NORMALIZATION AS GAUGE FIXING: A VARIATIONAL CHARACTERIZATION OF BATCH NORMALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Normalization layers appear in nearly every deep network, yet why they help is still unsettled: BN's original internal-covariate-shift explanation has been challenged, and the six later accounts each rest on measurements, yet none is load-bearing alone, since networks train competitively without a normalizer. Recent work has pinned down the geometry of what the operator produces, the centered sphere its outputs lie on, but not the problem that geometry solves. We supply that problem. Rescaling a neuron's incoming weights leaves a normalized network's function unchanged, so many parameter settings compute the same map and normalization picks one: gauge fixing, in the language of constrained mechanics. For one batch and one channel, batch normalization's two steps are the exact closed-form solution of a strictly convex selection problem over this redundancy, verified symbolically and numerically. Short proofs follow: why a zero-variance channel breaks the operator, what epsilon repairs, why depth needs no iteration, and when a centering folds into the next layer for free. The objective also says why this operator and not a near neighbour: other divisors remove the same redundancy and leave the forward pass unchanged, but only the standard deviation has a backward pass that distorts no direction, and the alternatives lose accuracy in proportion to their distortion, up to 7.7 points. The same structure splits the benefit into four parts, one of which we measure cleanly: holding the forward pass identical and changing only the backward pass, the batch-coupled gradient terms are worth 7.8 to 23.2 accuracy points on CIFAR-10.

1 INTRODUCTION

Normalization layers are among the few components that nearly every modern network keeps. Batch normalization (BN) helped train deep convolutional networks faster (Ioffe & Szegedy, 2015). Layer normalization normalizes features within each example and is used in transformer architectures (Ba et al., 2016; Xiong et al., 2020); RMSNorm removes mean subtraction (Zhang & Sennrich, 2019), and group normalization addresses small batches (Wu & He, 2018). Across a decade of architecture search the normalizer has been moved and simplified, but rarely removed.

Despite this, why normalization helps is still open, and its original answer has been challenged. BN arrived with a mechanism attached: it was said to reduce internal covariate shift, the drift in a layer's input distribution caused by updates to the layers below it. Santurkar et al. (2018) challenged it as a sufficient explanation by re-injecting distribution drift after the normalizer and observing no loss of benefit.

In its place six partial accounts now coexist: loss-landscape smoothing (Santurkar et al., 2018), automatic learning-rate tuning from scale invariance (Arora et al., 2019; Li & Arora, 2020), preconditioning and whitening (Desjardins et al., 2015; Huang et al., 2018), signal and gradient propagation at initialization (Yang et al., 2019), length–direction decoupling (Salimans & Kingma, 2016), and regularization by batch-statistics noise (Hoffer et al., 2017; Luo et al., 2019). Each is supported by measurements, and each is a real property of normalized training. Unfortunately none is load-bearing on its own, since networks with no normalizer reach competitive accuracy (Brock et al., 2021; Zhu et al., 2025). The original story remains insufficient, and the six later accounts do not add up to a mechanism.

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.

A question then arises: if each of the six names a real property and none of them is necessary, what is the operator itself doing? None of the six answers that, because none is a statement about the operator. The geometry of what BN produces is known: Sun et al. (2020) write its output as the centered pre-activation rescaled to a fixed radius, Cho & Lee (2017) give the Riemannian quotient this induces on the weights, and Roburin et al. (2022) carry it into the optimizer's step. However, that geometry does not by itself say which problem the operator is solving, and writing the problem down is what lets one calculation answer questions usually treated apart: why a channel with no variance breaks the operator, what ε is for, why depth needs no iteration, and when a centering can be folded away. The gap is also practical: every normalization-free design bets that nothing essential was dropped, and the bet cannot be priced until the benefit is itemized.

This paper supplies the missing objective. The starting point is an exact redundancy: for a channel followed by BN, rescaling the incoming weights and bias and shifting the bias leaves the network function unchanged. In constrained mechanics such a redundancy is a gauge group, and an operator that selects one representative from each redundancy class is a gauge fixing. We show that BN's two steps are the closed-form solution of that selection problem: centering returns the minimum-norm point of the translation orbit, and scaling minimizes a strictly convex function whose minimizer is exactly $1/\sigma$. The argument is elementary, two one-dimensional convex minimizations, and it borrows no theorem.

Having the problem rather than the formula is what pays. Four facts that are normally established one at a time now read off a single objective: a channel with no variance has no minimizer at all, which is what a dead channel is; ε is exactly what restores one; the learnable scale only moves the target radius, so expressivity and gauge fixing come apart; and across layers the conditions are triangular, so one forward pass in topological order settles all of them. A fifth fact comes from the same place and is new: among all the divisors that would fix this redundancy, the standard deviation is the only one whose backward pass is an orthogonal projection, which says why BN and not one of its close neighbours. The identity itself we check to 3.5×10^{-13} across four data types and three batch sizes. The same calculation settles by rule what is usually settled by rule of thumb, namely when a fixed centering folds into the next layer's bias: exact to 10^{-15} before a dense layer or a circular-padding convolution, and off by 6.6×10^{-2} to 1.2×10^{-1} before a zero-padded one, whose border outputs never see the shift. We are explicit throughout about what was already known and what is ours (Section 6).

Turning from what the operator *is* to what it *buys*, the same gauge structure separates normalization's benefit into four parts, three of which something outside the operator can supply. Sizing them proved harder than naming them, and only one admits an intervention that changes nothing else: run the exact BN forward pass in every arm and alter only the backward pass. Done that way on CIFAR-10 with a ResNet-8, the two batch-coupled terms of BN's gradient are worth 7.8 and 23.2 accuracy points. An optimizer-side substitute recovers 95.3% of the preconditioning gap and the independent-noise proxy tested here is inequivalent to BN at all four batch sizes, both carrying confounds we report where they arise; Section 5 then returns to the six accounts, of which three follow as consequences, one loses the specific form it is usually given, and two we can name but not close.

