DEPTH-DEPENDENT SELECTION GEOMETRY FOR CERTIFIED STRUCTURED SPARSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Structured pruning deletes whole channels, attention heads or neurons, and a line of work makes such deletions happen during ordinary training by writing each weight as a product of D factors shared across a channel or head, so that weight decay drives whole groups to zero (*deep gating*). None of these methods says whether a particular deletion was justified: at a group whose factors are all zero the training objective is stationary whatever the data say. Its curvature, however, depends sharply on depth: with two factors the curvature at a deleted group, in that group's own coordinates, still sees the data, in exactly the shape of the group lasso's optimality condition, so for almost every initialization a deletion that survives to a convergent gradient-descent limit carries a certificate costing one gradient to check. With three or more, the data drop out of that curvature and a deleted group is locally stable in its own coordinates, justified or not. Because a network trained at one depth can be rewritten exactly at another, *Certified Progressive Gating* decides at depth two, records which decisions the certificate supports, and compresses deeper. On CIFAR-10 and CIFAR-100 it has the highest mean accuracy of eleven evaluated methods in five of six settings, on CIFAR-10 it matches a tuned dependency-aware pruner on two thirds of that pruner's training budget, on BERT-base it has the highest mean accuracy of six evaluated structured-pruning methods, within noise, and, to our knowledge, it is the only method that records which of its deletions satisfy their optimality condition.

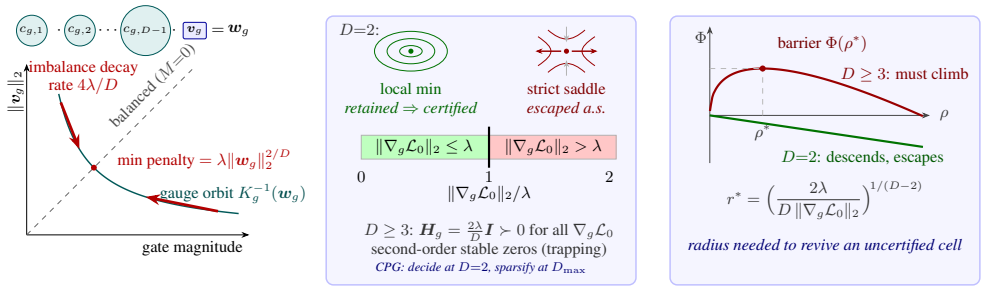
1 INTRODUCTION

Deep networks are routinely trained much larger than they need to be and made smaller before deployment. The form of shrinking that pays off on ordinary hardware is *structured*: deleting whole channels, attention heads or neurons leaves a genuinely smaller dense network (Hoefler et al., 2021). One strategy scores each channel or head after training and removes the lowest-scoring ones (Liu et al., 2017; He et al., 2017), with dependency-aware frameworks adding group-sparse training (Fang et al., 2023); another writes removal into the objective as a *group penalty* (Yuan & Lin, 2006; Wen et al., 2016), whose kink at zero is something stochastic gradient descent handles poorly. A productive line of work removes the kink by *reparametrizing*: write each weight as a product of D factors, and weight decay on the factors acts as a sparsity penalty on the product (Hoff, 2017; Ziyin & Wang, 2023; Kolb et al., 2025b; 2026). Its structured form, *D-Gating* (Kolb et al., 2025a), shares $D-1$ scalar gates across each channel or head and induces a group penalty of exponent $2/D$, whose minimizers, for a disjoint partition, correspond exactly to those of the reparametrized problem.

What none of this says is whether any *particular* deletion was a good one. For the reparametrized family the gap is sharp: deletions happen while training runs and are irreversible—once a group's factors all reach zero, memoryless SGD with coupled weight decay never brings it back (Remark 1 for momentum)—and the usual remedy of training to the end, searching for a threshold and fine-tuning is where roughly half of structured pruning's cost sits. The optimizer cannot help: a group whose factors are all zero is a stationary point of the training objective whatever the data say, since every data-gradient term is multiplied by a zeroed factor. The informative quantity, the risk gradient $\nabla_g \mathcal{L}_0$ on the group's collapsed weights, can reach a deleted group only through the *curvature*.

There the answer depends sharply on depth. A weight is a degree-D polynomial in its factors, and only at degree two does the data term curve at the origin. At $D{=}2$ it does so in exactly the shape

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) mechanism: gate stack & gauge fixing **(b) core result: the depth dichotomy** **(c) how large the $D \ge 3$ trap is**

Figure 1: **The depth-dependent selection geometry.** *(a)* A cell is $D-1$ shared gates and a carrier; weight decay contracts the gauge orbit onto the balanced manifold, where the penalty equals $\lambda\|\boldsymbol{w}_g\|_2^{2/D}$. *(b)* At $D{=}2$ the data gradient enters the group Hessian, so a removed cell's block has a downhill direction exactly when the group-lasso KKT condition fails; at $D \ge 3$ it disappears and the block is positive definite whatever the data say. *(c)* The leading-order barrier, of factor radius r^* (Section 3.3); CPG and other notation in Sections 3.1–3.4.

of the group lasso's optimality condition: with λ the penalty strength, a removed group is a strict saddle *in its own coordinates* precisely when $\|\nabla_g\mathcal{L}_0\|_2 \le \lambda$ fails, so for almost every initialization a removal still standing at a convergent gradient-descent limit satisfies it and carries a *certificate* costing one gradient to check. At $D \ge 3$ the data leave the Hessian: every removed group is a strict local minimum in its own coordinates regardless of the data, held—one cell at a time—inside a barrier whose leading-order size we give in closed form (Figure 1). *$D{=}2$ certifies a removal but shrinks what it keeps; $D \ge 3$ compresses hard but traps whatever reaches zero.*

Certified Progressive Gating (CPG), the method we propose, turns this dichotomy into a division of labor in which each depth does the job it suits (Figure 2). This is possible because depth is cheap to change: on the balanced manifold, which weight decay attracts, a stack of gates is fixed by a single magnitude, so a trained network can be rewritten at another depth *exactly*, by a closed-form re-lift that leaves the network function and the risk unchanged and matches the target penalty at the new exponent (Proposition 2). CPG first trains at $D{=}2$, where a removal the data contradict is a strict saddle that gradient descent escapes, and decides there, at the *boundary*: cells at the latent origin are committed as removals, and one gradient of the risk labels each removal certified or not in a per-removal *ledger*. It then re-lifts the survivors to $D{=}3$, where the more aggressive exponent $2/3$ prunes while shrinking the survivors far less than the group-lasso penalty, and removed cells stay removed because the latent origin is absorbing. **The certified stage decides; the deep stage compresses.** The pruning decision is thus taken during training, which removes the end-of-training threshold search, lets the second stage train an already smaller network and costs one extra gradient evaluation; and the output comes with a record of which removals the optimality condition supports, which, to our knowledge, no other structured pruner provides.

Our contributions. *First*, a second-order classification of removed cells in structured deep gating over disjoint partitions—the $D{=}2$ certificate and the $D \ge 3$ trapping barrier—that makes depth a discrete control over the reliability of a structural decision, not merely a regularization exponent. *Second*, Certified Progressive Gating: an exact re-lift between depths, a commitment rule that records the certificate for every boundary decision, and a two-stage algorithm (Section 3). *Third*, experiments in which CPG has the highest mean accuracy in five of six settings on CIFAR-10 and CIFAR-100, on CIFAR-10 matches a tuned dependency-aware pruner on two thirds of its budget, on BERT-base has the highest mean accuracy of six structured-pruning methods, within noise, and is, to our knowledge, the only method that records which of its deletions satisfy the optimality condition.

2 Related Work

Smooth reparametrizations turn weight decay into a sparsity penalty (Hoff, 2017; Ziyin & Wang, 2023; Kolb et al., 2025b; 2026); D-Gating (Kolb et al., 2025a) is the structured version we build on, and implicit-bias work studies related product parametrizations without regularization (Gunasekar et al., 2018; Woodworth et al., 2020). **Structured pruning** selects cells by a group penalty (Wen et al., 2016; Yuan & Lin, 2006), a scale parameter (Liu et al., 2017), layer reconstruction (He et al.,

2017) or a dependency-aware framework (Fang et al., 2023; Chen et al., 2023), and at LLM scale by prune-then-retrain (Xia et al., 2024; Muralidharan et al., 2024) or semi-structured mask learning (Fang et al., 2024); attachment grammars place deep gating in the generalized-lasso lineage (Fei et al., 2023; Bian et al., 2024; Tibshirani & Taylor, 2011). That a two-factor parametrization turns spurious stationary points into **strict saddles**, which gradient descent almost surely escapes, is known (Poon & Peyré, 2021; 2023; Boumal et al., 2016), and CPG is a **graduated non-convexity** scheme (Blake & Zisserman, 1987) related to gradual sparsity schedules (Zhu & Gupta, 2018). *Our position.* These works locate the minimizers of the gated problem or select structure by a criterion, relaxation or proximal step; to our knowledge none characterizes how the geometry at a zero depends on depth or attaches a per-removal optimality statement to its output, and unlike continuation, CPG changes depth by an exact re-lift rather than a re-solve (full discussion: Appendix A).

3 Deep Gating and the Geometry of a Removed Cell

3.1 Setup: the deep-gating parametrization

Let $\boldsymbol{w} \in \mathbb{R}^p$ be the p parameters of a network with empirical risk $\mathcal{L}_0(\boldsymbol{w})$, and let $\mathcal{G}$ be a disjoint partition of $[p] = \{1, \dots, p\}$ into $|\mathcal{G}|$ structural cells (channels, heads, neurons), with $\boldsymbol{w}_g \in \mathbb{R}^{|g|}$ the sub-vector of cell g. The target is the group quasi-norm problem

$$\min_{\boldsymbol{w}} \ \mathcal{L}_{w,\lambda}(\boldsymbol{w}) := \mathcal{L}_0(\boldsymbol{w}) + \lambda \sum_{g \in \mathcal{G}} \|\boldsymbol{w}_g\|_2^{2/D}, \qquad D \in \mathbb{N}_{\geq 2}, \tag{1}$$

with penalty strength $\lambda > 0$ and *depth* D, the number of factors whose product forms each weight; it is non-smooth wherever some $\boldsymbol{w}_g = \mathbf{0}$ and non-convex for $D > 2$ (Frank & Friedman, 1993). *Deep gating* (Kolb et al., 2025a) assigns each cell a carrier $\boldsymbol{v}_g \in \mathbb{R}^{|g|}$ and scalar gates $\boldsymbol{c}_g = (c_{g,1}, \dots, c_{g,D-1})$—the index d always ranges over $1, \dots, D-1$—with $\boldsymbol{w}_g = K_g(\boldsymbol{\omega}_g) := (\prod_d c_{g,d})\, \boldsymbol{v}_g$, and trains

$$\min_{\boldsymbol{\omega}} \ \mathcal{L}_\lambda(\boldsymbol{\omega}) := \mathcal{L}_0\big(K(\boldsymbol{\omega})\big) + \frac{\lambda}{D} \sum_{g \in \mathcal{G}} \Big(\sum_{d=1}^{D-1} c_{g,d}^2 + \|\boldsymbol{v}_g\|_2^2 \Big), \tag{2}$$

where $\boldsymbol{\omega} = (\boldsymbol{c}, \boldsymbol{v})$, with $\boldsymbol{\omega}_g = (\boldsymbol{c}_g, \boldsymbol{v}_g)$, collects the factors and $\boldsymbol{w} = K(\boldsymbol{\omega})$ is the collapsed parameter. Rescaling the factors against one another leaves the collapsed weight $\boldsymbol{w}_g$ untouched and changes only the penalty, so each nonzero $\boldsymbol{w}_g$ sits on a $(D-1)$-dimensional gauge orbit—and, because the gates are shared by the whole cell, an orbit along which $\|\boldsymbol{v}_g\|_2$ can move but its direction, up to sign, cannot. A factorization is *balanced* if $c_{g,1}^2 = \cdots = c_{g,D-1}^2 = \|\boldsymbol{v}_g\|_2^2$; the balanced representative with non-negative gates is the *canonical gauge*, and by AM–GM it is where the penalty on the orbit is smallest, equal there to $\lambda \|\boldsymbol{w}_g\|_2^{2/D}$ (Lemma 1), giving

$$\mathcal{L}_\lambda(\boldsymbol{\omega}) \ = \ \mathcal{L}_{w,\lambda}\big(K(\boldsymbol{\omega})\big) \ + \ \lambda\, M(\boldsymbol{\omega}), \qquad M(\boldsymbol{\omega}) \geq 0, \quad M(\boldsymbol{\omega}) = 0 \iff \boldsymbol{\omega} \text{ balanced}, \tag{3}$$

with M the gauge slack. The gated and target problems share their minimizers, imbalance decays as $e^{-4\lambda t/D}$ in gradient-flow time t (Figure 1a), and under memoryless SGD with coupled weight decay the *latent origin* $\boldsymbol{\omega}_g = \mathbf{0}$ is absorbing (Appendix C); none of this describes the zero state itself.

We write $\nabla_g \mathcal{L}_0 := \nabla_{\boldsymbol{w}_g} \mathcal{L}_0(K(\boldsymbol{\omega}))$ throughout—the risk gradient with respect to the *collapsed* weights of cell g—write $A_{\text{latent}}(\boldsymbol{\omega}) = \{g : \boldsymbol{\omega}_g \neq \mathbf{0}\}$ and $A_{\text{deploy}}(\boldsymbol{\omega}) = \{g : K_g(\boldsymbol{\omega}_g) \neq \mathbf{0}\}$ for the latent and deployed architectures, and distinguish the latent origin from a *collapsed zero* ($K_g(\boldsymbol{\omega}_g) = \mathbf{0}$ with $\boldsymbol{\omega}_g \neq \mathbf{0}$), which need not be absorbing; the re-lift of Section 3.4 converts the second into the first. Second-order statements assume $\mathcal{L}_0$ is C^2 near the points considered (for ReLU networks, away from the kink set), and certificates concern gradient descent with random initialization (Lee et al., 2016; 2019a); their behavior under SGD is measured rather than proved.

3.2 The depth dichotomy: when the data can question a deletion

At a configuration where cell g sits at its latent origin the penalty contributes a multiple of the identity, so everything depends on how the data term $\mathcal{L}_0 \circ K$ curves there—and that is decided by an algebraic degree count. $\boldsymbol{w}_g(\boldsymbol{\omega}_g) = (\prod_d c_{g,d})\boldsymbol{v}_g$ is homogeneous of degree D in the factors, so about a point $\bar{\boldsymbol{\omega}}$ with $\bar{\boldsymbol{\omega}}_g = \mathbf{0}$, the other coordinates frozen, we have $\mathcal{L}_0(K(\boldsymbol{\omega})) =$

$\mathcal{L}_0(K(\bar{\boldsymbol{\omega}})) + \langle \nabla_g \mathcal{L}_0, \boldsymbol{w}_g(\boldsymbol{\omega}_g)\rangle + O(\|\boldsymbol{\omega}_g\|_2^{2D})$: the data term enters at degree D, and only a degree-2 term contributes to a Hessian at $\mathbf{0}$. The data gradient therefore reaches $\nabla^2_{\boldsymbol{\omega}_g}(\mathcal{L}_0 \circ K)(\mathbf{0})$ when $D{=}2$ and cannot when $D \geq 3$, however large it is—this single algebraic fact drives everything below.

Theorem 1 (Second-order classification of removed groups)**.** *Let $\mathcal{G}$ be a disjoint partition and let $\bar{\boldsymbol{\omega}}_g = \mathbf{0}$. Because the Jacobian of K_g vanishes at $\boldsymbol{\omega}_g = \mathbf{0}$ (it is homogeneous of degree $D{-}1 \geq 1$), the Hessian of $\mathcal{L}_\lambda$ at $\bar{\boldsymbol{\omega}}$ has no mixed terms between group g and any other coordinate, whatever the coupling in $\mathcal{L}_0$, and its g-block, ordered $(c_{g,1}, \ldots, c_{g,D-1}, \boldsymbol{v}_g)$, is:*

$$\begin{aligned} &D = 2: \ \boldsymbol{H}_g = \begin{pmatrix} \lambda & \nabla_g \mathcal{L}_0^\top \\ \nabla_g \mathcal{L}_0 & \lambda \boldsymbol{I} \end{pmatrix}, \qquad \text{spec} = \{\lambda \pm \|\nabla_g \mathcal{L}_0\|_2,\ \lambda\ (\times |g|{-}1)\}; \\ &D \geq 3: \ \boldsymbol{H}_g = \tfrac{2\lambda}{D}\, \boldsymbol{I} \succ 0. \end{aligned} \tag{4}$$

Hence at $D = 2$, and at a point where $\bar{\boldsymbol{\omega}}$ is stationary for $\mathcal{L}_\lambda$, the removed group g is a strict saddle in its own group subspace *iff $\|\nabla_g \mathcal{L}_0\|_2 > \lambda$—precisely the violation of the non-smooth stationarity condition $\mathbf{0} \in \nabla_g \mathcal{L}_0 + \lambda\, \partial \|\cdot\|_2(\mathbf{0})$, where $\partial\|\cdot\|_2(\mathbf{0})$ is the subdifferential of the Euclidean norm at the origin, the closed unit ball. For $D \geq 3$ every removed group is* strictly *second-order stable in its group subspace for any $\nabla_g \mathcal{L}_0$ (Hessian $\frac{2\lambda}{D}\boldsymbol{I} \succ 0$), exactly mirroring the local minima of the non-convex $\ell_{2/D}$ target at zero. Both halves are statements about the g-block; a point at which one cell is second-order stable may still be a saddle of $\mathcal{L}_\lambda$ because of another cell.*

At $D{=}2$ the penalty puts λ on the diagonal of the cell's block and the data gradient sits off it, so the smallest eigenvalue of that block is $\lambda - \|\nabla_g \mathcal{L}_0\|_2$—negative, a downhill direction out of the zero, exactly when the data push harder than the penalty. At $D \geq 3$ the data term enters at too high a degree to reach a second derivative at all, leaving $\frac{2\lambda}{D}$ times the identity: uphill in every direction *within the cell's own coordinates*, whatever the data. The proof is in Appendix D.

Example 1 (Soft-thresholding from saddle geometry alone)**.** *One scalar cell w and $\mathcal{L}_0 = \frac{1}{2}(w{-}\mu)^2$ with $\mu \in \mathbb{R}$, so that $\nabla_g \mathcal{L}_0 = -\mu$ at $w = 0$. At $D{=}2$ the origin is a strict saddle iff $|\mu| > \lambda$: gradient descent leaves it exactly when soft-thresholding would keep the weight, and stays exactly when soft-thresholding would kill it (at $|\mu| = \lambda$ the origin is a degenerate but strict local minimizer, by a direct fourth-order check). This is the lasso's selection rule, obtained from saddle geometry with no proximal operator. At $D{=}3$ the same point has Hessian $\frac{2\lambda}{3}\boldsymbol{I} \succ 0$ for* every *μ, so a run started near zero stays there even when the global minimizer is far from it—trapping in its simplest instance, and the effect Section 4.3 measures at scale.*

Since D is an integer this is a discrete switch, not a continuous lowering of the exponent $q = 2/D$: at $D{=}2$ gradient descent performs the group-lasso KKT test without forming it, and from $D{=}3$ on a run that comes close to a zero is drawn into it whether or not the data would have kept the cell.

Corollary 1 (Certified pruning at depth two)**.** *Let $D = 2$ and suppose $\mathcal{L}_0$ is C^2 and bounded below, so that $\mathcal{L}_\lambda$ is coercive. Fix $c > \inf \mathcal{L}_\lambda$, so that $U_c := \{\mathcal{L}_\lambda < c\}$ is nonempty and $S_c := \{\boldsymbol{\omega} : \mathcal{L}_\lambda(\boldsymbol{\omega}) \leq c\}$ is compact; let $R := \sup\{\|\boldsymbol{\omega}\|_2 : \boldsymbol{\omega} \in S_c\}$, and let L_∇ be a Lipschitz constant of $\nabla\mathcal{L}_\lambda$ on the ball $B(\mathbf{0}, 3R)$. Run gradient descent on $\mathcal{L}_\lambda$ at a fixed step $\eta < 1/L_\nabla$ from an initialization drawn from a distribution absolutely continuous with respect to Lebesgue measure on U_c. Then for* almost every *initialization, at any convergent limit $\hat{\boldsymbol{\omega}}$ every group at the latent origin satisfies the exact groupwise KKT condition $\|\nabla_g \mathcal{L}_0(K(\hat{\boldsymbol{\omega}}))\|_2 \leq \lambda$ of the structured $q{=}1$ problem (Lee et al., 2016; 2019a); we call this its* certificate*. The certificate is a first-order statement about each removed group, rather than about the whole network, which is exactly the decision a pruner takes; ReLU networks fall outside the C^2 hypothesis, and for them Section 4.3 measures the condition directly. For $D \geq 3$ no such certificate exists: a convergent limit may hold $\boldsymbol{\omega}_g = \mathbf{0}$ with $\|\nabla_g \mathcal{L}_0(K(\hat{\boldsymbol{\omega}}))\|_2 > \lambda$ (an* uncertified *removal).*

So at a convergent $D{=}2$ limit the deletions still standing are those the data did not push hard enough to undo, which one gradient verifies: the *certificate ratio* $\gamma_g = \|\nabla_g \mathcal{L}_0\|_2/\lambda_g$ is at most one ($\lambda_g = \lambda$ unless (6) is used). No such guarantee exists at $D \geq 3$, where a deleted cell stays deleted whatever its gradient. At a finite horizon the test is $\|\nabla_g \mathcal{L}_0\|_2 \leq \lambda_g + \delta$ with slack $\delta \geq 0$ (Appendix E.3): a removal passing with $\delta > 0$ is *certificate-compatible* (Definition 1), and *certified* if $\delta = 0$.

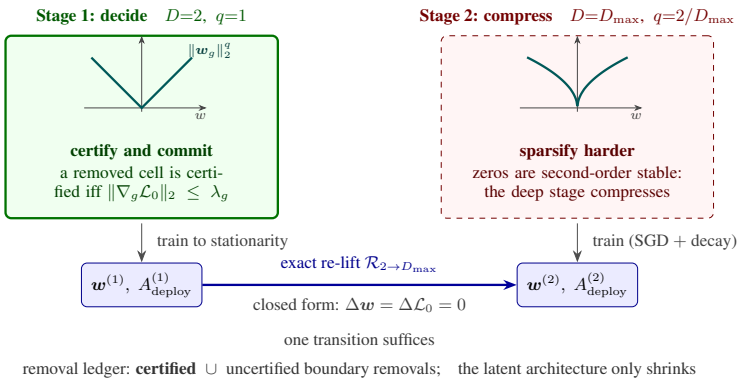


Figure 2: **Certified Progressive Gating.** Stage 1 trains at $D{=}2$ to stationarity (superscripts index stages); every latent zero is committed there and the certificate records which commitments it justifies. The exact re-lift (Proposition 2) rewrites the survivors at the target depth $D_{\max}$ in closed form, leaving $\boldsymbol{w}$ and the deployed architecture unchanged, so stage 2 starts warm and compresses.

