BENIGN LOSS LANDSCAPES CAN COEXIST WITH WORST-CASE HARDNESS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural networks are expressive enough to contain worst-case targets that can be *evaluated* in polynomial time but cannot be *learned* in polynomial time by gradient descent. For practical tasks they nonetheless learn well, raising the question of what non-generic structure of real-world targets enables this. Existing surrogate models cannot pose this question because they either lack hard-to-learn targets entirely (deep linear networks) or cannot evaluate such targets efficiently (kernel methods, infinite-width limits). We study tree tensor networks (TTNs), a model class that generalizes deep linear networks and Tucker decompositions. We show they embed arbitrary read-once Boolean formulas, and thus contain polynomial-size targets that cannot be learned by gradient descent in polynomial time under the same mechanism as neural networks. Despite this, we prove that their loss landscapes are conditionally benign for every realizable target: every local minimum that is *minimum-norm* is global. Thus, surprisingly, bad local minima are not what distinguishes between typical and worst-case problems in TTNs. Instead, learning difficulty in TTNs can arise from high-order degenerate saddle points, which we show are caused by rank-deficiency. This is explored through a case study of the parity function, illustrating the potential for TTNs to relate landscape geometry to computational hardness.

1 INTRODUCTION

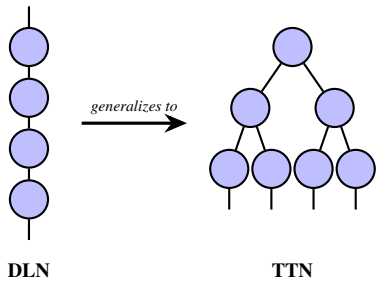
One of the core mysteries of deep learning theory is why gradient-based optimizers reliably find high-quality minima, even when navigating highly non-convex loss landscape. This phenomenon led to the hypothesis that these landscapes are *benign*, in the sense that all local minima are global, or at least that non-increasing escape paths exist for any suboptimal minima. Such results have been proven for models like deep linear networks (Baldi & Hornik, 1989; Kawaguchi, 2016; Lu & Kawaguchi, 2017) or very wide networks (Venturi et al., 2018; Nguyen, 2019; Petzka & Sminchisescu, 2021).

However, any account of why training succeeds must contend with the following fact: there exist targets like the *parity function* which neural networks can express perfectly, but cannot efficiently learn via gradient descent (Shalev-Shwartz et al., 2017; Shamir, 2018). Given the empirical success of deep learning, we are forced to conclude that practical targets have some property, not shared by worst-case targets, that enables learning them in polynomial time. Consequently, existing benign landscape results, and existing surrogate models more broadly, do not preserve the phenomenon we want to study. Doing so requires two properties of the surrogate: (A) its hypothesis class must contain worst-case targets which cannot be learned in polynomial time by gradient descent, like the parity function; and (B) hypotheses in the class must themselves be evaluable in polynomial time; otherwise the learning difficulty is trivial and unsurprising.

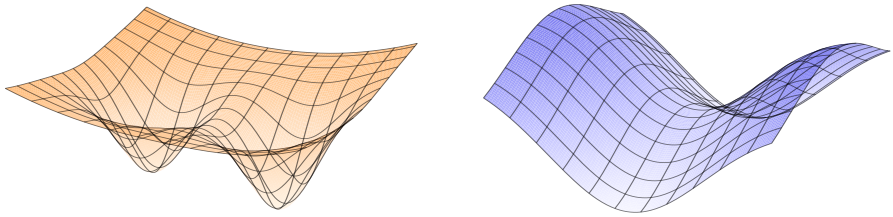
Every commonly-studied surrogate we are aware of fails either (A) or (B). Deep linear networks parameterize only linear functions and contain no hard-to-learn targets, failing (A). Infinite-width limits (without closed-form kernels) cannot be analytically computed in the first place, failing (B). Kernel methods (random feature models, neural tangent kernels, etc.) must fail either the first or the second condition because a kernel hypothesis $\hat{f}(x) = \sum_i a_i k(x, x_i)$ takes the same resources to evaluate as to learn, so it cannot be polynomial-time evaluable and super-polynomial to learn.

In this paper, we study ***tree tensor networks*** (TTNs) and demonstrate that they pass both properties (A) and (B). A TTN is defined by a tree whose nodes carry multilinear maps (tensors), with the model's

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.



	Benign loss landscape	Can express hard functions	Can evaluate hard functions in poly time
Typical neural network	empirically	✓	✓
Deep linear network	provably	×	-
Infinite-width limits	provably	✓	×
Kernel approximations	provably	✓	×
Tree tensor network	provably	✓	✓



Not benign *Benign*

Figure 1: **Top:** a deep linear network (DLN) is a chain of tensors; a tree tensor network (TTN) generalizes the DLN to a tree topology (left). The TTN captures nonlinear features and functions which are hard-to-learn while retaining a provably benign loss landscape (right). **Bottom:** our main results, Theorem 5.2 and Theorem 4.3, prove that under some conditions, the only critical points of the loss other than global minima are saddle points (right); there are no local minima which are not global (left).

output obtained by composing maps along edges (Figure 1). TTNs are an immediate nonlinear generalization of deep linear networks (DLNs) and Tucker decomposition. Most importantly, TTNs satisfy our two desiderata for surrogate models: they can express any read-once Boolean formula, including worst-case targets like the parity function (A), and any TTN can be evaluated in time polynomial in the tree size and bond dimension (B). TTNs therefore contain targets that cannot be learned in polynomial time by gradient descent (Section 3).

Our contribution is to leverage the analytic tractability of TTNs to understand how worst-case difficulty manifests in their loss landscapes. We show that, surprisingly, even for worst-case targets, their loss landscapes are conditionally benign, so all minimum-norm local minima are global minima (Theorem 4.3). Even without the minimum-norm condition, we prove that from any non-optimal parameter point there exists a continuous non-increasing loss path to strictly lower loss, that is, there are no "bad valleys" (Corollary 4.4).

Yet, if the loss landscapes of such a model are benign, how is that compatible with worst-case hardness? We suggest that such difficulty can instead come from high-order degenerate saddle points, as suggested by Dauphin et al. (2014); Sankar & Balasubramanian (2017). To demonstrate, we characterize these saddle points, proving that any critical points which are not global minima must occur at Tucker-rank-deficient points, where the model is "effectively smaller" than its maximum capacity (Theorem 5.2). The parity function provides a concrete case study of a known hard-to-learn function to illustrate this phenomenon, proving that the order at which a descent path can escape the saddle point around the "halfway learned" solution must be lower bounded by a linearly increasing function of the number of inputs n (making the loss function exponentially flat in n).

We hope that this paper will help establish TTNs as a new surrogate model for deep learning training dynamics. Because TTNs express computationally nontrivial functions, they provide the first analytically tractable model class designed to systematically study how computational structure of the target shapes the geometry of the optimization landscape.

Contributions. Our precise contributions are thus:

- **We prove TTNs lack bad local minima,** under a *minimum-norm* assumption. Despite expressing targets which cannot be learned efficiently by gradient descent (Corollary 3.4), we show that for

any realizable target, every minimum-norm local minimum is globally optimal (Theorem 4.3), and all other parameters admit non-increasing escape paths (Corollary 4.4).
- **We show rank-deficiency instead creates bad saddle points.** In particular, we prove that all non-optimal critical points are Tucker-rank-deficient (Theorem 5.2). Through a case study involving the parity function (Proposition 5.3), we show that these saddles can be highly degenerate with the potential to slow optimization.
- **We connect this geometry to broader theory.** As an auxiliary technical contribution, we show that our minimum-norm condition coincides with the *Kempf-Ness condition* from geometric invariant theory, generalizes a well-known *balancedness* condition in deep linear networks, and is preserved by gradient flow (Appendix F).

We verify the two main theorems (Theorem 4.3, Theorem 5.2) using the LEAN proof assistant (de Moura & Ullrich, 2021; mathlib Community, 2020): links to the corresponding LEAN code are provided in the right margin, and more details are given in Appendix J.

Ultimately, we establish the following picture of the TTN loss landscape, to inform the broader question of how training behavior depends on the function being learned:

> For regularized solutions, there are no spurious local minima. However, challenging saddle points can still occur when the model is low rank. These can impede convergence to worst-case targets, making them hard to learn.

2 RELATED WORK

Tensor networks Tensor networks originated in quantum state and many-body physics as a memory-efficient approximation to high-dimensional multi-particle states with path-graph (White, 1992; Östlund & Rommer, 1995; Schollwöck, 2011; Vidal, 2003), lattice (Verstraete & Cirac, 2004; Cirac et al., 2021), and tree topology (Shi et al., 2006). In machine learning, tensor networks have been employed both for supervised (Stoudenmire & Schwab, 2016) and unsupervised learning (Han et al., 2018; Miller et al., 2021) (in both cases later extended to tree topologies (Stoudenmire, 2018; Cheng et al., 2019)) leveraging their ability to efficiently represent high-dimensional feature tensors or multi-feature interactions (Novikov et al., 2016). Interestingly, Pearce et al. (2024) and Dooms et al. (2025) show that neural networks with bilinear activation function are equivalent to tree tensor networks with a copying operation, which provides a natural direction for future work.

Benign loss landscapes. Baldi & Hornik (1989) proved that every local minimum of the square loss is global for a two-layer linear network. Kawaguchi (2016) extended it to arbitrary depth and widths, and further showed that every critical point which is not a global minimum is a saddle. Later work extended and improved upon these results in deep linear networks (Lu & Kawaguchi, 2017; Yun et al., 2017; Zhou & Liang, 2017; Achour et al., 2024; Trager et al., 2019; Mehta et al., 2021; Bah et al., 2022; Arora et al., 2018a). Beyond DLNs, Frandsen & Ge (2022) show a similar lack of local minima for Tucker decomposition, though requiring additional assumptions such as regularization and realizability. Tree tensor networks generalize both Tucker decomposition and DLNs; we prove conditional benign loss landscape results for arbitrary tree tensor networks (Theorem 4.3).

It is known that some nonlinear neural networks *can* have spurious local minima (Ding et al., 2019; Swirszcz et al., 2016; Safran & Shamir, 2018; Christof & Kowalczyk, 2024); beyond the technical fact that such results do not apply to TTNs, our results do not contradict this even in spirit as they are conditional (see Appendix D.7), instead showing that such local minima are not *inherent* to nonlinear networks and can be avoided in a specific regime. For many types of neural networks, one can prove a weaker "no bad valleys" condition (Venturi et al., 2018; Nguyen, 2019; Nguyen et al., 2018; Petzka & Sminchisescu, 2021), similar to our Corollary 4.4, but only under extreme overparameterization.

Learning computationally hard targets. Kearns (1998) showed that parities are not learnable in polynomial time by statistical query algorithms; this applies to full-batch gradient descent, as it can be formulated as a statistical query algorithm (Feldman et al., 2021; Goel et al., 2020; Shalev-Shwartz et al., 2017). Abbe et al. (2021b) showed that such results transfer to minibatch SGD, under limits on gradient precision and minibatch size. A more recent line of work characterizes how such hard

targets affect gradient-based learning dynamics. Barak et al. (2022) use the problem of sparse parity learning to study the learning dynamics of emergent capabilities. Malach & Shalev-Shwartz (2019) study gradient-based learning when the target is a *tree-structured Boolean circuit*, and explore what differentiates parity targets from learnable targets (a *local correlation assumption*). Abbe et al. (2022) and Abbe et al. (2023) explore a similar question for Boolean functions, and relate computational properties (*leap complexity*) of the target to saddle-to-saddle learning dynamics. However, while these results explicitly predict gradient learning dynamics, they apply only for particular families of targets; our results apply for every realizable target, including worst-case ones.

Degenerate saddle points. For neural networks, it has been argued that the primary obstructions to learning are saddle points rather than local minima (Choromanska et al., 2015), and specifically *degenerate* saddle points (Dauphin et al., 2014; Sankar & Balasubramanian, 2017). That these saddles determine gradient dynamics for small initializations has been shown for both deep linear (Saxe et al., 2013; Arora et al., 2019; Chou et al., 2024; Jacot et al., 2021; Pesme & Flammarion, 2023) and neural networks (Zhang et al., 2026a). Guarantees for escaping saddle points in polynomial time apply only to *strict* saddles, i.e., those which have a strictly negative curvature direction (Ge et al., 2015; Jin et al., 2017; Daneshmand et al., 2018). Anandkumar & Ge (2016) show that without strictness, finding a fourth-order local minimum[1] can be NP-hard. We are able to explicitly link a hard-to-learn target, the parity function with n inputs, to saddles of order at least $n/2$ (Proposition 5.3). We conjecture that a similar mechanism may occur for hard targets more generally.