2 Setup and the exact identity

The plan is to name the redundancy, write down the function whose minimizer picks one representative from each redundancy class, and check that BN's two steps compute that minimizer. Everything here is for one channel and one fixed batch, the narrowest setting in which the statement is exact.

Notation. A batch is $X \in \mathbb{R}^{B \times d}$; one channel has weights $w \in \mathbb{R}^d$, bias b, and pre-activations $z = Xw + b\mathbf{1} \in \mathbb{R}^B$. Write $P = I - \frac{1}{B}\mathbf{1}\mathbf{1}^\top$ for the centering projection, $v = \|Pz\|^2 = \|PXw\|^2$ (the bias drops out since $P\mathbf{1} = 0$), and $\sigma^2 = v/B$. Training-mode BN with $\varepsilon = 0$ maps $z \mapsto Pz/\sigma$, and $L(w)$ is the loss restricted to one channel's block with everything else fixed.

The gauge action. BN is invariant under $z \mapsto cz + a\mathbf{1}$ for $c > 0$ and $a \in \mathbb{R}$, since the centered vector and σ both scale by c. Pulled back to parameters, $(w, b) \mapsto (cw, cb + a)$ leaves every downstream activation, hence the loss, unchanged, so the loss is positively homogeneous of degree

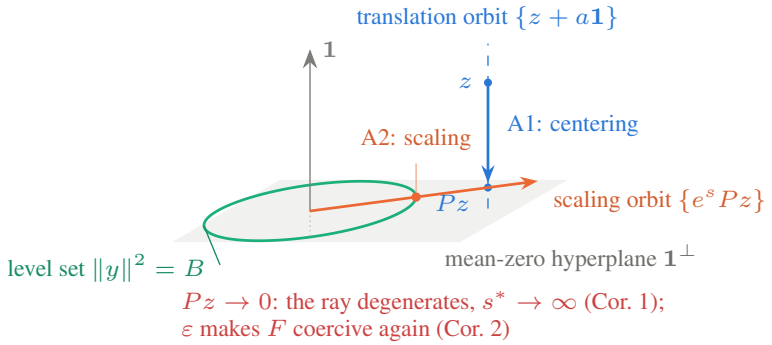

Figure 1: **BN selects one point of an orbit.** The pre-activation vector $z \in \mathbb{R}^B$ of one channel on one batch has an affine orbit $\{e^s(z + a\mathbf{1})\}$ along which the network function is constant. Centering (A1) picks the minimum-norm point Pz, and scaling (A2) picks the orbit point $\mathrm{BN}(z) = e^{s^*} Pz$ on the level set $\|y\|^2 = B$ (Theorem 1); when $Pz \to 0$ the ray degenerates and the minimizer escapes.

zero in each block: $L(cw, cb) = L(w, b)$. Per layer the group is $(\mathbb{R}_{>0} \ltimes \mathbb{R})^C$, one affine group per channel. We call this the gauge action, and any operator selecting one point of each orbit a gauge fixing; LN and RMSNorm carry different groups (Appendix D).

Lemma 1 (Homogeneity calculus)**.** *If $L(cw) = L(w)$ for all $c > 0$ and L is twice differentiable, then for all w: (i) $\langle w, \nabla L(w)\rangle = 0$; (ii) $\nabla L(cw) = c^{-1}\nabla L(w)$; (iii) $\nabla^2 L(w)\, w = -\nabla L(w)$, hence $w^\top \nabla^2 L(w)\, w = 0$.*

Item (iii) is the precise shape of the pathology gauge fixing removes: the radial direction carries no loss information, has exactly zero curvature, and is nevertheless coupled to the gradient through $\nabla^2 L\, w = -\nabla L$.

Proof. Each item comes from differentiating the invariance in a different variable. Differentiate $L(cw) = L(w)$ in c at $c = 1$ for (i), and in w for (ii). Differentiate (i) in w: $\nabla L + \nabla^2 L\, w = 0$, which is (iii); contract with w and use (i). □

Theorem 1 (BN solves one convex problem per channel)**.** *Fix a batch and a channel with $v = \|Pz\|^2 > 0$.*

(A1) Centering. $\arg\min_{a\in\mathbb{R}} \|z + a\mathbf{1}\|^2 = -\mathrm{mean}(z)$*, and the minimizer is Pz: the centered vector is the minimum-norm representative of the translation orbit.*

(A2) Scaling. *On the scaling orbit $\{e^s Pz\}$, consider the function at level B defined as:*

$$F(s) \;:=\; e^{2s} v - 2Bs. \tag{1}$$

Then F is strictly convex on $\mathbb{R}$ with unique minimizer $e^{s^*} = \sqrt{B/v} = 1/\sigma$.

Consequently $\mathrm{BN}(z) = e^{s^*} Pz$*: the two steps return the point of the affine orbit that lies on the level set $\|y\|^2 = B$ and is orthogonal to $\mathbf{1}$, which is the scaled image of the shortest point of the translation orbit. Proposition 1 below writes the two as one objective.*

Intuitively, the theorem says BN is not applying a formula but solving a problem: which point of the orbit is shortest, subject to having the right length? Centering answers the translation half by deleting the component along $\mathbf{1}$, the only direction a translation can move. Scaling balances the two forces in equation 1, the growth term penalizing outputs that are too large and the linear term penalizing outputs that are too small, with balance point exactly $1/\sigma$ and the level B setting where it falls.

