$, then

$$F(a, b) = \Pr[g(X, Y) = 1 \mid X \sim \mathrm{Ber}(a),\, Y \sim \mathrm{Ber}(b)].$$

This is the *Weighted Model Count* (WMC) (De Raedt et al., 2007) of gate g under independent Bernoulli atoms, the same computation used in ProbLog and probabilistic circuits (Choi et al., 2020). For a single 2-input gate, WMC and MLE evaluation are the same operation.

B.4 Corner basis as a sum-product network

The corner basis functions $\phi_{ij}(a, b) = a^i(1-a)^{1-i}\, b^j(1-b)^{1-j}$ used by IWP (Rüttgers et al., 2026) are exactly the joint probability mass function of two independent Bernoullis. Evaluating the IWP forward pass $z = \sum_{ij} s_{ij}\, \phi_{ij}(a, b)$ is equivalent to running a shallow sum-product network with 4 product nodes (one per corner) and a single sum node. The multilinear basis $\{1, a, b, ab\}$ and the corner basis span the same space and are related by the invertible linear map of §3 / Appendix C.

Table 4: Summary of all methods compared. Params = learnable parameters per neuron. DG = discretization gap (structural, not empirical).

Method	Params	Train fwd	Eval fwd	Backward	DG
Soft-Mix (Petersen et al., 2022)	16	soft mixture	argmax	autograd	> 0
Gumbel-ST (Jang et al., 2017)	16	hard (Gumbel)	argmax	Gumbel-ST	small
IWP (Rüttgers et al., 2026)	4	bilinear	bilinear	exact	> 0
IWP-STE (ablation)	4	snap	snap	corner STE	0
Multilinear-STE (ours)	**4**	snap	snap	poly STE	**0**
Multilinear-CovJac (ours)	**4**	soft-VQ	snap	CovJac	**0** (emp.)

Table 5: Per-neuron computational cost. All methods deploy the same hard Boolean gate.

Method	Params/neuron	Train (ops/sample)	Deploy (ops/sample)	Memory (floats)
Soft-Mix	16	$\sim$175	7	48
Gumbel-ST	16	$\sim$175	7	48
M-STE (ours)	4	7	7	12
M-CovJac (ours)	4	7	7	12

All methods have small batch-independent overhead ($<$0.4 ops/sample at $B{=}512$; see §C.3 summary).

B.5 Unifying principle

The three views (symbolic Boolean logic, probabilistic WMC, and differentiable Multilinear learning) compute the same quantity under different conventions:

$$\{0,1\} \text{ inputs (Boolean logic)} \longleftrightarrow [0,1] \text{ inputs (WMC / Bayes net)} \longleftrightarrow \text{gradient flows (Multilinear layer).}$$

The Multilinear layer (instantiated as Multilinear-STE or Multilinear-CovJac; §3) makes this explicit: it directly parameterizes the MLE coefficients and snaps to the nearest Boolean gate, so gradient-based training and Boolean inference are unified within a single algebraic object.

C Computational Cost and Corner-Polynomial Bijection

C.1 Method comparison

Table 4 summarizes all methods compared in this paper, including parameter counts, forward/backward semantics, and discretization-gap properties.

C.2 Per-neuron cost summary

Table 5 compares the per-neuron computational cost of all four methods. At deployment, all methods evaluate a single hard Boolean gate ($\sim$7 ops: compute ab, multiply by 4 coefficients, sum). The training cost differs: Soft-Mix and Gumbel-ST evaluate all 16 gate outputs and compute a softmax mixture (174 ops); M-STE evaluates only the snapped polynomial (7 ops, the same as deployment); M-CovJac computes the soft-VQ proximity weights over 16 codebook entries (174 ops, similar to Soft-Mix). The $4\times$ parameter reduction (16$\to$4 per neuron) reduces the Adam optimizer state from 48 to 12 floats per neuron (4 parameters $\times$ 3: value + first-moment + second-moment).

C.3 Computational cost: full op counts

Per-sample forward (Soft-Mix). Soft-Mix evaluates all 16 gate polynomials per sample:

- Softmax over 16 logits: 16 exp + 15 add + 16 div = 47 ops.
- Compute ab: 1 multiply (shared across all 16 gates).
- 16 gate evaluations $g_j(a,b) = c_{0,j} + c_{a,j}a + c_{b,j}b + c_{ab,j}ab$: each needs 3 multiply ($c_{a,j}a$, $c_{b,j}b$, $c_{ab,j} \cdot ab$) + 3 add = 6 ops; $16 \times 6 + 1 = 97$ ops.
- Weighted sum $z = \sum_j \pi_j g_j$: 16 multiply + 15 add = 31 ops.

- **Total:** $47 + 97 + 31 = 175$ **ops/sample/neuron.**

Backward: $\partial\mathcal{L}/\partial\pi_j = \delta \cdot g_j(a, b)$ for each j, plus softmax Jacobian: ≈ 100 ops/neuron.

Per-sample forward (Gumbel-ST). Same 16-gate evaluation as Soft-Mix, plus Gumbel noise:

- Sample 16 Gumbel variables ($-\log(-\log u)$): $16 \times 3 = 48$ ops.
- Add to logits and take argmax: $16 + 15 = 31$ ops.
- Evaluate the selected gate polynomial: 7 ops.
- **Total:** $48 + 31 + 7 = 86$ **ops (hard forward);** $\sim$**174 ops with soft backward (Gumbel-softmax).**

The hard forward is cheaper than Soft-Mix (one gate, not 16), but the Gumbel-softmax backward still requires all 16 gate values, bringing the effective training cost to $\sim$174 ops.

Per-sample forward (Multilinear-STE). Evaluates *one* committed polynomial per sample:

- $z = c_0 + c_a a + c_b b + c_{ab} ab$: 1 multiply ($ab$) + 3 multiply ($c_k \cdot \text{basis}_k$) + 3 add = 7 ops/sample/neuron.
- Gate selection (snap) is computed once per layer, amortized across the batch (§C.3 below).

Backward STE: $\nabla_{\mathbf{c}}\mathcal{L} = \delta \cdot [1, a, b, ab]^\top = [\delta,\ \delta a,\ \delta b,\ \delta \cdot ab]$: 3 multiply ($\delta$ is free for c_0) = 3 ops/neuron.

Per-sample forward (Multilinear-CovJac). The soft-VQ weights ω_j and the soft coefficient $\mathbf{c}_{\text{soft}} = \sum_j \omega_j \mathbf{G}_j$ depend only on $\mathbf{c}$ and the fixed codebook $\mathbf{G}$, not on the input (a, b). They are therefore computed **once per neuron per batch**: distances $\|\mathbf{c} - \mathbf{G}_j\|^2$ for 16 gates ($\sim$120 ops, using $\|\mathbf{c}\|^2 - 2\mathbf{c}^\top \mathbf{G}_j + \|\mathbf{G}_j\|^2$ with $\|\mathbf{G}_j\|^2$ precomputed), softmax (47 ops), weighted sum ($\sim$30 ops); total batch overhead $\sim$200 ops/neuron. Per sample, each neuron evaluates $z = \mathbf{c}_{\text{soft}}^\top \boldsymbol{\psi}(a, b) = \sim$7 ops (same as STE). Amortized over batch B=512: $200/512 \approx 0.4$ extra ops/sample, giving an effective per-sample cost of $\sim$7.4 ops. The CovJac backward adds a 4×4 Jacobian–vector product ($\sim$28 ops/neuron, also batch-independent). At deployment, CovJac uses the same hard gate as STE ($\sim$7 ops).

Summary. Per sample at training: STE $\sim$7 ops; CovJac $\sim$7 ops; Soft-Mix $\sim$175 ops. STE and CovJac are both $\sim 25\times$ cheaper than Soft-Mix per sample. All methods also incur a small batch-independent overhead per neuron: M-STE $\sim$120 ops (snap distances $\|\mathbf{c} - \mathbf{G}_j\|^2$), M-CovJac $\sim$200 ops (distances + softmax + weighted sum), Soft-Mix $\sim$47 ops (softmax on logits). At B=512 these amortize to $<$0.4 ops/sample for all methods. At deployment all methods evaluate a single hard gate: $\sim$7 ops.

Layer-level L_2 snap. The L_2 snap is computed *once per layer* and shared across the batch, so its $O(N \cdot 64)$ cost is batch-independent (approximately 5% of total at B=512). Using the identity

$$\|\mathbf{c} - \mathbf{G}_j\|^2 = \|\mathbf{c}\|^2 - 2\mathbf{c}\mathbf{G}^\top + \|\mathbf{G}_j\|^2,$$

the snap reduces to a single cuBLAS matmul.