3 Tree Tensor Networks

The theoretical study of deep learning dynamics is most commonly conducted using simple *toy models* retaining some essential feature(s) of the problem while discarding extraneous details. Among the most popular of these is the *deep linear network (DLN)*, defined for a depth L as a composition of L linear maps $W_i : \mathbb{R}^{n_{i-1}} \to \mathbb{R}^{n_i}$:

$$T = W_L W_{L-1} \cdots W_1. \tag{DLN}$$

Naturally, a DLN is itself a linear map. However, as a product of weight matrices $W_i \in \mathbb{R}^{n_i \times n_{i-1}}$, a DLN is nonlinear and nonconvex in all of its weights, and any loss function involving a DLN exhibits the same multiplicative interactions between layers that are characteristic of deep neural networks. Consequently, DLNs preserve the nonlinear dynamics characteristic of deep learning, while enjoying an easily interpretable, and theoretically tractable, relationship over its inputs. This isolation has been productive. It is known that every local minimum is a global minimum, and rank-deficient critical points are saddle points (Kawaguchi, 2016; Yun et al., 2017). The gradient flow admits exact solutions (Saxe et al., 2013), and its dynamics decompose into the independent evolution of singular modes, exhibiting stagewise learning of components from largest to smallest. Implicit regularization toward low-rank solutions has been characterized (Gunasekar et al., 2017; Arora et al., 2019; Li et al., 2020), and much of these phenomena appear to occur analogously in more standard nonlinear networks (Kalimeris et al., 2019; Huh et al., 2023; Zhang et al., 2026a).

Despite their success, DLNs exhibit fundamental properties that limit their mimicry of deep learning dynamics. The intermediate nonlinearities of neural networks allow them to build complex, hierarchical logic, where earlier computations gradually assemble to make later mechanisms useful. This is central to modern accounts of how deep networks learn (Olah et al., 2020; Abbe et al., 2021a; 2023). In contrast, the linearity of DLNs enforces that each component of the target is learned independently of any other. Because their end-to-end map is linear and their gradient flow is structurally decoupled, DLNs can neither represent this hierarchical logic as a function nor exhibit it during training.

Tree tensor networks (TTNs) extend the DLN strategy to a class of models which support compositional structure while preserving the algebraic features that make DLN analysis tractable. The construction relaxes the DLN in two key ways: (i) each matrix is replaced by a higher-order tensor; and (ii) the chain structure is generalized to a tree. A matrix represents a linear map between two vector spaces, while a tensor with k indices represents a k-linear map, separately linear in each input and *nonlinear* jointly. The product $(x, y) \mapsto xy$ is the simplest example: linear in x for fixed y, linear in y for fixed x, but *not linear* as a function of the pair. Furthermore, a matrix represents a map

[1] A critical point of a function f is a *p-th order local minimum* when $f(y) - f(x) \geq -C\|x - y\|^{p+1}$ near x.

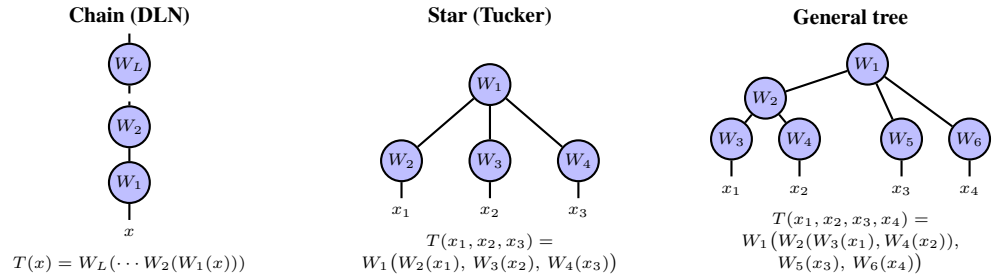


Figure 2: Examples of tree tensor networks, with every node a trainable weight tensor W_v. A path graph (left) recovers the deep linear network. A star graph (center) recovers the Tucker decomposition: leaf maps W_2, W_3, W_4 act on inputs and feed their outputs into a multilinear core W_1. General trees (right) are hierarchical, combining the outputs of disjoint subtrees through a hierarchy of internal multilinear maps. The formula under each panel gives the explicit nested expression for that topology.

with one input and one output, so composing matrices forces a chain topology, where each matrix takes its predecessor's output as its input. Multilinear maps have several inputs, and can therefore receive them from several distinct sources at once. This enables branching, with trees comprising the simplest branching topology. For examples of tree tensor networks, including deep linear networks and Tucker decomposition, see Figure 2.

Definition 3.1 (Tree tensor network)**.** A *tree tensor network* consists of:

- finite sets V of *nodes* and E of *edges*, where each $e \in E$ has one or two endpoints in V, such that the subnetwork of two-ended edges is a directed tree on V, with edges oriented towards the root;
- a dimension $d_e \in \mathbb{Z}_{>0}$ for each edge $e \in E$; and
- a trainable multilinear map $W_v : \prod_{e \in \mathrm{in}(v)} \mathbb{R}^{d_e} \longrightarrow \mathbb{R}^{d_{\mathrm{out}(v)}}$ at each node $v \in V$, where $\mathrm{in}(v)$ is the set of edges oriented into v, and $\mathrm{out}(v)$ is the unique outgoing edge from v to its parent (with the convention $d_{\mathrm{out}(v)} = 1$ if v is the root).

Edges with two endpoints are *internal* (or *bonds*, with *bond dimension* $r_e := d_e$) and edges with one endpoint *external*. To parameterize the model, we may choose bases to identify each W_v with a real tensor, as multilinear maps form a tensor product space. We denote $\theta = \{W_v\}_{v \in V}$ for the collection of all node maps and Θ for the parameter space. Given vectors on the external edges, the *represented tensor* $T(\theta)$ is a multilinear map obtained by function composition: composing the maps W_v from the leaves to the root according to the directed graph structure. By multilinearity, in coordinates this composition is given by tensor contraction, and it is independent of the choice of root (Appendix A).

We study learning an unknown target tensor T^* from random samples of its entries.[2] That is, let $\mathcal{X} = [d_1] \times \cdots \times [d_n]$ index the entries of T^*, and sample inputs $x \in \mathcal{X}$ uniformly from $\mathcal{X}$, with labels $y \in \mathbb{R}$ generated via evaluating the tensor contraction $y = T^*(x)$. Define the per-sample loss:

$$\ell(x, y; \theta) = \frac{1}{2}(y - T(\theta)(x))^2,$$

Our results concern the population average loss[3]

$$\mathcal{L}(\theta) = \mathbb{E}_{x,y}[\ell(x, y; \theta)] = \frac{1}{2|\mathcal{X}|}\|T(\theta) - T^*\|_F^2. \tag{1}$$

The main loss-landscape results assume that the target is *realizable* by the TTN architecture, that is, there exists at least one parameter point $\theta^* \in \Theta$ such that $T(\theta^*) = T^*$ (equivalently, T^* lies in the image of the TTN parameterization). For TTNs, this condition can also be characterized directly as a type of rank condition on T^* (see Appendix B).

[2]Note that our choice here is not unique: we could choose a different empirical learning problem with the same population loss. We could instead choose to consider some external modes as outputs rather than all being inputs, add label noise, sample inputs from a different distribution, etc; our choice here is purely for concreteness.

[3]See Appendix I for discussion on what these results imply for empirical learning with SGD.

Tree tensor networks can perform computation. Unlike deep linear networks, TTNs are *not* functionally linear, and in fact can perform basic computation. Consider any read-once Boolean formula F: a tree of s unary and binary gates in which each input variable appears at most once. We will implement F within a TTN with s nodes and bond dimension 2 (Figure 3). First, encode the Boolean values 0 and 1 by the standard basis vectors $e_0 = (1,0)^\top$ and $e_1 = (0,1)^\top$ in $\mathbb{R}^2$. Associate a multilinear node map W_g to each gate g. For a binary gate g, define its bilinear map W_g by specifying its action on basis vectors:

$$W_g(e_a, e_b) = e_{g(a,b)}, \qquad \text{equivalently} \qquad (W_g)_{a,b,c} = \mathbf{1}\{c = g(a,b)\}, \quad a,b,c \in \{0,1\}.$$

For example, $W_{\mathrm{AND}}(e_1, e_1) = e_1$, while the other three basis input pairs yield e_0. An analogous procedure applies to unary gates; for instance, NOT is represented by the linear map $W_{\mathrm{NOT}} e_a = e_{1-a}$.

Then, connect the gates together: whenever the output of a gate h is an argument of g, connect the output of W_h to the corresponding input of W_g by a dimension-2 bond; evaluating the network substitutes W_h's output vector into that argument of W_g. An argument which is a formula variable x_i instead becomes an external input edge receiving e_{x_i}. One may verify that the overall output of the network on k inputs $e_{x_1}, \ldots e_{x_k}$ is then given by $e_{F(x_1,\ldots,x_k)}$.[4] For example, for the formula in Figure 3, the resulting composition is $W_{\mathrm{OR}}\big(W_{\mathrm{AND}}(e_{x_1}, e_{x_2}), W_{\mathrm{AND}}(e_{x_3}, e_{x_4})\big) = e_{(x_1 \wedge x_2) \vee (x_3 \wedge x_4)}$.

Tree tensor networks are worst-case hard to learn. Perhaps surprisingly, the mere fact that TTNs can perform computation directly implies learning difficulty for gradient-based methods; and for the exact same reason as it does in real neural networks. Following the standard modeling of gradient descent as a statistical query algorithm (cf. Shamir, 2018; Goel et al., 2020), we say that a concept class is *learnable by gradient descent in polynomially many steps* if it is efficiently learnable from statistical queries in the sense of Kearns (1998), with each query restricted to an expected gradient (or value) of a bounded per-sample loss of a polynomial-size differentiable model (for example, the gradient of our loss function $\mathcal{L}$).[5] Merely expressing parity functions, for instance, implies such hardness, as per Theorem 3.2.

Theorem 3.2 (Theorem 5, Kearns (1998))**.** *Let $\mathcal{F}_n$ denote the class of all parity functions over n Boolean variables (where each concept is the parity of some unknown subset of the Boolean variables $x_1, \ldots x_n$), and let $\mathcal{F} = \bigcup_{n \geq 1} \mathcal{F}_n$. Then $\mathcal{F}$ is not efficiently learnable from statistical queries.*

A fortiori, the class $\mathcal{F}$ is not learnable by gradient descent in polynomially many steps. Note that this hardness is inherited by any superclass which realizes parity families. Standard neural networks realize parity at linear size (Shalev-Shwartz et al., 2017), implying Corollary 3.3. The same logic applies to tree tensor networks, which can realize every parity at bond dimension 2 (as they are read-once Boolean formulae), implying Corollary 3.4.

Corollary 3.3. *ReLU networks of width $O(n)$ are not learnable by gradient descent in polynomially many steps.*

Corollary 3.4. *Polynomial-size TTNs with bond dimensions ≥ 2 are not learnable by gradient descent in polynomially many steps.*

These are *concept class* learnability statements, as they imply that a worst-case target always exists for gradient descent, even if potentially most targets can be learned efficiently. Note that the hardness of both neural networks and TTNs follows from the same underlying mechanism: any model class rich enough to realize Boolean formulae automatically contains such hard targets. In particular, this is a property of the hypothesis class alone and does not depend on the choice of gradient method.

[4] Strictly speaking, for simplicity, Definition 3.1 is formulated in such a way that a TTN must have a scalar output. If one wishes to avoid modifying Definition 3.1, this can be achieved by setting the root node to be $W_g(e_a, e_b, e_c) = e_{g(a,b)} \cdot e_c$, where the "output" e_c is instead treated mathematically as an input to the TTN.

[5] Note that for ease of argument, this definition includes full-batch gradient descent, but not stochastic gradient methods. Nevertheless problems which are hard for SQ learners are also provably hard for methods like SGD at moderate batch size (Abbe et al., 2021b).