Proof. We show that each of BN's two steps is the unique minimizer of a strictly convex one-dimensional problem. For (A1), write $\|z + a\mathbf{1}\|^2 = \|z\|^2 + 2a\langle z, \mathbf{1}\rangle + a^2 B$, a strictly convex quadratic in a with critical point $a^* = -\langle z, \mathbf{1}\rangle / B$ and $z + a^*\mathbf{1} = Pz$; orthogonality $\langle Pz, \mathbf{1}\rangle = 0$

identifies it as the minimum-norm point of the affine line. For (A2), $F'(s) = 2(e^{2s}v - B)$ and $F''(s) = 4e^{2s}v > 0$ whenever $v > 0$, so the unique critical point $e^{2s^*} = B/v$ is the global minimum; then $e^{s^*}Pz = Pz/\sigma$ and $\|e^{s^*}Pz\|^2 = B$. □

This function has a name, and the name carries content. Writing $\mu(y) = \|y\|^2$ gives $F'(s) = 2(\mu(e^s Pz) - B)$, so equation 1 is the *Kempf–Ness function* of this action at level B, and its stationary points are exactly the orbit points sitting on the moment level set (Appendix A). We use the term from here on. Any level $\tau > 0$ gives a convex F_τ, so the identity fixes which problem BN solves and leaves the level open.

Corollary 1 (Zero-variance channels are unstable orbits)**.** *If $v \to 0^+$ then $s^* = \frac{1}{2}\log(B/v) \to \infty$, and at $v = 0$ we have $F(s) = -2Bs$, unbounded below with no minimizer. The Kempf–Ness representative escapes along the one-parameter subgroup $s \to \infty$. This is the Hilbert–Mumford picture of an unstable point (Mumford et al., 1994), and it is the "dead or near-constant channel" failure of BN. The condition is a property of the current batch, not of the unit on the data distribution.*

Unfortunately the same closed form carries its own failure mode. Put concretely: when a channel is constant across the batch its scaling orbit collapses to the single point 0, so no representative reaches the prescribed level at all, and equation 1 reports this by sending the scale to infinity rather than by failing.

Corollary 2 (ε is coercification)**.** *For $\varepsilon > 0$ let $F_\varepsilon(s) = e^{2s}(\sigma^2 + \varepsilon) - 2s$, which is equation 1 divided by B with σ^2 shifted to $\sigma^2 + \varepsilon$; the division does not move the minimizer. Then F_ε is strictly convex and coercive for every z, with unique minimizer $e^{s^*} = (\sigma^2 + \varepsilon)^{-1/2}$, the scale factor of BN as implemented. The ε hyperparameter is exactly the coercification of the variational problem.*

Fortunately the repair costs one term: a positive ε bounds the quadratic coefficient away from zero, so F_ε grows in both directions for every input and a finite minimizer always exists. Section 3 returns to what it does *not* fix.

Proof of both corollaries. Direct from F and F_ε; $F_\varepsilon'' = 4e^{2s}(\sigma^2 + \varepsilon) > 0$ and $F_\varepsilon \to \infty$ as $|s| \to \infty$. □

One more thing, since the two-step presentation invites the opposite guess.

Proposition 1 (One joint problem for the whole affine group)**.** *For $\varepsilon \geq 0$, a scale $a > 0$ and a shift c, define:*

$$J_\varepsilon(a, c) \;:=\; \tfrac{1}{2B}\|az + c\mathbf{1}\|^2 - \log a + \tfrac{\varepsilon}{2}a^2. \tag{2}$$

If $\sigma^2 + \varepsilon > 0$ then J_ε is strictly convex on $\{a > 0\} \times \mathbb{R}$, with unique minimizer $a^* = (\sigma^2 + \varepsilon)^{-1/2}$, $c^* = -a^* \operatorname{mean}(z)$, so $a^* z + c^* \mathbf{1} = \mathrm{BN}_\varepsilon(z)$.

What this says is that the two steps need not be sequential: one convex problem in the two affine parameters returns BN at once, with ε entering as a ridge term rather than a patch. Equation 2 is a Gaussian location–scale negative log-likelihood under $(a, c) = (1/s, -m/s)$; Theorem 1 adds the moment-map reading the rest of the paper uses.

The learnable affine output plays no part in the selection. For $\gamma > 0$ it re-selects the level, moving the representative to mean β and squared radius $\gamma^2 B$ while leaving the variational problem otherwise unchanged, so expressivity and gauge fixing are decoupled (Corollary 1, Appendix C).

The objective picks the centered sphere, and one may ask what another pick would cost. Any positive divisor homogeneous of degree one fixes the same redundancy and gives the same invariance, the batch's mean absolute deviation and its range included, and nothing in the forward pass tells them apart. The backward pass does.

Proposition 2 (The sphere is the one gauge fixing that is a gradient field)**.** *Let $N(z) = Pz/\rho(Pz)$ with $\rho > 0$ homogeneous of degree one, so that N fixes the same gauge action. Then $\mathrm{D}N$ is a positive multiple of a projection Π of rank $B - 2$ (written out in Appendix I) whose kernel is $\operatorname{span}\{\mathbf{1}, N(z)\}$, the two gauge directions. That projection is symmetric, hence orthogonal, exactly when ρ is a positive multiple of the Euclidean norm, that is exactly when $N = \lambda\,\mathrm{BN}$ for some $\lambda > 0$; equivalently, exactly when N is a gradient field, since $\mathrm{BN}(z) = \nabla_z[B\,\sigma(z)]$.*

Said without symbols: every gauge fixing of this shape differentiates to a projection that deletes the two redundant directions, and only the sphere deletes them at right angles. Corollary 2 in Appendix I says why: BN is the gradient of B times the batch standard deviation, so its Jacobian is a Hessian and cannot be anything but symmetric. The radius is left open, since λ is free, and that is where the level B sits: not forced by the geometry, but the one degree of freedom γ re-selects.