3.3 The size of the $D \ge 3$ trap: how firmly a zero is held

Proposition 1 (Trapping barrier, $D \ge 3$)**.** *Let $D \ge 3$ and $\nabla_g \mathcal{L}_0 \ne \mathbf{0}$.* Fix all coordinates outside group g *and consider the one-cell subproblem $\boldsymbol{\omega}_g \mapsto \mathcal{L}_\lambda(\boldsymbol{\omega})$ around the latent origin. The cost of moving cell g away from it is bounded below (Lemma 1) by the* reduced profile *in the collapsed radius $\rho := \|\boldsymbol{w}_g\|_2$,*

$$\Phi(\rho) \;=\; \lambda\,\rho^{2/D} \;-\; \|\nabla_g \mathcal{L}_0\|_2\,\rho \;-\; \psi(\rho), \qquad \textit{for some } \psi \textit{ with } 0 \le \psi(\rho) \le \tfrac{L}{2}\rho^2,$$

with L a bound on $\|\nabla^2_{\boldsymbol{w}_g} \mathcal{L}_0\|$ near $\boldsymbol{w}_g = \mathbf{0}$. (i) First-order local model. *If $\mathcal{L}_0$ is affine in $\boldsymbol{w}_g$ with slope $\nabla_g \mathcal{L}_0$, then $\psi \equiv 0$, Φ increases on $(0, \rho^*)$ and decreases beyond it, and*

$$\textit{with } A := \tfrac{2\lambda}{D\|\nabla_g \mathcal{L}_0\|_2}: \qquad \rho^* = A^{\frac{D}{D-2}}, \qquad r^* := (\rho^*)^{1/D} = A^{\frac{1}{D-2}}, \qquad \Phi(\rho^*) = \tfrac{D-2}{D}\,\lambda\,A^{\frac{2}{D-2}},$$

where r^* is the corresponding balanced factor magnitude. The latent origin is then a strict local minimizer in the frozen subproblem, and every path from it to $\|\boldsymbol{w}_g\|_2 \ge \rho^*$ must raise $\mathcal{L}_\lambda$ by at least $\Phi(\rho^*)$, over the whole cell subspace with the other coordinates held fixed. (ii) Floor criterion. *A group initialized in canonical gauge at balanced factor magnitude ε (collapsed norm ε^D; the floor of Appendix E.2 is applied to the collapsed norm instead) starts inside the barrier iff $\|\nabla_g \mathcal{L}_0\|_2 < 2\lambda/(D\varepsilon^{D-2})$, to leading order.*

With the rest of the network held fixed, a cell at zero sits at the bottom of a well whose lip is at collapsed radius ρ^*, factor magnitude r^*, and leaving it costs $\Phi(\rho^*)$; a balanced cell started below r^* along $-\nabla_g \mathcal{L}_0$ returns to the origin (Lemma 3). Appendix D.7 proves this, bounds the error term—small when $L\,A^{2(D-1)/(D-2)} \ll \lambda$ with $A = 2\lambda/(D\|\nabla_g \mathcal{L}_0\|_2)$, in particular for $A \ll 1$ at fixed curvature, the regime the penalty is operated in—and shows the profile degenerating at $D{=}2$ to $(\lambda - \|\nabla_g \mathcal{L}_0\|_2)\rho$, recovering Theorem 1. The proposition explains at the landscape level why CPG decides at $D{=}2$ before deepening, and why deep element-wise factorizations need initialization floors (Kolb et al., 2025b), and r^*_g (r^* of cell g) is the radius beyond which an uncertified cell must be re-placed to escape, which the optional *revival* rule does (Appendix E.3).

3.4 Separating selection from sparsification

Selection and sparsification want different depths. $D{=}2$ **admits a groupwise KKT test but shrinks what it keeps**: the convex penalty that removes nuisance cells also shrinks the retained ones, and support recovery needs irrepresentable-type conditions (Zhao & Yu, 2006), while non-convex ℓ_q, $q < 1$, recovers sparse signals under weaker restricted-isometry conditions (Chartrand, 2007). $D \ge 3$ **sparsifies hard but traps**: the barrier holds a cell at zero whatever the data say, which is what makes a deep stage compress and what leaves no certificate for what it *chooses*. CPG separates the two (Figure 2), because on the balanced manifold a gate stack is determined by a single magnitude and depth can be changed *exactly*.

Proposition 2 (Exact canonical re-lift)**.** *Let $\boldsymbol{\omega}$ be* any *depth-D configuration with collapsed weights $\boldsymbol{w} = K(\boldsymbol{\omega})$. For any integer $D' \geq 2$ define the depth-D' lift $\mathcal{R}_{D\to D'}(\boldsymbol{\omega}) =: \boldsymbol{\omega}'$, acting groupwise so that $\boldsymbol{\omega}'_g$ depends only on $\boldsymbol{\omega}_g$, by*

$$\begin{aligned} \boldsymbol{w}_g \neq \mathbf{0}: \quad & c'_{g,d} = s_{g,d}\|\boldsymbol{w}_g\|_2^{1/D'} \ (d = 1, \ldots, D'-1), \\ & \boldsymbol{v}'_g = \Big(\textstyle\prod_d s_{g,d}\Big)\, \boldsymbol{w}_g\, \|\boldsymbol{w}_g\|_2^{(1-D')/D'}; \\ \boldsymbol{w}_g = \mathbf{0}: \quad & \boldsymbol{\omega}'_g = \mathbf{0}, \end{aligned} \tag{5}$$

for any signs $s_{g,d} \in \{\pm 1\}$ (we take $s_{g,d} = +1$, the canonical gauge). Then the re-lift is exact*: $K(\boldsymbol{\omega}') = \boldsymbol{w}$, so the network function, $\mathcal{L}_0$ and the deployed architecture A_{deploy} are unchanged, and the depth-D' objective at the new point equals the target objective at exponent $2/D'$. It is also* rigid*: $\boldsymbol{\omega}'$ is balanced with every factor magnitude fixed uniquely at $\|\boldsymbol{w}_g\|_2^{1/D'}$, only gate signs free, and with $s_{g,d} = +1$ it is the canonical gauge. And it never enlarges the latent architecture: $A_{\text{latent}}(\boldsymbol{\omega}') = A_{\text{deploy}}(\boldsymbol{\omega}) \subseteq A_{\text{latent}}(\boldsymbol{\omega})$, with equality exactly when $\boldsymbol{\omega}$ has no collapsed zeros, so the map converts every collapsed zero into a latent one.*

Give the $D'-1$ gates and the carrier of a cell the same magnitude, the D'-th root of the weight's norm, with the carrier along the weight; a cell with zero weight goes to the latent origin. The network computes exactly what it did before, with D' factors instead of D: a reparametrization, not a re-solve.

The method. CPG (Algorithm 1, Appendix E.1) has five steps. *Decide*: train at $D{=}2$ until stationary; cells within a commitment tolerance of the latent origin are removal candidates. *Certify*: label each candidate certified if $\|\nabla_g \mathcal{L}_0\|_2 \leq \lambda_g$ and uncertified otherwise. *Commit* every candidate permanently and record its label in a ledger. *Re-lift* the survivors to the target depth $D_{\max}$ by Proposition 2. *Compress*: train at $D_{\max}$ and remove what reaches zero there. In the default BOUNDARY mode, certification is a property of the recorded $D{=}2$ decision, not a requirement for committing the removal; subsequent $D \geq 3$ removals are explicitly uncertified (a STRICT variant that commits only certified cells is in Definition 1). Commitments are permanent because the re-lift maps $\mathbf{0} \mapsto \mathbf{0}$ and the latent origin is absorbing. In the operating regime $2\lambda_g/(3\|\nabla_g \mathcal{L}_0\|_2) < 1$, which includes every cell with $\|\nabla_g \mathcal{L}_0\|_2 > \lambda_g$, the barrier height of Proposition 1 increases with every step in D (Appendix D.7), so a deeper second stage compresses harder; we take $D_{\max} \in \{3, 4\}$, and 3 by default in the experiments (Appendix F.5).

A per-cell penalty. Cells differ by an order of magnitude in compute cost. Relaxing the budget constraint $\sum_g m_g \mathbb{1}[\boldsymbol{w}_g \neq \mathbf{0}] \leq b$, with m_g the multiply–accumulate operations (MACs) of cell g and b the budget, replaces the global multiplier by

$$\lambda_g \;=\; \lambda\, m_g/\bar{m}, \qquad \bar{m} = |\mathcal{G}|^{-1} \textstyle\sum_g m_g, \quad \text{with } m_g, \bar{m} \text{ fixed at initialization.} \tag{6}$$

The objective keeps the form $\mathcal{L}_0 + \sum_g \lambda_g \|\boldsymbol{w}_g\|_2^{2/D}$, so Theorem 1 and Corollary 1 hold verbatim with λ_g; it is the certificate being a *per-cell* condition that licenses a per-cell penalty. The per-cell penalty is part of CPG, not of any baseline, and Appendix E.4 separates its effect from the schedule's.

4 Experiments

We go from general to specific: how CPG compares with related methods (Section 4.1), what deciding at the boundary buys against controls (Section 4.2), whether the predicted geometry holds in trained networks (Section 4.3), and what the certificate is good for (Section 4.4).

Settings. A synthetic group-sparse regression with known support; ResNet-18/56 with channel gates on CIFAR-10/100; LeNet-5 on MNIST for theory checks; BERT-base on SST-2 and QNLI with head and FFN-neuron gates. Baselines—group lasso / SSL (Wen et al., 2016), spred (Ziyin & Wang, 2023), DWF (Kolb et al., 2025b), Hadamard (Kolb et al., 2026), Powerpropagation (Schwarz et al., 2021), Network Slimming (Liu et al., 2017), DepGraph (Fang et al., 2023), fixed-depth D-Gating (Kolb et al., 2025a), and on BERT first-order importance (Michel et al., 2019) and group-norm magnitude—are re-run in one pipeline with strengths swept on validation. Under the *landing* protocol, used for Table 1's baselines, the network is thresholded onto the target at the end and fine-tuned; under the *boundary* protocol CPG's $D{=}2$ decision is the pruning decision. Cost is counted in *dense-epoch equivalents*, charging a pruned-network epoch by its MACs (Appendix F.1).

Table 1: **Comparison with related methods.** Mean ± population std over 3 seeds (5 for the synthetic column, fixed-depth D-Gating on CIFAR-10 and CPG on CIFAR-10 ResNet-18) at the stated MAC reduction; synthetic: F_1 of the recovered support. Strengths are selected on validation, and no baseline uses the per-cell penalty (6). CIFAR: landing protocol for the baselines; CPG uses one validation-selected configuration per dataset (Appendix F.1). Bold: best; underline: second best.

	Synth.	CIFAR-10↑				CIFAR-100↑	
		ResNet-56		ResNet-18		ResNet-56	ResNet-18
Method	F_1 ↑	4×	8×	8×	16×	4×	8×
Group lasso / SSL	0.359	0.8081 ± .0066	0.6867 ± .0252	0.8357 ± .0116	0.7778 ± .0093	0.5294 ± .0118	0.5432 ± .0231
spred	0.194	0.8204 ± .0055	0.7288 ± .0103	0.8267 ± .0120	0.7331 ± .0269	0.4661 ± .0314	0.5525 ± .0175
DWF	0.215	0.8047 ± .0032	0.6784 ± .0289	0.7473 ± .0236	0.6215 ± .0180	0.4783 ± .0199	0.5211 ± .0335
Hadamard	0.218	0.7552 ± .0130	0.6531 ± .0129	0.6630 ± .0158	0.5488 ± .0054	0.4964 ± .0064	0.5343 ± .0212
Powerpropagation	0.189	0.8143 ± .0125	0.7040 ± .0186	0.7635 ± .0600	0.7207 ± .0174	0.4966 ± .0109	0.5717 ± .0051
Network Slimming	n/a	0.8352 ± .0103	0.7945 ± .0170	0.8472 ± .0116	0.8194 ± .0184	0.5273 ± .0080	0.5806 ± .0189
DepGraph, swept	n/a	**0.8563 ± .0021**	0.8269 ± .0057	0.8873 ± .0048	0.8687 ± .0058	0.5391 ± .0227	0.6293 ± .0088
Structured deep gating (shared gates; what CPG builds on)							
D-Gating, fixed D=2	0.370	0.8436 ± .0077	0.8191 ± .0166	0.8694 ± .0036	0.8154 ± .0068	0.5441 ± .0007	0.5892 ± .0203
D-Gating, fixed D=3	0.807	0.8407 ± .0130	0.8236 ± .0054	0.8569 ± .0127	0.7915 ± .0305	0.5247 ± .0253	0.6078 ± .0101
D-Gating, fixed D=4	0.430	0.8420 ± .0097	0.8087 ± .0171	0.8534 ± .0112	0.8034 ± .0095	0.5379 ± .0044	0.5819 ± .0060
CPG (ours)	**1.000**	0.8523 ± .0078	**0.8524 ± .0032**	**0.8959 ± .0023**	**0.8854 ± .0029**	**0.5577 ± .0175**	**0.6382 ± .0051**

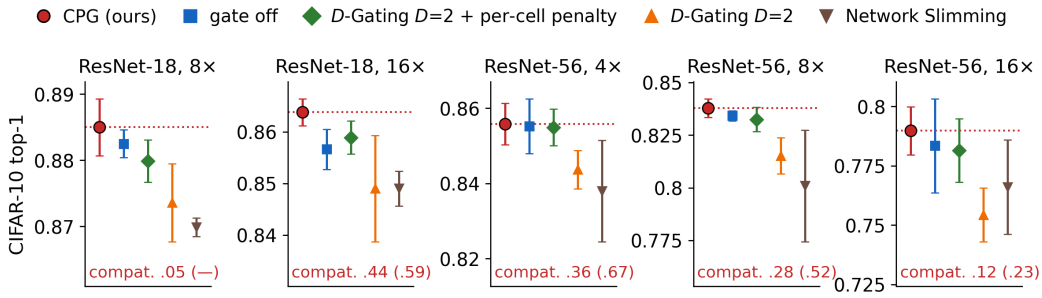


Figure 3: **Pruning at the boundary** (CIFAR-10, boundary protocol, 24–29 dense-epoch equivalents). Top-1, mean ± population std; dotted line: CPG. Controls: *gate off* (2→4, uniform penalty, no settle or certificate) and D-Gating D=2 with the per-cell penalty (6). *Compat.*: CPG's certificate-compatible share, plain risk (augmented objective in parentheses). Full table: Table 6.

4.1 Comparison with related methods

Table 1 compares CPG with ten methods. It is the only one that recovers the synthetic support exactly, it is ahead of fixed-depth D-Gating in every image column, by 0.009 to 0.070, and it has the highest mean accuracy in five of the six image settings, with DepGraph level with it at the sixth. DepGraph's group-sparsity strength is left to the user; sweeping it is worth two to five points of top-1 and makes DepGraph the strongest baseline, so we compare the two at two budgets. Given DepGraph's own budget, CPG is ahead at four of five CIFAR-10 settings (Table 1's four and ResNet-56 16×), by 0.009–0.048 (Welch $t = 2.4$–6.4), and level at the fifth ($t = -0.7$); on two thirds of that budget it matches DepGraph at four settings and leads at the fifth (Appendix F.1). The second stage's learning-rate multiplier for the longer budget comes from a grid added for this comparison and is selected on validation; Appendix F.1 details its selection, the runs without it, and how the budget is counted, which on MAC counts favors CPG while wall-clock time does not reverse the comparison. On CIFAR-100 CPG leads the best baseline at both cells, by 0.014 and 0.009 ($t = 1.1$ and 1.2 over three seeds), and at two further, higher-compression cells, by 0.012 and 0.017 (Appendix F.1). On BERT-base CPG has the highest mean accuracy of six structured-pruning methods, with differences within noise (Section 4.4). Unlike the others, it records which removals the certificate supports.

4.2 Deciding at the boundary

Figure 3 isolates the boundary decision. With no threshold search at the end of training and no fine-tuning of a still-dense network, CPG costs 24–29 dense-epoch equivalents against about 49 for the landing protocol, and at that cost it has the highest mean top-1 of the five at all five points.

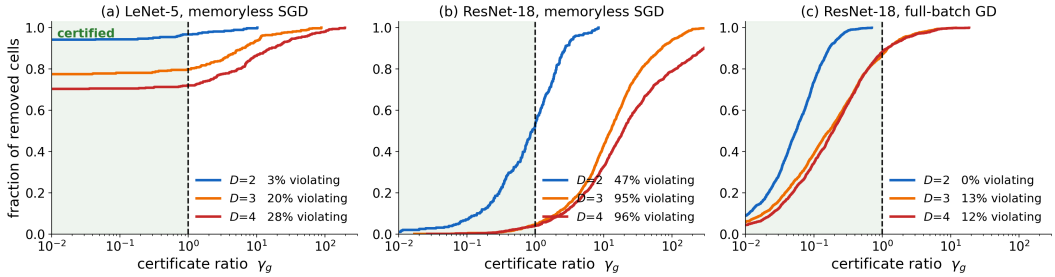


Figure 4: **The dichotomy in trained networks.** Cumulative distribution of the certificate ratio over removed cells; where a curve crosses the dashed line at 1 gives the certified fraction. *(a,b)* Memoryless SGD: $D{=}2$ violates the condition on 3% (LeNet-5) and 47% (ResNet-18) of removals, $D \geq 3$ on 20–28% and 95–96%. *(c)* Full-batch gradient descent, the optimizer setting of Corollary 1: $D{=}2$ violates it on none of 1366 removals, $D \geq 3$ on 12–13%.

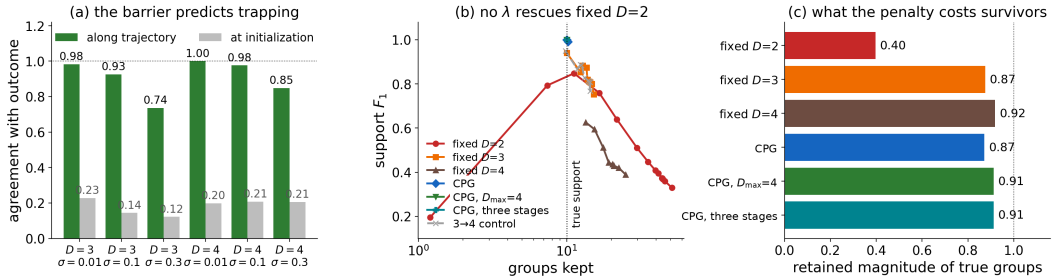


Figure 5: **The mechanism, on the synthetic problem.** *(a)* Read along the trajectory, the barrier of Proposition 1 predicts which cells a deep stage traps (σ: initialization scale). *(b)* Support F_1 against groups kept over a two-decade λ sweep: every schedule that decides at $D{=}2$ sits at 1.000, while fixed $D{=}2$ peaks at 0.85 at the correct sparsity and no fixed depth reaches 0.95. *(c)* At matched sparsity fixed $D{=}2$ keeps 0.40 of the true groups' magnitude, the progressive schedules 0.87–0.91.

Against fixed-depth gating with the *same* per-cell penalty it leads by $+0.001$ to $+0.008$, and under the landing protocol the schedule's margin over that control on ResNet-18 grows as the network narrows, reaching $+0.035$ at a quarter of the standard width (Appendix F.1): a thin block has no slack to absorb a removal made in error, which is where a certified decision pays most. Appendix E.4 separates the contributions of the penalty and the schedule. The run also yields a ledger: up to two thirds of CPG's removals are certificate-compatible under the gradient of the augmented objective (0.05–0.44 under the plain risk); on ResNet-18 an operating point with a majority of compatible removals stays within 0.001–0.011 of CPG's validation-selected top-1 (Table 6), and the share can be traded against accuracy along a frontier (Figure 6). Two design choices—a short large-batch *settle* before the boundary, and not re-admitting uncertified cells—are ablated in Appendix F.2. The pruned networks are faster in practice as well: at $16\times$ MAC reduction the ResNet-18 network runs $\approx 2.2\times$ faster on GPU at batch 512 and $\approx 5.7\times$ on a single CPU thread at batch 1 (Table 7).

4.3 Does the predicted geometry hold?

The Hessian and the certificate. On a problem small enough to form group Hessians exactly, the $D{=}2$ minimum eigenvalue tracks $\lambda - \|\nabla_g \mathcal{L}_0\|_2$ and crosses zero at the KKT threshold, while the $D \geq 3$ eigenvalue is flat at $2\lambda/D$ for every gradient, both to double precision across 6750 (cell, λ, D) points. The selection consequence carries over to real training (Figure 4): under memoryless SGD, $D{=}2$ satisfies the condition on far more removals than $D \geq 3$ (97% against 72–80% on LeNet-5, 53% against 4–5% on ResNet-18), and under full-batch gradient descent on ResNet-18—the optimizer setting of Corollary 1—it satisfies it on *every* removal, while $D \geq 3$ leaves over a tenth uncertified. The dichotomy belongs to the parametrization, not to the gradient estimator.

The barrier, permanence, and why depth must change. Proposition 1 holds in the frozen one-cell setting along the balanced ray, to numerical precision at $D{=}2$ and $D{=}4$ and within 4% at $D{=}3$. Read along the trajectory, it predicts the per-cell outcome for three quarters to all of the cells (Fig-

Table 2: **BERT-base, pruned at the boundary.** GLUE validation accuracy (mean ± population std over 3 seeds) at 2× and 4× encoder-MAC reduction, and the share of attention heads and FFN neurons each method removes (mean over tasks). Every row lands exactly on the target within one 4-epoch budget, strengths selected on held-out training data. Bold: best; underline: second best.

	SST-2		QNLI			Heads removed (%)		FFN removed (%)	
Method	2×	4×	2×	4×	**Avg.**	2×	4×	2×	4×
Dense (reference)	0.9190 ± .0038		0.8494 ± .0036		0.8842	0	0	0	0
First-order importance	0.9186 ± .0025	0.9193 ± .0046	<u>0.8495</u> ± .0037	<u>0.8503</u> ± .0045	0.8844	0	28	76	100
Magnitude (group ℓ_2)	<u>0.9224</u> ± .0029	0.9197 ± .0041	0.8467 ± .0039	0.8484 ± .0041	0.8843	0	28	76	100
Network Slimming	**0.9228 ± .0046**	0.9178 ± .0019	0.8493 ± .0046	0.8491 ± .0027	0.8848	70	81	40	72
DepGraph-style, swept	<u>0.9224</u> ± .0035	**0.9213 ± .0019**	**0.8497 ± .0036**	0.8481 ± .0044	<u>0.8854</u>	0	28	76	100
D-Gating, fixed D=2	0.9136 ± .0014	0.9132 ± .0029	0.8460 ± .0024	0.8456 ± .0037	0.8796	61	79	45	73
CPG (ours)	0.9221 ± .0029	<u>0.9203</u> ± .0018	<u>0.8495</u> ± .0020	**0.8511 ± .0036**	**0.8858**	77	85	37	72

ure 5a): no cell starts inside its barrier, and trapped cells enter one early, once others have captured the residual, a race that deciding at D=2 removes. Removals are also permanent in practice: 0 of 18,520 re-lifted latent zeros regrew under momentum. Nor can a larger λ at D=2 replace the change of depth: fixed D=2 peaks at the correct sparsity without recovering the support (Figure 5b), and by then keeps 0.40 of the true groups' magnitude against 0.87–0.91 for the progressive schedules (Figure 5c): one depth cannot both select reliably and preserve magnitude.