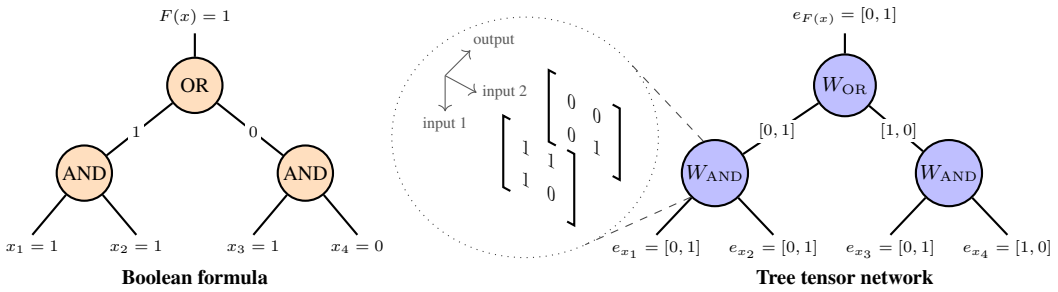


Figure 3: Embedding a Boolean formula into a tree tensor network. **Left:** the syntax tree of $F = (x_1 \wedge x_2) \vee (x_3 \wedge x_4)$, evaluated on the assignment $(x_1, x_2, x_3, x_4) = (1, 1, 1, 0)$. Inputs are external legs at the leaves; each internal edge is labelled with the output of the gate below it. **Right:** the corresponding TTN, obtained by replacing each gate with its $\{0, 1\}$-indicator tensor and each Boolean value with its one-hot encoding ($0 \mapsto e_0 = [1, 0]$, $1 \mapsto e_1 = [0, 1]$). Each internal bond carries the encoded value of the corresponding subformula. **Callout:** the entries of W_{AND}.

4 The Loss Landscape is Benign

Corollary 3.4 implies that there exist realizable TTN targets which cannot be learned in polynomial time by gradient descent on $\mathcal{L}$, but it does not specify a concrete mechanism by which gradient descent fails on such targets (and similarly, the absence of this mechanism on other realizable targets). It is clear that efficient learning must fail in *some* way, but we do not know in *what* way. The TTN loss function is non-convex, and we know that optimization can easily become trapped in local minima for such functions (Jain & Kar, 2017). Historically, this failure mode has been a central focus of the neural network optimization literature (Auer et al., 1996; Choromanska et al., 2015; Safran & Shamir, 2018). Thus, we may naturally suspect that optimization fails for worst-case targets by getting stuck in underperforming local minima.

We show that this explanation *cannot be complete*, at least in many cases. For *any* realizable TTN target, including worst-case targets, every local minimum is global (Theorem 4.3) if one assumes a regularization condition, or at least there always exists a non-increasing-loss escape path to lower loss (Corollary 4.4) if one does not assume the condition. Thus, even in the presence of worst-case hardness, the loss landscape can be benign.

Definition 4.1 (Minimum-norm point)**.** A point in parameter space θ is *minimum norm* if it has minimum L^2 norm among all parameters producing the same output tensor. Explicitly, $\|\theta\| \leq \|\theta'\|$ for all θ' with $T(\theta') = T(\theta)$, where $\|\theta\|^2 = \sum_{v \in V} \|W_v\|_F^2$.

A minimum-norm point is the most parameter-efficient representation of a given output tensor. Intuitively, it ensures that invisible "useless" structure cannot exist, so there is always sufficient internal capacity in the network to continue learning. We believe this condition to be a natural one. For example, Appendix F studies it in detail, showing that it coincides with the *Kempf–Ness condition* of geometric invariant theory (Kempf & Ness, 1979) for the gauge group of the network. It is also equivalent to a *balancedness* condition which generalizes the well-known balancedness condition in deep linear networks (Arora et al., 2018b), and reproduces known canonical forms of tree tensor networks (Acuaviva et al., 2023). We believe this condition to be a significant technical contribution of our work. Importantly, the minimum-norm assumption does not appear unduly restrictive for the purposes of discussing the difficulty of gradient-based optimization due to Proposition 4.2.

Proposition 4.2. *Let $\theta(t)$ be a solution to the gradient flow equation $\dot{\theta}(t) = -\nabla\mathcal{L}(\theta(t))$. If $\theta(0)$ is a minimum-norm point, then $\theta(t)$ is a minimum-norm point for all t, and so is every limit point of $\theta(t)$.*

See Appendix F for the proof[6]. Thus, if the network is initialized to be minimum-norm (such as if it is initialized near the origin), it will remain so under gradient flow. In fact, merely *locally minimizing* the norm over parameters producing the same $T(\theta)$ is equivalent to the minimum-norm condition (Proposition F.10), and L^2*-regularized* gradient flow converges to minimum-norm points

[6]Note the setup in Appendix F allows a strictly more general loss $\mathcal{L}$ than the square loss we consider in (1).

at an exponential rate (Corollary F.13). Under the minimum-norm assumption, in Theorem 4.3, we prove the absence of spurious local minima.

Theorem 4.3. *Every local minimum of $\mathcal{L}$ that is a minimum-norm point is a global minimum with $\mathcal{L} = 0$.*

See Appendix D for proof. As for points without the minimum-norm assumption, in Corollary 4.4, we establish the absence of bad valleys at every point in parameter space.

Corollary 4.4. *From any θ_0 with $T(\theta_0) \neq T^*$, there exists a continuous path $\gamma : [0, 1] \to \Theta$ with $\gamma(0) = \theta_0$ such that $t \mapsto \mathcal{L}(\gamma(t))$ is non-increasing and $\mathcal{L}(\gamma(1)) < \mathcal{L}(\gamma(0))$.*

We prove Corollary 4.4 in Appendix E. Crucially, both Theorem 4.3 and Corollary 4.4 apply for *all* realizable targets, despite Corollary 3.4.

It is worth remarking on the assumptions and setup we use in proving Theorem 4.3, namely minimum-norm parameters, realizability, square loss, and acyclicity of the network topology. Our core claim, that benign landscapes can coexist with worst-case computational hardness, is an existence claim, for which these assumptions are sufficient. However, to identify which parts of our argument are critical for future generalizations, Appendix D.7 demonstrates that arbitrary relaxations of these conditions permit spurious local minima. We suspect, however, that several of these constraints can be significantly weakened without compromising the underlying landscape geometry.

5 Where Can Hardness Arise in TTNs?

Our results in Section 4 suggest that local minima, and the resulting failure of convergence, is not the mechanism behind worst-case optimization difficulty in TTNs. We propose instead that even if the loss landscape is free from spurious local minima, it is not free from optimization difficulty, which can still arise from highly degenerate (very flat) *saddle points* (Dauphin et al., 2014; Sankar & Balasubramanian, 2017; Zhang et al., 2026a). Rather than causing outright failure of convergence, such saddle points can instead cause convergence to be *exponentially slow* (Du et al., 2017). We show that in TTNs, saddle points can occur only where the network is rank-deficient - that is, where it uses only a fraction of its capacity - since they are confined to the Tucker-rank-deficient subset of parameter space. Finally, we provide a case study of the parity function to concretely demonstrate this degenerate saddle phenomenon in a known hard-to-learn target.

5.1 Rank Deficiencies in Spurious Critical Points

Famously, there are several inequivalent notions of "rank" for tensors, much unlike the situation for matrices, where these notions coincide (Lim, 2021). For our purposes, the relevant notion of tensor rank is the *Tucker rank* (also called *multilinear rank*).

Definition 5.1 (Full Tucker rank)**.** A parameter point θ has *full Tucker rank* if, for every internal edge e and each of its endpoints v, the mode-e unfolding $\mathrm{mat}_e(W_v)$ has rank equal to the bond dimension r_e.

The following Theorem 5.2 shows that spurious critical points can only occur under rank deficiency.

Theorem 5.2. *Any critical point of $\mathcal{L}$ which is not a global minimum is not full Tucker rank.*

See Appendix C for the proof. In particular, any saddle points must be confined to rank-deficient points. Intuitively, such rank-deficient points represent points of "partial progress" where the model's effective capacity is lower than its actual capacity, and could be realized by a TTN with lower bond dimensions. This generalizes the phenomenon observed in DLNs where saddle points occur when the model has learned some fraction of the singular values of the target (Saxe et al., 2013).

5.2 The Parity Example

Parity functions are a canonical example of a hard-to-learn target: the class of parities is not learnable by gradient descent in polynomially many steps (Theorem 3.2), and parities are frequently analyzed as

a difficult limiting scenario for training dynamics (Malach & Shalev-Shwartz, 2019; Barak et al., 2022; Abbe et al., 2023). Concretely, consider the parity function on all n input bits, $x \mapsto x_1 \oplus \cdots \oplus x_n$. It is a read-once Boolean formula, and thus is realizable as a TTN of bond dimension 2, by the construction of Section 3.[7] To see this, place the XOR tensor $W \in (\mathbb{R}^2)^{\otimes 3}$ given by

$$W_{0,0,0} = W_{0,1,1} = W_{1,0,1} = W_{1,1,0} = 1, \qquad W_{0,0,1} = W_{0,1,0} = W_{1,0,0} = W_{1,1,1} = 0,$$

at every node of a balanced binary tree with $n/2$ leaves and one root, for a total of $|V| = n - 1$ nodes. The contraction at input $(x_1, \ldots, x_n)$ produces the one-hot encoding of $x_1 \oplus \cdots \oplus x_n$ at the output mode, so the full output tensor is

$$T^*_{x_1,\ldots,x_n,y} \;=\; \mathbf{1}[\, y = x_1 \oplus \cdots \oplus x_n \,] \;\in\; (\mathbb{R}^2)^{\otimes(n+1)}.$$

Theorem 4.3 therefore applies, asserting that no (minimum-norm) spurious local minima obstruct learning of parity. However, this does not imply the learning problem here is without difficulty.

Consider a parameter θ_0 obtained by restricting every bond in the TTN to dimension 1 and minimizing the squared loss within this rank-1 stratum (a best rank-1 approximation). For concreteness, we take the specific θ_0 at which every node tensor W has uniform entries

$$W_{a,b,c} \;=\; \tfrac{1}{2} \quad \text{for all } a, b, c \in \{0, 1\}.$$

This represents a state of having "halfway" learned the parity target. Nevertheless, even with partial learning progress, further learning from this point may be quite difficult.

Proposition 5.3. *The point θ_0 is a critical point of $\mathcal{L}$, and a saddle of order at least $n/2$. In particular, there exist constants $C, \rho > 0$ such that*

$$\mathcal{L}(\theta) - \mathcal{L}(\theta_0) \;\geq\; -C\,\|\theta - \theta_0\|^{n/2+1} \qquad \textit{for all } \theta \in \Theta \textit{ such that } \|\theta - \theta_0\| \leq \rho.$$

Corollary 4.4 shows that a descent path from θ_0 exists, but Proposition 5.3 shows that any such path has order at least $n/2$. Consequently, the saddle point is highly "flat", with flatness growing in input size. The proof of Proposition 5.3 is in Appendix H.

6 Conclusion

In this work, we propose tree tensor networks as a toy model family that can host more nontrivial learning behavior than existing toy models while remaining analytically tractable. To this end, we show that despite hosting worst-case targets that cannot be learned by gradient descent in polynomial time (Section 3), tree tensor networks (conditionally) lack suboptimal local minima (Section 4). We suggest that what makes worst-case targets hard to learn may be that they have very degenerate saddle points in their loss landscape (Section 5).

Tractable toy models have repeatedly shaped our understanding of deep learning, and the development of better toy models remains one of the most important open problems in the field (Simon et al., 2026). Deep linear networks alone have helped to clarify the roles of depth, initialization, and implicit regularization (Saxe et al., 2013; Kawaguchi, 2016; Arora et al., 2019). We believe tree tensor networks can play this role for questions existing toy models have not been able to address.

Empirical results suggest that many capabilities in deep neural networks are performed by mechanistic structures (Olah et al., 2020; Olsson et al., 2022; Nanda et al., 2023), but our understanding of how and why such structures emerge over the course of training is limited. A toy model can only address this question if it can represent such structures in the first place. Deep linear networks, which compute only linear maps, cannot, while tree tensor networks can, since they are able to represent nonlinear computation (read-once Boolean formulas) while remaining analytically tractable.

Our analysis of the parity function (Section 5.2), for example, provides a proof-of-concept for how computational structure can influence loss landscape geometry in tree tensor networks. We hope that future work can study this interplay more systematically, and expect the mathematical tools developed here (e.g. Appendix F) to be useful in that effort. For instance, Saxe et al. (2013) provided closed-form solutions to learning dynamics in deep linear networks, and an analogous result in tree tensor networks would allow us to characterize exactly how, why, and when computational structure emerges in such networks.

[7] We take n to be a power of two, and $n \geq 4$; the construction may be extended by adjusting the tree.