Obliqueness is a number, so this is a prediction. Normalize the smallest nonzero singular value of Π to one and call the largest $s_{\max}$, the worst ratio by which the backward pass stretches one tangent direction against another. For the sphere it is 1 at every batch size; for the mean absolute deviation it saturates at $\sqrt{\pi/2} \approx 1.25$; for the range it keeps growing with the batch, from 1.28 at $B = 8$ to 3.98 at $B = 256$ (Appendix I). So the prediction is not that the alternatives are worse across the board, but that the range divisor should worsen as the batch grows while the other should not. Trained in BN's place on CIFAR-10 with a ResNet-8, the range divisor trails the sphere by 6.3 points at $B = 32$ and 7.7 at $B = 256$, while the mean absolute deviation stays within a point of it at both. The ordering across divisors is far outside the seed spread; the widening with the batch, 1.3 points at two seeds, is in the predicted direction but we report it as suggestive (Appendix I).

Proposition 3 (Shift absorption and its boundary)**.** *Let a centering site subtract a fixed vector m from its input, let T be the map applied immediately downstream with linear part T_0, and let $\mathcal{C}$ be the space of biases T can carry: all of $\mathbb{R}^{\text{out}}$ for a dense layer, and one scalar per output channel, shared across spatial positions, for a convolution. Then centering followed by T equals T with its bias changed by $-T_0 m$, and so carries no expressive content, exactly when $T_0 m \in \mathcal{C}$.*

Read as a checklist, the phrase "commutes with constant shifts" separates the cases by what each one turns on:

(i) a dense layer always absorbs it, since $\mathcal{C}$ is everything;
(ii) a convolution absorbs it when its boundary condition is equivariant to constant shifts, so $T_0 m$ is the same at every position and lies in $\mathcal{C}$; this depends on the padding, not the weights;
(iii) under zero padding $T_0 m$ varies across positions, so no per-channel bias matches it, and the residual depends on how much of each receptive field falls outside the image;
(iv) a nonlinearity between the site and the bias breaks the absorption for every $m \neq 0$.

We check this at the operator level, weights and m held fixed so no optimization path enters: 15 cells in float64, 256 input batches each, exactness criterion 10^{-10} (Figure 2(b); per-cell values in Appendix G). Dense layers and circular-padding convolutions are exact to 10^{-15}, zero-padding convolutions leave 6.6×10^{-2} to 1.2×10^{-1}, and ReLU sites are untouched. The residual is a boundary effect and not a failure of the correction, since the same convolution with circular padding is exact. Centering is therefore free before a dense layer and not free before the zero-padded convolutions standard image networks are built from.

Corollary 3 (Depth is a triangular solve)**.** *On an acyclic computation graph, layer ℓ's constraint depends only on the outputs of layers before ℓ, which are already normalized. The per-layer problems are therefore lower-triangular in topological order, a single forward pass satisfies all constraints simultaneously, and re-solving any layer on its own output is a fixed point.*

Proof. We induct on the topological order, so it is enough to show that re-solving a layer changes nothing: $\mathrm{BN}(\mathrm{BN}(z)) = \mathrm{BN}(z)$, because $\mathrm{BN}(z)$ already has mean 0 and variance 1. □

Depth therefore costs nothing here, because the per-layer problems are never simultaneous: by the time a layer is reached, everything its constraint depends on has been fixed.

3 Constraint structure

Theorem 1 describes one forward pass. This section asks what the same redundancy does to training, in the least Hamiltonian language that carries content. Pair each weight w_i with a cotangent variable p_i under the bracket $\{f, g\} = \sum_i (\partial_{w_i} f\, \partial_{p_i} g - \partial_{p_i} f\, \partial_{w_i} g)$. The scaling action lifts to $(w, p) \mapsto (cw, c^{-1}p)$, generated by $\mu = \langle w, p \rangle$. Evaluated on the gradient field, $p = -\nabla L$, Lemma 1(i) gives $\mu = 0$ at every point, so $\mu \approx 0$ is a first-class constraint in Dirac's sense (Dirac, 1964) and its flow is the gauge transformation itself.

We are deliberate about which vector p is, because the natural shorthand is wrong: it is the gradient, not the optimizer's update. Gradient descent is not a Hamiltonian system, and under momentum or Adam the update has a nonzero radial part, which is Proposition 4(iii) below. So $\mu \approx 0$ is a property of the gradient field that every optimizer inherits pointwise, not a conserved quantity of the discrete dynamics.

Proposition 4 (Exact conservation and the norm recursion)**.** *Let L be scale invariant on the block w and let $g = \nabla L(w)$.*

(i) $\langle w, g\rangle = 0$ at every point, hence at every iterate of any optimizer. For SGD the per-batch loss is itself scale invariant (normalization is computed within the batch), so the identity holds per step, not in expectation.

(ii) Under gradient descent $w^+ = w - \eta g$, and with decoupled weight decay $w^+ = (1-\eta\lambda)w - \eta g$, the squared norm obeys exactly:

$$\|w^+\|^2 = \|w\|^2 + \eta^2\|g\|^2 \qquad \textit{and} \qquad \|w^+\|^2 = (1-\eta\lambda)^2\|w\|^2 + \eta^2\|g\|^2. \tag{3}$$

(iii) For any update $w^+ = w + \Delta w$, $\|w^+\|^2 - \|w\|^2 - \|\Delta w\|^2 = 2\langle w, \Delta w\rangle$: zero for (S)GD, nonzero for momentum (where $\Delta w = -\eta m$ and the buffer m of past gradients is not orthogonal to the current w) and for Adam. The break is predicted, not incidental.

Taken together the three items say this: the gradient never points along the weight, so no optimizer can change the scale on purpose; the scale nevertheless changes, by exactly the squared step length; and any optimizer whose update is not the raw gradient must violate equation 3, by an amount we can write down in advance.