4.4 What the certificate is good for

As a control signal. The certificate ratio γ_g is defined for surviving cells as well as removed ones, and a small γ_g on a survivor says the data push weakly on it relative to the penalty—for a cell being kept, a reason to stop shrinking it. So at the re-lift we exclude from the second-stage penalty every survivor with $\gamma_g < 0.25$, a threshold fixed before the runs, and change nothing else. On ResNet-18 at 16× (landing protocol) this is worth +0.008 top-1 ($t = 3.80$), replicated on independent seeds; the gain concentrates where survivors are few and every one of them matters (Appendix F.5).

At transformer scale. The certificate is computable over the 37,008 heads and FFN neurons of BERT-base: converged D=2 gating satisfies it on 99.9% of its removals, and under the landing protocol 73% of the removals committed by a ledger-keeping CPG run compressing at D=4 are certificate-compatible when committed (Table 8). Against structured-pruning baselines at an exact encoder-MAC target under the boundary protocol (Table 2), CPG has the highest average accuracy, is best at QNLI 4× and within noise of the best elsewhere ($|t| \leq 0.6$), and matches or exceeds the dense model, while removing 77–85% of the attention heads, where importance, group-norm and DepGraph-style scores remove FFN neurons first; there cells are removed before reaching the latent origin, so the comparison is one of accuracy (Appendix F.4).

5 Conclusion

Factorization depth is usually read as a regularization exponent. We have shown that it is also a discrete control over the reliability of a structural decision: with two factors the curvature at a deleted group reproduces the group lasso's optimality condition, so a deletion standing at a convergent gradient-descent limit carries, for almost every initialization, a certificate that costs one gradient to check, while with three or more a deleted group is second-order stable in its own coordinates whatever the data say. Certified Progressive Gating uses each depth for what it does well, deciding and certifying at depth two and compressing deeper after an exact change of depth, without an end-of-training threshold search, and records which deletions the certificate supports.

Several directions extend the principle. Carrying the saddle-avoidance argument to momentum and adaptive optimizers would put the certified stage on the same footing under production training, as the BERT measurements already suggest. The per-cell penalty invites budgets beyond multiply–accumulate counts, such as latency or memory, each with its own certificate. Variance-reduced gradients would bring the certificate down to a single minibatch. And the certify-then-sparsify principle applies wherever a smooth factorization exposes an optimality condition at zero, including attachments beyond disjoint partitions (Remark 4).

AI USE STATEMENT

We used generative AI tools in three roles, the first two of which fall under ICLR's required-disclosure categories. *Experiment execution and analysis:* AI assistance ran the experimental campaign and analyzed its output—launching and sequencing runs from author-specified configurations, collecting results, computing the summary statistics and significance tests reported here, and producing the figures from the resulting data. Every experimental design decision was made by the authors, the protocol was fixed before the reported runs (with the one exception disclosed in Appendix F.1), and every number is generated from the archived result files rather than transcribed. *Proof development:* the theorems and propositions were formulated by the authors, and parts of their proofs were developed with AI assistance, including derivations such as the Hessian blocks of (4), the leading-order closed forms in Proposition 1 and the algebra of the re-lift, as well as the checking of intermediate steps and the write-up; the authors verified every proof. *Drafting and editing:* initial prose for parts of the abstract, introduction, section text and captions was produced with AI assistance and then rewritten and approved by the authors, together with copy-editing and consistency checking of terminology, cross-references and quoted numbers against the result tables.

Generative AI was not used to develop the theoretical model or the conceptual framing, to decide what the theorems should say, to propose or refine the hypotheses, or to design the methodology or the experimental protocol; the synthetic benchmark was generated by a standard statistical procedure specified by the authors, not by a generative model, and no dataset was cleaned, reformatted or translated with AI assistance. The authors have verified every derivation, independently re-derived the headline quantities, and checked each table cell against its source record; they are responsible for the correctness of each statement and its proof, and take full responsibility for the content of this work, including every claim, number and artifact produced with AI assistance.

ETHICS STATEMENT

This work studies the optimization geometry of structured sparsification and evaluates it on standard public benchmarks (MNIST, CIFAR-10, CIFAR-100, ImageNet-1k, and SST-2 and QNLI from GLUE) together with a synthetic regression problem of our own construction. It involves no human subjects, no user studies, no personally identifying or otherwise sensitive data, and no deployment to end users. We foresee no ethical risks beyond those already inherent in model compression generally, and we are not aware of any conflict of interest. The authors have read and adhere to the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

Every theoretical claim in the main text is stated with its assumptions in Section 3 and proved in full in Appendix D (a few short proofs in Appendix C); the practical refinements the implementation adds to Algorithm 1, and the finite tolerances under which the certificate is evaluated, are specified in Appendix E.3, with their values in Appendix F.6. Experimental protocols, dataset splits, selection rules, seed counts, the certificate-measurement procedure and the hardware used are given in Appendix F.6, with every configuration in the released code, and the full result tables behind the main-text results are in Appendix F. Source code for CPG, the baselines and the evaluation pipeline is provided as anonymized supplementary material for the reviewing process, and will be made public upon acceptance.

REFERENCES

Carlos M Alaíz and José R Dorronsoro. The generalized group lasso. In *International Joint Conference on Neural Networks*, pp. 1–8, 2015.

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Hedy Attouch, Jérôme Bolte, and Benar Fux Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods. *Mathematical Programming*, 137(1-2):91–129, 2013.

Ke Bian, Lu Sun, and Dengji Zhao. Learning compact neural networks via generalized structured sparsity. In *European Conference on Artificial Intelligence*, pp. 2605–2612, 2024.

Andrew Blake and Andrew Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, MA, 1987.

Jérôme Bolte, Aris Daniilidis, and Adrian Lewis. The łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, 17(4):1205–1223, 2007a.

Jérôme Bolte, Aris Daniilidis, Adrian Lewis, and Masahiro Shiota. Clarke subgradients of stratifiable functions. *SIAM Journal on Optimization*, 18(2):556–572, 2007b. doi: 10.1137/060670080.

Nicolas Boumal, Vlad Voroninski, and Afonso S. Bandeira. The non-convex Burer–Monteiro approach works on smooth semidefinite programs. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Samuel Burer and Renato D.C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003.

Rick Chartrand. Exact reconstruction of sparse signals via nonconvex minimization. *IEEE Signal Processing Letters*, 14(10):707–710, 2007.

Tianyi Chen, Luming Liang, Tianyu Ding, Zhihui Zhu, and Ilya Zharkov. OTOv2: Automatic, generic, user-friendly. In *International Conference on Learning Representations (ICLR)*, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

Gongfan Fang, Xinyin Ma, Mingli Song, Michael Bi Mi, and Xinchao Wang. DepGraph: Towards any structural pruning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16091–16101, 2023.

Gongfan Fang, Hongxu Yin, Saurav Muralidharan, Greg Heinrich, Jeff Pool, Jan Kautz, Pavlo Molchanov, and Xinchao Wang. MaskLLM: Learnable semi-structured sparsity for large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Luhuan Fei, Lu Sun, Mineichi Kudo, and Keigo Kimura. Structured sparse multi-task learning with generalized group lasso. In *European Conference on Artificial Intelligence*, pp. 692–699, 2023.

Luhuan Fei, Xinyi Wang, Jiankun Wang, Lu Sun, and Yuyao Zhang. Multi-level sparse network lasso: Locally sparse learning with flexible sample clusters. *Neurocomputing*, 635: 129898, 2025. ISSN 0925-2312. doi: 10.1016/j.neucom.2025.129898. URL `https://www.sciencedirect.com/science/article/pii/S0925231225005703`.

Ildiko E Frank and Jerome H Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, 35(2):109–135, 1993.

Daniel Gissin, Shai Shalev-Shwartz, and Amit Daniely. The implicit bias of depth: How incremental learning drives generalization. In *International Conference on Learning Representations (ICLR)*, 2020.

Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *IEEE International Conference on Computer Vision*, pp. 1389–1397, 2017.

Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *Journal of Machine Learning Research*, 22(241):1–124, 2021.

Peter D Hoff. Lasso, fractional norm and structured sparse estimation using a Hadamard product parametrization. *Computational Statistics & Data Analysis*, 115:186–198, 2017.

Lu Hou, Zhiqi Huang, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. DynaBERT: Dynamic BERT with adaptive width and depth. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Chris Kolb, Laetitia Frost, Bernd Bischl, and David Rügamer. Differentiable sparsity via D-gating: Simple and versatile structured penalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a. Spotlight.

Chris Kolb, Tobias Weber, Bernd Bischl, and David Rügamer. Deep weight factorization: Sparse learning through the lens of artificial symmetries. In *International Conference on Learning Representations*, 2025b.

Chris Kolb, Christian L Müller, Bernd Bischl, and David Rügamer. Smoothing the edges: Smooth optimization for sparse regularization using Hadamard overparametrization. *Machine Learning*, 115(4):87, 2026. doi: 10.1007/s10994-026-06997-0.

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

Krzysztof Kurdyka. On gradients of functions definable in o-minimal structures. *Annales de l'Institut Fourier*, 48(3):769–783, 1998. doi: 10.5802/aif.1638.

Jason D Lee, Max Simchowitz, Michael I Jordan, and Benjamin Recht. Gradient descent only converges to minimizers. In *Conference on Learning Theory*, pp. 1246–1257, 2016.

Jason D Lee, Ioannis Panageas, Georgios Piliouras, Max Simchowitz, Michael I Jordan, and Benjamin Recht. First-order methods almost always avoid strict saddle points. *Mathematical Programming*, 176:311–337, 2019a.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. SNIP: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*, 2019b.

Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *IEEE International Conference on Computer Vision*, pp. 2736–2744, 2017.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Saurav Muralidharan, Sharath Turuvekere Sreenivas, Raviraj Joshi, Marcin Chochowski, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, Jan Kautz, and Pavlo Molchanov. Compact language models via pruning and knowledge distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Wenqing Ouyang, Yuncheng Liu, Ting Kei Pong, and Hao Wang. Kurdyka–łojasiewicz exponent via Hadamard parametrization. *SIAM Journal on Optimization*, 35(1):62–91, 2025.

Clarice Poon and Gabriel Peyré. Smooth bilevel programming for sparse regularization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Clarice Poon and Gabriel Peyré. Smooth over-parameterized solvers for non-smooth structured optimization. *Mathematical Programming*, 201(1):897–952, 2023. doi: 10.1007/s10107-022-01923-3.

Victor Sanh, Thomas Wolf, and Alexander M Rush. Movement pruning: Adaptive sparsity by fine-tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Jonathan Schwarz, Siddhant M Jayakumar, Razvan Pascanu, Peter E Latham, and Yee Whye Teh. Powerpropagation: A sparsity inducing weight reparameterisation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Ryan J Tibshirani and Jonathan Taylor. The solution path of the generalized lasso. *The Annals of Statistics*, 39(3):1335–1371, 2011.

Tomas Vaskevicius, Varun Kanade, and Patrick Rebeschini. Implicit regularization for optimal sparse recovery. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Huan Wang, Can Qin, Yulun Zhang, and Yun Fu. Neural pruning via growing regularization. In *International Conference on Learning Representations*, 2021.

Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Blake Woodworth, Suriya Gunasekar, Jason D Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*, pp. 3635–3673, 2020.

Mengzhou Xia, Zexuan Zhong, and Danqi Chen. Structured pruning learns compact and accurate models. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. Sheared LLaMA: Accelerating language model pre-training via structured pruning. In *International Conference on Learning Representations (ICLR)*, 2024.

Lu Yin, You Wu, Zhenyu Zhang, Cheng-Yu Hsieh, Yaqing Wang, Yiling Jia, Gen Li, Ajay Kumar Jaiswal, Mykola Pechenizkiy, Yi Liang, Michael Bendersky, Zhangyang Wang, and Shiwei Liu. Outlier weighed layerwise sparsity (OWL): A missing secret sauce for pruning LLMs to high sparsity. In *International Conference on Machine Learning (ICML)*, 2024.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, 68(1):49–67, 2006.

Peng Zhao and Bin Yu. On model selection consistency of Lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006.

Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression. In *ICLR Workshop*, 2018.

Liu Ziyin and Zihao Wang. spred: Solving L_1 penalty with SGD. In *International Conference on Machine Learning*, pp. 43407–43422, 2023.

APPENDIX ROADMAP

The appendix follows the order of the paper: literature and background first, then theory, then the method, then experiments.

- **Appendix A, extended related work:** the full discussion behind Section 2.
- **Appendix B, background:** the generalized group lasso and generalized structured sparsity, the convex and multiplicative predecessors of deep gating, and where CPG sits among them (Table 3).
- **Appendix C, deferred statements for Section 3:** the variational representation and the exact correspondence of minimizers, gauge attraction and the absorbing latent origin, the certified-commitment definition and the optional revival rule, and the scope conditions on initialization and optimizers.
- **Appendix D, proofs** of Theorem 1, Corollary 1 and Propositions 1 and 2, of the deferred lemmas and the correspondence theorem, and a proof sketch of single-point convergence; a few short proofs stay with their statements in Appendix C.
- **Appendix E, CPG in detail:** Algorithm 1, the balanced initialization, the practical refinements (finite tolerances, the settle, optional revival), and a landing-protocol ablation that separates the per-cell penalty (6) from the schedule.
- **Appendix F, additional experiments,** in the order of Section 4: the comparison with related methods (F.1), the boundary protocol in full (F.2), the geometry checks (F.3), BERT (F.4), what makes CPG work (F.5), and protocols and hyperparameters (F.6).

Names used in the tables. Unless a row label or caption says otherwise, *CPG* is the method of Section 3.4: decide at $D{=}2$, compress at $D{=}3$, the per-cell penalty (6), a two-epoch settle before the boundary, no revival, BOUNDARY mode. Captions state the protocol and any configuration specific to a setting (the CIFAR-10 entries of Table 1, for instance, use DepGraph's budget, and the CIFAR-100 entries a $6{+}30$ split of the 36 epochs). An ablation is labeled by what it changes, and an indented row modifies the row above it: *w/o settle*, $+$ *revival*, *uniform* λ (a single λ in place of (6); these runs also compress at $D{=}4$), $D_{\max}{=}4$, *three stages* ($D{=}2 \to 3 \to 4$), *momentum throughout*, a different split of the 36 epochs between the two depths ($18{+}18$ unless stated), or *ledger kept* (the decision ledger recorded at slack $\delta = 0.05\lambda_g$; in BOUNDARY mode this removes the same cells). None of these is a separate method. Fixed-depth rows are D-Gating, with or without the per-cell penalty; *gate off* and $3{\to}4$ (no $D{=}2$ stage) are controls.

A EXTENDED RELATED WORK

Smooth reparametrizations for sparsity. Hadamard-product parametrizations turn weight decay into sparsity penalties: the lasso in statistics (Hoff, 2017) and *spred* (Ziyin & Wang, 2023). Related parametrizations obtain sparsity from the gradient dynamics instead, through a power reparametrization (Powerpropagation; Schwarz et al., 2021) or through small initialization and early stopping (Vaskevicius et al., 2019). Kolb et al. (2026) give an optimization-transfer theory with matching local minima and observe that for depth $k > 2$ the zero state acquires positive curvature; Deep Weight Factorization (Kolb et al., 2025b) studies element-wise depth-D factorization and the floors and learning rates it needs; and *D-Gating* (Kolb et al., 2025a) gives the structured shared-gate version we build on. Implicit-bias work (Gunasekar et al., 2018; Woodworth et al., 2020; Arora et al., 2019; Gissin et al., 2020) studies related product parametrizations without explicit regularization. *This line establishes where the minimizers of the gated problem are; we ask what the geometry at a zero is.* We give the shared-gate Hessian in closed form on both sides of the $D{=}2$ versus $D \geq 3$ boundary, identify the $D{=}2$ side with the groupwise KKT condition, derive the trapping barrier, and construct an exact re-lift between depths. To our knowledge this second-order classification of the removed state, and its use as a removal certificate, are not established in prior work.

Structured pruning. Group penalties (Wen et al., 2016; Yuan & Lin, 2006), scale parameters (Liu et al., 2017), layer reconstruction (He et al., 2017), growing regularization (Wang et al., 2021), dependency-aware frameworks (Fang et al., 2023; Chen et al., 2023), and transformer and LLM pruning by prune-then-retrain (Xia et al., 2024; Muralidharan et al., 2024), mask learning (Xia et al., 2022) or distillation-based width adaptation (Hou et al., 2020) target the structures we gate; saliency at initialization (Lee et al., 2019b), movement pruning (Sanh et al., 2020) and layerwise sparsity

allocation (Yin et al., 2024) and N:M mask learning (Fang et al., 2024) act on finer units. GenGL (Fei et al., 2023), building on the generalized lasso (Tibshirani & Taylor, 2011) and its $\ell_{2,1}$ group version (Alaíz & Dorronsoro, 2015), supplies an attachment grammar, and generalized structured sparsity (Bian et al., 2024) proves an equivalence with a two-factor multiplicative decomposition solved proximally (see also Fei et al., 2025); Appendix B positions CPG in this lineage. *Every method here selects structure by a criterion, a relaxation or a proximal step; to our knowledge none attaches a per-removal optimality statement to its output.*

Strict saddles and continuation. That a two-factor parametrization turns spurious stationary points of a non-smooth objective into strict saddles is known (Poon & Peyré, 2021; 2023) and underlies Burer–Monteiro factorization (Burer & Monteiro, 2003; Boumal et al., 2016). Theorem 1 gives the shared-gate spectrum with the KKT threshold exact and shows the mechanism *disappears* beyond two factors, which is what turns depth into a usable control. CPG is a graduated non-convexity scheme (Blake & Zisserman, 1987) in $q = 2/D$, related to gradual sparsity schedules (Zhu & Gupta, 2018) and to continuation in the exponent of an ℓ_q penalty, but it differs in two ways: the stage transition is an exact closed-form re-lift rather than a re-solve, and the first stage is the only one whose zeros carry a certificate. The distinction is measurable—proximal continuation over the same exponents does not reproduce the support recovery while a fixed-exponent solver at $q = 1/2$ does, so the gain comes from entering the deep objective on a support selected under the $D{=}2$ geometry, not from the sequence of exponents (Appendix F.5).

B Background: Generalized Group Lasso and Generalized Structured Sparsity

Deep gating compiles two lines of structured-sparsity regularization into weight decay on gate stacks. This appendix states them self-containedly: the Generalized Group Lasso (GenGL) grammar (Fei et al., 2023) that deep gating uses as its *specification language*, and Generalized Structured Sparsity (GSS) (Bian et al., 2024), whose equivalence between a structured penalty and a *multiplicative weight decomposition* is the direct precursor of deep gating's landscape theorem. Both generalize the group lasso.

Generalized lasso and generalized group lasso.

Generalized lasso. For weights $\boldsymbol{w}$ and an analysis (penalty) matrix $\boldsymbol{D}$, the generalized lasso (Tibshirani & Taylor, 2011) regularizes $\Omega(\boldsymbol{w}) = \gamma\|\boldsymbol{D}\boldsymbol{w}\|_1$: $\boldsymbol{D} = \boldsymbol{I}$ is the lasso, first-difference $\boldsymbol{D}$ is the fused lasso / total variation, and higher-order differences give trend filtering. Replacing ℓ_1 by the $\ell_{2,1}$ norm (Alaíz & Dorronsoro, 2015) lifts this to the group level, recovering the group lasso ($\boldsymbol{D} = \boldsymbol{I}$) and group fused lasso.

Generalized group lasso. GenGL (Fei et al., 2023) makes the operator a first-class, per-mode object. For a weight tensor $\boldsymbol{W}$ with mode i of size p_i, it introduces partitions of the index set $[p_i]$ and their projection operators

$$\pi_i \in \begin{cases} \{\boldsymbol{I}_{p_i}\} & (\mathcal{P}_C,\ \text{coarsest: keep the whole mode}), \\ \{\boldsymbol{e}_j : j \in [p_i]\} & (\mathcal{P}_F,\ \text{finest: select index } j), \\ \{\boldsymbol{e}_j - \boldsymbol{e}_k : j < k,\ j,k \in [p_i]\} & (\mathcal{P}_{F^2},\ \text{pairwise: cluster } j,k), \end{cases} \tag{7}$$

where $\boldsymbol{e}_j$ is a unit vector. A selection operator $\pi_i = \boldsymbol{e}_j$ penalizes $\|\boldsymbol{W} \times_i \boldsymbol{e}_j^\top\|_F$—group sparsity along mode i—whereas a difference operator $\pi_i = \boldsymbol{e}_j - \boldsymbol{e}_k$ penalizes $\|\boldsymbol{W} \times_i (\boldsymbol{e}_j - \boldsymbol{e}_k)^\top\|_F$, a fused / clustering effect. Taking one operator per mode, the GenGL penalty is a sum of group-ℓ_2 norms of linearly transformed weights,

$$\Omega(\boldsymbol{W}) = \gamma \sum_{\pi_1 \in \mathcal{P}_1} \cdots \sum_{\pi_N \in \mathcal{P}_N} \left\|\boldsymbol{W} \times_1 \pi_1^\top \cdots \times_N \pi_N^\top\right\|_F = \gamma \sum_{\pi} \|\boldsymbol{D}_\pi \operatorname{vec}(\boldsymbol{W})\|_2, \\ \boldsymbol{D}_\pi = \pi_N^\top \otimes \cdots \otimes \pi_1^\top. \tag{8}$$

Two composition rules complete the grammar: the weight may be decomposed into h components either *additively*, $\boldsymbol{W} = \sum_l \boldsymbol{W}_l$ (multi-component / latent decomposition), or *multiplicatively*, $\boldsymbol{W} = \prod_l \boldsymbol{W}_l$ (multi-level lasso and deep models), each component carrying its own operators $\{\pi_{l,i}\}$ and strength γ_l, yielding hierarchical sparsity. Unstructured sparsity ($\mathcal{P}_F$ on every mode), channel /

neuron / head group sparsity ($\mathcal{P}_F$ on the channel, neuron or head mode, $\mathcal{P}_C$ on the others), and fused / clustering ($\mathcal{P}_{F^2}$) are all corners of this single specification.

Generalized structured sparsity and multiplicative decomposition.