AI USE STATEMENT

In this work, we used generative AI tools to help develop theoretical models, formulate mathematical claims, provide ingredients for proving mathematical claims, assist in the writing of proofs, and refine hypotheses. We have not used generative AI tools for assistance with translation or interpretation of results. Generating synthetic data sets, designing or providing feedback on research methodology or experiments, implementing methods, cleaning and reformatting datasets, and supporting qualitative and thematic data analysis, are not applicable to this work. Additionally, we used generative AI tools to create scientific figures, draft parts of the paper, and identify related work.

We have reviewed all AI-assisted work. For any mathematical proof steps produced by AI, such proof was checked by the authors and edited significantly; further, we verified our main results using the LEAN proof assistant (see Appendix J for details). AI tools assisted in finding counterexamples (e.g. Appendix G) and suggesting ways to modify mathematical claims, but did not come up with their own mathematical claims not suggested by the authors. All AI-generated text was edited for clarity of exposition; in areas where clarity is most critical, only human-written text was used. Any related work suggested by AI was checked for relevance by the authors, and citations were not generated by AI. For AI assistance in editing, we used version control tools to manually review proposed changes. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

This is a theoretical paper. Every result stated in the main text is proved in full in the appendices; the setup and notation used throughout the proofs are collected in Appendix A and Appendix B. The two central results, Theorems 4.3 and 5.2, are additionally machine-checked in the LEAN 4 proof assistant. More information on this formalization process is available in Appendix J, and the formalization itself is available at `https://anonymous.4open.science/r/ttn-loss-landscape-EC57`.

REFERENCES

Emmanuel Abbe, Enric Boix-Adserà, Matthew S. Brennan, Guy Bresler, and Dheeraj Nagaraj. The staircase property: How hierarchical structure can guide deep learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 26989–27002, 2021a.

Emmanuel Abbe, Pritish Kamath, Eran Malach, Colin Sandon, and Nathan Srebro. On the power of differentiable learning versus PAC and SQ learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24340–24351. Curran Associates, Inc., 2021b.

Emmanuel Abbe, Enric Boix-Adserà, and Theodor Misiakiewicz. The merged-staircase property: a necessary and nearly sufficient condition for SGD learning of sparse functions on two-layer neural networks. In *Conference on Learning Theory*, pp. 4782–4887. PMLR, 2022.

Emmanuel Abbe, Enric Boix-Adserà, and Theodor Misiakiewicz. SGD learning on neural networks: leap complexity and saddle-to-saddle dynamics. In *Conference on Learning Theory*, pp. 2552–2623. PMLR, 2023.

El Mehdi Achour, François Malgouyres, and Sébastien Gerchinovitz. The loss landscape of deep linear neural networks: a second-order analysis. *Journal of Machine Learning Research*, 25(242): 1–76, 2024.

Miguel Acosta. On GIT quotients and real forms. *arXiv preprint arXiv:2005.10512*, 2020.

Arturo Acuaviva, Visu Makam, Harold Nieuwboer, David Pérez-García, Friedrich Sittner, Michael Walter, and Freek Witteveen. The minimal canonical form of a tensor network. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 328–362. IEEE, 2023.

Andrei Aleksandrov and Kim Völlinger. Formalizing Piecewise Affine Activation Functions of Neural Networks in Coq, January 2023.

Animashree Anandkumar and Rong Ge. Efficient approaches for escaping higher order saddle points in non-convex optimization. In *Conference on Learning Theory*, pp. 81–102. PMLR, 2016.

Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. *arXiv preprint arXiv:1810.02281*, 2018a.

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International Conference on Machine Learning*, pp. 244–253. PMLR, 2018b.

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

Peter Auer, Mark Herbster, and Manfred K. Warmuth. Exponentially many local minima for single neurons. In *Advances in Neural Information Processing Systems*, volume 8, pp. 316–322. MIT Press, 1996.

Bubacarr Bah, Holger Rauhut, Ulrich Terstiege, and Michael Westdickenberg. Learning deep linear neural networks: Riemannian gradient flows and convergence to global minimizers. *Information and Inference: A Journal of the IMA*, 11(1):307–353, 2022.

Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, 1989.

Boaz Barak, Benjamin Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: SGD learns parities near the computational limit. *Advances in Neural Information Processing Systems*, 35:21750–21764, 2022.

Alexander Bentkamp, Jasmin Christian Blanchette, and Dietrich Klakow. A Formal Proof of the Expressiveness of Deep Learning. *Journal of Automated Reasoning*, 63(2):347–368, August 2019. ISSN 1573-0670. doi: 10.1007/s10817-018-9481-5.

Nils Berglund and Barbara Gentz. The Eyring-Kramers law for potentials with nonquadratic saddles. *arXiv preprint arXiv:0807.1681*, 2008.

Jacob D Biamonte, Stephen R Clark, and Dieter Jaksch. Categorical tensor network states. *AIP Advances*, 1(4), 2011.

Jacek Bochnak, Michel Coste, and Marie-Françoise Roy. *Real algebraic geometry*, volume 36. Springer Science & Business Media, 2013.

Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

Ziang Chen, Yingzhou Li, and Jianfeng Lu. Tensor ring decomposition: optimization landscape and one-loop convergence of alternating least squares. *SIAM Journal on Matrix Analysis and Applications*, 41(3):1416–1442, 2020.

Song Cheng, Lei Wang, Tao Xiang, and Pan Zhang. Tree tensor networks for generative modeling. *Physical Review B*, 99(15):155131, 2019.

Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 192–204. PMLR, 2015.

Hung-Hsu Chou, Carsten Gieshoff, Johannes Maly, and Holger Rauhut. Gradient descent for deep matrix factorization: Dynamics and implicit bias towards low rank. *Applied and Computational Harmonic Analysis*, 68:101595, 2024.

Constantin Christof and Julia Kowalczyk. On the omnipresence of spurious local minima in certain neural network training problems. *Constructive Approximation*, 60(2):197–224, 2024.

Matteo Cipollina, Michail Karatarakis, and Freek Wiedijk. Formalized Hopfield Networks and Boltzmann Machines, December 2025.

J Ignacio Cirac, David Pérez-García, Norbert Schuch, and Frank Verstraete. Matrix product states and projected entangled pair states: Concepts, symmetries, theorems. *Reviews of Modern Physics*, 93(4):045003, 2021.

Hadi Daneshmand, Jonas Kohler, Aurelien Lucchi, and Thomas Hofmann. Escaping saddles with stochastic gradients. In *International Conference on Machine Learning*, pp. 1155–1164. PMLR, 2018.

Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *Advances in Neural Information Processing Systems*, 27, 2014.

Leonardo de Moura and Sebastian Ullrich. The Lean 4 Theorem Prover and Programming Language. In *Automated Deduction – CADE 28: 28th International Conference on Automated Deduction, Virtual Event, July 12–15, 2021, Proceedings*, pp. 625–635, Berlin, Heidelberg, July 2021. Springer-Verlag. ISBN 978-3-030-79875-8. doi: 10.1007/978-3-030-79876-5_37.

Tian Ding, Dawei Li, and Ruoyu Sun. Spurious local minima exist for almost all over-parameterized neural networks. *Optimization Online*, 2019.

Thomas Dooms, Ward Gauderis, Geraint A Wiggins, and José Oramas. Compositionality unlocks deep interpretable models. *arXiv preprint arXiv:2504.02667*, 2025.

Simon S Du, Chi Jin, Jason D Lee, Michael I Jordan, Aarti Singh, and Barnabas Poczos. Gradient descent can take exponential time to escape saddle points. *Advances in Neural Information Processing Systems*, 30, 2017.

Simon S Du, Wei Hu, and Jason D Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. *Advances in Neural Information Processing Systems*, 31, 2018.

Henry Eyring. The activated complex in chemical reactions. *The Journal of Chemical Physics*, 3(2): 107–115, 1935.

Vitaly Feldman, Cristóbal Guzmán, and Santosh Vempala. Statistical query algorithms for mean vector estimation and stochastic convex optimization. *Mathematics of Operations Research*, 46(3): 912–945, 2021.

Abraham Frandsen and Rong Ge. Optimization landscape of Tucker decomposition. *Mathematical Programming*, 193(2):687–712, 2022.

Mark Iosifovich Freidlin and Alexander D Wentzell. Random perturbations. In *Random perturbations of dynamical systems*, pp. 15–43. Springer, 1998.

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on Learning Theory*, pp. 797–842. PMLR, 2015.

Surbhi Goel, Aravind Gollakota, Zhihan Jin, Sushrut Karmalkar, and Adam Klivans. Superpolynomial lower bounds for learning one-layer neural networks using gradient descent. In *International Conference on Machine Learning*, pp. 3587–3596. PMLR, 2020.

Lars Grasedyck and Wolfgang Hackbusch. An introduction to hierarchical (H-) rank and TT-rank of tensors with examples. *Computational Methods in Applied Mathematics*, 11(3):291–304, 2011.

Suriya Gunasekar, Blake E Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Implicit regularization in matrix factorization. *Advances in Neural Information Processing Systems*, 30, 2017.

Zhao-Yu Han, Jun Wang, Heng Fan, Lei Wang, and Pan Zhang. Unsupervised generative modeling using matrix product states. *Physical Review X*, 8(3):031012, 2018.

Liam Hodgkinson and Michael Mahoney. Multiplicative noise and heavy tails in stochastic optimization. In *International Conference on Machine Learning*, pp. 4262–4274. PMLR, 2021.

Minyoung Huh, Hossein Mobahi, Richard Zhang, Brian Cheung, Pulkit Agrawal, and Phillip Isola. The low-rank simplicity bias in deep networks. *Transactions on Machine Learning Research*, 2023. https://openreview.net/forum?id=bCiNWDmlY2.

Vasily Ilin. Semi-Autonomous Formalization of the Vlasov-Maxwell-Landau Equilibrium, March 2026.

Arthur Jacot, François Ged, Berfin Şimşek, Clément Hongler, and Franck Gabriel. Saddle-to-saddle dynamics in deep linear networks: Small initialization training, symmetry, and sparsity. *arXiv preprint arXiv:2106.15933*, 2021.

Prateek Jain and Purushottam Kar. Non-convex optimization for machine learning. *Foundations and Trends in Machine Learning*, 10(3–4):142–336, 2017.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *International Conference on Machine Learning*, pp. 1724–1732. PMLR, 2017.

Dimitris Kalimeris, Gal Kaplun, Preetum Nakkiran, Benjamin L. Edelman, Tristan Yang, Boaz Barak, and Haofeng Zhang. SGD on neural networks learns functions of increasing complexity. In *Advances in Neural Information Processing Systems*, volume 32, pp. 3491–3501, 2019.

Kenji Kawaguchi. Deep learning without poor local minima. *Advances in Neural Information Processing Systems*, 29, 2016.

Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6): 983–1006, 1998.

George Kempf and Linda Ness. The length of vectors in representation spaces. In *Algebraic Geometry: Summer Meeting, Copenhagen, August 7–12, 1978*, pp. 233–243. Springer, 1979.

Eleftherios Kofidis and Phillip A Regalia. On the best rank-1 approximation of higher-order super-symmetric tensors. *SIAM Journal on Matrix Analysis and Applications*, 23(3):863–884, 2002.

Hendrik Anthony Kramers. Brownian motion in a field of force and the diffusion model of chemical reactions. *Physica*, 7(4):284–304, 1940.

Daniel Kunin, Javier Sagastuy-Brena, Surya Ganguli, Daniel LK Yamins, and Hidenori Tanaka. Neural mechanics: Symmetry and broken conservation laws in deep learning dynamics. *arXiv preprint arXiv:2012.04728*, 2020.

Lean FRO. Comparator: A trustworthy judge for Lean proofs. https://github.com/leanprover/comparator, 2026. Accessed: 2026-09-07.

Qianxiao Li, Cheng Tai, and Weinan E. Stochastic modified equations and adaptive stochastic gradient algorithms. In *International Conference on Machine Learning*, pp. 2101–2110. PMLR, 2017.

Zhiyuan Li, Yuping Luo, and Kaifeng Lyu. Towards resolving the implicit bias of gradient descent for matrix factorization: Greedy low-rank learning. *arXiv preprint arXiv:2012.09839*, 2020.

Lek-Heng Lim. Tensors in computations. *Acta Numerica*, 30:555–764, 2021.

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

Haihao Lu and Kenji Kawaguchi. Depth creates no bad local minima. *arXiv preprint arXiv:1702.08580*, 2017.

Eran Malach and Shai Shalev-Shwartz. Learning Boolean circuits with neural networks. *arXiv preprint arXiv:1910.11923*, 2019.

Sibylle Marcotte, Rémi Gribonval, and Gabriel Peyré. Abide by the law and follow the flow: Conservation laws for gradient flows. *Advances in Neural Information Processing Systems*, 36: 63210–63221, 2023.