Proof. (i) is Lemma 1(i). (ii) follows by expanding and using $\langle w, g\rangle = 0$ to kill the cross term. (iii) is the polarization identity. All three are statements about the $\varepsilon = 0$ operator, whose scale invariance is exact; with $\varepsilon > 0$ the invariance is approximate and so are the identities, which is why Table 2 verifies them at $\varepsilon = 0$. □

Table 2 and Figure 1(c), in Appendix E, show all three at machine precision, including the predicted break. Item (i) and the decay-free case of (ii) are not new: the orthogonality is folklore for scale-invariant blocks (van Laarhoven, 2017), the recursion is Lemmas 2 and 3 (eq. 16) of Sun et al. (2020), and the effective rate $\eta/\|w\|^2$ was studied by Arora et al. (2019). Wan et al. (2021); Kosson et al. (2024) study equilibrium angular motion with weight decay; under the stationary-norm premise, the finite-step relation $\sin^2 \Delta\theta = \eta\lambda(2-\eta\lambda)$ follows algebraically from (ii). We add (iii), where the recursion stops holding and by how much, verified per step across four optimizers, and the reading of (ii) as an effective-rate schedule that Section 4 takes up.

The constraints are jointly first class: the per-neuron scaling and translation momenta Poisson-commute and the interface actions close in the Lie algebra, so the multi-layer reduction is well defined (Theorem 1, Appendix C). That does not give order-independent gauge fixing, which additionally needs the orbits not to share a variable. What makes one forward pass enough is the triangularity of Corollary 3, and Table 2 accordingly reports random sweeps converging in three to four passes while the topological order converges in one.

Proposition 5 (Double degeneracy)**.** *On a channel let $\chi_t = \mathrm{mean}(z)$ and $\chi_s = \|Pz\|^2 - B$ be the gauge-fixing conditions imposed by BN, and $\mu_t = p_b$, $\mu_s = \langle w, p_w\rangle + bp_b$ the translation and scaling momenta. Then $\{\chi_t, \mu_t\} = 1$ and $\{\chi_s, \mu_s\} = 2v$. Hence $v = 0$ simultaneously (i) destroys the coercivity of F (Corollary 1) and (ii) makes the Dirac pairing of the scaling constraint singular, so the gauge-fixing condition ceases to be transverse to the gauge flow. One degeneracy, two symptoms.*

Proof. We compute the two brackets directly. Since χ_s depends only on w and is homogeneous of degree two in it, Euler's identity gives $\{\chi_s, \mu_s\} = \sum_i w_i \partial_{w_i}\chi_s = 2\|PXw\|^2 = 2v$; the translation case is $\{\chi_t, \mu_t\} = \partial_b \chi_t = 1$. Both are verified symbolically. □

Recall that the pairing measures whether the condition BN imposes is transverse to the direction the gauge can move. So the proposition says the variance condition stops being a valid choice

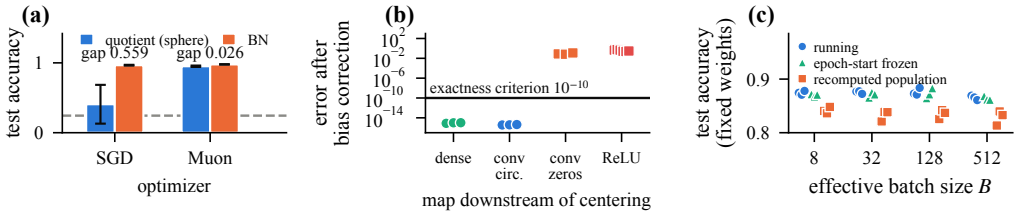

Figure 2: **Testing the ledger.** **(a)** ③ supplied from the optimizer side on anisotropic `digits` (5 seeds, mean ± std; dashed, the bare network). The gap between the Riemannian quotient ("sphere", $1/t$ schedule) and BN falls from 0.559 to 0.026 once the optimizer approximately orthogonalizes update matrices: the 95.3% of Section 4.2, with the cautions there. **(b)** Operator-level test of Proposition 3 (15 cells, float64): dense layers and circular-padding convolutions absorb the centering into the downstream bias, zero padding and ReLU do not. **(c)** At fixed weights, rate and checkpoint, changing only the evaluation-time statistics moves test accuracy in all 12 cells.

of representative at exactly the batch where the representative stopped existing. Both follow from $v = 0$, so this is not independent evidence for the identification; what it buys is one name for two symptoms usually handled apart.

One warning, since the natural guess is wrong. Corollary 2 does not also repair this degeneracy: shifting χ_s by the constant $B\varepsilon$ leaves its gradient unchanged, so the pairing stays $2v$; and $\varepsilon > 0$ in any case removes the exact scale invariance whose regularity was at issue. It is a numerical stabilizer, not a structural repair (Appendix C).

Weight decay has a second role in the gauge directions. For a ReLU-type neuron's intrinsic rescaling the imbalance $Q = \|w_{\text{in}}\|^2 - \|w_{\text{out}}\|^2$ is conserved at $\lambda = 0$ (Du et al., 2018) and decays as $Q(0)e^{-2\lambda t}$ under gradient flow with decoupled decay (Kunin et al., 2021, eq. 20). Here $Q = \frac{1}{2}F'(0)$ for that orbit's Kempf–Ness function, so the along-orbit component of the decay force is the orbit gradient of F and the transverse component is not (Prop. 1; the deep linear version is Lindsey & Menon, 2025).

4 The Payload Ledger

Sections 2 and 3 describe what normalization *is*; what it *buys* is a separate question. We answer it as a decomposition with one term per piece of the gauge structure, calling the terms *payloads* and the list a *ledger*. ① **Quotient of the flat gauge directions** (Lemma 1), in two halves: $①_{\text{fwd}}$ keeps the iterate on the chosen representatives, $①_{\text{grad}}$ keeps the update in the tangent space of the conditions that choose them, which BN gets by differentiating through its own statistics. ② **Effective-rate schedule** $\eta/\|w\|^2$ from Proposition 4(ii), with the angular thermostat when weight decay is on. ③ **Data-coupled diagonal preconditioning**: the representative BN picks, the *section* in bundle vocabulary, is chosen by data statistics, so per-channel standardization is the diagonal approximation of whitening (Desjardins et al., 2015; Huang et al., 2019). ④ **Section estimation from the batch**: the section is a function of the batch, not of each sample, and Section 4.4 tests an independent-noise proxy for it. Each is supplied independently below.