GSS (Bian et al., 2024) targets *multi-level* (nested) grouping. With a first-level partition $\mathcal{G}$ and, for each $g \in \mathcal{G}$, a second-level partition $\mathcal{G}_g$,

$$\Omega_{\mathrm{GSS}}(\boldsymbol{w}) = \sum_{g\in\mathcal{G}} \Big(\sum_{g'\in\mathcal{G}_g} \|\boldsymbol{w}_{g,g'}\|_2 \Big)^2, \tag{9}$$

an ℓ_2-ℓ_1-ℓ_2 hierarchy that alternates inter-group cooperation with intra-group competition across levels; group lasso ($|\mathcal{G}| = 1$, in squared form), exclusive lasso ($|g'| = 1$), and the squared ℓ_1/ℓ_2 norms are special cases. A general form parametrized by $p, q > 0$ interpolates convex and *non-convex* regularizers.

Equivalence to multiplicative decomposition (the precursor to deep gating). Bian et al. (2024) prove that regularizing the factors of a multiplicative decomposition $\boldsymbol{w}_{g,g'} = \alpha_{g,g'}\,\tilde{\boldsymbol{w}}_{g,g'}$,

$$\min_{\alpha,\tilde{\boldsymbol{w}}} \ \mathcal{L}(\boldsymbol{w}) + \lambda \sum_{g\in\mathcal{G}} \sum_{g'\in\mathcal{G}_g} \|\tilde{\boldsymbol{w}}_{g,g'}\|_p^p \quad \text{s.t.} \quad \boldsymbol{w}_{g,g'} = \alpha_{g,g'}\,\tilde{\boldsymbol{w}}_{g,g'},\ \|\alpha_g\|_q^q = 1,\ \alpha_g \geq 0, \tag{10}$$

is *equivalent* to penalizing $\boldsymbol{w}$ by the general (non-convex) GSS regularizer (their Theorem 1). This variational identity—a structured penalty realized by decaying the factors of a product parametrization—is the direct ancestor of Lemma 1 and Theorem 2. Multi-level sparse network lasso (Fei et al., 2025) uses the same principle, splitting each parameter into a coarse group-level factor and a fine entry-level factor, $\boldsymbol{\theta}_\ell = \alpha_\ell\,\boldsymbol{\beta}_\ell$, and again proving equivalence to a jointly regularized model.

Where deep gating and CPG sit in this lineage.

Deep gating (Kolb et al., 2025a) takes the GSS multiplicative-decomposition equivalence and makes it deep: the two-factor product $\alpha\,\tilde{\boldsymbol{w}}$ becomes a depth-D stack of shared scalar gates, the induced exponent is any $2/D$, and the gauge acts on one radial coordinate per group, which is the source of the exact correspondence of minimizers (Theorem 2), the distribution-matching initialization and the vanishing overhead. CPG adds the second-order account of the removed state that, to our knowledge, no proximal or gated treatment provides—the D=2 strict-saddle/KKT classification and its disappearance at $D \geq 3$ (Theorem 1), the trapping barrier (Proposition 1), the exact canonical re-lift (Proposition 2)—and the two-stage certify-then-sparsify algorithm built on them. GenGL supplies the *what* (which structures); GSS and deep gating supply the *how* (a multiplicative, then deep, realization); CPG supplies the *when to trust a removal* (Table 3).

Table 3: **Positioning.** CPG against deep gating and their shared predecessors in the generalized-(group-)lasso / structured-sparsity lineage. "Decomposition" is how the parameters are factorized; "Exponent q" is the induced penalty exponent (convex at q=1, non-convex for q<1); "Cert." is a per-removal groupwise KKT certificate for removals made by plain gradient training of a factorized network (a convex group-lasso solution satisfies the KKT conditions by construction). Deep gating supplies the depth-tunable exponent $q = 2/D$ under plain SGD with weight decay; CPG adds the certificate and the certified continuation. Marks record what each paper establishes.

Method	Decomposition	Depth	Exponent q	Optimization	Cert.
Generalized lasso (Tibshirani & Taylor, 2011)	none (analysis $\boldsymbol{D}\boldsymbol{w}$)	—	1 (convex)	dual path / proximal	✗
Group lasso (Yuan & Lin, 2006)	none (predefined groups)	—	1 (convex)	proximal / block-CD	✗
GenGL (Fei et al., 2023)	additive *or* multiplicative	h comp.	1 (convex $\ell_{2,1}$)	smoothing proximal gradient	✗
GSS (Bian et al., 2024)	multiplicative (2 factors)	2	≤ 1 (convex & non-convex)	proximal / alternating	✗
MSN Lasso (Fei et al., 2025)	multiplicative (group $\times$ entry)	2	$1/2$ at k=p=1 (+ network lasso)	alternating / smoothing prox.	✗
Hadamard overparam. (Kolb et al., 2026)	multiplicative (element-wise, k factors)	$k \geq 2$	$2/k$	SGD + weight decay	✗
DWF (Kolb et al., 2025b)	multiplicative (element-wise, depth D)	$D \geq 2$	$2/D$	SGD + weight decay	✗
D-Gating (Kolb et al., 2025a)	multiplicative, *shared depth-D gates*	$D \geq 2$	$2/D$ (any)	SGD + weight decay	✗
CPG (ours)	same parametrization as D-Gating	$2 \to D_{\max}$	$1 \to 2/D_{\max}$	**SGD + weight decay**	✓ (D=2)

C Deferred Statements and Material for Section 3

latent zero	$\boldsymbol{\omega}_g = \mathbf{0}$; absorbing under memoryless SGD, what the theory tracks
collapsed zero	$K_g(\boldsymbol{\omega}_g) = \mathbf{0}$ with $\boldsymbol{\omega}_g \neq \mathbf{0}$; not necessarily absorbing
certified removal	a latent zero with $\|\nabla_g \mathcal{L}_0\|_2 \leq \lambda_g$ (Corollary 1)
certificate-compatible	the same test at finite training tolerance, $\leq \lambda_g + \delta$
trapped removal	a $D \geq 3$ zero that violates the test but is held by the barrier (Proposition 1)
thresholded removal	a deployment-time hard prune, carrying no certificate

This appendix collects the statements referenced from Section 3 — prior results restated in our notation (Kolb et al., 2025a; 2026) and the supporting lemmas for Certified Progressive Gating — followed by the initialization and optimizer notes, the scope of the attachments covered, and the discrete-time counterparts of the gauge results. Proofs are in Appendix D.

Lemma 1 (Variational representation)**.** *For every $\boldsymbol{w}_g$ and $D \geq 2$, $\min_{(\prod_d c_{g,d})\boldsymbol{v}_g = \boldsymbol{w}_g} \frac{1}{D}\big(\sum_d c_{g,d}^2 + \|\boldsymbol{v}_g\|_2^2\big) = \|\boldsymbol{w}_g\|_2^{2/D}$, attained exactly at balanced factorizations. Hence, with the* gauge slack *$M(\boldsymbol{\omega}) := \frac{1}{D}\sum_g(\sum_d c_{g,d}^2 + \|\boldsymbol{v}_g\|_2^2) - \sum_g \|K_g(\boldsymbol{\omega}_g)\|_2^{2/D} \geq 0$,*

$$\mathcal{L}_\lambda(\boldsymbol{\omega}) \;=\; \mathcal{L}_{w,\lambda}\big(K(\boldsymbol{\omega})\big) \;+\; \lambda\, M(\boldsymbol{\omega}), \qquad M(\boldsymbol{\omega}) = 0 \iff \boldsymbol{\omega} \text{ balanced}. \tag{11}$$

Theorem 2 (Exact correspondence of minimizers)**.** *Let $D \geq 2$ and $\mathcal{G}$ a disjoint partition.* (i) *If $\hat{\boldsymbol{w}}$ is a local (global) minimizer of* (1)*, every balanced $\hat{\boldsymbol{\omega}} \in K^{-1}(\hat{\boldsymbol{w}})$ is a local (global) minimizer of* (2)*.* (ii) *Every local (global) minimizer $\hat{\boldsymbol{\omega}}$ of* (2) *is balanced and $K(\hat{\boldsymbol{\omega}})$ is a local (global) minimizer of* (1)*. Objective values coincide in both directions.*

Lemma 2 (Exponential gauge attraction; absorbing origin)**.** (a) *If $\mathcal{L}_0$ is C^1 with locally Lipschitz gradient and bounded below, then along gradient flow on* (2) *every pairwise imbalance within a group obeys $\frac{d}{dt}(c_{g,d}^2 - \|\boldsymbol{v}_g\|_2^2) = -\frac{4\lambda}{D}(c_{g,d}^2 - \|\boldsymbol{v}_g\|_2^2)$, so imbalances decay as $e^{-4\lambda t/D}$: the balanced manifold is globally exponentially attracting* in the imbalance coordinates *(directions and sign patterns are unconstrained).* (b) *Under memoryless SGD with coupled weight decay, $\boldsymbol{\omega}^{(t+1)} = \boldsymbol{\omega}^{(t)} - \eta_t \nabla \mathcal{L}_\lambda(\boldsymbol{\omega}^{(t)}; \xi_t)$, if $\boldsymbol{\omega}_g^{(t)} = \mathbf{0}$ then $\boldsymbol{\omega}_g^{(t')} = \mathbf{0}$ for all $t' > t$, for any data, loss and step sizes: the latent origin is absorbing and $A_{\text{latent}}(\boldsymbol{\omega}^{(t)})$ is non-increasing.*

Remark 1 (Forward invariance with momentum)**.** *Part (b) holds because both the data gradient (through K) and the coupled decay vanish at $\boldsymbol{\omega}_g = \mathbf{0}$; an optimizer with state inherits it once its state is handled at the origin. Under heavy ball with buffer $\boldsymbol{m}_g$, a group reaching the origin with $\boldsymbol{m}_g \neq \mathbf{0}$ is displaced by $-\eta\beta\boldsymbol{m}_g$ at the next step. Either of two exact measures achieves this: run the cells memorylessly, or zero $\boldsymbol{m}_g$ when a group is absorbed, which we do whenever momentum is used. Part (b) concerns the latent origin once reached (reaching it is measured in Appendix F.3), and a* collapsed *zero ($K_g(\boldsymbol{\omega}_g) = \mathbf{0}$, $\boldsymbol{\omega}_g \neq \mathbf{0}$) is a distinct state that can regrow (Remark 3); our absorption measurements therefore track the full latent state.*

Remark 2 (Exact versus practical certification)**.** *Corollary 1 concerns an exact latent zero at a convergent GD limit and the exact condition $\|\nabla_g \mathcal{L}_0\|_2 \leq \lambda_g$. The algorithm uses the* tolerance *test for the latent origin ($\|\boldsymbol{\omega}_g\|_2 \leq \varepsilon_c \bar{r}$) and the* practical *certificate with slack δ; practical commitments are approximate certificates that become exact as $\varepsilon_c, \delta, \tau \to 0$, with τ the stationarity tolerance of stage* 1*. Under SGD, alignment with the certificate is an empirical finding (Section 4.3).*

Definition 1 (Certified commitment, and the optional revival variant)**.** *At the D=2 stage boundary, with commitment tolerance ε_c and slack $\delta \geq 0$, each group g with $\|\boldsymbol{\omega}_g\|_2 \leq \varepsilon_c \bar{r}$—at a balanced point $\|\boldsymbol{\omega}_g\|_2 = \sqrt{D}$ times the factor magnitude, so the constant absorbs a $\sqrt{D}$—is classified by its certificate ratio $\gamma_g := \|\nabla_g \mathcal{L}_0(K(\boldsymbol{\omega}))\|_2/\lambda_g$, and is recorded as* certified *(*certificate-compatible *when $\delta > 0$) iff $\gamma_g \leq 1 + \delta/\lambda_g$ and as* uncertified *otherwise. In* BOUNDARY *mode (Algorithm 1, the default) it is committed either way (set $\boldsymbol{\omega}_g \leftarrow \mathbf{0}$ exactly; absorbing by Lemma 2(b)); in* STRICT *mode only certified cells are committed and uncertified ones return to the network (an exact or collapsed zero, which the re-lift sends to $\mathbf{0}$, can leave only through revival).* The optional revival variant *(revival margin $\kappa > 1$) instead returns an uncertified near-zero to the network: with $\boldsymbol{u}_g = -\nabla_g \mathcal{L}_0/\|\nabla_g \mathcal{L}_0\|_2$ the steepest-descent direction of the collapsed weight (well defined, since $\|\nabla_g \mathcal{L}_0\|_2 > \lambda_g + \delta > 0$), set $\bar{\boldsymbol{w}}_g = r_g^{D_{\max}} \boldsymbol{u}_g$ with factor radius $r_g = \min\{\max\{\varepsilon, \kappa r_g^*\},\ \bar{r}\}$, $r_g^* = (2\lambda_g/(D_{\max}\|\nabla_g \mathcal{L}_0\|_2))^{1/(D_{\max}-2)}$ and $\bar{r}$ the median depth-$D_{\max}$ factor magnitude $\|\boldsymbol{w}_g\|_2^{1/D_{\max}}$ of the cells not at the latent origin, i.e. the factor magnitude*

the survivors have after $\mathcal{R}_{2\to D_{\max}}$ (taken as $+\infty$, i.e. no cap, if that set is empty; the commitment tolerance above uses the same median at depth 2), re-lift it with the survivors, and log the violation. The cap keeps the revived cell inside the regime in which Proposition 1 is accurate and guards against divergence on cells whose certificate fails only marginally. Revival can release only cells with $\bar{r} > r_g^*$; when $\bar{r} < r_g^*$ the capped cell lies inside *its own barrier and, by Lemma 3, returns to zero. Revival is not used by default (Appendix E.3). Groups with $\|\boldsymbol{\omega}_g\|_2 > \varepsilon_{\mathrm{c}}\bar{r}$ are survivors and are re-lifted by $\mathcal{R}_{2\to D_{\max}}$, which fixes the latent origin and carries a collapsed zero to it. At any later boundary ($D \geq 3$) no certificate exists, and a group is committed on the numerical condition $\|\boldsymbol{\omega}_g\|_2 \leq \varepsilon_{\mathrm{c}}\bar{r}$ alone, which we label an* uncertified *removal.*

Lemma 3 (Revival escapes the trap)**.** *In the frozen first-order model of Proposition 1(i) at depth $D \geq 3$, consider a balanced group with factor magnitude m and carrier direction $\boldsymbol{u}_g = -\nabla_g\mathcal{L}_0/\|\nabla_g\mathcal{L}_0\|_2$. Gradient flow on* (2) *preserves balance and direction, and the common factor magnitude obeys $\dot{m} = \|\nabla_g\mathcal{L}_0\|_2\, m^{D-1} - \frac{2\lambda_g}{D}\, m$, which is positive iff $m > r_g^*$. Hence a group placed at any $m > r_g^*$—at $r_g \geq \kappa r_g^*$ with $\kappa > 1$, or at the cap $\bar{r}$ when $r_g^* < \bar{r} < \kappa r_g^*$—moves monotonically away from the latent origin (in this frozen model, for $D \geq 3$, without bound, which is what the cap addresses), while one placed at $m < r_g^*$ returns to it.*

Proof. With all gates equal to m and $\boldsymbol{v}_g = m\boldsymbol{u}_g$, (14) gives $\partial_{c_{g,d}}(\mathcal{L}_0 \circ K) = m^{D-1}\langle\nabla_g\mathcal{L}_0, \boldsymbol{u}_g\rangle = -m^{D-1}\|\nabla_g\mathcal{L}_0\|_2$ and $\partial_{\boldsymbol{v}_g}(\mathcal{L}_0 \circ K) = -m^{D-1}\|\nabla_g\mathcal{L}_0\|_2\boldsymbol{u}_g$, so every block receives the same radial update and the direction is unchanged; the sign of $\dot{m}$ is that of $m^{D-2} - (r_g^*)^{D-2}$. ■

Corollary 2 (Certified-commitment invariant)**.** *Let $C_1 = \{g : \boldsymbol{\omega}_g = \mathbf{0},\ \|\nabla_g\mathcal{L}_0(K(\boldsymbol{\omega}))\|_2 \leq \lambda_g\}$ at the stage-1 limit of Algorithm 1 (exact certification, $\varepsilon_{\mathrm{c}} = \delta = 0$). Then* (a) *by construction every $g \in C_1$ satisfies the groupwise KKT condition of the q=1 structured problem at that limit (Corollary 1);* (b) *under the optimizer of Lemma 2(b), $\boldsymbol{\omega}_g = \mathbf{0}$ for all $g \in C_1$ at every later iterate, so C_1 is a subset of the permanently removed groups and A_{latent} is non-increasing across the schedule when revival is off (the default), coinciding with A_{deploy} immediately after every re-lift by Proposition 2; and* (c) *every retained or revived group* whose collapsed weight is nonzero *is re-lifted to an interior balanced point, so such a group does not* begin *stage* 2 *at the absorbing origin. A survivor that is a collapsed zero ($\boldsymbol{\omega}_g \neq \mathbf{0}$ with $K_g(\boldsymbol{\omega}_g) = \mathbf{0}$) is carried to the latent origin by Proposition 2 and is thereafter removed like any other zero. Property (b) keeps the groups certified in (a) at zero through depth $D_{\max}$; the KKT condition in (a) itself is a property of the stage-1 solution.*

C.1 Initialization, optimizers, and scope

The construction of Appendix E.2 ties the gated initialization to the standard one, so that the results reflect the mechanism: draw the collapsed weight $\tilde{\boldsymbol{w}}_g$ from the standard initializer, apply the floor to the collapsed sample, $\bar{\boldsymbol{w}}_g = \tilde{\boldsymbol{w}}_g \max(1, \varepsilon/\|\tilde{\boldsymbol{w}}_g\|_2)$, and lift it exactly, $c_{g,d} = \|\bar{\boldsymbol{w}}_g\|_2^{1/D}$, $\boldsymbol{v}_g = \bar{\boldsymbol{w}}_g\|\bar{\boldsymbol{w}}_g\|_2^{(1-D)/D}$, so that forward statistics at initialization match standard training in distribution except on the floor event $\|\tilde{\boldsymbol{w}}_g\|_2 < \varepsilon$ (of positive but negligible probability), while the floor keeps the group off the absorbing origin. Training then uses SGD with a large initial learning rate and cosine decay (Loshchilov & Hutter, 2017); because gate gradients aggregate across a cell they empirically show lower stochastic variability than element-wise factors, broadening the stable learning-rate range.

Remark 3 (Scope of the absorbing property)**.** *Lemma 2(b) says the* latent origin $\boldsymbol{\omega}_g = \mathbf{0}$ *is forward-invariant; whether a group initialized away from zero reaches it is measured in Appendix F.3. The latent origin is distinct from a* collapsed *zero, $K_g(\boldsymbol{\omega}_g) = \mathbf{0}$ with $\boldsymbol{\omega}_g \neq \mathbf{0}$: at D=3, $c_1 = 0$ with $c_2, \boldsymbol{v} \neq \mathbf{0}$ collapses to $\boldsymbol{w}_g = \mathbf{0}$, yet ∂_{c_1} is proportional to $c_2\langle\nabla_g\mathcal{L}_0, \boldsymbol{v}\rangle$ and need not vanish—such a state is* not *absorbing and can regrow. Every monotonicity and irreversibility claim here concerns the latent origin (Section 3.1). Three notions therefore stay separate: the* theoretical *zero $\boldsymbol{\omega}_g = \mathbf{0}$, at which Theorem 1 is computed; the* numerical *zero, identified in finite precision by $|P_g|\,\|\boldsymbol{v}_g\|_2 <$ machine epsilon—a* collapsed *criterion, so Appendix F.3 checks per group whether the full latent state approaches the origin or only one factor collapses, and whether such groups regrow; and* deployment *thresholding, used when an external budget must be met exactly, whose removals carry no certificate and so count as uncertified in coverage. Precisely: the exact latent origin is irreversible, and certified only if it passed the D=2 test; numerically collapsed groups are treated as removed, with persistence measured in Appendix F.3; thresholded removals are neither.*

What "exact" means. *Exact:* the variational representation (Lemma 1); the minima correspondence for disjoint partition-type attachments (Theorem 2); the gauge rate $4\lambda/D$ under gradient flow (Lemma 2); forward-invariance of the latent origin (Lemma 2(b)); the balanced re-lift (Proposition 2); and the second-order classification (Theorem 1). *Conditional:* the spectral correspondence (smooth rank stratum; stated only in the extended version, Remark 4), certificates (under GD; Corollary 1), and architecture selection (under monotone shrinking). *Empirical:* that training reaches the latent origin (Remark 1, Appendix F.3), behavior under adaptive optimizers (Appendix F.4), and all reported accuracies.

Proposition 3 (A common mechanism across partition-type attachments)**.** *Let the carrier map be the identity and let $\mathcal{G}$ be* any *disjoint partition of the parameter vector—entries, rows, channels, attention heads, or residual-branch parameter blocks $\boldsymbol{\theta}_b$. Then: (i) the induced regularizer is $\sum_g \|\boldsymbol{w}_g\|_2^{2/D}$ (Lemma 1); (ii) the smooth gated and non-smooth penalized objectives have corresponding local and global minima (Theorem 2); (iii) reaching the latent origin of a cell is irreversible under memoryless SGD (Lemma 2(b)), so training performs monotone descent through A_{latent}; and (iv) at $D = 2$, under the hypotheses of Corollary 1, every cell at the latent origin at a convergent GD limit carries its certificate, for almost every initialization.*

Proof. None of the four proofs refers to what a cell *means*; each uses only that $\mathcal{G}$ is a disjoint partition of the coordinates, so that the penalty is separable over cells, the fibers of K are products of per-cell fibers, and the Jacobian $\boldsymbol{J}_g(\mathbf{0}) = \mathbf{0}$ makes the Hessian block-isolated. ■

Remark 4 (Scope of the proposition)**.** *A* spectral *attachment, in which the gate multiplies a singular value of $\Psi(\boldsymbol{U}, \boldsymbol{g}, \boldsymbol{V}) = \boldsymbol{U}\,\mathrm{diag}(\boldsymbol{g})\,\boldsymbol{V}^\top$ rather than a partition cell of an identity carrier, lies outside the scope of this proposition. There Ψ is not the identity, the cells share the factors $\boldsymbol{U}, \boldsymbol{V}$ rather than partitioning a parameter vector, and each of the four steps above would need a separate argument: the induced penalty equals $\sum_i \sigma_i^{2/D}$ only when $\boldsymbol{U}, \boldsymbol{V}$ are orthonormal, a property that training must then maintain; the fibers of Ψ no longer partition into independent cells; and the Hessian at a zeroed bond acquires cross terms with $\boldsymbol{U}$ and $\boldsymbol{V}$, so the block-isolation step of Theorem 1 needs a separate treatment. This is the precise content of the* smooth rank stratum *condition in "What 'exact' means" above, and the extended version of this work treats it with the additional hypotheses it needs.*

Proof sketch of Theorem 2. Lifting: a local minimizer $\hat{\boldsymbol{w}}$ of (1) lifts to any balanced factorization $\hat{\boldsymbol{\omega}}$; on a K-preimage ball, (3) gives $\mathcal{L}_\lambda(\boldsymbol{\omega}') \geq \mathcal{L}_{w,\lambda}(K(\boldsymbol{\omega}')) \geq \mathcal{L}_{w,\lambda}(\hat{\boldsymbol{w}}) = \mathcal{L}_\lambda(\hat{\boldsymbol{\omega}})$. *Projection:* a local minimizer $\hat{\boldsymbol{\omega}}$ of (2) is balanced (gauge-move argument); if $K(\hat{\boldsymbol{\omega}})$ were not locally minimal, a strictly better $\boldsymbol{w}'$ nearby would lift via Lemma 5 to a balanced $\boldsymbol{\omega}'$ in the minimality neighborhood with $\mathcal{L}_\lambda(\boldsymbol{\omega}') = \mathcal{L}_{w,\lambda}(\boldsymbol{w}') < \mathcal{L}_\lambda(\hat{\boldsymbol{\omega}})$, a contradiction. □

Continuity of the balanced selection (Lemma 5) is where group structure pays: in element-wise DWF the analogous step requires a per-coordinate sign-pattern construction over $p(D-1)$ factors (Kolb et al., 2025b), whereas the gated fiber has *one* radial degree of freedom per group, so the selection reduces to continuity of $t \mapsto t^{1/D}$ and the sign pattern is reused verbatim. The gate overhead is $|\mathcal{G}|(D-1)$ scalars—for channel-level structure, a vanishing fraction of p.