Optimizer state. Adam stores per parameter: value + first-moment + second-moment = 3 floats. Multilinear: $4 \times 3 = 12$ floats/neuron. Soft-Mix: $16 \times 3 = 48$ floats/neuron. $4\times$ **reduction in optimizer memory.**

Table 6: STE basis failure catalogue on MNIST ($k = 64$k, $L = 6$, 3 seeds unless noted). **Bold** = reference (canonical basis).

Basis	Description	Acc (%)	Interpretation
Canonical $(1, a, b, ab)$	Multilinear-STE (reference)	**98.11 ± 0.03**	$(\rho, \kappa, B) = (1/4, 1, 0)$, Part (i)
Walsh $\{-1, +1\}$: $\phi(x)=2x-1$	$\alpha=-1, \beta=2$, centered	94.92 ± 0.01	Sub-case (ii-b), $p=-q$, $\kappa=0$, coherence cancellation
Smoothed $\epsilon=0.2$: $(1, a+\epsilon, b+\epsilon, (a+\epsilon)(b+\epsilon))$	$\phi(x)=x+0.2$	10.14 ± 2.63	Sub-case (ii-a), $p, q>0$, $B>0$, collapses at $k=64$k
Expected-value $(1, 0.5, 0.5, 0.25)$	$\beta=0$; basis is constant	13.30*	Degenerate, $\beta=0$ excluded in Prop. 7
Uniform $(1, 1, 1, 1)$	$\alpha=1, \beta=0$	12.42*	Degenerate

C.4 Corner–polynomial bijection (explicit matrices)

Proposition (Bijection). The multilinear parameterization $\mathbf{c} = [c_0, c_a, c_b, c_{ab}]$ and the corner parameterization $\mathbf{s} = [s_{00}, s_{10}, s_{01}, s_{11}]$ are related by the invertible linear map $\mathbf{c} = M\mathbf{s}$ where

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 1 & -1 & -1 & 1 \end{pmatrix}, \qquad M^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

Both parameterizations span the same function space and represent the same 16 Boolean gates. Consequences: (a) the 17pp IWP-STE vs Multilinear-STE accuracy gap at k=64k (§5.2) cannot arise from function class or expressive capacity; it must arise from optimization dynamics, specifically the STE gradient structure (verified by the 2×2 experiment in Appendix F and by Lemma 6 in Appendix A.2).

C.5 Voronoi geometry of the L_2 snap

The Voronoi cells of the 16 gate vectors are convex polyhedra in $\mathbb{R}^4$; nearest-gate distances between pairs of gates range from 1 (adjacent gates, e.g. AND to $a \wedge \neg b$) to $\sqrt{11}$ (XOR $(0, 1, 1, -2)$ to AND $(0, 0, 0, 1)$). No geometric uniformity is required: what matters is that every boundary of the continuous $\mathbf{c}$ space lies in the interior of some Voronoi cell, so the snap is well-defined almost everywhere.

D STE Basis Failure Catalogue

D.1 Alternative STE bases tested

All experiments below use the Multilinear-STE forward pass (L_2 snap to the nearest Boolean gate) and swap only the backward basis $\tilde{\boldsymbol{\psi}}(a, b)$ used in the STE gradient $\delta \cdot \tilde{\boldsymbol{\psi}}(a, b)$. Results at k=64k, L=6, 3 seeds, last-10 hard accuracy (Table 6).

*Single seed, k=8k (degenerate bases collapse to near-random regardless of scale; re-run at k=64k not informative).

D.2 Walsh cancellation is the cleanest predicted failure

Setting $\alpha = -\beta/2$ (Walsh basis) gives $p = -q$, so the corner values of $\tilde{\psi}_3 = \phi(a)\phi(b)$ are $(p^2, -p^2, -p^2, p^2)$. The expectation under uniform Boolean (a, b) is $\mathbb{E}[\tilde{\psi}_3] = (1/4)(p^2 - p^2 - p^2 + p^2) = 0$, producing *exact* coherence cancellation $\kappa = 0$. Proposition 7 predicts this basis cannot deliver a coherent c_{ab} update; the 94.92% accuracy (3.2pp below canonical at k=64k) is the empirical consequence.

D.3 Smoothed basis fails for a different reason

With $\phi(x) = x + \epsilon$, ϵ=0.2, we have p=0.2, q=1.2>0, so κ=1 but the basis components differ from canonical: specifically, $\tilde{\psi}_3 = (0.2+a)(0.2+b) = 0.04 + 0.2(a+b) + ab$ contains a non-removable constant and linear term, making $B>0$ in the Proposition 7 sense. The 10.14% accuracy at k=64k (near

random chance on 10-class MNIST) is a catastrophic failure (far worse than Walsh), confirming that STE bias ($B>0$) is more destructive than coherence loss ($\kappa=0$).

D.4 Degenerate $\beta = 0$ bases

The expected-value and uniform bases have $\beta=0$, meaning $\tilde{\psi}_k$ no longer depends on (a, b) in the linear coordinates. These are outside the scope of Proposition 7 (which assumes $\beta \neq 0$), but we include them for completeness. Both collapse to random-guess-level accuracy (12–13% on 10-class MNIST-bin), confirming that non-informative backward bases simply stop learning. They are instructive negative controls but not interesting counterexamples.

E Experimental Details and Additional Results

This appendix contains: training protocol and hyperparameters (§E.1); binarization analysis and architecture baselines (§E.1); discretization gap (§E.2); gate-entropy diagnostic (§E.3); width scaling (§E.4); CIFAR-10 full table (§E.5); scaling to larger datasets (§E.6); gradient-norm diagnostic (§E.7); interaction-task scaling (§E.9); depth scaling (§E.10); gate codebook distribution (§E.12); and temperature sensitivity (§E.13).

E.1 Hyperparameters and training protocol

All methods are trained with Adam (lr $= 0.01$, $\beta_1 = 0.9$, $\beta_2 = 0.999$) for 50,000 iterations with batch size 512. Gate logits are initialized i.i.d. $\mathcal{N}(0, \sigma^2)$ with $\sigma = 1.0$ (the standard deviation of the initialising normal distribution for each 16-dim Soft-Mix logit vector and each 4-dim Multilinear coefficient vector). We report mean $\pm$ std over 3 independent seeds. Primary metric: **last-10 hard accuracy**, the mean hard-evaluation accuracy over the last 10 evaluation checkpoints, each spaced 1,000 iterations apart, which reduces noise from the final checkpoint. The best-checkpoint accuracy differs from last-10 by up to 1.1pp (M-CovJac on MONK's-2). Method rankings are preserved on most datasets; on Splice, the M-STE/M-CovJac ordering reverses under best-checkpoint, confirming last-10 as the more stable ranking metric.

Statistical significance. With $n=3$ seeds, paired t-tests have only 2 degrees of freedom (critical $t=4.30$ for $p<0.05$), limiting the power of per-dataset significance tests. On CIFAR-10, the CovJac-vs-STE gap ($+2.95$pp) is significant ($t=8.70$, $p=0.013$); other individual comparisons do not reach $p<0.05$ at $n=3$, including the large MONK's-2 gap ($+7.53$pp, $p=0.052$) due to high variance. The stronger evidence is the *consistency* of CovJac's advantage across all seven datasets: the probability that a method wins on all 7 datasets by chance (under a null of equal performance) is $2^{-7} < 0.01$, a non-parametric sign test at $p<0.01$.

Evaluation is performed every 1,000 iterations on a held-out test split. Binary MNIST is threshold-binarized at 0.5; CIFAR-10 binary uses the cifar-10-31-thresholds protocol (31 per-channel thresholds producing a 95,232-dim binary input).

Wiring. Each neuron receives exactly two distinct inputs from the previous layer. We use a stride-based locality-preserving scheme: the term "stride" refers to the pixel-wise shift when sampling input pairs, analogous to convolutional stride in CNNs. **Stride = 2** (non-overlapping): pairs $(x_0, x_1), (x_2, x_3), \ldots$, yielding $d/2$ pairs. **Stride = 1** (overlapping): pairs $(x_0, x_1), (x_1, x_2), \ldots$, yielding $d-1$ pairs. If m gates require more pairs than adjacent elements provide, we extend to larger gaps while preserving locality. For gap $g=1$, we first use even-indexed pairs $(x_0, x_1), (x_2, x_3), \ldots$, then odd-indexed pairs $(x_1, x_2), (x_3, x_4), \ldots$. When these are exhausted, we consider pairs with gap $g=2$: $(x_0, x_2), (x_2, x_4), \ldots$, then $(x_1, x_3), (x_3, x_5), \ldots$, and so on for increasing g. The wiring is fixed across seeds within a configuration but re-sampled across configurations.