4.1 ① and ②: Working Directly on the Quotient

If ① is only the removal of the flat direction, working on the quotient should reproduce BN's angular solution with no normalizer. Per-filter Riemannian gradient descent on the product of spheres, with ② restored as a $1/t$ schedule, matches that solution to four digits on a toy problem. It recovers 93% of the BN benefit there and 62% on a CIFAR-10 BN-CNN, at one seed, and the sphere arm is not literally the BN network's quotient, since deleting the normalizer changes the forward map (Appendix H.1).

4.2 ③ CAN BE SUPPLIED FROM THE OPTIMIZER SIDE

If ③ is conditioning and not a property of the activation statistics as such, an optimizer that approximately orthogonalizes update matrices should close most of the gap between the quotient method and BN; we fixed $\text{gap}_{\text{Muon}} \ll \text{gap}_{\text{SGD}}$ before the run, refuted at absorption $\leq 60\%$. On anisotropic `digits` with an MLP, five seeds and Muon (Jordan et al., 2024), $\text{gap}_{\text{SGD}} = 0.559$ falls to 0.026, absorption 95.3%, strongly sub-additive (Figure 2(a), Appendix H). Three cautions, none small: the bare and sphere arms carry near-untrainable seeds and both took the smallest rate in the grid, so part of the gap is trainability until a wider grid rules that out; Muon also changes update magnitudes as it orthogonalizes directions, so 95.3% is an upper bound rather than a measurement; and the residual 2.6 points is unattributed.

4.3 THE TWO HALVES OF ①

Separating $①_{\text{grad}}$ from $①_{\text{fwd}}$ needs an intervention that changes the backward pass and leaves everything else alone. BN's gradient with respect to its input splits into three terms, written for $\hat{y} = (x - \mu)/s$ with $s = \sqrt{\text{var} + \varepsilon}$ and $g = \partial L/\partial \hat{y}$, as:

$$\frac{\partial L}{\partial x} = \frac{1}{s}\big[\, g - \overline{g} - \hat{y}\,\overline{g\hat{y}} \,\big], \tag{4}$$

with the bar denoting the mean over the axes the statistics are taken over. Term by term, the first is what any fixed rescaling would give, the second is what differentiating through the batch mean contributes, and the third is what differentiating through the batch standard deviation contributes; the third is also the tangent projection $①_{\text{grad}}$ names. We therefore run the exact BN forward pass in all four arms, including the same running-statistic rule, and drop the second and third terms in the backward pass only. Every arm then computes the same function at fixed weights, and the `full` arm reproduces PyTorch's BN to 1.8×10^{-15} in the input gradient (Appendix H.3).

On CIFAR-10 with a ResNet-8 the two terms carry most of what ① is worth: dropping the standard-deviation term costs 7.8 points, with a 95% paired interval of $[7.3, 8.2]$; dropping the mean term costs 23.2; dropping both costs 17.9 and widens the seed spread from 0.3 points to 8 (Table 6). Every arm selected the interior rate 0.1 from its own grid, so no baseline sits at the edge of a search. The arms do not look additive either: removing both hurts less than removing the mean term alone, by 5.3 points, though at three seeds the four-arm interaction is 13.1 points with a 95% interval of $[-12.8, +39.0]$, so that is a pattern to check and not a finding. On `digits` all four arms sit above 97% and the ablated ones are if anything slightly ahead, by at most 0.6 points; there is little for the gradient to win on a saturated problem.

What an arm removes has a closed form, and that form fixes the scale these numbers should be read against. Dropping the mean term does not merely delete a contribution: it re-injects $\partial L/\partial \beta$ into the weights through a leak proportional to the input mean, and dropping the standard-deviation term re-injects $\partial L/\partial \gamma$ through the input covariance, both exact to 2×10^{-15} in float64 (Appendix H.3). So the effect is largest where activations carry a large mean, which is every site after the first, and it vanishes when a site's input is centered. The figures above are the size of that leak at the input means this network has, not a constant of the operator. The arms also differ in kind: BN's update field is curl-free and each ablated field is not, so no arm but `full` descends any objective at all.

An earlier route of ours, replacing BN with a section tracker, changes the section estimator too, so we report the CIFAR-10 figures instead (Appendix H.4).

4.4 WHAT THE SECTION-ESTIMATION PAYLOAD IS

Hoffer et al. (2017); Teye et al. (2018) motivate a stochastic reading of batch statistics, but do not specify the proxy we test. We define it as full-batch BN plus sample-independent Gaussian noise with variances set to the first-order fluctuations of the batch estimates for Gaussian activations. A general distribution adds a kurtosis factor to the scale term, so this experiment tests our Gaussian proxy rather than every independent-noise account (Appendix H states the model). Criterion: agreement within pooled 1σ at each $B \in \{8, 32, 128, 512\}$ on CIFAR-10 with a ResNet-8, three seeds, 160 epochs, frozen before the run. The model is inequivalent at all four batch sizes, with gaps 0.070, 0.067, 0.114, 0.188, and the direction is the surprise: the gaps grow with B, so the discrepancy is

largest exactly where the injected noise is smallest (Figure 2(b)). Batch size is not the only moving part, and pooled 1σ is a threshold fixed in advance rather than a formal equivalence test. Four diagnostics and a fixed-selection counterfactual then localize what the proxy misses, and none closes more than two of the four cells (Appendix H). So this is a negative result about one model, not a measurement of ④; it fails in a way consistent with ④ being batch coupling of the representative, since every sample's normalization depends on the whole batch and on the state the evaluation path carries. A quantitative model of the term stays open.