Proposition 4 (Discrete-time attraction up to $O(\eta^2)$)**.** *Write $a := c_{g,d}^2$ and $b := \|\boldsymbol{v}_g\|_2^2$ (or two gate squares) for the squared blocks of a group. One gradient step of size η gives*

$$a^+ - b^+ = (1 - 2\eta\lambda/D)^2(a - b) \;+\; \eta^2\big(\|\boldsymbol{G}_a\|_2^2 - \|\boldsymbol{G}_b\|_2^2\big),$$

where $\boldsymbol{G}_\bullet$ is the data-fit partial of the corresponding block. The first-order data-fit terms cancel identically*—this is the discrete shadow of the conserved quantity—so the imbalance contracts geometrically up to an $O(\eta^2)$ residual driven by the spread of squared block gradients. The cancellation is exact, and the factor $(1 - 2\eta\lambda/D)^2$ is strictly less than one for every $0 < \eta < D/\lambda$ (for $\eta < D/(2\lambda)$ the blocks are moreover scaled by a positive factor), so the contraction holds up to that residual on that whole step-size range.*

Corollary 3 (Gauge-fixed training)**.** *Initialize on the balanced manifold (Appendix C.1); then by Proposition 4 the imbalance $a - b$ of every pair of squared blocks in a group after t steps is bounded, for $\eta < D/\lambda$, by $\eta^2 \sum_{s<t}(1 - 2\eta\lambda/D)^{2(t-1-s)}\,|\delta G_s| \leq \eta\,\frac{D \max_s |\delta G_s|}{4\lambda - 4\eta\lambda^2/D}$, where δG_s is the squared-gradient spread at step s: the imbalance of squared blocks stays $O(\eta)$, so on any region where the*

block magnitudes are bounded away from zero the trajectory remains within $O(\eta)$ of the balanced manifold, and there, by (3)*, the gauge slack obeys $\lambda M = O(\eta^2)$ and the smooth objective agrees with the non-smooth target to that order. Two conditions delimit the statement: the $O(\eta)$ proximity holds away from the origin, since the map from squared to unsquared imbalance is not Lipschitz there (the origin itself is the region Theorem 1 treats directly); and δG_s is assumed bounded along the trajectory. Under these conditions, gated training is an $O(\eta)$-faithful gauge-fixed realization of* (1) *rather than a relaxation of it.*

Equivalence sanity check (Theorem 2). On synthetic group-sparse regression (p=1000, G=100, s=10), gated SGD is compared against a proximal-gradient solver on the same objective (1) at $\lambda = 0.02$, in a separate run with its own data draws, so its support recoveries are read on their own rather than against Table 9. **Result:** at D=2 the objective values agree to 2.7×10^{-6} and both recover the support exactly ($F_1 = 1.000$). Theorem 2 establishes correspondence between the minimizers of the two formulations; that the two solvers select the same support is an empirical observation. At D=3 and D=4 the objective gap is large (0.45 and 0.54) and fixed-depth gated SGD reaches support recovery 0.314 and 0.286 against 1.000 for the proximal solver, as the theory predicts: the correspondence is between minimizers, and at $D \geq 3$ gradient descent on the smooth objective is trapped at a stable non-global point that the proximal solver escapes. This is the same phenomenon Theorem 1 predicts and CPG is designed to avoid. Support recovery as a function of sample size is reported empirically for all depths (Figure 7h, Table 10).

D Full Proofs

Throughout, notation follows Section 3.1; $B(\boldsymbol{x}, \delta)$ is the open Euclidean ball. Define the gauge slack

$$M(\boldsymbol{\omega}) = \frac{1}{D}\sum_g \Big(\sum_d c_{g,d}^2 + \|\boldsymbol{v}_g\|_2^2\Big) - \sum_g \|K_g(\boldsymbol{\omega}_g)\|_2^{2/D} \;\geq\; 0, \tag{12}$$

so that $\mathcal{L}_\lambda(\boldsymbol{\omega}) = \mathcal{L}_{w,\lambda}(K(\boldsymbol{\omega})) + \lambda M(\boldsymbol{\omega})$ once Lemma 1 is established.

D.1 Proof of Lemma 1

Derivation 1 (AM–GM). Fix a feasible point; abbreviate $c_d := c_{g,d}$, $\boldsymbol{v} := \boldsymbol{v}_g$. The D non-negative blocks $c_1^2, \ldots, c_{D-1}^2, \|\boldsymbol{v}\|_2^2$ satisfy

$$\frac{1}{D}\Big(\sum_{d=1}^{D-1} c_d^2 + \|\boldsymbol{v}\|_2^2\Big) \;\geq\; \Big(\prod_d c_d^2 \cdot \|\boldsymbol{v}\|_2^2\Big)^{1/D} = \Big(\big|\prod_d c_d\big|\,\|\boldsymbol{v}\|_2\Big)^{2/D} = \|\boldsymbol{w}_g\|_2^{2/D}, \tag{13}$$

the last step using feasibility $\|\boldsymbol{w}_g\|_2 = |\prod_d c_d| \cdot \|\boldsymbol{v}\|_2$. AM–GM is tight iff all blocks coincide, i.e. at balanced points. For $\boldsymbol{w}_g \neq \mathbf{0}$ the explicit point $c_d = \|\boldsymbol{w}_g\|_2^{1/D}$, $\boldsymbol{v} = \boldsymbol{w}_g\|\boldsymbol{w}_g\|_2^{(1-D)/D}$ is feasible, balanced, and attains the bound; for $\boldsymbol{w}_g = \mathbf{0}$, $\boldsymbol{\omega}_g = \mathbf{0}$ attains 0.

Derivation 2 (radial reduction and Lagrange). For $\boldsymbol{w}_g \neq \mathbf{0}$ (the case $\boldsymbol{w}_g = \mathbf{0}$ is trivial, the objective being non-negative and 0 at $\boldsymbol{\omega}_g = \mathbf{0}$), the objective and constraint depend on $\boldsymbol{v}$ only through $(\|\boldsymbol{v}\|_2, \boldsymbol{v}/\|\boldsymbol{v}\|_2)$, and feasibility pins the direction to $\pm\boldsymbol{w}_g/\|\boldsymbol{w}_g\|_2$, so the problem reduces to minimizing $\frac{1}{D}\sum_{d=1}^D m_d^2$ over $m_d > 0$ with $\prod_d m_d = \|\boldsymbol{w}_g\|_2$ ($m_d = |c_d|$ for $d < D$, $m_D = \|\boldsymbol{v}\|_2$). Stationarity of $\frac{1}{D}\sum m_d^2 - \mu \log\prod m_d$ gives $m_d^2 = \mu D/2$ constant in d; the constraint forces $m_d = \|\boldsymbol{w}_g\|_2^{1/D}$ and the value $\|\boldsymbol{w}_g\|_2^{2/D}$; coercivity on the constraint set makes the stationary point the global minimum. The identity $\mathcal{L}_{w,\lambda}(\boldsymbol{w}) = \min_{K(\boldsymbol{\omega})=\boldsymbol{w}} \mathcal{L}_\lambda(\boldsymbol{\omega})$ follows by adding the fiber-constant term $\mathcal{L}_0(K(\boldsymbol{\omega})) = \mathcal{L}_0(\boldsymbol{w})$ and applying the per-group result, since penalty and constraint decouple over g. ■

D.2 Balance at local minima

Lemma 4 (Balance)**.** *Every local minimizer $\hat{\boldsymbol{\omega}}$ of $\mathcal{L}_\lambda$ is balanced; hence $\mathcal{L}_\lambda(\hat{\boldsymbol{\omega}}) = \mathcal{L}_{w,\lambda}(K(\hat{\boldsymbol{\omega}}))$.*

Proof. If some group is unbalanced, two of its D blocks differ, say $a^2 \neq b^2$ with (a, b) a pair of gates or a gate and $\|\boldsymbol{v}_g\|_2$ (if one of the two blocks is zero while the other is not—a zero gate, or

$\boldsymbol{v}_g = \mathbf{0}$ with a gate nonzero—then $K_g \equiv \mathbf{0}$ along the whole curve and shrinking the nonzero block strictly decreases the penalty). Along the gauge curve $\gamma(\alpha) = (\alpha a, \alpha^{-1} b)$, K and hence $\mathcal{L}_0 \circ K$ are constant while the penalty contribution $h(\alpha) = \alpha^2 a^2 + \alpha^{-2} b^2$ has $h'(1) = 2(a^2 - b^2) \neq 0$; thus every neighborhood contains a strictly better point, contradicting local minimality. The objective identity is (12) with $M(\hat{\boldsymbol{\omega}}) = 0$. □

D.3 Continuity of the balanced selection

Lemma 5 (Balanced selection). *Let $\hat{\boldsymbol{\omega}}$ be balanced with $K(\hat{\boldsymbol{\omega}}) = \hat{\boldsymbol{w}}$. For every $\varepsilon > 0$ there is $\delta > 0$ such that every $\boldsymbol{w}' \in B(\hat{\boldsymbol{w}}, \delta)$ admits a balanced $\boldsymbol{\omega}' \in B(\hat{\boldsymbol{\omega}}, \varepsilon)$ with $K(\boldsymbol{\omega}') = \boldsymbol{w}'$.*

Proof. It suffices to construct, per group and per $\eta > 0$, a threshold δ_g such that $\|\boldsymbol{w}'_g - \hat{\boldsymbol{w}}_g\| < \delta_g$ yields a balanced $\boldsymbol{\omega}'_g$ with $K_g(\boldsymbol{\omega}'_g) = \boldsymbol{w}'_g$ and $\|\boldsymbol{\omega}'_g - \hat{\boldsymbol{\omega}}_g\| < \eta$; take $\eta = \varepsilon/\sqrt{|\mathcal{G}|}$, $\delta = \min_g \delta_g$.

Case $\hat{\boldsymbol{w}}_g \neq \mathbf{0}$. Balance gives $|\hat{c}_{g,d}| = \|\hat{\boldsymbol{v}}_g\| = \hat{m} := \|\hat{\boldsymbol{w}}_g\|^{1/D} > 0$; write $s_d = \operatorname{sign}(\hat{c}_{g,d})$, $\sigma = \prod_d s_d$, so $\hat{\boldsymbol{v}}_g = \hat{\boldsymbol{w}}_g/(\sigma \hat{m}^{D-1})$. For $\boldsymbol{w}'_g$ with $\|\boldsymbol{w}'_g - \hat{\boldsymbol{w}}_g\| < \delta_g \leq \frac{1}{2}\|\hat{\boldsymbol{w}}_g\|$, set $m' = \|\boldsymbol{w}'_g\|^{1/D}$, $c'_{g,d} = s_d m'$, $\boldsymbol{v}'_g = \boldsymbol{w}'_g/(\sigma (m')^{D-1})$. Then $K_g(\boldsymbol{\omega}'_g) = \boldsymbol{w}'_g$ and $\|\boldsymbol{v}'_g\| = m'$: balanced, with the original sign pattern reused verbatim. Distances: $|c'_{g,d} - \hat{c}_{g,d}| = |m' - \hat{m}| \to 0$ by continuity of $\boldsymbol{x} \mapsto \|\boldsymbol{x}\|^{1/D}$, and

$$\|\boldsymbol{v}'_g - \hat{\boldsymbol{v}}_g\| \leq \frac{\|\boldsymbol{w}'_g - \hat{\boldsymbol{w}}_g\|}{(m')^{D-1}} + \|\hat{\boldsymbol{w}}_g\| \Big| \frac{1}{(m')^{D-1}} - \frac{1}{\hat{m}^{D-1}} \Big| \;\longrightarrow\; 0,$$

since $m' \geq (\frac{1}{2}\|\hat{\boldsymbol{w}}_g\|)^{1/D} > 0$ on the ball. Choose δ_g so both discrepancies are below $\eta/\sqrt{D}$.

Case $\hat{\boldsymbol{w}}_g = \mathbf{0}$. Balance forces $\hat{\boldsymbol{\omega}}_g = \mathbf{0}$. For $\boldsymbol{w}'_g \neq \mathbf{0}$ set $m' = \|\boldsymbol{w}'_g\|^{1/D}$, $c'_{g,d} = m'$, $\boldsymbol{v}'_g = \boldsymbol{w}'_g/(m')^{D-1}$: balanced, feasible, and $\|\boldsymbol{\omega}'_g\|^2 = (D-1)(m')^2 + (m')^2 = D\|\boldsymbol{w}'_g\|^{2/D} \to 0$; choose δ_g with $D\delta_g^{2/D} < \eta^2$. □

D.4 Proof of Theorem 2

(i) Lifting. Let $\hat{\boldsymbol{w}}$ be a local minimizer of $\mathcal{L}_{w,\lambda}$ with radius ε_0, and let $\hat{\boldsymbol{\omega}}$ be any balanced factorization (exists by Lemma 1): $M(\hat{\boldsymbol{\omega}}) = 0$ and $\mathcal{L}_\lambda(\hat{\boldsymbol{\omega}}) = \mathcal{L}_{w,\lambda}(\hat{\boldsymbol{w}})$. By continuity of the polynomial map K, pick δ_0 with $K(B(\hat{\boldsymbol{\omega}}, \delta_0)) \subseteq B(\hat{\boldsymbol{w}}, \varepsilon_0)$. For $\boldsymbol{\omega}' \in B(\hat{\boldsymbol{\omega}}, \delta_0)$: $\mathcal{L}_\lambda(\boldsymbol{\omega}') = \mathcal{L}_{w,\lambda}(K(\boldsymbol{\omega}')) + \lambda M(\boldsymbol{\omega}') \geq \mathcal{L}_{w,\lambda}(K(\boldsymbol{\omega}')) \geq \mathcal{L}_{w,\lambda}(\hat{\boldsymbol{w}}) = \mathcal{L}_\lambda(\hat{\boldsymbol{\omega}})$.

(ii) Projection. Let $\hat{\boldsymbol{\omega}}$ be a local minimizer of $\mathcal{L}_\lambda$ with radius ε_0; by Lemma 4 it is balanced, so $\mathcal{L}_\lambda(\hat{\boldsymbol{\omega}}) = \mathcal{L}_{w,\lambda}(\hat{\boldsymbol{w}})$, $\hat{\boldsymbol{w}} = K(\hat{\boldsymbol{\omega}})$. If $\hat{\boldsymbol{w}}$ were not a local minimizer, apply Lemma 5 with $\varepsilon = \varepsilon_0$ to get δ_0; choose $\boldsymbol{w}' \in B(\hat{\boldsymbol{w}}, \delta_0)$ with $\mathcal{L}_{w,\lambda}(\boldsymbol{w}') < \mathcal{L}_{w,\lambda}(\hat{\boldsymbol{w}})$ and its balanced lift $\boldsymbol{\omega}' \in B(\hat{\boldsymbol{\omega}}, \varepsilon_0)$; then $\mathcal{L}_\lambda(\boldsymbol{\omega}') = \mathcal{L}_{w,\lambda}(\boldsymbol{w}') < \mathcal{L}_\lambda(\hat{\boldsymbol{\omega}})$, a contradiction.

Global minima. Taking $\min_{\boldsymbol{w}}$ in the pointwise identity of Lemma 1 and exchanging minimization over fibers (the fibers partition the domain of the surjective K) gives equal optimal values, with the inner minimum attained at balanced points. ■

D.5 Dynamics: proofs of Lemma 2(a),(b) and Proposition 4

With $\nabla_g \mathcal{L}_0 := \nabla_{\boldsymbol{w}_g} \mathcal{L}_0(K(\boldsymbol{\omega}))$ and $P_g := \prod_d c_{g,d}$, the chain rule gives

$$\frac{\partial(\mathcal{L}_0 \circ K)}{\partial c_{g,d}} = \Big(\prod_{d' \neq d} c_{g,d'} \Big) \langle \nabla_g \mathcal{L}_0, \boldsymbol{v}_g \rangle, \qquad \frac{\partial(\mathcal{L}_0 \circ K)}{\partial \boldsymbol{v}_g} = P_g \, \nabla_g \mathcal{L}_0. \tag{14}$$

Attraction. The gradient constraint that the rescaling symmetry imposes (Kunin et al., 2021), $c_{g,d}\, \partial_{c_{g,d}}(\mathcal{L}_0 \circ K) = P_g \langle \nabla_g \mathcal{L}_0, \boldsymbol{v}_g \rangle = \langle \boldsymbol{v}_g, \partial_{\boldsymbol{v}_g}(\mathcal{L}_0 \circ K) \rangle$ is identical across all D blocks. Along $\dot{\boldsymbol{\omega}} = -\nabla \mathcal{L}_\lambda$,

$$\tfrac{d}{dt} c_{g,d}^2 = -2P_g \langle \nabla_g \mathcal{L}_0, \boldsymbol{v}_g \rangle - \tfrac{4\lambda}{D} c_{g,d}^2, \qquad \tfrac{d}{dt} \|\boldsymbol{v}_g\|_2^2 = -2P_g \langle \nabla_g \mathcal{L}_0, \boldsymbol{v}_g \rangle - \tfrac{4\lambda}{D} \|\boldsymbol{v}_g\|_2^2,$$

so the data-fit terms cancel in every pairwise difference, leaving linear ODEs with solution $\Delta(t) = e^{-4\lambda t/D}\Delta(0)$; at $\lambda = 0$ the differences are conserved—they are the Noether-type quantities of that symmetry—and weight decay makes them decay.

Absorbing states. At $\boldsymbol{\omega}_g = \mathbf{0}$: every gate partial in (14) vanishes ($\boldsymbol{v}_g = \mathbf{0}$ kills the inner product; for $D \geq 3$ the gate product also vanishes), the direction partial vanishes ($P_g = 0$), and the decay terms vanish; the exact (minibatch) gradient restricted to group g is zero, so the update fixes it; induct over iterations.

Discrete identity. For a gate, $c^+ = c - \eta(G + \frac{2\lambda}{D}c)$ with $cG = N := P_g\langle\nabla_g\mathcal{L}_0, \boldsymbol{v}_g\rangle$, hence $(c^+)^2 = (1 - \frac{2\eta\lambda}{D})^2c^2 - 2\eta(1 - \frac{2\eta\lambda}{D})N + \eta^2G^2$; identically, $\|\boldsymbol{v}^+\|^2 = (1 - \frac{2\eta\lambda}{D})^2\|\boldsymbol{v}\|^2 - 2\eta(1 - \frac{2\eta\lambda}{D})N + \eta^2\|\partial_{\boldsymbol{v}}\mathcal{L}_0\|^2$. Subtracting any two such identities cancels the N terms exactly. ■

D.6 Proof of Theorem 1 and Corollary 1

Write $F(\boldsymbol{\omega}_g) := \mathcal{L}_0(K(\bar{\boldsymbol{\omega}}_{-g}, \boldsymbol{\omega}_g)) + \frac{\lambda}{D}(\sum_d c_{g,d}^2 + \|\boldsymbol{v}_g\|_2^2)$ with the other coordinates held at $\bar{\boldsymbol{\omega}}_{-g}$. The collapsed weight $\boldsymbol{w}_g(\boldsymbol{\omega}_g)$ is a homogeneous polynomial of degree D, so its Jacobian $\boldsymbol{J}(\boldsymbol{\omega}_g)$ vanishes to order $D - 1 \geq 1$, giving $\boldsymbol{J}(\mathbf{0}) = \mathbf{0}$.

Block isolation. For any other block $\boldsymbol{\omega}_{g'}$,

$$\frac{\partial^2(\mathcal{L}_0 \circ K)}{\partial\boldsymbol{\omega}_{g'}\partial\boldsymbol{\omega}_g} = \frac{\partial}{\partial\boldsymbol{\omega}_{g'}}\big[\boldsymbol{J}(\boldsymbol{\omega}_g)^\top\nabla_g\mathcal{L}_0(K(\boldsymbol{\omega}))\big] = \boldsymbol{J}(\boldsymbol{\omega}_g)^\top\frac{\partial\nabla_g\mathcal{L}_0(K(\boldsymbol{\omega}))}{\partial\boldsymbol{\omega}_{g'}} \overset{\boldsymbol{\omega}_g=\mathbf{0}}{=} \mathbf{0},$$

since the other product-rule term would differentiate $\boldsymbol{J}$ in $\boldsymbol{\omega}_{g'}$, on which it does not depend, and $\boldsymbol{J}(\mathbf{0}) = \mathbf{0}$; the penalty is separable.

The g-block. Taylor-expand $\mathcal{L}_0$ in the collapsed weight: $\mathcal{L}_0 = \text{const} + \langle\nabla_g\mathcal{L}_0, \boldsymbol{w}_g(\boldsymbol{\omega}_g)\rangle + O(\|\boldsymbol{w}_g(\boldsymbol{\omega}_g)\|^2)$, and the remainder is $O(\|\boldsymbol{\omega}_g\|^{2D})$, contributing nothing to the Hessian at $\mathbf{0}$ ($2D \geq 4$). The degree-D form $\langle\nabla_g\mathcal{L}_0, \boldsymbol{w}_g(\boldsymbol{\omega}_g)\rangle$ has vanishing second derivatives at $\mathbf{0}$ for $D \geq 3$, leaving $\frac{2\lambda}{D}\boldsymbol{I} \succ 0$. For $D = 2$ it is the bilinear $c_g\langle\nabla_g\mathcal{L}_0, \boldsymbol{v}_g\rangle$, whose Hessian is the symmetric off-diagonal pair $(\nabla_g\mathcal{L}_0, \nabla_g\mathcal{L}_0^\top)$; adding $\lambda\boldsymbol{I}$ yields (4), with spectrum $\lambda \pm \|\nabla_g\mathcal{L}_0\|_2$ and λ (multiplicity $|g| - 1$) from the rank-two off-diagonal block. Since $\partial\|\cdot\|_2(\mathbf{0})$ is the unit ball, $\mathbf{0} \in \nabla_g\mathcal{L}_0 + \lambda\,\partial\|\cdot\|_2(\mathbf{0}) \Leftrightarrow \|\nabla_g\mathcal{L}_0\|_2 \leq \lambda$: the strict-saddle threshold is the non-smooth stationarity boundary. The local-minimum claim of the non-smooth $\ell_{2/D}$ ($D \geq 3$) objective at zero follows since $\lambda\|\boldsymbol{w}_g\|^{2/D}$ dominates $\langle\nabla_g\mathcal{L}_0, \boldsymbol{w}_g\rangle$ as $\boldsymbol{w}_g \to \mathbf{0}$.