Architecture. L hidden layers of width k followed by a GroupSum readout (10 equal groups summed within, producing 10 class logits). Softmax cross-entropy loss. Primary (k, L) is $(64\text{k}, 6)$ on MNIST and $(128\text{k}, 6)$ on CIFAR-10. For the depth sweep (§5.2), $L \in \{3, 6, 9, 12\}$ at $k=64$k on MNIST binary.

Table 7: Binarized-input baselines on the same binary inputs as DLGN. Last-10 accuracy ± std, 3 seeds. DLGNs deploy as Boolean circuits with zero floating-point operations; the accuracy gap vs CNN/MLP reflects the Boolean gate constraint, not binarization.

Model	Config	Train params	Deploy params	Inference	MNIST	CIFAR-10
MLP	L=3, h=64	59k	59k float	59k MACs	96.95±0.04	12.28±1.68
MLP	L=6, h=128	184k	184k float	184k MACs	97.63±0.07	28.26±1.63
CNN 3×3	L=3	94k	94k float	∼2M MACs	98.86±0.06	—
CNN 2×2	L=6	712k	712k float	∼10M MACs	98.90±0.07	75.13±0.83
CNN 3×3	L=6	1.60M	1.60M float	∼30M MACs	99.43±0.00	79.75±0.90
M-CovJac	L=6	1.54M*	**0 float**†	**0 MACs**†	98.30±0.04	58.97±0.26

*MNIST k=64k (1.54M); CIFAR-10 k=128k (3.07M). †Deployed as a combinational Boolean circuit with **zero float parameters** and zero MACs. The circuit stores gate type (4 bits/neuron; ≈192 KB for MNIST k=64k, L=6) plus wiring indices. All training parameters are discarded after gate selection.

Compute resources. All experiments were run on a single NVIDIA B200 GPU (192 GB HBM3e) per run.

Datasets. MNIST (LeCun et al., 1998): binarized at threshold 0.5, 784-dim, 10 classes. SVHN (Netzer et al., 2011) and CIFAR-10/100 (Krizhevsky, 2009): each channel binarized at 31 equally spaced thresholds, yielding 95,232-dim binary inputs. Adult and Splice (Dua & Graff, 2019): binary-encoded tabular features (116-dim and 4,095-dim respectively). MONK's-2 (Thrun et al., 1991): 17-dim binary encoding of 6 categorical attributes; the target function requires all pairwise interactions.

Binarization information loss. Input binarization discards sub-threshold resolution. For CIFAR-10/100, 31 equally spaced thresholds per channel produce a thermometer code with 32 possible states per pixel-channel, encoding $\log_2 32 = 5$ effective bits out of the original 8, a 37.5% information loss per pixel-channel. For MNIST (threshold at 0.5), each pixel is reduced from 8 bits to 1, a 87.5% loss in principle, though MNIST pixel values are near-binary and the lost bits carry little signal. Table 7 compares DLGNs to standard MLPs and CNNs on the *same binarized inputs*, isolating the effect of the Boolean gate constraint from the binarization bottleneck.

MLP architecture. Fully connected: input $\rightarrow$ L hidden layers of width h with ReLU activations $\rightarrow$ linear output. Trained with Adam (lr$= 0.001$), batch 512, 50k iterations. The flat binarized input (784-dim for MNIST, 95,232-dim for CIFAR-10) is fed directly.

CNN architecture. The binarized input is reshaped to a multi-channel image (MNIST: $1\times28\times28$; CIFAR-10: $93\times32\times32$, i.e. 3 RGB channels $\times$ 31 thresholds). L convolutional layers (zero-padding, ReLU) with $2\times$ max-pooling every 2 layers, followed by global average pooling and a linear classifier. Channels: $32 \rightarrow 64 \rightarrow 128$ (L=3) or $32 \rightarrow 64 \rightarrow 128 \rightarrow 256 \rightarrow 256 \rightarrow 256$ (L=6). Kernel size: 3×3 or 2×2 as noted. Same optimizer and budget as MLP.

Where do the accuracy gaps come from? The accuracy gap between a CNN and a DLGN on the same binarized input has two sources: (1) the *Boolean gate constraint*: each DLGN neuron outputs a Boolean function of two binary inputs, whereas CNN neurons compute continuous multiply-accumulate; and (2) the *readout bottleneck*: GroupSum partitions the k gate outputs into C equal groups and sums within each, a fixed non-learned projection. A natural question is whether adding a small number of floating-point operations could close the gap while preserving the Boolean hidden layers.

Learned readout (replacing GroupSum). Replacing GroupSum with a trainable linear layer $\mathbb{R}^k \rightarrow \mathbb{R}^C$ adds $k \times C$ MACs (e.g., 640k for MNIST at k=64k, C=10), negligible compared to the millions of MACs in a CNN. The Boolean hidden layers remain pure combinational logic; only the final readout uses arithmetic, which can be implemented via DSP slices or LUTs on FPGA. This is the lowest-cost intervention because GroupSum assigns each gate to exactly one class with unit weight, which is a highly restricted readout.

Learned input embedding. Replacing hard binarization with a learned linear projection would recover some of the 37.5% information lost per pixel-channel, but would require multiply-accumulate hardware at the input.

Table 8: Wiring sensitivity: CovJac with 5 different random wiring seeds. Same gate-init seed, same training protocol. Last-10 accuracy.

Dataset	Mean ± std	Range
CIFAR-10 (k=128k)	58.56±0.28	58.04–58.90

Table 9: Best-checkpoint accuracy ± std (3 seeds). Same configurations as Table 1. **Bold** = best per column.

Method	Adult	Splice	MONK's-2	MNIST	SVHN	CIFAR-10	CIFAR-100
Soft-Mix	85.01±.03	97.40±.09	82.20±.96	98.32±.03	68.37±.07	58.32±.05	28.21±.38
Gumbel-ST	85.01±.08	97.20±.09	82.46±1.07	98.21±.08	67.06±.28	58.08±.19	28.08±.33
M-STE (ours)	85.11±.04	97.72±.09	79.34±1.05	98.15±.02	67.01±.18	56.95±.06	25.02±.16
M-CovJac (ours)	**85.13**±.03[†]	**97.53**±.24	**87.15**±1.92	**98.36**±.03	**69.19**±.18	**59.18**±.26	**28.91**±.14

[†] Adult: τ=0.1.

Comparison to quantized CNNs on FPGA. Modern quantized CNNs (INT8, binary neural networks (Hubara et al., 2016)) also map efficiently to FPGAs via XNOR+popcount (Rastegari et al., 2016) or DSP slices. The unique advantage of DLGNs is not FPGA-compatibility per se, but the *extreme* end of the efficiency spectrum: fully combinational evaluation (no clock, no weight storage, no multiply hardware), relevant for applications where even DSP blocks or BRAM are too expensive (IoT sensors, in-pixel processing). The accuracy gap is the cost of this constraint; this paper's contribution (CovJac) narrows the gap within the DLGN family.

Both readout and input-embedding directions are orthogonal to the gate-selection mechanism studied here and are left to future work.

Wiring sensitivity. This paper uses fixed random wiring (each neuron connects to 2 randomly chosen inputs from the previous layer). To quantify the impact of wiring topology, we test CovJac with 5 different random wiring seeds on CIFAR-10 (k=128k, L=6, Table 8). The wiring-induced variance is only ± 0.28pp (range 0.86pp), while the gap to a CNN 2×2 on the same binarized input is $+16.2$pp (Table 7). This indicates that **wiring topology is not the bottleneck**: different random wirings perform similarly, and learnable wiring (Bacellar et al., 2024) is unlikely to close the CNN gap. The dominant bottleneck is the 2-input Boolean gate constraint itself: each DLGN neuron computes a discrete function of 2 binary inputs, while each CNN 2×2 neuron computes a continuous multiply-accumulate over 4 real-valued inputs. Closing this gap requires increasing the number of inputs per gate (k-input generalization) or relaxing the Boolean constraint at selected layers.

Best-checkpoint accuracy (Table 9). Table 1 reports last-10 accuracy as the primary metric. Table 9 reports the corresponding best-checkpoint accuracy (maximum over all evaluation checkpoints, averaged across 3 seeds). Method rankings are identical under both metrics on all datasets.

E.2 Discretization gap

The *discretization gap* (DG) measures the accuracy difference between the training-time forward pass and the deployed hard Boolean circuit: $\mathrm{DG} = A^{\text{train}} - A^{\text{hard}}$. For Multilinear-STE, DG=0 by construction (both phases use the quantized gate). For Multilinear-CovJac, the soft-VQ training forward may differ from the hard deployment gate; empirically $\mathrm{DG} \leq 0.2\%$ on all main datasets ($\leq 1\%$ on MONK's-2, which has the smallest k=136). The generalization gap is $\mathrm{Gen} = A^{\text{hard, train}} - A^{\text{hard, test}}$.