The mathlib Community. The lean mathematical library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*, CPP 2020, pp. 367–381, New York, NY, USA, January 2020. Association for Computing Machinery. ISBN 978-1-4503-7097-4. doi: 10.1145/3372885.3373824.

Dhagash Mehta, Tianran Chen, Tingting Tang, and Jonathan D Hauenstein. The loss surface of deep linear networks viewed through the algebraic geometry lens. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5664–5680, 2021.

Russell Merris. *Multilinear algebra*. CRC Press, 1997.

Lazar Milikic, Simon Guilloud, Khanh Nguyen, and Viktor Kuncak. LeanFlow: A Case Study in Workflow-Driven Lean Autoformalization, June 2026.

Jacob Miller, Guillaume Rabusseau, and John Terilla. Tensor networks for probabilistic sequence modeling. In *International Conference on Artificial Intelligence and Statistics*, pp. 3079–3087. PMLR, 2021.

David Mumford, John Fogarty, and Frances Kirwan. *Geometric invariant theory*, volume 34. Springer Science & Business Media, 1994.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.

Quynh Nguyen. On connected sublevel sets in deep learning. In *International Conference on Machine Learning*, pp. 4790–4799. PMLR, 2019.

Quynh Nguyen, Mahesh Chandra Mukkamala, and Matthias Hein. On the loss landscape of a class of deep neural networks with no bad local valleys. *arXiv preprint arXiv:1809.10749*, 2018.

Thanh Huy Nguyen, Umut Şimşekli, Mert Gürbüzbalaban, and Gaël Richard. First exit time analysis of stochastic gradient descent under heavy-tailed gradient noise. *Advances in Neural Information Processing Systems*, 32, 2019.

Alexander Novikov, Mikhail Trofimov, and Ivan Oseledets. Exponential machines. *arXiv preprint arXiv:1605.03795*, 2016.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom In: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001. `https://distill.pub/2020/circuits/zoom-in`.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Stellan Östlund and Stefan Rommer. Thermodynamic limit of density matrix renormalization. *Physical Review Letters*, 75(19):3537, 1995.

Michael T Pearce, Thomas Dooms, Alice Rigg, José M Oramas, and Lee Sharkey. Bilinear MLPs enable weight-based mechanistic interpretability. *arXiv preprint arXiv:2410.08417*, 2024.

Roger Penrose. Applications of negative dimensional tensors. *Combinatorial mathematics and its applications*, 1(221-244):3, 1971.

Scott Pesme and Nicolas Flammarion. Saddle-to-saddle dynamics in diagonal linear networks. *Advances in Neural Information Processing Systems*, 36:7475–7505, 2023.

Henning Petzka and Cristian Sminchisescu. Non-attracting regions of local minima in deep and wide neural networks. *Journal of Machine Learning Research*, 22(143):1–34, 2021.

Roger Wolcott Richardson and Peter J Slodowy. Minimum vectors for real reductive algebraic groups. *Journal of the London Mathematical Society*, s2-42(3):409–429, 1990.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

Steven Roman. *Advanced linear algebra*. Springer, 2005.

Itay Safran and Ohad Shamir. Spurious local minima are common in two-layer ReLU neural networks. In *International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4433–4441. PMLR, 2018.

Adepu Ravi Sankar and Vineeth N Balasubramanian. Are saddles good enough for deep learning? *arXiv preprint arXiv:1706.02052*, 2017.

Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96–192, 2011.

Shai Shalev-Shwartz, Ohad Shamir, and Shaked Shammah. Failures of gradient-based deep learning. In *International Conference on Machine Learning*, pp. 3067–3075. PMLR, 2017.

Ohad Shamir. Distribution-specific hardness of learning neural networks. *Journal of Machine Learning Research*, 19(32):1–29, 2018.

Yaoyun Shi, Luming Duan, and Guifré Vidal. Classical simulation of quantum many-body systems with a tree tensor network. *Physical Review A*, 74(2):022320, 2006.

Jamie Simon, Daniel Kunin, Alexander Atanasov, Enric Boix-Adserà, Blake Bordelon, Jeremy Cohen, Nikhil Ghosh, Florentin Guth, Arthur Jacot, Mason Kamb, Dhruva Karkada, Eric J. Michaud, Berkan Ottlik, and Joseph Turnbull. There will be a scientific theory of deep learning. *arXiv preprint arXiv:2604.21691*, 2026.

Umut Şimşekli, Levent Sagun, and Mert Gürbüzbalaban. A tail-index analysis of stochastic gradient noise in deep neural networks. In *International Conference on Machine Learning*, pp. 5827–5837. PMLR, 2019.

E Miles Stoudenmire. Learning relevant features of data with multi-scale tensor networks. *Quantum Science and Technology*, 3(3):034003, 2018.

Edwin Stoudenmire and David J Schwab. Supervised learning with tensor networks. *Advances in Neural Information Processing Systems*, 29, 2016.

Grzegorz Swirszcz, Wojciech Marian Czarnecki, and Razvan Pascanu. Local minima in training of neural networks. *arXiv preprint arXiv:1611.06310*, 2016.

Matthew Trager, Kathlén Kohn, and Joan Bruna. Pure and spurious critical points: a geometric study of linear networks. *arXiv preprint arXiv:1910.01671*, 2019.

Luca Venturi, Afonso S Bandeira, and Joan Bruna. Spurious valleys in two-layer neural network optimization landscapes. *arXiv preprint arXiv:1802.06384*, 2018.

Frank Verstraete and J Ignacio Cirac. Renormalization algorithms for quantum-many body systems in two and higher dimensions. *arXiv preprint cond-mat/0407066*, 2004.

Guifré Vidal. Efficient classical simulation of slightly entangled quantum computations. *Physical Review Letters*, 91(14):147902, 2003.

Steven R White. Density matrix formulation for quantum renormalization groups. *Physical Review Letters*, 69(19):2863, 1992.

Yuhuai Wu, Albert Q. Jiang, Wenda Li, Markus N. Rabe, Charles Staats, Mateja Jamnik, and Christian Szegedy. Autoformalization with Large Language Models, May 2022.

Chulhee Yun, Suvrit Sra, and Ali Jadbabaie. Global optimality conditions for deep neural networks. *arXiv preprint arXiv:1707.02444*, 2017.

Yedi Zhang, Andrew M. Saxe, and Peter E. Latham. Saddle-to-saddle dynamics explains a simplicity bias across neural network architectures. In *International Conference on Learning Representations*, 2026a. arXiv:2512.20607.

Yuanhe Zhang, Yuekai Sun, Taiji Suzuki, Jason D. Lee, and Fanghui Liu. LeanMarathon: Toward Reliable AI Co-Mathematicians through Long-Horizon Lean Autoformalization, June 2026b.

Yi Zhou and Yingbin Liang. Critical points of neural networks: Analytical forms and landscape properties. *arXiv preprint arXiv:1710.11205*, 2017.

Appendix

Background and setup.

- **Appendix A** reviews the multilinear algebra used throughout, reconciling the feedforward (composition) view of TTNs from the main text with the undirected (tensor-contraction) view used in the proofs.
- **Appendix B** records basic setup used throughout the proofs: edge matricizations, cut factorizations, gauge freedom, and the characterization of realizability as a rank condition.

Landscape results.

- **Appendix C** proves Theorem 5.2: critical points that are not global minima cannot have full Tucker rank.
- **Appendix D** proves the main theorem, Theorem 4.3: every minimum-norm local minimum is a global minimum.
- **Appendix E** proves Corollary 4.4: even without the minimum-norm assumption, from every non-optimal point there is a continuous non-increasing path that strictly decreases the loss.

The minimum-norm condition.

- **Appendix F** studies the minimum-norm condition itself, using geometric invariant theory: minimum norm is equivalent to the *Kempf-Ness condition* and a *balancedness* condition (Proposition F.10), and it is preserved by gradient flow (Proposition 4.2).
- **Appendix G** shows the assumption is necessary: an explicit three-leaf star (Tucker) TTN with a spurious local minimum (Theorem G.1).

Hardness in practice.

- **Appendix H** proves Proposition 5.3, characterizing the parity saddle point from Section 5.2.
- **Appendix I** discusses why population-loss geometry bears on sample-based learning, and how degenerate saddles can slow stochastic gradient methods.

Formal verification.

- **Appendix J** describes the autoformalization of Theorems 4.3 and 5.2 in LEAN, its verification and correspondence with the manuscript, and the case for formalizing theoretical results about toy models in machine learning.

A BACKGROUND: MULTILINEAR ALGEBRA AND TENSOR NETWORKS

The main text introduces tree tensor networks (TTNs) intuitively as feedforward models: directed trees where each node computes a multilinear map, and the network output is obtained by function composition. However, the geometric and algebraic proofs in Appendix B through Appendix H rely on an undirected, coordinate-based framework where networks are evaluated via *tensor contraction*.

This appendix reviews the basic multilinear algebra foundations that justify transitioning between these two perspectives. Our treatment is necessarily abridged, and we direct the reader towards standard resources such as Merris (1997); Roman (2005); Lim (2021) for proofs and further discussion.

A.1 TENSORS, ARRAYS, AND MULTILINEAR MAPS

In the context of machine learning, we work over finite-dimensional Euclidean spaces $\mathbb{R}^d$ equipped with the standard inner product and standard basis. The standard inner product gives an isomorphism

$$\mathbb{R}^d \cong (\mathbb{R}^d)^*,$$

and the standard basis identifies elements of tensor product spaces with multidimensional arrays. Throughout the paper, we use these standard identifications freely.

Thus, in our setting, we move between three equivalent representations of the same data:

1. **Tensors:** elements of a tensor product space $\mathbb{R}^{d_1} \otimes \cdots \otimes \mathbb{R}^{d_k}$.
2. **Arrays:** multidimensional grids of real numbers $W \in \mathbb{R}^{d_1 \times \cdots \times d_k}$.
3. **Multilinear maps:** functions $\mathbb{R}^{d_1} \times \cdots \times \mathbb{R}^{d_n} \to \mathbb{R}^{d_{n+1}} \otimes \cdots \otimes \mathbb{R}^{d_k}$ linear in each of their n arguments when all other arguments are held fixed.

The equivalence between tensors and arrays is the usual coordinate representation. The equivalence between tensors and multilinear maps uses the Euclidean identifications $\mathbb{R}^{d_i} \cong (\mathbb{R}^{d_i})^*$. Once these identifications are fixed, we may partition the k modes of an array into a set of input modes $\mathcal{I}$ and output modes $\mathcal{O}$.

Proposition A.1. *Fix a partition $\mathcal{I} \sqcup \mathcal{O} = \{1, \ldots, k\}$. Using the standard Euclidean identifications of input spaces with their duals, there is a standard isomorphism between $\bigotimes_{i=1}^{k} \mathbb{R}^{d_i}$ and the space of multilinear maps*

$$W : \prod_{i \in \mathcal{I}} \mathbb{R}^{d_i} \longrightarrow \bigotimes_{j \in \mathcal{O}} \mathbb{R}^{d_j}.$$

This licenses us to switch between viewing W as a static array of parameters and as a computational map. The choice of directionality, meaning which modes are regarded as inputs and which are regarded as outputs, is not an intrinsic property of the tensor W, and can be changed arbitrarily.

Example A.2. An order-3 tensor $W \in \mathbb{R}^{d_1} \otimes \mathbb{R}^{d_2} \otimes \mathbb{R}^{d_3}$ can be interpreted in several equivalent ways:

- A tensor in $\mathbb{R}^{d_1} \otimes \mathbb{R}^{d_2} \otimes \mathbb{R}^{d_3}$ (zero inputs, three outputs)
- A linear map in $\mathbb{R}^{d_1} \to \mathbb{R}^{d_2} \otimes \mathbb{R}^{d_3}$ (one input, two outputs).
- A bilinear map in $\mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \to \mathbb{R}^{d_3}$ (two inputs, one output).
- A trilinear map in $\mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \times \mathbb{R}^{d_3} \to \mathbb{R}$ (three inputs, zero outputs).

A.2 Composition Is Tensor Contraction

When we view tensors as multilinear maps, we compose them via ordinary function composition. When we view them as arrays, we compose them via tensor contraction.