5 THE SIX EXPLANATIONS, PLACED

The six accounts of the introduction are readings of one reduction where we can establish the link, and named but open connections where we cannot. One we can derive: the gradient-norm and smoothness improvements reported by Santurkar et al. (2018) contain a radial factor that follows from homogeneity times the norm growth of Proposition 4(ii), with no appeal to the landscape ($\|w\|$ grows from 4.0 to 17.7 in our CIFAR run), and what is left after dividing it out is the conditioning of L on the sphere, which is ③. We do not claim the whole effect, since u_t moves too (Appendix K). Table 11 places the other five.

6 RELATED WORK

Section 1 listed the work on normalization's consequences (Cho & Lee, 2017; Sun et al., 2020; Arora et al., 2019; Wan et al., 2021; Kunin et al., 2021; Ziyin et al., 2025). Here we say, result by result, who already had it; Appendix J is the long form. **The geometry.** Nearest is Sun et al. (2020): the centered sphere, its radius, and the affine parameters as a change of radius and center are all there. We add the objective that selects the point, and with it non-existence at zero variance, a degenerate pairing there, a movable level, and the uniqueness of the sphere among divisors. **The backward terms.** Work on normalization-statistics gradients includes detachment for LayerNorm (Xu et al., 2019), stabilization of BN backward statistics (Yan et al., 2020), and an online substitute (Chiley et al., 2019); Section 4.3 adds their reading as the gradient half of a gauge quotient and a closed form for what detaching leaks. **The decomposition.** Taking BN apart by intervention is not new either (Dauphin & Cubuk, 2021); our ledger differs in where its terms come from, one per piece of gauge structure, not in the technique. **The optimizer and weight decay.** Heo et al. (2021) project out the same radial component Proposition 4(iii) measures, and Lindsey & Menon (2025) use Kempf–Ness for the balanced manifold of a deep linear network, of which Proposition 1 is the single-neuron case. **Shift absorption.** Folding a mean into a downstream bias is standard under other names (Nagel et al., 2019; Jiang et al., 2023); Proposition 3 contributes the boundary. **What we do not subsume.** The depth-and-initialization line (Yang et al., 2019; Daneshmand et al., 2020; De & Smith, 2020; Lubana et al., 2021) explains phenomena our reduction says nothing about, and Table 11 marks that row open. Nearest in vocabulary is Terin (2026), who reads neuron-wise rescaling as a gauge freedom and adds a soft orbit-selection penalty where our selection is the operator itself.

7 SCOPE, CLAIM TIERS, AND CONCLUSION

Claim tiers and limitations. The theorems are proved and checked symbolically and in float64, not in a proof assistant (Appendices C, E, G). They concern a fixed batch, a per-channel affine action, block-wise scale invariance and $\varepsilon = 0$, with non-homogeneous components outside the invariant subspace as in Arora et al. (2019). Proposition 3 assumes fixed weights and shift, not retraining or training-mode BN. The backward ablation covers one architecture and budget; its truncated Jacobian also lets weight norms drift, and its effect depends on input mean. The divisor test uses two batch sizes at 30 epochs; the other shares are small-scale and single-dataset; and ④ rejects only one noise proxy.

Conclusion. BN is a closed-form Kempf–Ness gauge section: its objective explains zero-variance failure, shift absorption, and why the sphere alone has an orthogonal backward pass among the divisors studied here. The four-payload ledger lists what a normalization-free design must supply. Only the gradient half of the first has a defensible measured share: 7.8 to 23.2 CIFAR-10 points under an unchanged forward pass.

ETHICS STATEMENT

This work is theoretical and uses only public benchmark datasets (`digits`, CIFAR-10). We see no ethical concerns beyond those of general machine-learning research.

REPRODUCIBILITY STATEMENT

All exact claims are checked symbolically and numerically by the scripts in the supplement (Appendix E); the suite runs on a CPU in under two minutes. The backward ablation of Section 4.3 is one self-contained script, `bn_backward_ablation.py`, which reproduces every arm of Table 6 from the seeds and grids recorded there; the divisor comparison of Proposition 2 is a second one, `bn_section_choice.py`, whose `-smax` flag recomputes Table 9 on a CPU in under a minute and whose training arms reproduce Table 10 from the seeds $\{3101, 3102\}$ and grids recorded there. On what was fixed when: the criteria of Sections 4.2 and 4.4 and the learning-rate table of Appendix H were fixed before the runs they judge, and we ship the file with its SHA-256 digests. Under anonymous review we cannot supply a third-party timestamp, so these should be read as criteria fixed before the final runs rather than as a registration, and two are weaker than that: the 30–50% band of Section 4.3 was drawn around the measurement it is compared with, and the learning-rate table was frozen only after an initial diagnostic round. Protocols, seeds and learning-rate tables are in Appendix H. Two precision notes, since they differ by experiment: the ④ runs of Section 4.4 use float64 under strict deterministic CUDA, with manifests and SHA-256 digests in the supplement, while the backward ablation of Section 4.3 trains in float32 and uses float64 only for the gradient-agreement check reported in Appendix H.3.

AI USE STATEMENT

In this work, we used generative AI tools to assist with the experimental work, with parts of the theoretical development, and to generate the first draft of the manuscript. We have reviewed all AI-assisted work: every theorem and proof was checked by the authors, all reported results were produced under the protocols recorded in the appendix and verified by the authors, and we checked the cited bibliography entries against primary records and revised source-dependent claims after a targeted attribution audit. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Sanjeev Arora, Zhiyuan Li, and Kaifeng Lyu. Theoretical analysis of auto rate-tuning by batch normalization. In *International Conference on Learning Representations (ICLR)*, 2019.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Andrew Brock, Soham De, Samuel L. Smith, and Karen Simonyan. High-performance large-scale image recognition without normalization. In *International Conference on Machine Learning (ICML)*, 2021.