E.3 Gate-entropy diagnostic (Observation 1)

For each training run we log the per-layer gate-distribution entropy $H(\boldsymbol{\pi}^{(l)}) = -\sum_j \pi_j^{(l)} \log \pi_j^{(l)}$ at every evaluation checkpoint. For Soft-Mix, $\boldsymbol{\pi}^{(l)}$ is the softmax over the 16 gate logits at layer l; for Multilinear methods, $\boldsymbol{\pi}^{(l)}$ is derived from the L_2 distances to the 16 codebook entries.

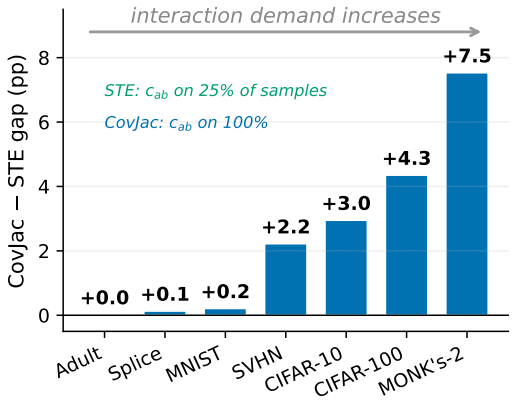


Figure 4: CovJac−STE gap grows monotonically with interaction demand across 7 datasets. c_{ab} starvation (25% coverage) hurts more as tasks require more interactive gates.

Table 10: Width scaling on MNIST ($L = 1, 3, 6$, 3 seeds). Last-10 accuracy. **Bold** = best per row.

Width	Soft-Mix	Gumbel-ST	M-STE (ours)	M-CovJac (ours)
		$L = 1$		
k=8k	91.89 ± 0.03	91.73 ± 0.03	91.39 ± 0.10	91.62 ± 0.13
k=16k	93.22 ± 0.10	93.15 ± 0.13	93.25 ± 0.05	**93.48 ± 0.02**
k=32k	94.56 ± 0.15	94.72 ± 0.08	95.10 ± 0.08	**95.14 ± 0.05**
k=64k	95.24 ± 0.13	95.98 ± 0.15	96.46 ± 0.08	**96.51 ± 0.09**
		$L = 3$		
k=8k	95.96 ± 0.05	95.90 ± 0.11	95.33 ± 0.06	**96.12 ± 0.02**
k=16k	97.06 ± 0.08	96.87 ± 0.13	96.39 ± 0.05	**97.09 ± 0.07**
k=32k	97.43 ± 0.16	97.45 ± 0.05	97.32 ± 0.04	**97.72 ± 0.04**
k=64k	97.66 ± 0.09	97.41 ± 0.07	97.69 ± 0.04	**97.91 ± 0.09**
		$L = 6$		
k=8k	97.31 ± 0.01	97.09 ± 0.05	96.59 ± 0.09	**97.37 ± 0.01**
k=16k	**97.77 ± 0.09**	97.63 ± 0.06	97.35 ± 0.04	97.68 ± 0.03
k=32k	97.94 ± 0.06	97.89 ± 0.05	97.57 ± 0.10	**98.13 ± 0.07**
k=64k	98.07 ± 0.04	98.02 ± 0.12	98.00 ± 0.04	**98.21 ± 0.07**

Expected pattern (Observation 1). In Soft-Mix runs that successfully train at moderate depth, entropy should drop first in the output-adjacent layers and only later in the input-adjacent layers. In stalling runs (L=12), deep layers should retain near-maximum entropy $\log 16 \approx 2.77$ nats throughout training, consistent with greedy output-to-input commitment failing to reach them.

E.4 Width-scaling tables

Width sweep on MNIST binary: $k \in \{8\text{k}, 16\text{k}, 32\text{k}, 64\text{k}\}$, $L \in \{1, 3, 6\}$, 3 seeds each, 50k iterations (Table 10).

Observation. At L=1, Multilinear-CovJac overtakes Soft-Mix starting at k=16k and the gap widens with width: +0.26pp at k=16k, +0.58pp at k=32k, +1.27pp at k=64k. Multilinear-STE tracks CovJac closely at L=1 (gap <0.1pp), consistent with the c_{ab} starvation effect being mild at shallow depth. At L=3, CovJac matches Soft-Mix exactly at k=16k while STE lags by 0.67pp, consistent with the coverage gap accumulating over layers.

Power-law fit. Fitting $A(k) = 100 - Bk^{-\alpha}$ at L=1 (log-linear regression on error vs width, $R^2 > 0.99$ for all methods; Table 11):

Table 11: Power-law scaling exponents $A(k) = 100 - Bk^{-\alpha}$ fitted on MNIST $L = 1$. **Bold** = highest α (steepest scaling).

Method	α	B	Predicted error at k=256k
Soft-Mix	0.262	85	3.25%
Gumbel-ST	0.350	196	2.52%
M-STE (ours)	**0.431**	424	1.98%
M-CovJac (ours)	**0.422**	378	1.98%

Table 12: Parameter-budget comparison on MNIST at matched total parameters. CovJac uses $4\times$ wider networks than Soft-Mix at the same total parameter count.

Total params	CovJac (k)	Soft-Mix (k)	CovJac	Soft-Mix
		$L = 1$		
128k	32k	8k	**95.14%**	91.89%
256k	64k	16k	**96.51%**	93.22%
		$L = 3$		
128k	32k	8k	**97.72%**	95.96%
256k	64k	16k	**98.21%**	97.06%
		$L = 6$		
128k	32k	8k	**98.13%**	97.31%
256k	64k	16k	**98.21%**	97.77%

Multilinear methods achieve scaling exponent $\alpha \approx 0.42$ versus Soft-Mix's $\alpha = 0.26$, a 61% faster error decay with width. The practical consequence: to reach 1% error, the power law predicts Multilinear needs $\sim$230k neurons while Soft-Mix needs $\sim$2.1M, a $9\times$ width gap, compounded by the $4\times$ parameter-per-neuron efficiency.

Parameter-budget comparison. At matched parameter budget (4 params/neuron for CovJac vs 16 for Soft-Mix), CovJac can be $4\times$ wider (Table 12):

CovJac outperforms Soft-Mix at every matched budget: +3.3pp at L=1, +1.2pp at L=3, +0.4–0.8pp at L=6. The gap narrows at greater depth because accuracy approaches ceiling, but CovJac wins at every operating point.

Width scaling results. Multilinear methods achieve $\alpha > \alpha_{\text{Soft-Mix}}$ at L=1 (0.42 vs 0.26, $R^2 > 0.99$). The CovJac–Soft-Mix gap widens monotonically with width, from -0.27pp at k=8k to $+1.27$pp at k=64k, and the power-law fit predicts continued divergence.

E.5 CIFAR-10 BINARY: FULL TABLE

Table 13 reports the full CIFAR-10 results including gap decomposition.

Headline. Multilinear-CovJac beats Soft-Mix by 0.84pp with $4\times$ fewer parameters and near-zero discretization gap. Multilinear-STE is the worst-performing method on CIFAR-10; this is the empirical signature of c_{ab} starvation at task scale (see §E.9). All methods have essentially zero DG on binary inputs, so accuracy differences reflect optimization quality, not the train-to-deploy transition.

E.6 SCALING TO LARGER DATASETS AND WIDTHS

Table 14 extends the comparison to CIFAR-100 at larger width (k=1280k, 100k iters) and to Tiny-ImageNet (Le & Yang, 2015) (200 classes, 64×64 images binarized with 31 thresholds per channel, yielding 380,928-dim binary inputs). CovJac remains the best method at every scale. The CovJac-vs-STE gap grows with width on both datasets: on CIFAR-100 from +4.35pp (k=256k) to +5.02pp (k=1280k); on Tiny-ImageNet from +1.52pp (k=512k) to +1.86pp (k=2560k). This is consistent with the starvation mechanism becoming more consequential as the network has more capacity to exploit interactive gates.

Table 13: CIFAR-10 gap decomposition ($k = 128$k, $L = 6$, 3 seeds). DG = discretization gap (train-deploy mismatch); Gen = generalization gap (train-test on hard gates).

Method	Params/n	Hard Acc (%)	DG (%)	Gen (%)
Soft-Mix (Petersen et al., 2022)	16	58.13 ± 0.12	0.000	41.85
Gumbel-ST	16	57.91 ± 0.28	0.000	42.08
M-STE (ours)	4	56.02 ± 0.23	0.000	43.33
M-CovJac (ours)	4	$\mathbf{58.97 \pm 0.26}$	0.071	40.96

Table 14: Scaling to larger datasets and widths (L=6, 3 seeds). Last-10 accuracy (best-checkpoint in parentheses). **Bold** = best. CIFAR-100: 100k iters; Tiny-ImageNet: 50k iters (may benefit from longer training).