Definition A.3 (Tensor contraction)**.** Let

$$A \in \mathbb{R}^{d_1 \times \cdots \times d_p \times r}, \qquad B \in \mathbb{R}^{r \times e_1 \times \cdots \times e_q}.$$

The *contraction* of A and B along their shared r-dimensional mode is the tensor

$$C = A \times_r B \in \mathbb{R}^{d_1 \times \cdots \times d_p \times e_1 \times \cdots \times e_q}$$

with entries

$$C_{i_1,\ldots,i_p,j_1,\ldots,j_q} = \sum_{\alpha=1}^{r} A_{i_1,\ldots,i_p,\alpha} B_{\alpha,j_1,\ldots,j_q}.$$

More generally, one may contract any chosen mode of A with any chosen mode of B of the same dimension, after reordering the modes.

If A and B are vectors, contraction is the standard inner product, while for matrices A and B, contraction over one mode is standard matrix multiplication. Crucially, contraction is the algebraic mechanism that computes the composition of multilinear maps (Proposition A.4).

Proposition A.4. *Let f and g be multilinear maps such that the output space of f matches one of the input spaces of g. Let A_f and A_g be their corresponding coordinate arrays. Then the multilinear map obtained by composition,*

$$(x, y) \mapsto g(f(x), y),$$

corresponds exactly to the array obtained by contracting A_f and A_g along their shared mode.

For a fixed pattern of internal contractions, the final tensor is independent of the order in which those contractions are performed, up to the ordering of the remaining uncontracted modes. Thus tensor contraction effectively performs function composition, but without requiring a global choice of computational direction.

A.3 Equivalence of TTN Definitions

We can now bridge the directed, composition-based definition of TTNs used in the main text (Definition 3.1) with the undirected, contraction-based definition used in the proofs (Definition A.5). Definitions of this form are more common in the literature, and generalize better to non-tree networks.

Definition A.5 (Tree tensor network, equivalent definition)**.** A *tree tensor network* consists of:

- a finite set V of *nodes* and a finite set E of *edges*, where each $e \in E$ has either one or two endpoints in V, such that the subnetwork of two-ended edges is a tree on V;
- a dimension $d_e \in \mathbb{Z}_{>0}$ for each edge $e \in E$; and
- a tensor $W_v \in \bigotimes_{e \ni v} \mathbb{R}^{d_e}$ at each node $v \in V$, with one mode for each incident edge e.

Edges with two endpoints are *internal* (or *bonds*, with *bond dimension* $r_e := d_e$) and edges with one endpoint *external*. The *represented tensor* $T(\theta)$ is obtained by tensor contraction along all internal edges.

Equivalence. To recover Definition 3.1 from Definition A.5, choose a root node. This choice orients each internal edge toward the root. For each non-root node v, the edge connecting v to its parent is regarded as the output mode, while the remaining incident modes are regarded as input modes. At the root, all incident internal modes are regarded as input modes, and the root tensor produces a scalar.

By Proposition A.1, each node tensor can therefore be interpreted as a multilinear map. By Proposition A.4, composing these multilinear maps from the leaves to the root computes the same represented tensor as contracting the undirected network along its internal edges.

Consequently, the root and orientation used in the main-text intuition are presentation choices. The undirected, contraction-based definition Definition A.5 gives the same tensor network formalism as Definition 3.1, but is often easier to manipulate algebraically. We therefore adopt this undirected perspective for the remainder of the appendix.

Remark A.6. This algebraic flexibility is most clearly seen within graphical notation such as the *Penrose graphical calculus*, or *string diagrams* in the symmetric monoidal category **FdVect**, which formalize the standard visual language for tensor networks (Penrose, 1971; Biamonte et al., 2011). Tensors are drawn as nodes, and modes as edges. Contraction is represented by connecting edges. Once the tensors and their labelled modes are fixed, the resulting contraction depends on which modes are connected, not on an arbitrary choice of data-flow direction. This is why tensor network diagrams are often drawn as undirected graphs.

B Additional Setup

Here we record some facts and definitions around tree tensor networks which will be useful for the proofs: edge matricizations, cut factorizations, gauge freedom, and an equivalent characterization of realizability.

Notation. For each node $v \in V$, we write $n_v := \prod_{e \in E_v^{\mathrm{ext}}} d_e$ for the combined dimension of the external edges incident to v, where E_v^{ext} denotes the set of external edges at v (with $n_v = 1$ when v has no external edges). We denote the *residual* of the loss $\mathcal{L}(\theta) = \frac{1}{2|\mathcal{X}|}\|T(\theta) - T^*\|_F^2$ by $R := T(\theta) - T^*$. Since the normalization constant $\frac{1}{|\mathcal{X}|}$ is fixed and positive, it has no effect on the loss landscape: rescaling $\mathcal{L}$ by a positive constant preserves critical points, local and global minima, saddle orders, and descent directions. Throughout the appendices we therefore work with the un-normalized loss $\frac{1}{2}\|T(\theta) - T^*\|_F^2$, which (abusing notation) we continue to denote by $\mathcal{L}$.

Edge matricization. For any edge $e \in E$, removing e bipartitions the nodes into sets V_e and V_e^c, because the network is a tree. After choosing an ordering of the external modes on each side, this gives the *edge matricization*

$$T^{(e)} \in \mathbb{R}^{m_e \times m_e^c}, \qquad m_e = \prod_{v \in V_e} n_v, \qquad m_e^c = \prod_{v \in V_e^c} n_v.$$

Cut factorization. For a TTN parameter θ, cutting the network at e produces two subtrees, as the network is a tree. Contracting the two resulting subtrees with the bond index left open gives matrices $F_e \in \mathbb{R}^{m_e \times r_e}$, and $H_e \in \mathbb{R}^{m_e^c \times r_e}$, such that $T(\theta)^{(e)} = F_e H_e^\top$.

Gauge freedom. The cut factorization also exposes a basic non-identifiability of the parameterization. If $M \in \mathrm{GL}(r_e)$, then $F_e \mapsto F_e M$, and $H_e \mapsto H_e M^{-\top}$ leaves the represented tensor unchanged, since $(F_e M)(H_e M^{-\top})^\top = F_e H_e^\top$. Equivalently, this operation changes basis on the bond space associated with e and so one may apply M to the bond-e mode on one side of the edge and the inverse transformation on the other side. These transformations are the *gauge freedom* of the TTN parameterization. The *fiber* over an output tensor T_0 is $\mathcal{F}_{T_0} := \{\theta \in \Theta : T(\theta) = T_0\}$. Thus gauge transformations move within a fiber, although a fiber may contain parameter points not related by gauge transformations.

Realizability. Recall that we define a tensor T^* to be *realizable* by a given TTN $T(\theta)$ if there exists at least one parameter $\theta^* \in \Theta$ such that $T(\theta^*) = T^*$ (Section 3). We now provide an alternative characterization of the realizability condition. This characterization is well-known (see e.g. Grasedyck & Hackbusch (2011)) but we prove it here for our specific setup.

Proposition B.1. *Let $r_e^* := \mathrm{rank}(T^{*(e)})$ for each internal edge e. A tensor T^* is realizable by the TTN architecture with bond dimensions $\{r_e\}$ if and only if $r_e^* \le r_e$ for every internal edge e.*

Moreover, whenever these conditions hold, the realization may be chosen to use only r_e^* bond directions at each edge. More precisely, for any collection of r_e^*-dimensional subspaces $S_e \subseteq \mathbb{R}^{r_e}$, there exists a realization $\theta^* \in \Theta$ of T^* such that each endpoint tensor of e is supported on S_e in its e-mode. For this realization, the cut factors satisfy

$$\mathrm{rank}(F_e^*) = \mathrm{rank}(H_e^*) = r_e^*$$

for every internal edge e.

Proof. The forward direction follows from the $T(\theta)^{(e)} = F_e H_e^\top$ factorization above, which gives $\mathrm{rank}(T(\theta)^{(e)}) \le r_e$. For the converse, root the tree at an arbitrary internal node. For each directed edge e, let V_e denote the nodes in the subtree below e, and define $U_e := \mathrm{im}(T^{*(e)}) \subseteq \bigotimes_{v \in V_e} \mathbb{R}^{n_v}$. By assumption, $\dim U_e = \mathrm{rank}(T^{*(e)}) \le r_e$. We claim that these spaces are nested. Precisely, let e have lower endpoint u with child edges $e_1, \dots, e_k$ and n_u external dimensions. Then we claim that $U_e \subseteq \mathbb{R}^{n_u} \otimes U_{e_1} \otimes \cdots \otimes U_{e_k}$. Each element of U_e is T^* contracted against a fixed vector on the modes outside the subtree, hence a tensor over the modes of $V_e = \{u\} \cup \bigcup_j V_{e_j}$. Its fibers along a child edge e_j are themselves partial contractions of T^* across e_j, so they lie in $U_{e_j} = \mathrm{im}(T^{*(e_j)})$. A tensor all of whose mode-e_j fibers lie in U_{e_j} (for every j) lies in $\mathbb{R}^{n_u} \otimes U_{e_1} \otimes \cdots \otimes U_{e_k}$, which is the claim.

Now choose a basis of each U_e. The inclusion above writes each basis vector of U_e as a tensor in $\mathbb{R}^{n_u}$ and the child bases; take these coefficients as the node tensor at u. At the root ρ the same nestedness gives $T^* \in \mathbb{R}^{n_\rho} \otimes U_{e_1} \otimes \cdots \otimes U_{e_k}$; expand T^* in the chosen bases and use the coefficients as the root tensor. Contracting from the leaves upward, the open bond on each edge e ranges over the chosen basis of U_e, so the root expansion reconstructs T^*.

This constructs a realization with bond dimension exactly r_e^* on every internal edge. To place these bond spaces (dimension r_e^*) inside the prescribed ambient bond spaces (dimension r_e), choose for each e an isometry

$$\iota_e : \mathbb{R}^{r_e^*} \hookrightarrow \mathbb{R}^{r_e}$$

with image S_e, and apply ι_e to the e-mode of both endpoint tensors. Since each contraction across e then factors through $\iota_e^\dagger \iota_e = I_{r_e^*}$, the represented tensor remains T^*, while the endpoint tensors are supported on S_e.

Consequently, the corresponding cut factors F_e^* and H_e^* both vanish on $S_e^\perp$, and hence have rank at most r_e^*. On the other hand, the cut factorization

$$T^{*(e)} = F_e^* H_e^{*\top}$$

has rank r_e^*, so both factors must have rank at least r_e^*. Therefore

$$\operatorname{rank}(F_e^*) = \operatorname{rank}(H_e^*) = r_e^*.$$

□

We freely use this equivalent characterization of realizability going forward.

C No Spurious Critical Points at Full Tucker Rank

In this section we prove that full Tucker rank critical points must be global minima; or contrapositively, that critical points which are not global minima must be rank-deficient (Theorem 5.2). The proof proceeds by induction, stripping leaf nodes until the network is reduced to a single tensor, where criticality trivially guarantees global optimality. The induction relies on a property of full-rank networks: at any leaf, the remainder of the tree acts as a full-rank linear map (Lemma C.1). Informally, this non-degeneracy, together with criticality and realizability, constrains the network to match the target at each leaf. We are then able to inductively remove each leaf, transferring the loss, criticality, full-rank, and realizability conditions to a strictly smaller tree.

Lemma C.1. *Under the full Tucker rank assumption, for any edge e and either subtree on one side of e, the corresponding factor matrix (F_e or H_e) has full column rank r_e.*

Proof. The proof follows by induction on the size of the subtree. We prove the result for H_e, and the result for F_e follows identically. First, for the base case, if the subtree is a single leaf v, then $H_e = W_v \in \mathbb{R}^{n_v \times r_e}$, which has rank r_e by full Tucker rank. Now for the inductive step, suppose node u is the root of the subtree, with edge e connecting u to the rest of the tree and edges $e_1, \dots, e_k$ connecting u to its child subtrees (with factors $H_{e_1}, \dots, H_{e_k}$). Then

$$H_e = (I_{n_u} \otimes H_{e_1} \otimes \cdots \otimes H_{e_k}) \operatorname{mat}_e(W_u)$$

where the identity matrix I_{n_u} accounts for u's own external modes. By induction, each H_{e_j} has full column rank r_{e_j}, so the Kronecker product is full column rank. Since $\operatorname{mat}_e(W_u)$ has rank r_e by the full Tucker rank hypothesis, $\operatorname{rank}(H_e) = \operatorname{rank}(\operatorname{mat}_e(W_u)) = r_e$. □

Theorem 5.2. *Any critical point of $\mathcal{L}$ which is not a global minimum is not full Tucker rank.*

Proof. We prove the contrapositive that if $\theta \in \Theta$ is a full-Tucker-rank critical point, it is a global minimum. We proceed by strong induction on $|V|$. The base case of $|V| = 1$ is trivial, since with a single node the model is linear, and thus critical implies global minimality. Now suppose that $|V| \geq 2$ and $\mathcal{T}$ is a tree. Then there exists a leaf $v \in L$ with unique incident edge e connecting v to parent u. By the full Tucker rank hypothesis, the factor tensor W_v at node v has full column rank r_e. Considering its thin QR decomposition $W_v = Q_v S$, with $Q_v^\top Q_v = I_{r_e}$ and $S \in \mathbb{R}^{r_e \times r_e}$ non-singular, we absorb S into the neighboring tensor at node u via $W_u \leftarrow W_u \times_e S$. This reparameterization leaves the represented tensor T, the residual R, and all Tucker ranks invariant. Hence, we assume without loss of generality that $W_v^\top W_v = I_{r_e}$.