Corollary. $\mathcal{L}_\lambda \geq \inf\mathcal{L}_0 + \frac{\lambda}{D}\|\boldsymbol{\omega}\|_2^2$ is coercive, so S_c is compact and a global minimizer $\boldsymbol{\omega}^\star \in S_c$ exists. For $\boldsymbol{\omega} \in S_c$, $\|\nabla\mathcal{L}_\lambda(\boldsymbol{\omega})\|_2 = \|\nabla\mathcal{L}_\lambda(\boldsymbol{\omega}) - \nabla\mathcal{L}_\lambda(\boldsymbol{\omega}^\star)\|_2 \leq 2RL_\nabla$, so the step $\boldsymbol{\omega}^+ := \boldsymbol{\omega} - \eta\nabla\mathcal{L}_\lambda(\boldsymbol{\omega})$ has length at most $2R$ and the segment $[\boldsymbol{\omega}, \boldsymbol{\omega}^+]$ lies in $B(\mathbf{0}, 3R)$, where the descent lemma applies: $\mathcal{L}_\lambda(\boldsymbol{\omega}^+) \leq \mathcal{L}_\lambda(\boldsymbol{\omega}) - \frac{\eta}{2}\|\nabla\mathcal{L}_\lambda(\boldsymbol{\omega})\|_2^2$. Hence U_c is open and forward invariant, and on it $\|\nabla^2\mathcal{L}_\lambda\| \leq L_\nabla$, so $I - \eta\nabla^2\mathcal{L}_\lambda$ is invertible. The stable-manifold argument of Lee et al. (2019a, Theorem 2), applied to the gradient map on U_c (cf. Lee et al., 2016), gives that the initializations in U_c whose iterates converge to a strict saddle form a Lebesgue-null set. Any limit is a critical point, since $\boldsymbol{\omega}^k - \boldsymbol{\omega}^{k+1} = \eta\nabla\mathcal{L}_\lambda(\boldsymbol{\omega}^k) \to \mathbf{0}$, and at a limit that is not a strict saddle every latent-zero group satisfies $\|\nabla_g\mathcal{L}_0\|_2 \leq \lambda$, since by (4) the g-block is a decoupled principal submatrix whose smallest eigenvalue $\lambda - \|\nabla_g\mathcal{L}_0\|_2$ would otherwise be negative. ■

D.7 Proof of Proposition 1

Reduction to the collapsed radius. By Lemma 1, $\mathcal{L}_\lambda(\boldsymbol{\omega}) = \mathcal{L}_{w,\lambda}(K(\boldsymbol{\omega})) + \lambda M(\boldsymbol{\omega})$ with $M \geq 0$ and $M = 0$ exactly at balanced points, so for any path $\boldsymbol{\omega}(\cdot)$ leaving the latent origin of cell g, $\mathcal{L}_\lambda(\boldsymbol{\omega}) - \mathcal{L}_\lambda(\bar{\boldsymbol{\omega}}) \geq \Phi(\|\boldsymbol{w}_g\|_2)$ where $\Phi(\rho) := \lambda\rho^{2/D} + \min_{\|\boldsymbol{u}\|_2=1}\big[\rho\langle\nabla_g\mathcal{L}_0, \boldsymbol{u}\rangle\big] - \frac{L}{2}\rho^2 = \lambda\rho^{2/D} - \|\nabla_g\mathcal{L}_0\|_2\rho - \frac{L}{2}\rho^2$, with L a bound on $\|\nabla^2_{\boldsymbol{w}_g}\mathcal{L}_0\|$ on the ball in question (Taylor's theorem). The first-order part of the bound is attained along the balanced ray in the descent direction (the remainder aside), so the analysis below, carried out on that ray, is the minimum-energy escape profile and bounds every path. Strictness holds at collapsed zeros too: any $\boldsymbol{\omega}_g \neq \mathbf{0}$ with $\|\boldsymbol{w}_g\|_2 = 0$ is unbalanced, so $M(\boldsymbol{\omega}) > 0$ and $\mathcal{L}_\lambda$ is strictly raised by (3).

(i) First-order local model. With $\mathcal{L}_0$ affine in $\boldsymbol{w}_g$ with slope $\nabla_g\mathcal{L}_0$ the remainder vanishes and $\Phi'(\rho) = \frac{2\lambda}{D}\rho^{2/D-1} - \|\nabla_g\mathcal{L}_0\|_2$. For $D \geq 3$ the exponent $2/D-1$ is negative, so $\Phi' \to +\infty$ as $\rho \downarrow 0$, is strictly decreasing, and vanishes once, at $\rho^* = \big(2\lambda/(D\|\nabla_g\mathcal{L}_0\|_2)\big)^{D/(D-2)}$. Hence Φ increases on $(0,\rho^*)$ and decreases after. Writing $A := 2\lambda/(D\|\nabla_g\mathcal{L}_0\|_2)$, so $\rho^* = A^{D/(D-2)}$ and $(\rho^*)^{2/D} = A^{2/(D-2)}$,

$$\Phi(\rho^*) = \lambda A^{2/(D-2)} - \|\nabla_g\mathcal{L}_0\|_2 A^{D/(D-2)} = A^{2/(D-2)}\big(\lambda - \|\nabla_g\mathcal{L}_0\|_2 A\big) = \tfrac{D-2}{D}\,\lambda\, A^{2/(D-2)},$$

using $\|\nabla_g\mathcal{L}_0\|_2 A = 2\lambda/D$. The balanced factor magnitude with $\|\boldsymbol{w}_g\|_2 = \rho^*$ is $r^* = (\rho^*)^{1/D} = A^{1/(D-2)}$.

Error term (general smooth loss). Let $\Phi(\rho) = \lambda\rho^{2/D} - \|\nabla_g\mathcal{L}_0\|_2\rho - \psi(\rho)$ with $0 \leq \psi(\rho) \leq \frac{L}{2}\rho^2$, $|\psi'(\rho)| \leq L\rho$, L a bound on $\|\nabla^2_{\boldsymbol{w}_g}\mathcal{L}_0\|$ on the relevant ball. At the unperturbed root, $\Phi'(\rho^*) = O(L\rho^*)$, while $\Phi''(\rho) = \frac{2\lambda}{D}\big(\frac{2}{D}-1\big)\rho^{2/D-2}$ has magnitude $\frac{2(D-2)}{D^2}\lambda(\rho^*)^{2/D-2}$ at ρ^*. The implicit function theorem gives a root at $\rho^*\big(1 + O(L(\rho^*)^{2-2/D}/\lambda)\big)$. The relative error is therefore small when $L(\rho^*)^{2-2/D} \ll \lambda$, in particular when ρ^* is small, i.e. when $A = 2\lambda/(D\|\nabla_g\mathcal{L}_0\|_2) \ll 1$; the barrier height $\Phi(\rho^*)$ shifts by $\psi(\rho^*) = O(L(\rho^*)^2)$, the same relative order. The dependence on D is regime-specific. Write $A = A(D) = 2\lambda/(D\|\nabla_g\mathcal{L}_0\|_2)$, decreasing in D. Treating D as continuous, $\frac{d}{dD}\log r^* > 0$ iff $A < e^{-(D-2)/D}$ and $\frac{d}{dD}\log\Phi(\rho^*) = -2\log A/(D-2)^2 > 0$ iff $A < 1$, where $\Phi(\rho^*) = \frac{D-2}{D}\lambda A^{2/(D-2)}$. Since A decreases in D, if $A(D) < 1$ then the barrier height increases at every integer step from D onward, toward its limit λ, and if $A(D) < 1/e$ then r^* increases at every step from D onward; the collapsed radius $\rho^* = A^{D/(D-2)}$ is non-monotone in general, since the exponent decreases toward 1. (The converses fail for integer steps: from $D = 3$ to 4 the barrier increases iff $A(3) < 9/8$.) A cell with $A > 1$ has $\|\nabla_g\mathcal{L}_0\|_2 < 2\lambda/D < \lambda$ and already satisfies the KKT condition, so the growth of the barrier with D describes the operating regime $A < 1$. There, depth does not simply widen or narrow the trap; it *deepens* it.

(ii) Floor criterion. A canonical-gauge initialization at floor ε places the cell at $\|\boldsymbol{w}_g\|_2 = \varepsilon^D$; it is inside the barrier iff $\varepsilon^D < \rho^*$, i.e. iff $\varepsilon < r^*$, i.e. iff $\|\nabla_g\mathcal{L}_0\|_2 < 2\lambda/(D\varepsilon^{D-2})$—exactly under (i), to leading order for a general smooth loss (error term above). If instead the floor is applied to the *collapsed* sample, so that $\|\boldsymbol{w}_g\|_2 = \varepsilon$ (as in the balanced initialization of Appendix E.2), the corresponding test is $\|\nabla_g\mathcal{L}_0\|_2 < 2\lambda/(D\,\varepsilon^{(D-2)/D})$. The two differ, and Section 4.3 evaluates the criterion at each cell's measured factor radius rather than at either floor.

Degeneration at D=2. The same reduction gives $\Phi(\rho) = (\lambda - \|\nabla_g\mathcal{L}_0\|_2)\rho + O(\rho^2)$, so the sign of $\lambda - \|\nabla_g\mathcal{L}_0\|_2$ decides stability independently of ε, recovering Theorem 1. □

D.8 Proof of Proposition 2

Fix a cell g. If $\boldsymbol{w}_g \neq \mathbf{0}$, then by (5)

$$K_g(\boldsymbol{\omega}'_g) = \Big(\prod_{d=1}^{D'-1} c'_{g,d}\Big)\boldsymbol{v}'_g = \Big(\prod_d s_{g,d}\Big)^2 \|\boldsymbol{w}_g\|_2^{(D'-1)/D'}\,\boldsymbol{w}_g\,\|\boldsymbol{w}_g\|_2^{(1-D')/D'} = \boldsymbol{w}_g,$$

and if $\boldsymbol{w}_g = \mathbf{0}$ then $\boldsymbol{\omega}'_g = \mathbf{0}$ and $K_g(\boldsymbol{\omega}'_g) = \mathbf{0} = \boldsymbol{w}_g$. Hence $K(\boldsymbol{\omega}') = \boldsymbol{w}$, and the network function, $\mathcal{L}_0$ and A_{deploy}, which depend on the parameters only through K, are unchanged. For $\boldsymbol{w}_g \neq \mathbf{0}$ every gate has magnitude $|c'_{g,d}| = \|\boldsymbol{w}_g\|_2^{1/D'}$ and the carrier $\|\boldsymbol{v}'_g\|_2 = \|\boldsymbol{w}_g\|_2\,\|\boldsymbol{w}_g\|_2^{(1-D')/D'} = \|\boldsymbol{w}_g\|_2^{1/D'}$, so $\boldsymbol{\omega}'$ is balanced and the cell's depth-D' penalty is $\frac{\lambda}{D'}\big(\sum_d c'^2_{g,d} + \|\boldsymbol{v}'_g\|_2^2\big) = \lambda\|\boldsymbol{w}_g\|_2^{2/D'}$, the target penalty at exponent $2/D'$; summing over cells (with λ_g in place of λ under (6)) gives the statement about the objective. On the balanced manifold all D' factor magnitudes of a cell equal a common m with $m^{D'} = \|K_g(\boldsymbol{\omega}_g)\|_2 = \|\boldsymbol{w}_g\|_2$, so $m = \|\boldsymbol{w}_g\|_2^{1/D'}$ is forced, the carrier direction is fixed by $\boldsymbol{w}_g$ and the gate signs, and only the signs are free. Finally $\boldsymbol{\omega}'_g \neq \mathbf{0}$ if and only if $\boldsymbol{w}_g \neq \mathbf{0}$, so $A_{\mathrm{latent}}(\boldsymbol{\omega}') = A_{\mathrm{deploy}}(\boldsymbol{\omega})$; since $K_g(\boldsymbol{\omega}_g) \neq \mathbf{0}$ implies $\boldsymbol{\omega}_g \neq \mathbf{0}$, $A_{\mathrm{deploy}}(\boldsymbol{\omega}) \subseteq A_{\mathrm{latent}}(\boldsymbol{\omega})$, with equality exactly when no cell has $\boldsymbol{\omega}_g \neq \mathbf{0}$ and $K_g(\boldsymbol{\omega}_g) = \mathbf{0}$, that is, when $\boldsymbol{\omega}$ has no collapsed zeros. ■

D.9 CONVERGENCE (KURDYKA–ŁOJASIEWICZ)

Theorem 3 (Single-point convergence)**.** *Suppose the per-example loss and the network map are semialgebraic (more generally, definable in a common o-minimal structure), so that $\mathcal{L}_0$ is definable; suppose $\mathcal{L}_\lambda$ is C^1 with locally Lipschitz gradient, and $\mathcal{L}_0$ is bounded below; then $\mathcal{L}_\lambda$ is bounded below with bounded sublevel sets, since $\mathcal{L}_\lambda \geq \inf \mathcal{L}_0 + \frac{\lambda}{D}\|\boldsymbol{\omega}\|_2^2$ and its sublevel sets are therefore bounded. Then $\mathcal{L}_\lambda$ satisfies the Kurdyka–Łojasiewicz inequality at every point, and gradient descent started in a bounded sublevel set S with constant step $\eta < 1/L_S$, where L_S is a Lipschitz constant of $\nabla\mathcal{L}_\lambda$ on $S_R := S + \bar{B}(\mathbf{0}, R)$ for some $R > 0$ and $\eta \leq R/\sup_S \|\nabla\mathcal{L}_\lambda\|_2$ (so that each step stays inside S_R and the descent lemma returns the iterate to S), has finite trajectory length and converges to a* single *critical point $\hat{\boldsymbol{\omega}}$ (Kurdyka, 1998; Bolte et al., 2007b; Attouch et al., 2013). Definability covers polynomial and piecewise-polynomial networks and ones built from exp/log, sigmoid/tanh and softmax, with standard losses. The theorem is stated for C^1 objectives; ReLU networks are definable but not C^1, and for them the relevant non-smooth results of Attouch et al. (2013) cover proximal-type schemes satisfying a relative-error condition, to be verified separately for plain (sub)gradient steps. Two consequences follow. If $\hat{\boldsymbol{\omega}}$ is a local minimizer, it is balanced and $K(\hat{\boldsymbol{\omega}})$ locally minimizes the non-smooth target (Theorem 2). Independently, at $D = 2$ under the hypotheses of Corollary 1, for almost every initialization in the sublevel set fixed there the limit $\hat{\boldsymbol{\omega}}$ is not a strict saddle, so every group at the latent origin there carries the certificate—whether or not $\hat{\boldsymbol{\omega}}$ is a local minimizer.*

Proof sketch. If the per-example loss and the network map are definable, so is $\mathcal{L}_\lambda$ (K is polynomial and the penalty semialgebraic), hence the KL inequality holds at every point (Kurdyka, 1998; Bolte et al., 2007a;b); finite trajectory length and single-point convergence then follow from Attouch et al. (2013), whose global Lipschitz assumption can be replaced by L_S: with $\eta \leq R/\sup_S \|\nabla\mathcal{L}_\lambda\|_2$ every segment $[\boldsymbol{\omega}^k, \boldsymbol{\omega}^{k+1}]$ lies in S_R, so the descent lemma applies and keeps the iterates in S. The remaining statements combine Lemma 4, Theorem 2 and Corollary 1. Transferring the KL *exponent* between the gated and non-smooth objectives (cf. Ouyang et al., 2025, for the Hadamard case) and extending saddle avoidance to SGD are natural next steps.

E CERTIFIED PROGRESSIVE GATING: IMPLEMENTATION AND THE PER-CELL PENALTY

This appendix specifies the components of Algorithm 1 that Section 3.4 states without detail, and separates the contribution of the per-cell penalty (6) from that of the schedule.

E.1 THE ALGORITHM

Algorithm 1 states Certified Progressive Gating as summarized in Section 3.4; BOUNDARY mode is the default and the mode of every reported CPG result except the revival ablations.

Algorithm 1 Certified Progressive Gating

Require: partition $\mathcal{G}$, penalties λ_g, target depth $D_{\max} \in \{3, 4\}$, mode $\in \{\text{STRICT}, \text{BOUNDARY}\}$

1: **Decide.** Train at D=2 until stationary; a cell within the commitment tolerance of the latent origin is a removal candidate.
2: **Certify.** Label each candidate *certified* if $\|\nabla_g \mathcal{L}_0\|_2 \leq \lambda_g$ and *uncertified* otherwise.
3: **Commit** permanently: the certified candidates only, returning the rest to the network (STRICT); or every candidate (BOUNDARY). Labels go into a ledger.
4: **Re-lift.** $\boldsymbol{\omega}_g \leftarrow \mathcal{R}_{2\to D_{\max}}(\boldsymbol{\omega}_g)$ on survivors (Proposition 2); removed cells stay at $\mathbf{0}$.
5: **Compress.** Train at $D_{\max}$, then commit the remaining near-zeros as uncertified removals.

Ensure: collapsed model at exponent $2/D_{\max}$, and the ledger of its removals

E.2 BALANCED INITIALIZATION

CPG starts stage 1 on the balanced manifold, so that $M(\boldsymbol{\omega}) = 0$ at step zero and Lemma 1 is tight from the outset. The collapsed weight is drawn from the standard initializer and lifted exactly, so forward statistics match ordinary training.

For each group g, draw $\tilde{\boldsymbol{w}}_g$ from the base initializer $\mathcal{P}$, apply the floor $\varepsilon > 0$ to the collapsed sample, $\bar{\boldsymbol{w}}_g = \tilde{\boldsymbol{w}}_g \max\left(1, \varepsilon/\|\tilde{\boldsymbol{w}}_g\|_2\right)$, and set

$$c_{g,d} = \|\bar{\boldsymbol{w}}_g\|_2^{1/D} \quad (d = 1, \ldots, D{-}1), \qquad \boldsymbol{v}_g = \bar{\boldsymbol{w}}_g \, \|\bar{\boldsymbol{w}}_g\|_2^{(1-D)/D}.$$

Every factor then has magnitude $\|\bar{\boldsymbol{w}}_g\|_2^{1/D}$, so $M(\boldsymbol{\omega}) = 0$ exactly, and the product collapses to $K(\boldsymbol{\omega}) = \bar{\boldsymbol{w}}$, which equals $\tilde{\boldsymbol{w}}$ except on groups with $\|\tilde{\boldsymbol{w}}_g\|_2 < \varepsilon$, rescaled radially to norm ε.

E.3 PRACTICAL REFINEMENTS TO ALGORITHM 1

Three details make Algorithm 1 implementable without changing what it does. *(a) Finite tolerances.* A cell is taken to be at the latent origin when $\|\boldsymbol{\omega}_g\|_2 \le \varepsilon_c \bar{r}$, with $\bar{r}$ the median factor magnitude of the cells with $\boldsymbol{\omega}_g \neq \mathbf{0}$ at the depth of the comparison (Definition 1), which makes the tolerance scale-free; the certificate is applied with slack, $\|\nabla_g \mathcal{L}_0\|_2 \le \lambda_g + \delta$, giving certificate-compatible rather than exactly certified removals. *(b) The settle.* At a minibatch boundary the sampling noise in $\nabla_g \mathcal{L}_0$ is of the order of λ_g, enough to fail cells the full gradient would certify; under full-batch stage-1 training the uncertified fraction is at most 0.03. We therefore spend the last two stage-1 epochs at a large batch and a small learning rate, inside the same budget, which, at matched λ, lowers the uncertified fraction while matching the accuracy of ordinary epochs (Appendix F.2). *(c) Optional revival.* An uncertified near-zero may instead be returned to the network along $-\nabla_g \mathcal{L}_0$ at factor radius $\min\{\max\{\varepsilon, \kappa r_g^*\}, \bar{r}\}$, outside its own barrier whenever $\bar{r} > r_g^*$, so that it initially grows under the frozen first-order model (Lemma 3); the cap $\bar{r}$ keeps it in the regime where Proposition 1 is accurate. Revival suits settings in which the boundary is a mid-training diagnostic and pruning comes later; when the boundary decision *is* the pruning decision, committing every candidate performs better on thin networks (Table 6), so Algorithm 1 does not revive by default.

E.4 THE PER-CELL PENALTY AND THE SCHEDULE UNDER THE LANDING PROTOCOL

Equation (6) replaces the single λ of the target problem by a per-cell λ_g proportional to the cell's MAC cost. Substituting it into published pruners would measure methods that do not exist in the literature, so the main text runs every baseline as published and treats (6) as a component of CPG. Table 4 separates the contributions of the penalty and the schedule under the landing protocol; the boundary-protocol counterpart is the per-cell D-Gating row of Table 6.

Penalty and schedule. On the wide ResNet-18 the penalty is the larger lever and also lifts fixed-depth D-Gating ($\Delta_{\text{budget}} = +0.015$ and $+0.037$). On the thin ResNet-56 the schedule is the larger lever: its lead over fixed $D{=}2$ with the same penalty is 0.025 at $8\times$ and 0.038 at $16\times$ ($t = 5.6$ and 1.8). With a uniform λ the results echo the synthetic finding (Figure 5): on ResNet-18 at $16\times$ every schedule that decides at $D{=}2$ leads every fixed depth. The schedule protects against an unjustified removal, which pays off when the network is too thin to absorb a wrong one.

Effect on the parameter ratio. Targeting MACs weights early, high-resolution channels more heavily than late ones, which hold most of the parameters, so at a given MAC ratio the per-cell penalty realizes a parameter ratio up to about $2.4\times$ lower than a uniform λ (Table 6). Setting m_g to the parameter count of cell g, or to a measured per-channel latency, gives the same derivation against that budget (the parameter-count version is evaluated in Appendix F.1).

F ADDITIONAL EXPERIMENTAL RESULTS

F.1 COMPARISON WITH RELATED METHODS (SECTION 4.1)

Tuning DepGraph. DepGraph runs inside our pipeline with the same split, budget, target and fine-tuning, and its group-sparsity strength is selected on validation. `torch-pruning`'s default strengths are tuned for much longer sparse-training schedules and are far too strong under our budget

Table 4: **Penalty and schedule under the landing protocol**, CIFAR-10 top-1, mean ± population std over 5 seeds (3 where marked *, 2 where marked **), same budget and split for every row, λ selected per row and target on validation. Fixed-depth rows of the lower block use (6) and are controls for this ablation only; 3→4 omits the D=2 stage; *momentum throughout* replaces the memoryless default. Δ_{budget}: fixed D=2 with minus without (6); Δ_{CPG}: CPG minus fixed D=2, both with (6). Bold: column maximum; "—": not run.