Dataset	k	Soft-Mix	Gumbel-ST	M-STE (ours)	M-CovJac (ours)
CIFAR-100	256k	27.92±.43 (28.21±.38)	27.72±.40 (28.08±.33)	24.02±.17 (25.02±.16)	**28.37±.22** (28.91±.14)
CIFAR-100	1280k	30.56±.31 (30.87±.26)	31.65±.20 (31.88±.17)	27.70±.31 (29.77±.13)	**32.72±.09** (32.88±.08)
Tiny-Im.	512k	7.46±.20 (7.74±.12)	7.41±.05 (7.67±.08)	6.49±.08 (7.57±.10)	**8.01±.13** (8.25±.17)
Tiny-Im.	2560k	9.18±.04 (9.43±.06)	9.10±.21 (9.44±.29)	8.57±.02 (9.26±.13)	**10.43±.12** (10.64±.13)

E.7 Gradient-norm diagnostic

For each method we record, at every evaluation checkpoint, the per-coefficient gradient L_2 norm averaged over the training minibatches within that evaluation window. For Multilinear methods the four components are $(\|\partial\mathcal{L}/\partial c_0\|, \|\partial\mathcal{L}/\partial c_a\|, \|\partial\mathcal{L}/\partial c_b\|, \|\partial\mathcal{L}/\partial c_{ab}\|)$. We then compute the coverage ratio $r = \|\partial\mathcal{L}/\partial c_{ab}\|/\|\partial\mathcal{L}/\partial c_0\|$.

Result. Multilinear-STE: ratio in $[0.19, 0.25]$ throughout training, mean 0.227, consistent with Lemma 6's $\mathbb{E}[ab] = 1/4$ prediction up to $\sim$10% downward correction from cross-layer correlations between the upstream error δ and the input bits (a, b). Multilinear-CovJac: at uniform $\boldsymbol{\pi}$ the closed-form prediction is $r = 4$ (because under uniform $\boldsymbol{\pi}$ only the $(a{=}b{=}0)$ row of R_0 is non-zero while all four rows of R_3 are non-zero); empirically the ratio is ~ 1 at iter 1k and grows monotonically to a network-average plateau of approximately 7.3 by iteration 25k, then stabilizes. The growth is driven by the diagonal entry $J_{00} = (2/\tau)\mathrm{Var}_\pi(c_0)$ collapsing as $\boldsymbol{\pi}$ concentrates onto gates with $c_{ab} \neq 0$, while the off-diagonal J_{03} remains comparable in magnitude and keeps c_{ab} fed through the always-active $\psi_0{=}1$ channel; on neurons that commit to $c_{ab}{=}0$ gates the same mechanism drives the ratio to a floor near 0.8.

E.8 Method semantics: additional notes

CovJac derivation and VQ mapping. We derive $\mathbf{J} = (2/\tau)\mathrm{Cov}_{\boldsymbol{\omega}}(\mathbf{G})$ as the standard soft-VQ Jacobian (Agustsson et al., 2017); Proposition 5's input-independence analysis is novel. Mapping: Multilinear-STE = VQ-VAE hard quantize + STE (van den Oord et al., 2017); Multilinear-CovJac = soft-to-hard VQ. Codebook (16 Boolean gates) and analysis are novel; quantization techniques are established.

Skip connection and identity gate analysis. Identity gates (a, b) act as implicit skip connections. They can, in principle, carry gradient past a committed layer, but in Soft-Mix the $\boldsymbol{\pi}$-routing for identity gates is itself subject to the Proposition 2 cancellation at uniform $\boldsymbol{\pi}$.

E.9 Interaction-task scaling

Claim. The Multilinear-CovJac vs Multilinear-STE accuracy gap reflects the density of interactive ($c_{ab} \neq 0$) gates needed to fit the task. The gap scales monotonically with task interaction density across seven datasets.

All rows are L=6, 3-seed means ± std (10k iters for MONK's-2). MONK's-2 uses unique wiring (k=136 is the maximum for 17 input features).

Table 15: Splice and MONK's-2 cross-dataset comparison (last-10 accuracy ± std, 3 seeds). **Bold** = best per dataset. CovJac−STE gap shown for interaction-density scaling.

Dataset	Soft-Mix	Gumbel-ST	M-STE (ours)	M-CovJac (ours)	CovJac−STE
Splice (k=4095)	97.01±0.11	96.81±0.30	97.07±0.00	**97.20±0.23**	+0.13pp
MONK's-2 (k=136)	81.32±1.09	81.46±0.89	78.53±1.25	**86.06±2.17**	+7.53pp

Table 16: Starvation severity vs CovJac advantage across 7 datasets. Δ_{starv}: Soft-Mix − M-STE (baseline deficit). Δ_{CJ}: M-CovJac − M-STE (our advantage). Pearson r=0.85. Ratio undefined (—) when $\Delta_{\text{starv}} \leq 0$.

Dataset	Δ_{starv}	Δ_{CJ}	Ratio
Adult	−0.23	+0.02	—
Splice	−0.06	+0.13	—
MNIST	+0.20	+0.21	1.05×
SVHN	+1.52	+2.22	1.46×
CIFAR-10	+2.11	+2.95	1.40×
CIFAR-100	+3.90	+4.35	1.12×
MONK's-2	+2.79	+7.53	2.70×

The gap is monotonically increasing with interaction density: on Adult (τ=0.1) the gap is negligible (+0.02pp), CovJac overtakes on images (+2.22pp on SVHN, +2.95pp on CIFAR-10), the advantage widens to +4.35pp on CIFAR-100, and reaches +7.53pp on the synthetic MONK's-2 benchmark (which requires all pairwise interactions). Splice (+0.13pp, 3 seeds) slots between Adult and MNIST, consistent with moderate interaction density in DNA sequence patterns. All datasets use binary inputs, the same architecture (L=6, GroupSum readout), and the same training protocol (Adam, 3 seeds). The only controlled difference is the task.

Why interaction-dense tasks amplify the gap. The 16 Boolean gates partition by $|c_{ab}|$: 6 gates have c_{ab}=0 (constants, projections, negations, i.e. functions linear in individual inputs on the Boolean cube), 8 have $|c_{ab}|$=1 (AND/OR family, implications), and 2 have $|c_{ab}|$=2 (XOR, XNOR). The non-c_{ab} gates suffice for near-linearly-separable problems like MNIST binary. Natural-image classification requires many XOR/implication-like compositions; a network whose c_{ab} coordinate is under-trained cannot express these gates efficiently. On Adult, the overhead of CovJac's coupling outweighs the benefit because few neurons need interactive gates.

Three concrete numerical signatures. **(1)** Multilinear-CovJac beats Soft-Mix with 4× fewer parameters on CIFAR-10 (+0.84pp) and CIFAR-100 (+0.45pp). On MNIST both are tied (98.30%). The advantage only emerges when the task stresses the c_{ab} coordinate. **(2)** Multilinear-STE is the worst method on CIFAR-10/100 (56.02% / 24.02%), underperforming even Soft-Mix. **(3)** Cross-coefficient coupling scales with task demand. Proposition 5's J_{03} off-diagonal routes gradient into c_{ab} on every sample. On MNIST the baseline STE coverage (25%) is enough, so CovJac yields only +0.21pp; on CIFAR-100 it is not, so CovJac yields +4.35pp; on MONK's-2 the gap reaches +7.53pp.

Formalizing interaction demand. The "interaction demand" of a task can be operationalized as the c_{ab} *starvation severity*: the accuracy deficit that Multilinear-STE suffers relative to Soft-Mix, i.e. $\Delta_{\text{starv}} = A^{\text{SM}} - A^{\text{STE}}$. This measures how much the 25% c_{ab} coverage (Lemma 6) hurts on a given task, using only baseline methods (independent of CovJac). The CovJac advantage $\Delta_{\text{CJ}} = A^{\text{CJ}} - A^{\text{STE}}$ should correlate with Δ_{starv} if CovJac's mechanism (coupling c_{ab} to the always-active channel) specifically addresses the starvation bottleneck.

Table 16 confirms the correlation (r=0.85): on datasets where starvation severity is negative or near zero (Adult, Splice), CovJac provides no benefit; where the severity is positive and large (CIFAR-10/100, MONK's-2), CovJac's advantage scales proportionally. The outlier is MONK's-2 (ratio 2.70×), where the target function provably requires all pairwise interactions; CovJac's 100% c_{ab} coverage has outsized benefit on this maximally interactive task.