Vitaliy Chiley, Ilya Sharapov, Atli Kosson, Urs Koster, Ryan Reece, Sofia Samaniego de la Fuente, Vishal Subbiah, and Michael James. Online normalization for training neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Minhyung Cho and Jaehyung Lee. Riemannian approach to batch normalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Hadi Daneshmand, Jonas Kohler, Francis Bach, Thomas Hofmann, and Aurelien Lucchi. Batch normalization provably avoids ranks collapse for randomly initialised deep networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Yann N. Dauphin and Ekin Dogus Cubuk. Deconstructing the regularization of BatchNorm. In *International Conference on Learning Representations (ICLR)*, 2021.

Soham De and Samuel L. Smith. Batch normalization biases residual blocks towards the identity function in deep networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Guillaume Desjardins, Karen Simonyan, Razvan Pascanu, and Koray Kavukcuoglu. Natural neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Paul A. M. Dirac. *Lectures on Quantum Mechanics*. Belfer Graduate School of Science, Yeshiva University, 1964.

Simon S. Du, Wei Hu, and Jason D. Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Byeongho Heo, Sanghyuk Chun, Seong Joon Oh, Dongyoon Han, Sangdoo Yun, Gyuwan Kim, Youngjung Uh, and Jung-Woo Ha. AdamP: Slowing down the slowdown for momentum optimizers on scale-invariant weights. In *International Conference on Learning Representations (ICLR)*, 2021.

Elad Hoffer, Itay Hubara, and Daniel Soudry. Train longer, generalize better: Closing the generalization gap in large batch training of neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Lei Huang, Dawei Yang, Bo Lang, and Jia Deng. Decorrelated batch normalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Lei Huang, Yi Zhou, Fan Zhu, Li Liu, and Ling Shao. Iterative normalization: Beyond standardization towards efficient whitening. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, pp. 448–456, 2015.

Zixuan Jiang, Jiaqi Gu, Hanqing Zhu, and David Z. Pan. Pre-RMSNorm and pre-CRMSNorm transformers: Equivalent and efficient pre-LN transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks. `https://kellerjordan.github.io/posts/muon/`, 2024.

Atli Kosson, Bettina Messmer, and Martin Jaggi. Rotational equilibrium: How weight decay balances learning across neural networks. In *International Conference on Machine Learning (ICML)*, 2024.

Daniel Kunin, Javier Sagastuy-Brena, Surya Ganguli, Daniel L. K. Yamins, and Hidenori Tanaka. Neural mechanics: Symmetry and broken conservation laws in deep learning dynamics. In *International Conference on Learning Representations (ICLR)*, 2021.

Zhiyuan Li and Sanjeev Arora. An exponential learning rate schedule for deep learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Kathryn Lindsey and Govind Menon. Regularization implies balancedness in the deep linear network. *arXiv preprint arXiv:2511.01137*, 2025.

Ekdeep Singh Lubana, Robert P. Dick, and Hidenori Tanaka. Beyond BatchNorm: Towards a unified understanding of normalization in deep learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Ping Luo, Xinjiang Wang, Wenqi Shao, and Zhanglin Peng. Towards understanding regularization in batch normalization. In *International Conference on Learning Representations (ICLR)*, 2019.

David Mumford, John Fogarty, and Frances Kirwan. *Geometric Invariant Theory*. Springer, 3rd edition, 1994.

Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. Data-free quantization through weight equalization and bias correction. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Simon Roburin, Yann de Mont-Marin, Andrei Bursuc, Renaud Marlet, Patrick Pérez, and Mathieu Aubry. Spherical perspective on learning with normalization layers. *Neurocomputing*, 487:66–74, 2022.

Tim Salimans and Diederik P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Mądry. How does batch normalization help optimization? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Jiacheng Sun, Xiangyong Cao, Hanwen Liang, Weiran Huang, Zewei Chen, and Zhenguo Li. New interpretations of normalization methods in deep learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

Rodrigo Carmo Terin. Scale redundancy and soft gauge fixing in positively homogeneous neural networks. *arXiv preprint arXiv:2602.14729*, 2026.

Mattias Teye, Hossein Azizpour, and Kevin Smith. Bayesian uncertainty estimation for batch normalized deep networks. In *International Conference on Machine Learning (ICML)*, 2018.

Twan van Laarhoven. L2 regularization versus batch and weight normalization. *arXiv preprint arXiv:1706.05350*, 2017.

Ruosi Wan, Zhanxing Zhu, Xiangyu Zhang, and Jian Sun. Spherical motion dynamics: Learning dynamics of normalized neural network using SGD and weight decay. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Yuxin Wu and Kaiming He. Group normalization. In *European Conference on Computer Vision (ECCV)*, 2018.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning (ICML)*, 2020.

Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. Understanding and improving layer normalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Junjie Yan, Ruosi Wan, Xiangyu Zhang, Wei Zhang, Yichen Wei, and Jian Sun. Towards stabilizing batch statistics in backward propagation of batch normalization. In *International Conference on Learning Representations (ICLR)*, 2020.

Greg Yang, Jeffrey Pennington, Vinay Rao, Jascha Sohl-Dickstein, and Samuel S. Schoenholz. A mean field theory of batch normalization. In *International Conference on Learning Representations (ICLR)*, 2019.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Jiachen Zhu, Xinlei Chen, Kaiming He, Yann LeCun, and Zhuang Liu. Transformers without normalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Liu Ziyin, Yizhou Xu, Tomaso Poggio, and Isaac Chuang. Parameter symmetry potentially unifies deep learning theory. *arXiv preprint arXiv:2502.05300*, 2025.