	ResNet-18		ResNet-56		
Row	8×	16×	4×	8×	16×
Uniform λ					
D-Gating fixed D=2	0.8694 ± 0.0036	0.8154 ± 0.0068	0.8436 ± 0.0077	0.8191 ± 0.0166	0.7805 ± 0.0060
momentum throughout	0.8605 ± 0.0052	$0.8342 \pm 0.0058^*$	—	—	—
D-Gating fixed D=3	0.8569 ± 0.0127	0.7915 ± 0.0305	0.8407 ± 0.0130	0.8236 ± 0.0054	0.7683 ± 0.0164
D-Gating fixed D=4	0.8534 ± 0.0112	0.8034 ± 0.0095	0.8420 ± 0.0097	0.8087 ± 0.0171	0.7418 ± 0.0344
CPG, uniform λ	0.8591 ± 0.0019	0.8353 ± 0.0021	0.8509 ± 0.0031	0.8232 ± 0.0058	0.7773 ± 0.0100
three stages	0.8665 ± 0.0059	0.8518 ± 0.0036	$0.8296 \pm 0.0072^*$	$0.8294 \pm 0.0180^*$	$0.7651 \pm 0.0096^{**}$
momentum throughout	0.8551 ± 0.0065	0.8389 ± 0.0033	—	—	—
3→4 (control: no D=2 stage)	0.8444 ± 0.0188	0.8279 ± 0.0033	$0.8392 \pm 0.0027^*$	$0.8184 \pm 0.0155^*$	$\mathbf{0.7855 \pm 0.0113}^*$
With $\lambda_g \propto m_g$ (6)					
D-Gating fixed D=2	$\mathbf{0.8842 \pm 0.0023}$	0.8528 ± 0.0030	$0.8465 \pm 0.0053^*$	$0.8199 \pm 0.0061^*$	$0.7338 \pm 0.0298^*$
D-Gating fixed D=3	0.8730 ± 0.0042	$\mathbf{0.8530 \pm 0.0021}$	$0.8503 \pm 0.0053^*$	$0.8327 \pm 0.0087^*$	$0.7656 \pm 0.0144^*$
CPG	0.8778 ± 0.0050	0.8489 ± 0.0036	$\mathbf{0.8514 \pm 0.0132}$	$\mathbf{0.8446 \pm 0.0020}$	0.7721 ± 0.0122
Δ_{budget} *(penalty alone)*	+0.015	+0.037	+0.003	+0.001	−0.047
Δ_{CPG} *(schedule alone)*	−0.006	−0.004	+0.005	+0.025	+0.038

(table below), so we add two smaller values; the sweep is worth $+0.024$ to $+0.053$ over no regularization and makes DepGraph the strongest baseline, against which every comparison is stated. OTOv2 replaces the training loop with its own optimizer, so a number from it would not be comparable to Table 1; it is discussed as related work.

Model	Target	reg 0	$5{\cdot}10^{-5}$	10^{-4}	$5{\cdot}10^{-4}$	$5{\cdot}10^{-3}$	selected (bold)
ResNet-18	8×	0.8631	**0.8873**	0.8775	0.7819	0.3534	$5{\cdot}10^{-5}$
ResNet-18	16×	0.8370	**0.8687**	0.8706	0.7710	0.3705	$5{\cdot}10^{-5}$
ResNet-56	4×	0.8108	**0.8563**	0.8481	0.7938	0.3496	$5{\cdot}10^{-5}$
ResNet-56	8×	0.7739	0.8100	**0.8269**	0.7714	0.3447	10^{-4}
ResNet-56	16×	0.7362	0.7491	**0.7766**	0.7672	0.3185	10^{-4}

CPG at DepGraph's budget. DepGraph spends 39 dense-epoch equivalents (DEE), CPG 24–29 at the boundary, where it is within noise of DepGraph at four CIFAR-10 cells and ahead at the fifth (ResNet-56 8×, $+0.0109$, $t = 2.36$). To match DepGraph's budget we spend the extra epochs in stage 2. With its learning rate unchanged the two remain within noise at every cell; scaling it by a multiplier selected on validation from $\{1, 0.3, 0.1\}$ gives the CIFAR-10 CPG entries of Table 1, and 0.8243 ± 0.0054 at ResNet-56 16×, a column that table omits. CPG is then ahead of the swept DepGraph at four of the five cells, by $+0.0086$ to $+0.0477$ ($t = 2.41$ to 6.39; 5 seeds on ResNet-18, 3 on ResNet-56 and for DepGraph), and level at ResNet-56 4× (-0.0040, $t = -0.70$); Table 1 reports this configuration because it has the best mean seed-0 validation accuracy of CPG's three CIFAR-10 configurations (boundary budget, landing protocol, DepGraph's budget).

Three protocol details bear on this comparison. The multiplier grid was introduced after the first comparison with DepGraph, and validation selects its smallest value at every compared cell. A DEE charges CPG's stage-2 epochs, which run on the pruned network, by the MACs they keep, but DepGraph's 36+3 epochs in full, although its network also shrinks during its prune steps, so on MAC counts the accounting favors CPG. Wall-clock time on the same machine does not reverse the comparison: at 39 DEE, CPG takes 0.5–1.1 times DepGraph's median time per cell (measured without load control; DepGraph's times include the overhead of its regularizer).

Width. Scaling ResNet-18's channel counts under the landing protocol at 8×, the margin of CPG over fixed D=2 with the same penalty rises monotonically as the network narrows, reaching $+0.0350$ ($t = 3.68$) at a quarter of the standard width, as Section 4.2 predicts. On ResNet-56 it is positive at both widths tested ($+0.0142$ at 0.5, $+0.0247$ at 1).

CIFAR-100 (Table 5) uses the CIFAR-10 pipeline with the loader swapped and 3 seeds; every λ, DepGraph's strength and CPG's split of the 36 epochs are selected on validation at seed 0. CPG leads fixed D=2 by 0.0136 at ResNet-56 and DepGraph by 0.0089 at ResNet-18 ($t = 1.10$ and

1.24), and leads the best baseline at both of these cells under every split tried. At the two higher-compression cells, run with a reduced method set, it leads the best baseline by 0.0115 (fixed D=3, $t = 0.66$) and 0.0174 (DepGraph, $t = 4.64$). CPG's parameter compression (gray) follows from its MAC target: the per-cell penalty weights early, high-resolution channels, which hold few of the parameters, most heavily (Appendix E.4).

Table 5: **CIFAR-100 with the parameter compression each method realizes** (landing protocol, 3 seeds, mean ± population std; gray: parameter compression at the MAC target, not recorded for DepGraph). CPG's split of the 36 epochs between D=2 and D=3 is selected on validation: 6+30 at the first two targets and 9+27 at the two higher-compression targets. Bold: column maximum; "—": not run.

Method	ResNet-56 4×	ResNet-18 8×	ResNet-56 8×	ResNet-18 16×
Group lasso / SSL	0.5294 ± 0.0118 (2.9×)	0.5432 ± 0.0231 (6.3×)	0.4795 ± 0.0223 (6.6×)	0.4369 ± 0.0326 (13.2×)
spred	0.4661 ± 0.0314 (3.7×)	0.5525 ± 0.0175 (5.4×)	—	—
DWF	0.4783 ± 0.0199 (3.4×)	0.5211 ± 0.0335 (5.3×)	—	—
Hadamard	0.4964 ± 0.0064 (3.4×)	0.5343 ± 0.0212 (6.9×)	—	—
Powerpropagation	0.4966 ± 0.0109 (3.5×)	0.5717 ± 0.0051 (4.9×)	—	—
Network Slimming	0.5273 ± 0.0080 (3.0×)	0.5806 ± 0.0189 (6.1×)	0.4627 ± 0.0165 (6.1×)	0.5448 ± 0.0165 (12.7×)
DepGraph, swept	0.5391 ± 0.0227	0.6293 ± 0.0088	0.4653 ± 0.0068	0.6069 ± 0.0011
D-Gating, fixed D=2	0.5441 ± 0.0007 (3.3×)	0.5892 ± 0.0203 (9.6×)	0.4831 ± 0.0085 (6.6×)	0.4911 ± 0.0473 (24.7×)
D-Gating, fixed D=3	0.5247 ± 0.0253 (3.7×)	0.6078 ± 0.0101 (8.1×)	0.5062 ± 0.0128 (5.8×)	0.5544 ± 0.0121 (21.5×)
D-Gating, fixed D=4	0.5379 ± 0.0044 (3.5×)	0.5819 ± 0.0060 (12.4×)	—	—
CPG	$\mathbf{0.5577 \pm 0.0175}$ (2.1×)	$\mathbf{0.6382 \pm 0.0051}$ (3.8×)	$\mathbf{0.5177 \pm 0.0210}$ (5.0×)	$\mathbf{0.6243 \pm 0.0052}$ (8.8×)

Other budgets and scales. Setting m_g to cell g's parameter count gives the same construction against a parameter budget; against a fixed-D=2 control aligned the same way, CPG is ahead at three of four boundary-protocol cells, significantly at two ($t = 2.45$ and 8.40), and within noise at the fourth. On ResNet-50/ImageNet-1k (one seed, 32 epochs from scratch, boundary protocol), CPG lands 2× at 0.6619 and 4× at 0.6227, against 0.6467 and 0.6203 for fixed D=2 with the same penalty and a dense anchor of 0.7006 trained for the same 32 epochs. At 2×, raising λ from 0.006 to 0.02 brings the certificate-compatible share to 0.74 while staying within 0.0094 top-1 of the selected run.

F.2 Deciding at the boundary (Section 4.2)

Seeds and a sixth point. The headline rows of Table 6 (CPG, CPG + revival, CPG with a uniform λ, gate off, both D-Gating D=2 rows and Network Slimming) carry five seeds, except CPG + revival and Network Slimming on ResNet-18, which carry three; the other rows are ablation arms (Appendix F.6). ResNet-18 at 4× was also run under the boundary protocol (Figure 6) and is left out of Figure 3 because the gated rows lie within 0.003 of each other there; CPG again has the highest mean (0.8963 ± 0.0028).

The settle sharpens the certificate while matching the accuracy of ordinary epochs: over the ten cells of Table 6 it gains $+0.0035$ on average over the rows without it, level with the $+0.0037$ from returning its two epochs as ordinary epochs, and at matched λ it lowers the uncertified fraction at the boundary, as the theory predicts (ResNet-18 $0.42 \to 0.22$ at $\lambda = 0.033$), which is why it is the default.

Revival is off by default. At a minibatch boundary on a narrow network most near-zero cells report a certificate ratio above one, so revival returns them to the network and the landing later removes them by magnitude: at matched $\lambda \geq 0.06$ the uniform-λ row with revival leaves the boundary at 1.09–1.52× compression, the gate-off row at 2.04–7.03×. Reading the revival decision off the augmented gradient (Appendix F.3) brings the revival rows closer to CPG on ResNet-56, but CPG still ties or leads them at every cell of Figure 3, so the default does not revive.

The other defaults. Changing any one ingredient of CPG—the per-cell penalty, $D_{\max}$, revival, the settle—or replacing it by the gate-off control or by fixed D=2 with the same penalty lowers top-1 at its worst cell, by 0.0042 to 0.0857. The certificate slack δ is a hyperparameter: from $\delta = 0$ to ∞ the certificate-compatible fraction of near-zero cells rises from 0.460 to 1.000 on ResNet-18 (from 0.217 on ResNet-56); the default is $\delta = 0.05\lambda_g$. The certificate can also pick λ without a validation split: the knee of the compatible-share curve reproduces the validation-selected λ exactly

Table 6: **Deciding at the boundary, in full** (CIFAR-10, boundary protocol, 24–29 dense-epoch equivalents, λ selected on validation). *(a)* Top-1, mean $\pm$ population std; Figure 3 plots the CPG, gate-off, D-Gating and Slimming rows. *(b)* Certificate-compatible share of removals under the plain risk (for CPG, under the augmented objective in parentheses) / parameter ratio. *Majority-compatible*: top-1 / share / parameter ratio at the best validation point with at least half the removals certificate-compatible. The uniform-λ rows compress at $D{=}4$; gate off is their twin without the settle and any certificate machinery. "—": not recorded, or no λ reaches that share. Seed counts: Appendix F.2.

(a) Top-1

	ResNet-18		**ResNet-56**		
Row	8×	16×	4×	8×	16×
CPG	0.8850 ± 0.0043	0.8638 ± 0.0026	0.8558 ± 0.0055	0.8378 ± 0.0044	0.7897 ± 0.0101
w/o settle	0.8834 ± 0.0023	0.8655 ± 0.0026	0.8446 ± 0.0115	0.8403 ± 0.0025	0.7888 ± 0.0091
+ revival	0.8837 ± 0.0019	0.8620 ± 0.0037	0.8243 ± 0.0066	0.7521 ± 0.0055	0.7259 ± 0.0029
+ revival, w/o settle	0.8833 ± 0.0033	0.8534 ± 0.0129	0.8568 ± 0.0079	0.8223 ± 0.0156	0.7598 ± 0.0177
CPG, uniform λ	0.8786 ± 0.0069	0.8472 ± 0.0083	0.8502 ± 0.0103	0.8333 ± 0.0072	0.7910 ± 0.0050
w/o settle	0.8831 ± 0.0028	0.8507 ± 0.0059	0.8511 ± 0.0023	0.8211 ± 0.0085	0.7688 ± 0.0306
+ revival, w/o settle	0.8791 ± 0.0043	0.8512 ± 0.0083	0.8563 ± 0.0077	0.8300 ± 0.0072	0.7754 ± 0.0094
Gate off (control)	0.8825 ± 0.0021	0.8566 ± 0.0039	0.8552 ± 0.0073	0.8342 ± 0.0024	0.7834 ± 0.0198
D-Gating $D{=}2$, per-cell penalty	0.8799 ± 0.0032	0.8589 ± 0.0032	0.8549 ± 0.0049	0.8324 ± 0.0058	0.7814 ± 0.0134
D-Gating $D{=}2$	0.8736 ± 0.0059	0.8490 ± 0.0103	0.8437 ± 0.0051	0.8151 ± 0.0086	0.7543 ± 0.0113
Network Slimming	0.8699 ± 0.0014	0.8490 ± 0.0034	0.8380 ± 0.0135	0.8009 ± 0.0265	0.7660 ± 0.0198

(b) Certificate-compatible share / parameter ratio

	ResNet-18		**ResNet-56**		
Row	8×	16×	4×	8×	16×
CPG	.05 (—) / 6.7	.44 (.59) / 15.0	.36 (.67) / 2.5	.28 (.52) / 6.0	.12 (.23) / 17.8
w/o settle	.26 / 6.4	.23 / 15.0	.00 / 3.5	.28 / 5.7	.25 / 16.5
+ revival	.06 / 6.7	.05 / 16.2	.20 / 6.3	.17 / 20.7	.17 / 32.4
+ revival, w/o settle	.05 / 7.0	.22 / 21.4	.02 / 3.7	.02 / 8.8	.02 / 19.5
CPG, uniform λ	.58 / 16.1	.01 / 25.9	.24 / 4.2	.40 / 8.4	.55 / 21.0
w/o settle	.00 / 11.4	.48 / 26.9	.00 / 4.5	.00 / 9.7	.39 / 21.5
+ revival, w/o settle	.50 / 16.2	.48 / 27.2	.05 / 4.8	.04 / 9.8	.04 / 22.6
Gate off (control)	— / 15.2	— / 26.9	— / 4.4	— / 9.9	— / 20.8
D-Gating $D{=}2$, per-cell penalty	— / 6.3	— / 14.1	— / 2.8	— / 5.6	— / 17.6
D-Gating $D{=}2$	— / 13.7	— / 27.4	— / 4.3	— / 9.8	— / 22.4
Network Slimming	— / 7.7	— / 23.8	— / 3.9	— / 10.0	— / 24.4
Majority-compatible operating point (certificate-compatible share ≥ 0.5)					
CPG	.8742 / .73 / 4.9	.8626 / .64 / 12.8	—	—	—
CPG, uniform λ	.8786 / .58 / 16.1	.8485 / .90 / 31.0	.8278 / .63 / 7.4	.8253 / .58 / 10.5	.7910 / .55 / 21.0
+ revival, w/o settle	.8791 / .50 / 16.2	.8499 / .70 / 26.1	.8075 / .54 / 5.9	—	—

on ResNet-18, and on ResNet-56 the best of three simple certificate-based rules is within 0.0076 of it.

Table 7: **What the landed structures run at.** Compact ResNets rebuilt from the per-block channel counts of one boundary landing of CPG per cell (seed 0, at the λ selected when the timing was run, hence parameter ratios slightly different from Table 6), timed round-robin against the dense model on one GPU and on a single CPU thread. Speed-up: dense over landed time. The GPU is launch-bound below a few hundred $32{\times}32$ images, so its speed-up is largest at batch 512.

		GPU speed-up		**CPU speed-up**		
Model	Target	batch 64	batch 512	1 thread, b1	1 thread, b64	Param. ratio
ResNet-18	8×	1.46×	1.98×	4.37×	4.23×	6.6×
ResNet-18	16×	1.65×	2.17×	5.71×	4.37×	13.8×
ResNet-56	4×	1.06×	1.37×	1.93×	1.94×	2.7×
ResNet-56	8×	1.10×	1.46×	2.27×	2.41×	5.9×
ResNet-56	16×	1.12×	1.52×	2.46×	2.59×	16.8×

Cost. Deep gating adds $(D{-}1)|\mathcal{G}|$ scalars (0.053% of ResNet-18 at $D{=}4$), all removed at collapse, so the deployed model has exactly the baseline architecture, and each depth transition is $O(p + |\mathcal{G}|D')$ scalar operations (D' the new depth) with no optimization solve (Proposition 2). An epoch of fixed-depth deep gating or of three-stage CPG on ResNet-18/CIFAR-10 takes 1.27–1.28× the dense time, mostly gate-module launch overhead on small batches, so the landing protocol's 36

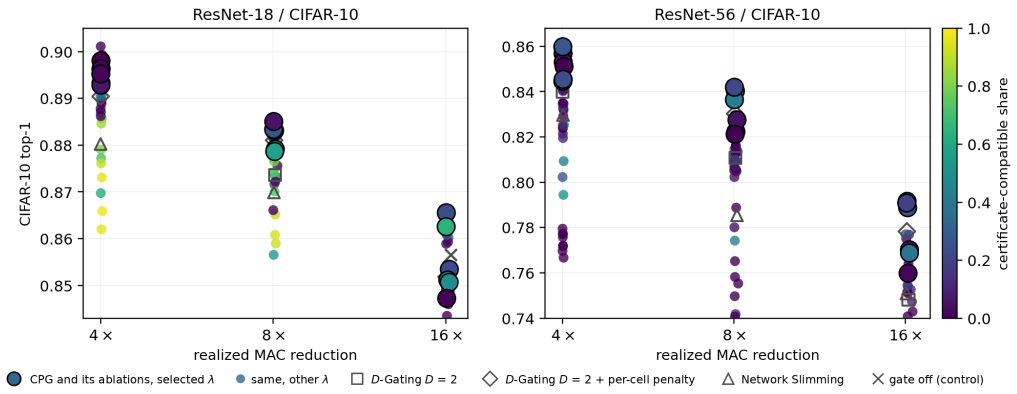


Figure 6: **The certified sparsification frontier.** Every eligible boundary landing, plotted as accuracy against realized MAC reduction with the certificate-compatible share encoded in color; large markers are the validation-selected points of CPG and each of its ablations in Table 6, small markers their other λ, gray the baselines and controls. Among the methods shown the CPG family traces the upper envelope, and the color axis is one no other method produces.

gated epochs and 3 fine-tuning epochs amount to about 49 dense-epoch equivalents; the re-lifts cost about 1% of one dense epoch and the certificate check 0.50% of the run (1.83% on BERT-base). The landed structures are also faster at inference (Table 7).

F.3 Does the predicted geometry hold? (Section 4.3)

Controlled checks (Figure 7). In a frozen landscape, with cells initialized in canonical gauge at factor magnitude ε, the escape–trapping boundary of Proposition 1 holds along the balanced ray to numerical precision at $D{=}2, 4$ and within 4% at $D{=}3$, and the barrier height matches $\Phi(\rho^*)$ to a median ratio of 1.07. The gauge-attraction rate $-4\lambda/D$ of Lemma 2 is recovered to within 0.4–4.1% in all nine (λ, D) cells, with an $O(\eta^2)$ discrete-time residual.

The latent origin is reached, and removals stay removed. Tracking $\|\boldsymbol{c}_g(t)\|$ and $\|\boldsymbol{v}_g(t)\|$ separately under memoryless SGD, on average 73.2% of removed cells follow the pure-decay envelope and 31.4% underflow to exact float32 zero without thresholding, rising to 100% wherever the step budget permits; the collapsed-only state of Remark 1 has incidence and regrowth rate 0.000 at every configuration. The momentum displacement $\eta\beta\|\boldsymbol{m}_g\|$ that Remark 1 predicts is $\sim 10^{-25}$ in natural training, because decay-driven absorption arrives with an already-decayed buffer, and buffer resetting removes it. Accordingly, across 121 stage-2 runs with momentum, none of the 18,520 cells at the latent origin after the re-lift was nonzero at the end of training—the permanence Corollary 2 relies on.

LeNet-5. Under memoryless SGD the violation rate on LeNet-5/MNIST is $3.3\% \pm 1.5\%$ at $D{=}2$ against $20.0\% \pm 1.4\%$ and $27.9\% \pm 1.0\%$ at $D{=}3, 4$ (3 seeds; Figure 4a). What carries over from the frozen landscape to real training is this $D{=}2$ versus $D \geq 3$ dichotomy, while the barrier's scalings are verified in the controlled checks above.

Initialization scale. Fixed $D{=}3$ improves monotonically with the initialization scale σ ($F_1 = 0.415$ to 0.903) while $D{=}4$ stays flat, at either floor: initial radii lie far above ε, so Proposition 1 is read along the trajectory rather than at initialization. Read this way—a cell is predicted trapped iff it lies inside its barrier when it first falls below half its radius—it agrees with the outcome for 0.92–1.00 of true cells at every depth and scale, and for 0.74–1.00 of all cells (Figure 5a), nuisance cells being the ones that can lie outside the leading-order regime. No cell starts inside its barrier; trapped true cells enter one after 50–150 steps, once their gradient has fallen by 50–140×. Rows that decide at $D{=}2$ reach $F_1 = 1.000$ at every setting.