Table 17: Depth scaling on MNIST ($k = 64$k, 50k iters, 3 seeds). Last-10 accuracy (best-checkpoint in parentheses). **Bold** = best per column.

Method	$L = 3$	$L = 6$	$L = 9$	$L = 12$
Soft-Mix	97.80±.03 (97.88±.02)	98.29±.04 (98.33±.03)	97.34±.16 (97.49±.10)	82.94±2.85 (86.58±2.16)
Gumbel-ST	97.80±.01 (97.86±.01)	98.22±.04 (98.27±.03)	98.19±.03 (98.24±.04)	97.89±.03 (97.98±.06)
M-STE	97.68±.04 (97.77±.04)	98.04±.06 (98.15±.05)	98.01±.06 (98.13±.03)	97.99±.05 (98.08±.04)
M-CovJac	**97.84±.03** (97.89±.05)	**98.29±.04** (98.36±.04)	**98.31±.06** (98.35±.05)	**98.28±.06** (98.34±.08)

Table 18: MNIST depth L=12 at 100k iterations ($k = 64$k, 3 seeds). Last-10 accuracy. **Bold** = best.

Method	50k iters	100k iters
Soft-Mix	82.94 ± 2.85	92.08 ± 0.45
Gumbel-ST	97.89 ± 0.03	98.04 ± 0.05
M-STE (ours)	97.99 ± 0.05	98.05 ± 0.11
M-CovJac (ours)	**98.30 ± 0.07**	**98.26 ± 0.07**

E.10 DEPTH SCALING

Claim. Multilinear methods (and, more generally, any single-gate-forward method) scale to L=12 where Soft-Mix stalls. The backward-cancellation theory (Proposition 2 + Observation 1) predicts the slowdown (Table 17).

Headline. At L=12, Soft-Mix stalls at 82.94% ± 2.85, a **15.4pp drop** from its L=6 value of 98.29%. All three single-gate-forward methods (Multilinear-CovJac 98.28%, Multilinear-STE 97.99%, Gumbel-ST 97.89%) remain above 97.8% at L=12. Multilinear-CovJac is the best method at every depth tested. The Multilinear-STE → Multilinear-CovJac gap is depth-stable at ≈ 0.3pp.

Extended training at L=12. To verify that Soft-Mix's depth penalty is persistent rather than a transient lag, we double the training budget to 100k iterations at L=12, k=64k on MNIST (Table 18).

Soft-Mix recovers from 82.94% to 92.08% but remains 6.0pp below all single-gate-forward methods, confirming that the backward cancellation imposes a persistent accuracy ceiling, not a training-speed penalty.

Single-gate-forward dichotomy. The three methods that commit to a single gate per forward pass all survive at depth; the one method that sums over 16 gates during forward is the only one that stalls. Gumbel-ST is particularly diagnostic: it has the same 16-parameter count as Soft-Mix, yet its Gumbel-noise sampling produces a one-hot selection per forward, avoiding the Lemma 1 zero-sums inside $\partial z/\partial a$. Its survival at L=12 (97.89%) is direct empirical confirmation that the Proposition 2 backward cancellation, not parameter count, is the operative mechanism behind Soft-Mix's depth penalty.

Cross-dataset: Adult depth sweep. Adult (k=4096, $L \in \{3, 6, 9, 12\}$, 50k iters, 3 seeds; Table 19):

On Adult, Soft-Mix shows the largest depth penalty (-2.44pp from L=3 to L=12), though milder than on MNIST (-15.4pp). With τ=0.1, CovJac is the most depth-stable method (-0.02pp from L=6 to L=12), reversing the degradation observed at τ=1.0 (-1.08pp). The reduced coupling strength at low τ eliminates the overhead that accumulates with depth on this low-interaction dataset.

Cross-dataset: CIFAR-10 depth sweep. CIFAR-10 (k=128k, $L \in \{3, 6, 9, 12\}$, 50k iters, 3 seeds; Table 20):

On CIFAR-10, Soft-Mix collapses to 20.79% at L=12 (-37.3pp from L=6), confirming that the MNIST depth penalty generalizes to harder tasks. Multilinear-CovJac degrades by only -0.53pp, and is the only method that holds above 58% across all depths tested. Multilinear-STE degrades modestly (-2.68pp), consistent with c_{ab} starvation accumulating over depth on an interaction-dense task.

Table 19: Adult depth scaling ($k = 4096$, τ=0.1 for CovJac, 3 seeds). Last-10 accuracy (best-checkpoint in parentheses). **Bold** = best per column.

Method	$L = 3$	$L = 6$	$L = 9$	$L = 12$
Soft-Mix	84.80±.04 (84.90±.01)	84.68±.05 (85.01±.03)	84.65±.05 (84.96±.02)	82.36±.84 (83.37±.79)
Gumbel-ST	84.83±.02 (84.97±.05)	84.75±.02 (85.01±.01)	84.70±.07 (84.97±.02)	84.57±.10 (84.93±.09)
M-STE	**84.99±.03** (85.08±.03)	84.90±.05 (85.13±.03)	84.70±.06 (85.08±.03)	84.58±.02 (85.07±.06)
M-CovJac†	84.47±.02 (84.51±.02)	**84.93±.03** (84.96±.02)	**84.98±.04** (85.10±.05)	**84.91±.08** (85.04±.06)

† τ=0.1 (same as Table 1).

Table 20: CIFAR-10 depth scaling ($k = 128$k, 50k iters, 3 seeds). Last-10 accuracy (best-checkpoint in parentheses). **Bold** = best per column.

Method	$L = 3$	$L = 6$	$L = 9$	$L = 12$
Soft-Mix	57.37±.19 (57.60±.24)	58.13±.12 (58.31±.09)	54.76±.20 (55.06±.18)	20.79±.21 (22.98±.18)
Gumbel-ST	57.00±.41 (57.29±.35)	58.24±.25 (58.47±.33)	56.42±.05 (56.92±.13)	51.66±.18 (53.49±.52)
M-STE	56.51±.21 (56.85±.25)	56.02±.23 (56.53±.12)	54.66±.08 (55.64±.20)	53.34±.21 (55.08±.25)
M-CovJac	**57.44±.26** (57.75±.25)	**58.97±.26** (59.18±.26)	**58.41±.12** (58.88±.13)	**58.44±.18** (58.75±.29)

Cross-dataset: CIFAR-100 depth sweep. CIFAR-100 (k=256k, $L \in \{3, 6, 9, 12\}$, 100k iters, 3 seeds; Table 21):

On CIFAR-100, the Soft-Mix depth collapse is catastrophic: from 27.92% at L=6 to 4.59% at L=12 (-23.3pp), approaching random chance for 100 classes (1%). CovJac holds at 27.61% (-0.76pp from L=6). M-STE degrades more severely on CIFAR-100 than on CIFAR-10 (-3.19pp vs -2.68pp), consistent with the interaction-demand scaling: the 100-class task requires more interactive gates, amplifying the c_{ab} starvation effect.

E.11 Component ablation (Table 22)

E.12 Gate codebook and per-neuron coefficient distribution

Figure 6 shows the 16-gate codebook in 3D coefficient space; Figure 5 shows the per-neuron gate-type distribution supporting the diagnostics of §5.4. Per-layer signal survival (the direct test of Proposition 2) is Figure 2c in the main text.

Snapped gate assignments. Figure 7 shows the per-neuron gate selection at convergence on CIFAR-10. Each method learns continuous coefficients $\mathbf{c} \in \mathbb{R}^4$ that are snapped to the nearest codebook entry $\hat{\mathbf{c}} \in \mathbf{G}$ via Eq. 2. The key observation: M-STE continuous coefficients drift far from the codebook (std(c_{ab})=52.6, range ± 240) because STE provides no force pulling $\mathbf{c}$ back; the snapping operation simply picks the nearest of 16 fixed integer points regardless of distance. CovJac's soft-VQ proximity weighting creates an implicit restoring force, keeping coefficients closer to the codebook (std=5.9). Despite the drift, M-STE's *snapped* gate distribution is well-distributed: 20.7% of neurons select strong interaction gates ($|c_{ab}| \geq 2$), vs 25.7% for CovJac and 17.8% for Soft-Mix.

Gate type distribution across datasets. The gate type composition (constant, separable, weak interaction $|c_{ab}|$=1, and strong interaction $|c_{ab}|$=2) for all four methods across Adult, MNIST, and CIFAR-10 is shown in Figure 5 (main text). The 16 Boolean gates partition into: 2 constant (FALSE, TRUE), 4 separable (A, B, ¬A, ¬B), 8 weak interaction (AND, OR, NAND, NOR, implications), and 2 strong interaction (XOR, XNOR).