Since v is a leaf, the cut factorization at edge e becomes $T^{(e)} = W_v H_e^\top$, where H_e has full column rank r_e by Lemma C.1, so $H_e^\top H_e$ is invertible and positive definite. Stationarity at node v implies that the residual $R^{(e)} = T^{(e)} - T^{*(e)}$ satisfies $R^{(e)} H_e = 0$. Therefore,

$$T^{*(e)} H_e = T^{(e)} H_e = W_v (H_e^\top H_e),$$

which has rank r_e and image $\operatorname{im}(W_v)$. By realizability, $\operatorname{rank}(T^{*(e)}) \leq r_e$, so $\operatorname{im}(T^{*(e)}) = \operatorname{im}(W_v)$. Since both $T^{(e)}$ and $T^{*(e)}$ have columns in $\operatorname{im}(W_v)$, so does $R^{(e)} = T^{(e)} - T^{*(e)}$.

Now define the *reduced residual* $\widetilde{R} := W_v^\top R$ and observe that $R = W_v \widetilde{R}$ since $W_v^\top W_v = I$. Define the reduced tree $\widetilde{\mathcal{T}}$ by removing v and e, where u's former bond mode along e becomes an external mode of dimension r_e. Observe that the residual of the loss on $\widetilde{T}$ becomes $\widetilde{R}$. We verify all hypotheses transfer:

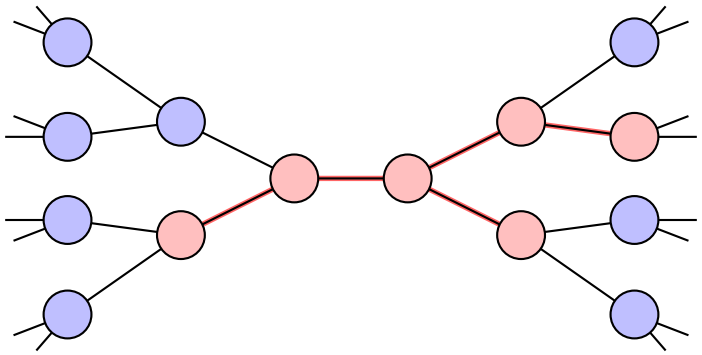


Figure 4: The rank-deficient core is a connected subgraph induced by rank-deficient edges (marked in red). As each vertex in this subgraph (light red) corresponds to a tensor that is perturbed in the escape path, the size of this subgraph equals the exponent of the escape direction in the saddle, so in this case $\sim \|\theta - \theta_0\|^6$. Note that there can be multiple, mutually disconnected rank-deficient cores in the whole tensor network. In that case, each core represents one escape path in the saddle point, and the *smallest* core delivers the least suppressed escape (Remark D.10).

- *Loss:* $\|R\|_F^2 = \|W_v \widetilde{R}\|_F^2 = \|\widetilde{R}\|_F^2$ (by $W_v^\top W_v = I$).
- *Criticality:* Any perturbation $\delta\widetilde{T}$ of the reduced network $\widetilde{T}$ lifts to a perturbation δT of the original network T (where W_v is left fixed), and $\delta T = W_v \delta\widetilde{T}$ by multilinearity. Hence
$$\langle \widetilde{R},\, \delta\widetilde{T}\rangle = \langle W_v^\top R,\, \delta\widetilde{T}\rangle = \langle R,\, W_v\, \delta\widetilde{T}\rangle = \langle R,\, \delta T\rangle,$$
and thus criticality transfers because $\langle \widetilde{R},\, \delta\widetilde{T}\rangle = \langle R,\, \delta T\rangle = 0$.
- *Full Tucker rank:* Unfolding ranks at remaining nodes are unchanged.
- *Realizability:* $\operatorname{rank}(\widetilde{T}^{*(e')}) \le \operatorname{rank}(T^{*(e')}) \le r_{e'}$ for edges e' in $\widetilde{\mathcal{T}}$.

By the inductive hypothesis, $\widetilde{R} = 0$, so $R = W_v \widetilde{R} = 0$. □

Remark C.2. The full-rank assumption on W_v is essential in the stripping argument. If $\operatorname{rank}(W_v) = s < r_e$, then stationarity still implies
$$T^{*(e)} H_e = W_v (H_e^\top H_e),$$
so the columns of $T^{*(e)} H_e$ lie in $\operatorname{im}(W_v)$. But this only tests $T^{*(e)}$ against the s-dimensional image of H_e. It does not force the entire image of $T^{*(e)}$ to lie in $\operatorname{im}(W_v)$. Thus the target unfolding may have components outside the model image at the leaf, and the residual need not factor through W_v.

D No Spurious Local Minima at Minimum-Norm Points

This section proves that every minimum-norm local minimum is globally optimal (Theorem 4.3). The key idea is that any non-optimal critical point must contain a *rank-deficient region*: a connected component of internal edges whose Tucker ranks are strictly smaller than their available bond dimensions. Intuitively, such a region is using only part of the capacity available to it, behaving as though those bonds had smaller dimensions than the architecture permits.

We isolate this connected component and collapse the surrounding full-rank parts of the network into fixed boundary tensors, producing a smaller *rank-deficient core* (Appendix D.4, Figure 4). The total loss decomposes into a constant term plus the loss of this core, so it suffices to study optimization within the core itself. Rank deficiency then creates latent directions in parameter space that are invisible to first order but can be activated through a coordinated perturbation of all tensors in the core. We construct such a perturbation explicitly and show that whenever the residual is nonzero it yields a strictly decreasing descent direction (Appendix D.5). Consequently, any minimum-norm

critical point with positive loss admits an arbitrarily local loss-decreasing perturbation, contradicting local optimality. The only remaining local minima are therefore global minima.

Note that while both Theorem 5.2 and Theorem 4.3 use stripping arguments to reduce the network, the reductions operate differently. In Theorem 5.2, the full-rank assumption forces the model to match the target's column space at every leaf. This alignment allows us to iteratively remove individual leaves while preserving the *exact* loss. Here, rank deficiency breaks this alignment, preventing us from collapsing the network entirely. Instead, we must rely on a weaker form of alignment (Appendix D.2), which allows us to isolate a rank-deficient core while preserving the loss only up to an *additive constant*. However, because we only need to construct a descent path to rule out a local minimum, rather than prove the loss is zero, this partial reduction is sufficient: any loss-decreasing path within the reduced core lifts to a loss-decreasing path in the full network (Theorem 4.3).

D.1 Dormant Subspaces

Definition D.1 (Dormant subspace)**.** For an internal edge $e = \{u, v\}$, the *dormant subspace* $D_e \subseteq \mathbb{R}^{r_e}$ at e is

$$D_e := \ker(\mathrm{mat}_e\, W_u) \cap \ker(\mathrm{mat}_e\, W_v),$$

the subspace along which both endpoint tensors vanish on their e-mode. That is, for every $d \in D_e$, contracting W_u or W_v against d on its e-mode gives the zero tensor.

Proposition D.2. *If θ is minimum norm, then for every internal edge $e = \{u, v\}$, we have*

$$D_e = \ker(\mathrm{mat}_e\, W_u) = \ker(\mathrm{mat}_e\, W_v) = \ker(F_e) = \ker(H_e).$$

Further, the dimension of the dormant subspace is determined purely by the rank of the model. In particular,

$$\dim D_e = r_e - \mathrm{rank}(T^{(e)}).$$

In other words, the minimum-norm property prevents parameter structure that does not contribute to the output: any slice of a node tensor annihilated by the rest of the network must itself be zero. In particular, the four notions of "dormancy" at an edge coincide, and are represented by the single object D_e.

Proof. We prove the kernel equalities by establishing the cycle of inclusions

$$\ker(\mathrm{mat}_e\, W_u) \subseteq \ker(F_e) \subseteq \ker(\mathrm{mat}_e\, W_v) \subseteq \ker(H_e) \subseteq \ker(\mathrm{mat}_e\, W_u).$$

The cycle forces all four subspaces to be equal, and hence also equal to $D_e = \ker(\mathrm{mat}_e\, W_u) \cap \ker(\mathrm{mat}_e\, W_v)$.

Each cut factor F_e, H_e is linear in its adjacent node tensor W_u, W_v. Concretely, F_e is obtained by contracting the remaining tensors on the u-side of the cut into the non-e modes of W_u; this operation is linear and leaves the e-mode untouched, so $F_e = E_u\ \mathrm{mat}_e\, W_u$ for some matrix E_u. Likewise $H_e = E_v\ \mathrm{mat}_e\, W_v$. This gives two of the four inclusions: $\ker(\mathrm{mat}_e\, W_u) \subseteq \ker(F_e)$ and $\ker(\mathrm{mat}_e\, W_v) \subseteq \ker(H_e)$.

The other two inclusions use the minimum-norm hypothesis. Consider $\ker(F_e) \subseteq \ker(\mathrm{mat}_e\, W_v)$, with $\ker(H_e) \subseteq \ker(\mathrm{mat}_e\, W_u)$ following via the same argument. Let $d \in \ker(F_e)$ be a unit vector, and let θ' be obtained from θ by zeroing the d-slice of W_v: that is, by replacing $\mathrm{mat}_e\, W_v$ with $(\mathrm{mat}_e\, W_v)(I - dd^\top)$. Since $H_e = E_v\ \mathrm{mat}_e\, W_v$, this changes H_e into $H_e(I - dd^\top)$, and the output does not change:

$$T(\theta')^{(e)} = F_e(I - dd^\top)H_e^\top = T(\theta)^{(e)} - (F_e d)(H_e d)^\top = T(\theta)^{(e)},$$

where the last equality uses $F_e d = 0$. On the other hand, $\|\theta'\|^2 = \|\theta\|^2 - \|(\mathrm{mat}_e\, W_v)\, d\|^2$. Because θ is minimum norm, $\|\theta'\| \geq \|\theta\|$, so $(\mathrm{mat}_e\, W_v)\, d = 0$; that is, $d \in \ker(\mathrm{mat}_e\, W_v)$.

It remains to compute $\dim D_e$, the dimension of the dormant subspace. In the factorization $T^{(e)} = F_e H_e^\top$, the image of $H_e^\top$ is $\ker(H_e)^\perp = D_e^\perp$. On $D_e^\perp$ the matrix F_e is injective, because $\ker(F_e) = D_e$. Therefore $\mathrm{rank}(T^{(e)}) = \dim(D_e^\perp) = r_e - \dim D_e$. □

Consequently, at a minimum-norm point every internal edge satisfies $\operatorname{rank}(\operatorname{mat}_e W_u) = \operatorname{rank}(\operatorname{mat}_e W_v) = \operatorname{rank}(T^{(e)})$, so e is rank-deficient in the sense of Definition 5.1 if and only if $\operatorname{rank}(T^{(e)}) < r_e$, if and only if $D_e \neq 0$. As every parameter considered in this appendix is minimum norm, we use these three characterizations of rank deficiency (and the corresponding characterizations of full-rank edges) interchangeably in what follows.

D.2 Active Bond Directions are Target-Aligned

We use the minimum-norm and criticality conditions to ensure that no structure in the model is "unused": informally, subtrees in the model must have non-zero alignment with their corresponding subtree in the target. Without such a property, counterexamples like Appendix G are possible.

Fix a root vertex and orient each non-root internal edge away from the root. Fix also an internal edge e with subtree factors F_e, H_e (model) and F_e^*, H_e^* (target), for an arbitrary realization of the target (Appendix B). Recall the dormant subspace $D_e = \ker(F_e) = \ker(H_e)$ from Proposition D.2, and write $A_e = D_e^\perp$ for the *active subspace* and $M_e := H_e^{*\top} H_e \in \mathbb{R}^{r_e \times r_e}$ for the *cross-Gram matrix* at e.

The cross-Gram matrix measures how the model's bond usage at e overlaps with the target's: the u-direction in bond space contributes to $\langle T, T^* \rangle$ only insofar as $M_e u \neq 0$. The following proposition says that at a minimum-norm critical point, every active bond direction has nonzero target alignment.