Which gradient the certificate is read with. Definition 1 tests the empirical-risk gradient, but a run trained with data augmentation descends a different objective. On ResNet-18 and ResNet-56 boundary runs, the certificate-compatible fraction of near-zero cells changes by at most 0.03 between a 10k-sample and the full plain gradient, whereas averaging over augmentations raises it by 0.11–

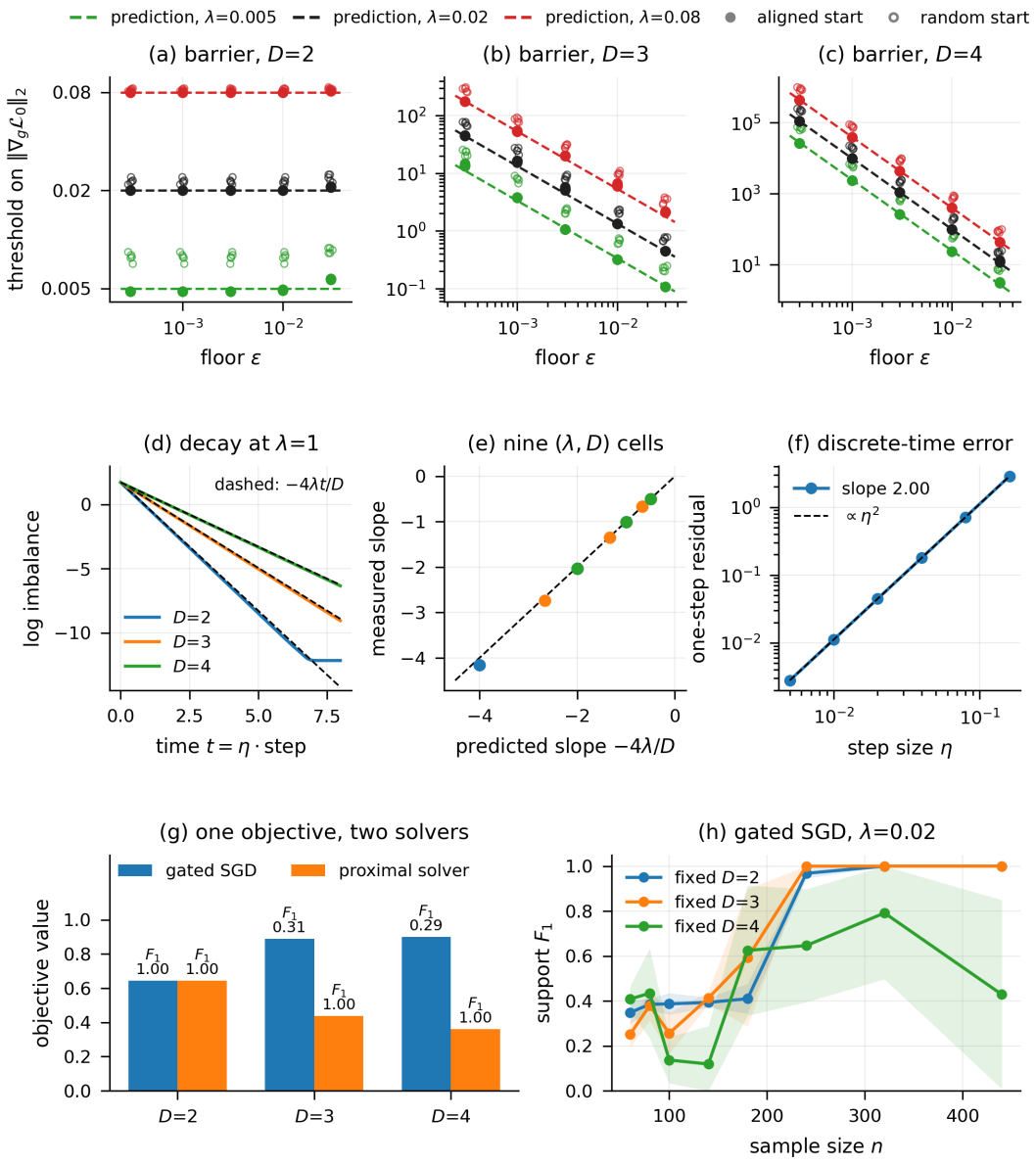


Figure 7: **Numerical checks of the theory in controlled settings.** *(a)–(c)* Proposition 1 in a frozen one-cell landscape: the gradient threshold between escape and trapping against the floor ε, for three values of λ; dashed lines are the prediction $2\lambda/(D\varepsilon^{D-2})$ of Proposition 1(ii). Cells started along the balanced ray (filled) sit on the prediction (median relative displacement 8.6×10^{-8} at D=2 and 2.5×10^{-7} at D=4, within 4% at D=3); cells started in a random direction (hollow) are displaced by an $O(1)$ alignment factor, and 86.6–97.9% of them still fall on the predicted side. The barrier height matches $\Phi(\rho^*)$ to a median ratio of 1.07. *(d)–(f)* Lemma 2: *(d)* log per-cell imbalance against continuous time at $\lambda = 1$, with the predicted slopes $-4\lambda/D$ dashed; *(e)* fitted against predicted slopes over the $\lambda \times D$ grid, agreeing to 0.4–4.1% in all nine cells; *(f)* the residual of the one-step discrete identity scales as $O(\eta^2)$ (log-log slope 2.00001). *(g)–(h)* Theorem 2 (correspondence of minimizers): *(g)* gated SGD against a proximal-gradient solver on the same objective at $\lambda = 0.02$; at D=2 the objectives agree to 2.7×10^{-6} and both are observed to recover the support exactly, while at $D \geq 3$ the gap is the trapping Theorem 1 predicts, which CPG avoids; *(h)* support recovery of fixed-depth gated SGD against sample size at the same λ (mean $\pm$ std over seeds).

Table 8: **BERT-base, landing protocol** (SST-2, 3 seeds, mean ± population std). *Coverage* is a diagnostic—the fraction of a run's final removals that satisfy the condition. *Decision share* is the operational quantity: of the removals committed at the boundary, the fraction that were certificate-compatible when committed. The $D_{\max}$=4 row keeps the decision ledger (BOUNDARY mode, $\delta = 0.05\lambda_g$). Bold: best accuracy and most heads removed, the dense reference excluded.

Method	Acc.	Heads rem.	FFN rem.	Coverage (diagnostic)	Decision share (cert. / uncert.)
Dense (reference)	0.9190 ± 0.0038	0%	0%	—	—
D-Gating, fixed D=2	$\mathbf{0.8238 \pm 0.0005}$	30%	75%	0.999 ± 0.001	—
CPG	0.8180 ± 0.0046	**45%**	75%	0.999 ± 0.000	—
$D_{\max}$=4, ledger kept	0.8165 ± 0.0095	36%	75%	—	0.73 (20,225 / 7,389)

0.32, saturating at four augmentations; the plain-risk shares we report are therefore conservative. Likewise the threshold must follow the trained penalty: the per-cell λ_g of (6) is in the objective, so Corollary 1 applies verbatim with λ_g in place of λ.

F.4 The certificate at transformer scale (Section 4.4)

Setup and results. LayerNorm uses no batch statistics, so on a transformer the gradient the certificate reads does not depend on the batch composition, as it does through BatchNorm in the CNNs. We gate the 144 heads and 36,864 FFN neurons of pretrained BERT-base (Devlin et al., 2019) and train on SST-2 without distillation, so absolute scores are not directly comparable with DynaBERT (Hou et al., 2020) or CoFi (Xia et al., 2022). Certificate coverage is 0.999 ± 0.001 over three seeds (Table 8), and CPG removes half again as many attention heads as fixed D=2 (45% against 30%) at comparable accuracy. The allocation is structured—early and mid-layer FFN is pruned, late-layer FFN and most heads are kept—and coverage stays near 1 at every λ that removes cells (Figure 8). Of the removals committed by the ledger-keeping run that compresses at D=4, 73% are certificate-compatible when committed; as in Corollary 2, the certificate attaches to the removal decision rather than to the final optimum. Corollary 1 applies at a convergent limit, where the certified fraction is 0.999.

Against structured-pruning baselines (Table 2, main text). Under the boundary protocol every row lands exactly on the encoder-MAC target within the same four-epoch budget. CPG has the highest average accuracy (0.8858 against 0.8854 for the best baseline), is best at QNLI 4× and within noise of the best baseline at the other three settings ($|t| \leq 0.6$), and matches or exceeds the dense model at every setting. The methods reach the target differently: first-order, magnitude and DepGraph-style scores remove FFN neurons first (no heads at 2×, 99–100% of FFN neurons at 4×), while CPG removes 77% and 85% of the attention heads on average. Under this protocol cells are removed before they reach the latent origin, so the comparison is one of accuracy, and the ledger belongs to the landing-protocol runs of Table 8.

Adaptive gate optimizers. The D=2 certificate is derived for gradient descent, so we changed only the gate optimizer. **AdamW on the gates prunes 100% of FFN neurons** against 75% for plain SGD at comparable accuracy (0.831 against 0.824), and its removals *still satisfy the certificate*: the uncertified fraction among the 30,524 cells it drives exactly to the latent origin is 0.0001. The adaptive optimizer changes the sparsity–accuracy balance, not the validity of the certificate, which suggests that the certified stage can carry over to a production optimizer; the theory-validation runs keep memoryless SGD, under which the latent origin is absorbing (Lemma 2(b)).

F.5 What makes CPG work

Four controls isolate the active ingredient, each holding the objective, the budget and the λ grid fixed and changing one thing. *Control 1: the depth transition, not the exponent.* Proximal continuation $1 \to \frac{2}{3} \to \frac{1}{2}$ reaches $F_1 = 0.891$ on the synthetic problem, while a proximal solver at *fixed* $q = 1/2$ reaches 1.000: the gain comes from entering the deep objective on a support selected under the D=2 geometry (Theorem 1), not from the sequence of exponents. *Control 2: the certified starting depth.* A 3→4 control that keeps the transition but omits the D=2 stage matches fixed D=3 (0.810 ± 0.237 against 0.807 ± 0.241), whereas every progressive schedule that begins at D=2 recovers the support exactly across the two-decade λ grid, where no fixed depth reaches 0.95 (Figure 5).

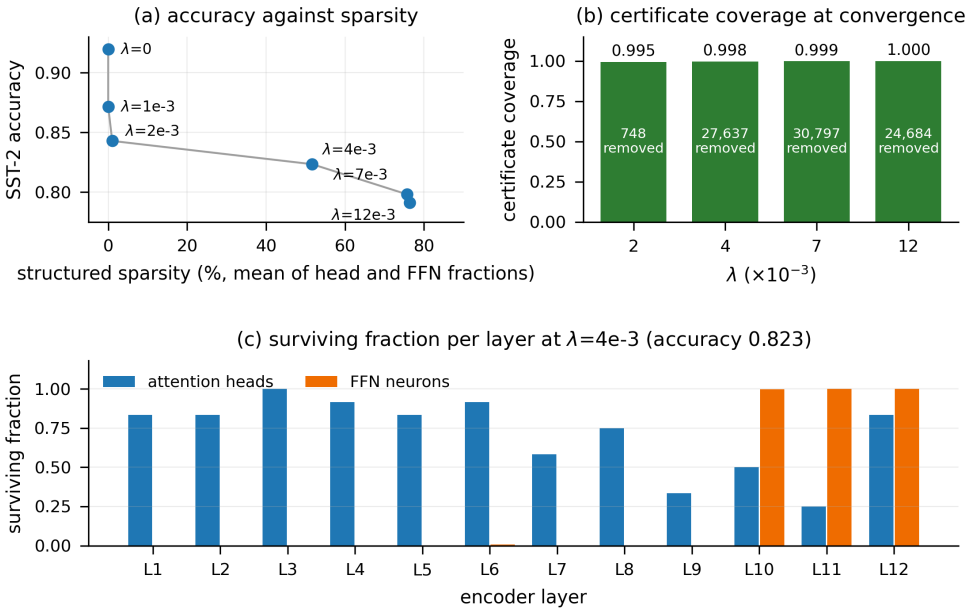


Figure 8: **BERT-base on SST-2** with D=2 gates on every head and FFN neuron, across a λ sweep (one seed). *(a)* Accuracy against structured sparsity, the mean of the pruned head and FFN fractions. *(b)* Certificate coverage: the fraction of removals satisfying $\|\nabla_g \mathcal{L}_0\|_2 \le \lambda$ at convergence, at every λ that removes cells, with the number of removed cells in each bar. *(c)* Surviving fraction per layer at $\lambda = 4{\times}10^{-3}$: early and mid-layer FFN is pruned, late-layer FFN and most heads are kept.

Control 3: the shared scalar gate. Element-wise factorizations reach $F_1 = 0.19$–0.22 (Table 1); the group-shared gate is what makes a group-level decision available and the group Hessian of Theorem 1 block-diagonal. *Control 4: the certificate at no cost in accuracy.* Across six stage-1 budgets the certificate's effect on synthetic F_1 is exactly 0.000, and no value of $\delta \in \{0, 0.05\lambda, 0.2\lambda, \infty\}$ or $\kappa \in \{1.5, 2, 4\}$ changes any metric: at a stationary boundary every near-zero cell already satisfies the condition, as Theorem 1 predicts, so the ledger is free information about the decisions.

The synthetic benchmark (Table 9) collects the fixed-depth rows, the CPG schedules and these controls. Its bottom block checks Theorem 2: smooth bilevel via L-BFGS (0.372), gated SGD (0.370) and the proximal group lasso (0.359) agree on the same D=2 objective, so fixed D=2's gap is not an optimizer artifact. Nor does any λ close it: fixed D=2 peaks at $F_1 = 0.848$ (recall 0.90) with its survivors shrunk to 0.40 of their true magnitude, and it reaches the correct sparsity only at a λ some 90–150$\times$ larger than the progressive schedules use to recover the support exactly (Figure 5b,c).

Hyperparameters. Stage 2 reuses the stage-1 penalty, $\lambda_2 = \lambda_1$, which gives higher accuracy and a lower uncertified rate than a larger stage-2 penalty. The target depth operates at $D_{\max} \in \{3, 4\}$, which is 4–6 points more accurate than $D \ge 6$ and keeps the synthetic support, and a third stage matches $D_{\max}$=4 on the synthetic problem without ordering consistently against it on CIFAR (Table 4), so Algorithm 1 uses a single transition. For the per-cell penalty, the linear weight m_g that the Lagrangian relaxation gives is the default; on ResNet-56 it is better than or equal to $\sqrt{m_g}$ and m_g^2 at every cell compared (one seed per cell, combined with gated freezing).

Sample size and irrepresentability (Table 10). On the benchmark design CPG reaches $F_1 = 1.000$ from $n = 160$, fixed D=2 and D=3 from $n = 400$. We then pair each true group with a nuisance group at correlation ρ and measure the group irrepresentable quantity $\mathrm{IC} = \max_{g \notin S} \|\boldsymbol{X}_g^\top \boldsymbol{X}_S (\boldsymbol{X}_S^\top \boldsymbol{X}_S)^{-1} \boldsymbol{z}_S\|_2$, $\boldsymbol{z}_S = (\boldsymbol{w}_g^*/\|\boldsymbol{w}_g^*\|_2)_{g \in S}$, which runs from 0.20 at $\rho = 0$ to 2.04–2.20 at $\rho = 0.9$ and exceeds 1 at $\rho \ge 0.45$. CPG reaches $F_1 \ge 0.95$ earlier than fixed D=2 at every ρ, by factors of 2.5, 4 and 10, and never later than fixed D=3; the convex group lasso needs $10\times$ more data than CPG at $\rho = 0.6$ and 0.9. A proximal solver at fixed $q = 1/2$ crosses at $n = 400$ everywhere, so the earlier crossing at $n = 160$ for $\rho \le 0.6$ comes from the depth transition rather than the exponent. At $n = 6400$ every row of this study (fixed D=2, D=3, group lasso,

Table 9: **The synthetic benchmark, in full** (p=1000, G=100, s=10 true groups, n=160; 5 seeds; one 12,000-step budget and one λ for every row; initialization $0.1\mathcal{N}(0,1)$, floor 10^{-3}). *Kept* of 100, *Prec.* with respect to the true groups, *Viol.* the fraction of pruned groups violating $\|\nabla_g \mathcal{L}_0\|_2 \le \lambda$—an unconverged escape at D=2, a trapped group at $D \ge 3$. The CPG rows ran with the ledger off (δ=∞); running Algorithm 1 in full, with the ledger at $\delta = 0.05\lambda$, changes no entry of the $D_{\max}$=4 row (Control 4). The lower blocks carry the controls of Appendix F.5. Bold: best per column (highest F_1 and precision, *Kept* closest to s=10, lowest violation rate).

Row	Kept	Prec.	F_1	Viol.
Fixed depth				
Fixed D=2	44.4 ± 3.9	0.227	0.370 ± 0.028	**0.021**
Fixed D=3	14.4 ± 5.4	0.754	0.807 ± 0.241	0.587
Fixed D=4	20.8 ± 0.7	0.319	0.430 ± 0.085	0.211
Ours				
CPG	$\mathbf{10.0 \pm 0.0}$	**1.000**	$\mathbf{1.000 \pm 0.000}$	0.420
$D_{\max}$=4	$\mathbf{10.0 \pm 0.0}$	**1.000**	$\mathbf{1.000 \pm 0.000}$	0.196
three stages	$\mathbf{10.0 \pm 0.0}$	**1.000**	$\mathbf{1.000 \pm 0.000}$	0.196
3→4 (control: no D=2 stage)	14.2 ± 5.2	0.757	0.810 ± 0.237	0.297
Is this ℓ_q continuation?				
Proximal group lasso (q=1)	46.0 ± 3.7	0.219	0.359 ± 0.022	—
Proximal ℓ_q, fixed q=2/3	12.2 ± 2.7	0.858	0.914 ± 0.106	—
Proximal ℓ_q, fixed q=1/2	$\mathbf{10.0 \pm 0.0}$	**1.000**	$\mathbf{1.000 \pm 0.000}$	—
Proximal ℓ_q continuation $1 \to \frac{2}{3} \to \frac{1}{2}$	13.0 ± 3.7	0.829	0.891 ± 0.134	—
Is this deep factorization?				
Element-wise D=2 (spred-like)	86.8 ± 3.5	0.108	0.194 ± 0.005	—
Element-wise D=3 (DWF-like)	73.8 ± 2.3	0.122	0.215 ± 0.015	—
Element-wise D=4 (Hadamard-like)	70.8 ± 1.3	0.125	0.218 ± 0.022	—
Powerpropagation α=2	87.4 ± 2.7	0.105	0.189 ± 0.014	—
Three solvers, one D=2 objective (against the fixed D=2 and proximal q=1 rows above)				
Smooth bilevel (L-BFGS on the D=2 split)	44.0 ± 4.0	0.229	0.372 ± 0.028	—

CPG; three seeds per design) reaches 1.000 at every ρ, even where IC = 2.04: the irrepresentable condition governs support-consistent recovery along the λ path, whereas a thresholded F_1 at large n only needs a small-λ solution to rank the true groups first. The advantage of depth over the convex penalty is therefore one of sample efficiency, which CPG attains while certifying its first-stage decisions.

Table 10: **Support recovery against sample size.** *Left*: support F_1 on the synthetic benchmark (5 seeds; the main-text runs use n=160). *Right*: the smallest n at which each row first reaches $F_1 \ge 0.95$ on the equicorrelated design, at nuisance correlation ρ. CPG compresses at D=4 here; keeping the ledger or not gives identical entries. Fixed D=4 is trap-limited at every n: at $n \ge 800$ it keeps only 4–8 cells, i.e. it prunes true ones. Bold: best per column; "—": not run.

	F_1 **at sample size** n (benchmark)							n **to reach** $F_1 \ge 0.95$		
Row	80	160	200	400	800	1600	3200	ρ=0	0.6	0.9
Fixed D=2	0.376	0.370	0.600	**1.000**	**1.000**	**1.000**	**1.000**	400	640	6400
Fixed D=3	0.395	0.807	0.897	**1.000**	**1.000**	**1.000**	**1.000**	**160**	250	640
Fixed D=4	0.369	0.430	0.434	0.440	0.400	0.800	0.400	400	640	640
Proximal group lasso (q=1)	—	0.359	—	—	—	—	—	400	1600	6400
Proximal q=1/2	—	**1.000**	—	—	—	—	—	400	400	**400**
CPG, $D_{\max}$=4	**0.504**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**160**	**160**	640

Using the certificate: gated freezing. Excluding survivors with $\gamma_g < 0.25$ from the stage-2 penalty gives 0.8573 ± 0.0037 against 0.8489 ± 0.0036 at ResNet-18 16× (+0.0083, $t = 3.80$, ten seeds against the five-seed control) and leaves 8× unchanged. The threshold was fixed before the runs and never swept, and the gain replicates on two independent seed blocks (+0.0069, $t = 2.96$, on seeds 0–4; +0.0098, $t = 3.73$, on seeds 5–9).

Revival. On hand-built populations, Lemma 3's prediction of initial motion holds exactly: cells placed below r^* shrink from the first step and cells above it grow. The cap $\bar{r}$ keeps revived cells in

the leading-order regime of Proposition 1; with it, every revived cell at a stochastic boundary is a nuisance cell and returns to zero, at $F_1 = 1.000$ on all seeds.

F.6 Protocols, hyperparameters and compute

Certificate measurement. At convergence we collapse the model, identify pruned groups ($|P_g|\|\boldsymbol{v}_g\| <$ machine epsilon), and compute $\nabla_g \mathcal{L}_0$ by a full-dataset backward pass through the *collapsed* network at the pruned configuration; we report the distribution of $\|\nabla_g \mathcal{L}_0\|_2/\lambda_g$ and the violation rate $\Pr[\|\nabla_g \mathcal{L}_0\|_2 > \lambda_g]$. Candidates are committed at tolerance $\varepsilon_c = 10^{-3}$, and ledgers are recorded or scored at slack $\delta = 0.05\lambda_g$; the landing-protocol CPG runs of Table 4 ran at $\delta = \infty$ with the ledger scored afterwards, and revival uses $\kappa = 2$.

Splits and selection. CIFAR-10 and CIFAR-100 runs use one fixed 45k/5k train/validation split, and no hyperparameter is selected on test data: every selected one (λ, DepGraph's strength, CPG's stage split and the stage-2 learning-rate multiplier) is selected on validation. For each row and target, λ is selected over a two-decade grid among the *eligible* landings, those within $\pm 10\%$ of the target, and every landing is bisected onto the target MACs exactly, so no result is interpolated; the landing protocol then fine-tunes for 3 epochs and recalibrates BatchNorm. On BERT, whose reported metric is GLUE validation accuracy, every strength in Table 2 is selected on 2k held-out training examples per task and target.

Optimizers and seeds. Stage 1 runs memoryless SGD by default, the setting of every theory check; on ResNet-18 this halves the boundary violation rate while moving top-1 by less than 0.004. The momentum rows of Table 4 reset a cell's buffer when it is absorbed. On BERT the gates train with SGD and the backbone with AdamW. On CIFAR-10, headline rows, re-run baselines and ablation arms carry 3 or 5 seeds, as stated with each result (2 for ** entries, 10 for gated freezing); CIFAR-100, LeNet-5 and BERT use 3 seeds, the synthetic benchmark 5, and ImageNet and the BERT λ sweep one.

Compute. Every experiment ran on one NVIDIA GeForce RTX 5090 GPU (32 GB) with an Intel Core Ultra 9 285 CPU and 128 GB of memory.