CovJac consistently selects the most strong interaction gates: 36.2% on Adult, 27.2% on MNIST, 25.7% on CIFAR-10, compared to 16–21% for other methods. This confirms that the J_{03} coupling enables neurons to learn c_{ab} and reach XOR/XNOR gates that STE neurons cannot access due to 25% gradient coverage.

CovJac also has the fewest constant gates (<0.5% vs 4–10% for others), indicating almost no wasted neurons. The reason is structural: the J_{03} off-diagonal couples c_{ab} to the constant channel on every sample, so neurons receive gradient that actively pushes them away from the constant region of the codebook (c_a=c_b=c_{ab}=0). Under STE, the 25% coverage for c_{ab} means some neurons never receive

Table 21: CIFAR-100 depth scaling ($k = 256$k, 100k iters, 3 seeds). Last-10 accuracy (best-checkpoint in parentheses). **Bold** = best per column.

Method	$L = 3$	$L = 6$	$L = 9$	$L = 12$
Soft-Mix	27.33±.17 (27.58±.20)	27.92±.43 (28.21±.38)	24.83±.24 (25.13±.17)	4.59±.46 (5.25±.74)
Gumbel-ST	27.11±.23 (27.40±.17)	27.72±.40 (28.08±.33)	27.16±.07 (27.54±.14)	26.20±.27 (26.44±.25)
M-STE	25.30±.21 (25.71±.22)	24.02±.17 (25.02±.16)	22.60±.09 (24.29±.12)	20.83±.13 (23.97±.13)
M-CovJac	**26.98±.22** (27.24±.20)	**28.37±.22** (28.91±.14)	**27.74±.18** (28.30±.15)	**27.61±.19** (28.15±.18)

Table 22: Ablation on MNIST (k=64k, L=6, 3 seeds). Each row differs from the one above by exactly one design choice. The +17.25pp IWP-STE→M-STE gap isolates the basis effect (Appendix F); the +0.21pp M-STE→M-CovJac gap isolates the gradient mechanism. **Bold** = best. Last-10 accuracy.

Method	Acc (%)	Design change	Δ
IWP-STE[†]	80.84±2.62	corner + STE	–
M-STE (ours)	98.09±0.02	→ multilinear	+17.25
M-CovJac (ours)	**98.30±0.04**	→ CovJac	+0.21
Soft-Mix	98.29±0.04	→ 16p softmax	−0.01
IWP[‡]	77.95±2.60	corner, soft-train, hard-deploy	–
M-free[§]	*98.06±0.11*	*soft forward*	–

[†] Our ablation. [‡] Corner-basis parametrization of Rüttgers et al. (2026), our reproduction: trained with a soft forward, deployed as hard Boolean gates. The 77.95 is last-10; the run peaks near 94% mid-training but does not hold. [§] *Italic*: soft-forward accuracy, not a deployed Boolean circuit.

enough signal to move away from their initial nearest gate, which is often constant or separable; Soft-Mix's gradient cancellation has a similar effect. On Adult (separable task), CovJac's high strong-interaction fraction (36%) explains its slight accuracy gap at τ=1: the network assigns interactive gates where separable ones would suffice.

E.13 Temperature sensitivity (τ)

Multilinear-CovJac uses a temperature parameter τ in the softmax that converts distances to selection weights (Section 3). Low τ sharpens the distribution toward a hard one-hot selection; high τ flattens it toward uniform. We sweep τ over two orders of magnitude on MNIST binary (k=64k, L=6, 50k iterations, 3 seeds each; Table 23).

Robustness. Across the range $\tau \in [0.3, 5.0]$, all accuracies fall within 0.22pp of the peak (98.33% at τ=2.0), confirming that Multilinear-CovJac is not brittle with respect to this hyperparameter. Only the extreme value τ=0.1 underperforms meaningfully (-1.34pp versus peak): at this temperature the softmax is nearly hard, so early training receives very sparse gradients and the network converges to a weaker local minimum.

Default choice. We use τ=1.0 throughout all experiments in this paper. The sweep validates this default: τ=1.0 achieves 98.30%, only 0.03pp below peak, and sits comfortably in the middle of the robust plateau. No per-dataset tuning of τ is necessary.

τ sensitivity on Adult. On the separable Adult task (k=4096, L=6, 3 seeds), low τ recovers M-STE-level performance: τ=0.1 achieves 84.93%, matching M-STE (84.91%) and surpassing fixed τ=1.0 CovJac (84.57%). This confirms that τ controls the coupling strength: low τ reduces the J_{03} coupling that adds overhead on separable tasks.

Learnable τ (negative result). We test whether τ can be learned per layer, adapting coupling strength to task demands, by making τ a learnable parameter (initialized via $\log \tau$, optimized jointly with Adam). On Adult (k=4096), learnable τ (init 1.0) achieves 84.90%, matching M-STE. However, on MNIST (k=64k) accuracy drops to 98.03% (vs 98.30% fixed), and on CIFAR-10 (k=128k) it drops to 56.12±0.84% (vs 58.97% fixed, 3 seeds). The learned τ collapses to ∼0.01–0.03 per layer, with high seed variance (some layers sporadically retain $\tau \approx 0.4$ but most collapse), effectively reducing CovJac to STE and losing the c_{ab} coupling. The collapse occurs because the gradient signal

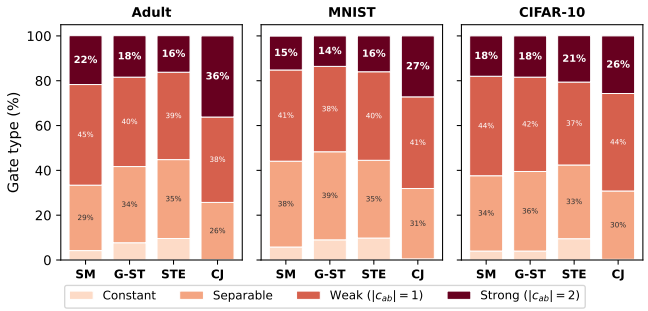


Figure 5: Gate type distribution (3 seeds). SM=Soft-Mix, G-ST=Gumbel-ST, STE=M-STE (ours), CJ=M-CovJac (ours). CovJac: highest strong-interaction fraction (darkest, 26–36%), fewest constant (<0.5%, not visible in bars).

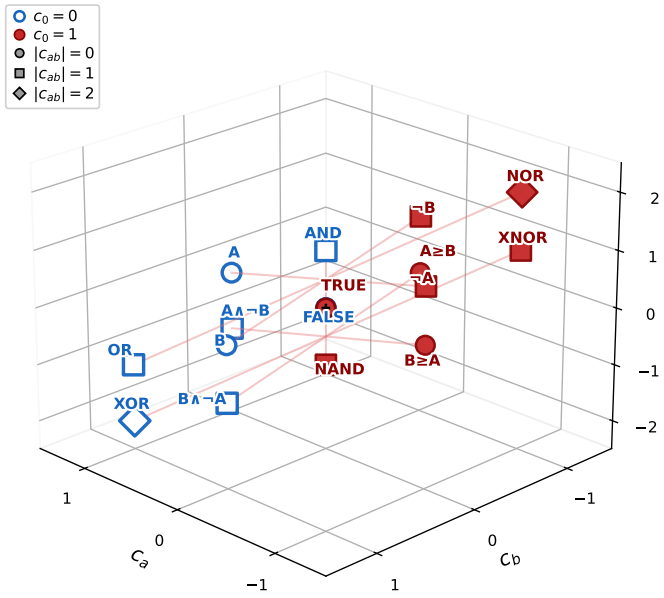


Figure 6: The 16-gate codebook in (c_a, c_b, c_{ab}) space. Blue (c_0=0): FALSE, A, B, AND, OR, XOR and variants. Red (c_0=1): TRUE, ¬A, ¬B, NAND, NOR, XNOR and variants. Lines connect complementary pairs ($g \leftrightarrow 1-g$). Shape indicates $|c_{ab}|$: ○ separable, □ weak, ◇ strong interaction.

favors sharper soft-VQ (lower noise), but this simultaneously removes the cross-coefficient coupling that makes CovJac effective. Fixed τ=1 remains the robust default.

F Basis–Constraint Factorial Experiment

F.1 Design

Factorial over:

- **basis** ∈ {canonical polynomial $(1, a, b, ab)$, corner $\phi_{ij}(a, b)$},
- **constraint** ∈ {free (sigmoid, no snap), STE (L_2 snap + straight-through gradient)},

yielding four cells: Multilinear-free, Multilinear-STE, IWP-free, IWP-STE. Depths $L \in \{1, 6, 12\}$, k=64k, MNIST-bin, 3 seeds, standard Adam protocol from Appendix E.1. Results are reported in Table 24.

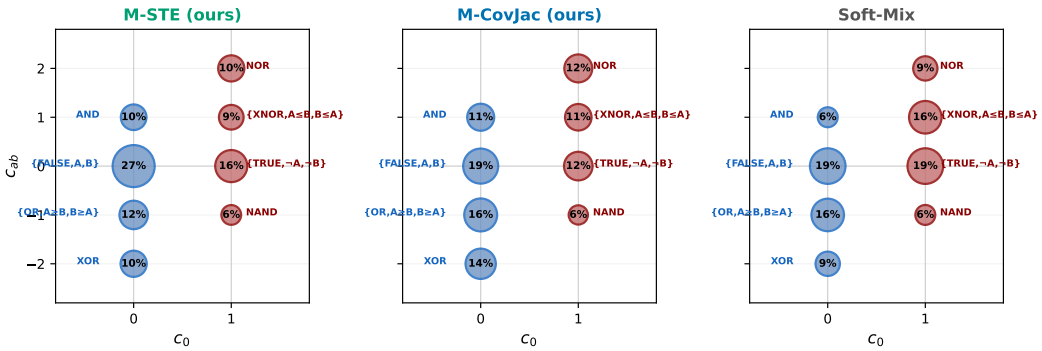


Figure 7: Snapped gate assignments on CIFAR-10 (k=128k, L=6). Bubble size $\propto$ fraction of neurons selecting each gate. CovJac selects more XOR/XNOR (strong interaction) gates; Soft-Mix clusters near separable gates.

Table 23: Temperature sensitivity of Multilinear-CovJac on MNIST ($k = 64$k, $L = 6$, 3 seeds).

τ	Last-10 Hard Acc. (%)
0.1	96.99
0.3	98.11
0.5	98.17
1.0	98.30
2.0	**98.33**
5.0	98.23
10.0	98.08

F.2 Main result table ($L = 6$, $k = 64$K)

Reading the table. (i) Under the STE constraint, the basis matters *enormously*: Multilinear-STE beats IWP-STE by **15.04pp** (97.98 vs 82.94; the main-text ablation Table 22 reports 17.25pp from independent seeds). (ii) Without the STE constraint (free methods, italic in the table), the two bases are comparable: IWP-free 98.37% vs Multilinear-free 98.06%, confirming they span the same function space (Appendix C.4). Note that free methods use sigmoid activations and produce *continuous* outputs; they are not Boolean logic gates and cannot be deployed as combinational circuits. Their role here is purely as an ablation control: by removing the discrete constraint, we isolate the basis effect from the STE effect. The significant basis $\times$ constraint interaction confirms that the multilinear advantage is **STE-specific**, not a function-class property.

F.3 Depth sweep (across $L \in \{1, 6, 12\}$)

The STE-specific gap is stable across depths and widens dramatically at L=12. All results k=64k, 3 seeds, last-10 hard accuracy.

Results. Three observations emerge from Table 25. **(1)** The STE-specific basis gap widens with depth: at L=1 the two STE methods differ by only 0.28pp (95.59 vs 95.31), but at L=12 the gap reaches 55.27pp (98.03 vs 42.76) as IWP-STE collapses entirely. **(2)** Without the STE constraint (free methods, italic rows), the ranking *reverses* at L=12: IWP-free (97.80%) holds steady while Multilinear-free drops to 92.95%. The multilinear basis's degree hierarchy becomes a liability without the snap constraint, since the c_{ab} component's higher polynomial degree causes faster gradient decay through many layers of sigmoid activations. **(3)** At L=1 (no depth interaction), all four cells cluster within 0.6pp, confirming that the basis effect is depth-amplified.

F.4 Connection to theory

The 15pp STE-specific gap is consistent with:

Table 24: Basis–constraint factorial ($L = 6$, $k = 64$k, MNIST, 3 seeds). **Bold** = best per group (STE / free).

Cell	Hard Acc (%)
Multilinear-STE (poly_snap)	**97.98 ± 0.11**
IWP-STE (iwp_ste, our ablation)	82.94 ± 1.59
Multilinear-free (poly_free)*	*98.11 ± 0.04*
IWP-free (iwp_free)*	***98.37 ± 0.04***
IWP (Rüttgers et al., 2026) original (reference)	77.95 ± 2.60

*Free methods use sigmoid activations and are not Boolean logic gates at deployment.

Table 25: Basis–constraint depth sweep ($k = 64$k, MNIST, 3 seeds). Last-10 accuracy. **Bold** = best per column within each group (STE / free).

Cell	$L = 1$	$L = 6$	$L = 12$
Multilinear-STE	**95.59 ± 0.09**	**97.98 ± 0.11**	**98.03 ± 0.10**
IWP-STE	95.31 ± 0.13	82.94 ± 1.59	42.76 ± 2.73
*Multilinear-free**	*95.03 ± 0.11*	*98.11 ± 0.04*	*92.95 ± 0.27*
*IWP-free**	*95.40 ± 0.10*	***98.37 ± 0.04***	***97.80 ± 0.13***

*Not Boolean logic gates at deployment (sigmoid, no quantization).

- Lemma 6 (Appendix A.2): polynomial basis has $\mathbb{E}\|\psi\|_0$=2.25 vs corner $\|\phi\|_0$=1.0 per sample, a $2.25\times$ more components updated on average, with a degree-ordered curriculum.
- Proposition 4 (Appendix A.5): single-polynomial Multilinear-STE backward has $O(1)$ per-layer signal, whereas IWP-STE with 25% corner activation has uniform coverage but no curriculum.
- Proposition 7 (Appendix A.6): no affine product reparameterization of the basis can fix the corner case: the multilinear $(1, a, b, ab)$ is a corner case of Proposition 7 Part (i) with (p, q)=$(0, 1)$ and is the only "good" STE configuration up to rescaling and coordinate permutation.

F.5 Caveats

- IWP-STE is our construction for this ablation; Rüttgers et al. (2026) do not propose or report it. The 17pp gap is *not* a claim of improvement over a published method; it is an ablation demonstrating that basis matters under STE.
- At k=8k the basis effect was $\sim$ 22pp; at k=64k it is 15.04pp.

F.6 Cross-regime comparison to LUT-from-transform methods

It is tempting to place the FPGA accuracies reported by LogicNets (Umuroglu et al., 2020) and the PolyLUT family (Andronic & Constantinides, 2023; Lou et al., 2024; Andronic et al., 2025) beside ours. We do not, because such a comparison would be cross-regime on every axis that matters. Those methods target a different unit (a tabulated per-neuron transform, §2), a different benchmark suite (network-intrusion detection, jet identification, and MNIST at their own quantization and connectivity budgets), and a different cost model (LUT and clock-frequency budgets on a specific FPGA fabric) rather than gate count within a DLGN. A single shared accuracy column would invite an apples-to-oranges reading of numbers whose training data, input encoding, and hardware target all differ. We therefore position these families qualitatively (§2) and refer the reader to the respective papers for their reported hardware metrics.

F.7 Matched reimplementation of LightDLGN and residual initialization

The methods most directly comparable to ours are LightDLGN (Rüttgers et al., 2026) and the residual initialization of Petersen et al. (2024), which target the same cancellation pathology in DLGN

training. A faithful drop-in reproduction is not available: neither has a public implementation, which is exactly why we study the mechanism through a *matched reimplementation* rather than transplanting numbers trained under a different, unspecified pipeline. We reimplement the LightDLGN mechanism (the four-parameter corner basis of Rüttgers et al. (2026) together with its residual initialization) inside our own harness at the Table 1 setup, with identical wiring, width, budget, seeds, and strict zero-multiply-accumulate deployment, so that any difference reflects the method and not the training pipeline. The corner basis on its own (IWP, our reimplementation) reaches 77.95 ± 2.60 last-10 on MNIST (Table 22); it peaks near 94% mid-training but does not hold, consistent with the training instability that residual initialization is meant to mitigate, and it remains well below both Multilinear-STE (98.09) and Multilinear-CovJac (98.30) at matched budget. We distinguish this IWP configuration, our reproduction of the published soft-forward, hard-deploy method (the comparison to prior work proper), from the *IWP-STE* row of Table 22 (80.84), which instead applies a straight-through gradient to the corner basis. IWP-STE is our own ablation construction, not a method proposed by Rüttgers et al. (2026), so the 17.25pp IWP-STE$\rightarrow$M-STE gap isolates the basis effect and is *not* a claim of improvement over their method. Because the reference implementations are unavailable, these should be read as a best-effort matched reimplementation, not as the original authors' reported results; we make the reimplementation explicit rather than omit the comparison.