Proposition D.3. *At a minimum-norm critical point, M_e is injective on A_e.*

Proof. We may assume that $F_e^\top F_e = P_{A_e}$, the orthogonal projector onto A_e. To arrange this, note that the Gram matrix $F_e^\top F_e$ is positive semidefinite, with kernel $\ker F_e = D_e$ (Proposition D.2), so it maps A_e to itself and restricts to a positive-definite operator there. Let $S \in \mathrm{GL}(r_e)$ act as the identity on D_e and as $\big((F_e^\top F_e)|_{A_e}\big)^{-1/2}$ on A_e; then S is symmetric and preserves both D_e and A_e. The gauge transformation $F_e \mapsto F_e S$, $H_e \mapsto H_e S^{-\top} = H_e S^{-1}$ at e (Appendix B) then yields $(F_e S)^\top (F_e S) = S(F_e^\top F_e)S = P_{A_e}$. This gauge leaves T unchanged, and, being an invertible linear change of parameters that preserves the loss, it maps critical points to critical points. It preserves the kernel equalities $\ker(F_e) = \ker(H_e) = D_e$ from Proposition D.2, since $\ker(F_e S) = S^{-1} D_e = D_e$ and $\ker(H_e S^{-1}) = S D_e = D_e$. And it replaces M_e by $M_e S^{-1}$, which is injective on A_e if and only if M_e is, because S^{-1} maps A_e bijectively to itself. The minimum-norm hypothesis itself need not survive the gauge, but the remainder of the proof uses only criticality and the equalities $\ker F_e = \ker H_e = D_e$.

Now suppose toward a contradiction that $M_e u = 0$ for some unit vector $u \in A_e$. Expanding the squared loss,

$$\mathcal{L} = \frac{1}{2}\langle T, T \rangle - \langle T, T^* \rangle + \frac{1}{2}\langle T^*, T^* \rangle. \tag{2}$$

We construct a smooth path $\theta(t)$ with $\theta(0) = \theta$ along which $\langle T, T^* \rangle$ is constant and $\langle T, T \rangle$ strictly decreases at $t = 0$; the loss then strictly decreases at $t = 0$, which is impossible at a critical point.

Let W_v be the endpoint tensor of e on the H_e-side, so that $H_e = E_v \operatorname{mat}_e W_v$ for a fixed matrix E_v, as in the proof of Proposition D.2. Define $\theta(t)$ by scaling down the u-component of W_v: that is, replacing $\operatorname{mat}_e W_v$ with $(\operatorname{mat}_e W_v)(I - t\, uu^\top)$, keeping every other node tensor fixed. Write $T(t) := T(\theta(t))$ and $H_e(t) := H_e(\theta(t))$. Along this path F_e is constant, while H_e changes according to

$$H_e(t) = H_e(I - t\, uu^\top) = H_e - t\,(H_e u)\, u^\top.$$

We first show the target alignment term $\langle T(t), T^* \rangle$ of Equation (2) is constant along the path. Through the matricization of T and T^* along edge e, $\langle T, T^* \rangle = \operatorname{tr}\big(T^{(e)\top} T^{*(e)}\big) = \operatorname{tr}\big(H_e F_e^\top F_e^* H_e^{*\top}\big)$, which is linear in H_e. Substituting $H_e(t)$,

$$\langle T(t), T^* \rangle = \langle T, T^* \rangle - t\, \operatorname{tr}\big((H_e u)\, u^\top F_e^\top F_e^* H_e^{*\top}\big) = \langle T, T^* \rangle - t\, u^\top F_e^\top F_e^* (M_e u) = \langle T, T^* \rangle,$$

where the second equality uses cyclicity of the trace and the definition $M_e = H_e^{*\top} H_e$, and the third uses $M_e u = 0$.

Now we show the self-overlap term $\langle T(t), T(t)\rangle$ of Equation (2) strictly decreases for small $t > 0$. By the same matricization identity and the normalization $F_e^\top F_e = P_{A_e}$,

$$\langle T(t), T(t)\rangle = \operatorname{tr}\big(H_e(t)\, P_{A_e}\, H_e(t)^\top\big) = \|H_e(t)\|_F^2,$$

where the second equality holds because every row of $H_e(t)$ lies in A_e: the rows of H_e span $(\ker H_e)^\perp = A_e$, and $u \in A_e$. Expanding the rank-one update, with $\|u\| = 1$,

$$\|H_e(t)\|_F^2 = \|H_e\|_F^2 - 2t\,\|H_e u\|^2 + t^2\,\|H_e u\|^2.$$

Moreover $H_e u \neq 0$, because u is a nonzero vector of $A_e = (\ker H_e)^\perp$.

Since the target alignment term $\langle T(t), T^*\rangle$ of Equation (2) is constant, the loss along the path is

$$\mathcal{L}(\theta(t)) = \mathcal{L}(\theta) - \Big(t - \frac{t^2}{2}\Big)\|H_e u\|^2,$$

and its derivative at $t = 0$ is $-\|H_e u\|^2 < 0$. But $t \mapsto \theta(t)$ is a smooth path with $\theta(0) = \theta$, so at a critical point this derivative is $\langle \nabla\mathcal{L}(\theta), \dot\theta(0)\rangle = 0$ — a contradiction. □

Corollary D.4. *At a minimum-norm critical point,* $\operatorname{rank}(T^{(e)}) \le r_e^*$ *for every internal edge* e.

Proof. Neither side of the claimed inequality refers to any particular realization of the target T^*; by Proposition B.1, we may choose a realization with $\operatorname{rank}(H_e^*) = r_e^*$ at every internal edge. By Proposition D.2, $\dim(A_e) = r_e - \dim(D_e) = \operatorname{rank}(T^{(e)})$. By Proposition D.3, M_e is injective on A_e, so $\dim(A_e) \le \operatorname{rank}(M_e)$. And $M_e = H_e^{*\top} H_e$ has rank at most $\operatorname{rank}(H_e^*) = r_e^*$. □

Corollary D.5. *At a minimum-norm critical point, if* $\operatorname{rank}(T^{(e)}) = r_e$ *then* M_e *is invertible.*

Proof. If $\operatorname{rank}(T^{(e)}) = r_e$, then $\dim(D_e) = r_e - \operatorname{rank}(T^{(e)}) = 0$ by Proposition D.2, so $A_e = D_e^\perp = \mathbb{R}^{r_e}$. By Proposition D.3, M_e is then injective on all of $\mathbb{R}^{r_e}$, and an injective square matrix is invertible. □

D.3 Reduction to Full Target Rank

Before going further, it is convenient to reduce to the case where the target T^* has full rank along every internal edge. This subsection justifies that reduction. Throughout, θ denotes the minimum-norm critical point under consideration.

The reduced problem. Note that we may assume without loss of generality that the target bond rank $r_e^* > 0$ for every internal edge e (equivalently, the target $T^* \neq 0$), since if this was not the case, Corollary D.4 would imply $\operatorname{rank}(T^{(e)}) \le r_e^* = 0$, and therefore $T(\theta) = 0 = T^*$. For each internal edge e, choose an r_e^*-dimensional subspace $S_e \subseteq \mathbb{R}^{r_e}$ containing A_e; this is possible by Corollary D.4. Apply Proposition B.1 with these subspaces to fix a target realization $\{W_v^*\}$ supported on the S_e directions.

After choosing orthonormal coordinates on each S_e, define a reduced TTN on the same tree with bond dimensions r_e^* by restricting every internal bond space from $\mathbb{R}^{r_e}$ to S_e. Equivalently, each node tensor is replaced by its component supported on the S_e directions along all incident internal edges. Let $\rho : \Theta \to \widetilde\Theta$ denote this orthogonal restriction map, and let $\iota : \widetilde\Theta \to \Theta$ denote the reverse coordinate inclusion, obtained by zero-padding the discarded $S_e^\perp$ slices. Set $\widetilde\theta := \rho(\theta)$, let $\widetilde T^*$ be the target represented by the restricted target tensors, and write $\widetilde{\mathcal{L}}$ for the squared loss of the reduced problem.

Lemma D.6. *The reduced problem satisfies:*

1. $\operatorname{rank}(\widetilde T^{*(e)}) = r_e^*$ *for every internal edge.*
2. *The zero-padding map* $\iota : \widetilde\Theta \to \Theta$ *is a linear isometry with*

$$\iota(\widetilde\theta) = \theta, \qquad T \circ \iota = \widetilde T, \qquad \mathcal{L} \circ \iota = \widetilde{\mathcal{L}}.$$

3. *$\widetilde{\theta}$ is a minimum-norm critical point of $\widetilde{\mathcal{L}}$.*

4. *$\widetilde{T}(\widetilde{\theta}) = \widetilde{T}^*$ if and only if $T(\theta) = T^*$.*

5. *If θ is a local minimum of $\mathcal{L}$, then $\widetilde{\theta}$ is a local minimum of $\widetilde{\mathcal{L}}$.*

Proof. Since $A_e \subseteq S_e$, we have $S_e^\perp \subseteq A_e^\perp = D_e$ by Proposition D.2. Both endpoint tensors of e therefore vanish on $S_e^\perp$. Thus θ is already supported on the chosen subspaces, so

$$\iota(\widetilde{\theta}) = \iota(\rho(\theta)) = \theta.$$

The target realization is supported on the same subspaces by construction, so restricting it does not change the represented external tensor: $\widetilde{T}^* = T^*$. Consequently, the edge matricizations of the reduced target have the same ranks as those of T^*:

$$\operatorname{rank}(\widetilde{T}^{*(e)}) = \operatorname{rank}(T^{*(e)}) = r_e^*.$$

This proves (1).

For any reduced parameter $\widetilde{\phi} \in \widetilde{\Theta}$, zero-padding commutes with TTN contraction, so $T(\iota(\widetilde{\phi})) = \widetilde{T}(\widetilde{\phi})$. Since $\widetilde{T}^* = T^*$, this gives $\mathcal{L}(\iota(\widetilde{\phi})) = \widetilde{\mathcal{L}}(\widetilde{\phi})$. The map ι is a linear isometry because it only inserts zero components in the orthogonal complements $S_e^\perp$. Hence (2) follows.

We now derive the remaining assertions from (2). If $\widetilde{\theta}'$ lies in the reduced fiber over $\widetilde{T}(\widetilde{\theta})$, then $\iota(\widetilde{\theta}')$ lies in the original fiber over $T(\theta)$. Therefore, by minimum-norm of θ and the isometry of ι,

$$\|\widetilde{\theta}\| = \|\theta\| \le \|\iota(\widetilde{\theta}')\| = \|\widetilde{\theta}'\|.$$

Thus $\widetilde{\theta}$ is minimum norm in its reduced fiber. Criticality also transfers: for every reduced tangent direction $\widetilde{\delta}$,

$$\left.\frac{d}{dt}\right|_{t=0} \widetilde{\mathcal{L}}(\widetilde{\theta} + t\widetilde{\delta}) = \left.\frac{d}{dt}\right|_{t=0} \mathcal{L}(\theta + t\,\iota(\widetilde{\delta})),$$

and the right-hand side vanishes because θ is critical. This proves (3).

Since $\iota(\widetilde{\theta}) = \theta$ and $\widetilde{T}^* = T^*$, (2) gives

$$\widetilde{T}(\widetilde{\theta}) - \widetilde{T}^* = T(\theta) - T^*,$$

which proves (4). Finally, if θ is a local minimum of $\mathcal{L}$, then the continuity of ι and the identity $\mathcal{L} \circ \iota = \widetilde{\mathcal{L}}$ imply that no sufficiently small reduced perturbation can decrease $\widetilde{\mathcal{L}}$. Hence $\widetilde{\theta}$ is a local minimum of $\widetilde{\mathcal{L}}$, proving (5). □

Thus, for the remainder of Appendix D, we may work in the reduced problem and assume full target rank without loss of generality. Any local descent path found after this reduction zero-pads to a local descent path for the original problem.

D.4 The Rank-Deficient Core

Here we reduce the network to a smaller *rank-deficient core* (Lemma D.7 and Figure 4). This is a second reduction, distinct from the full-target-rank reduction above (Appendix D.3). This reduction freezes all tensors outside one connected component of rank-deficient edges and absorbs them into the target tensor, in such a way as to preserve the loss up to an additive constant.

Lemma D.7. *Assume $T(\theta) \neq T^*$. Then there is a connected component C of rank-deficient internal edges and a reduced TTN, with parameter $\widetilde{\theta}$, target $\widetilde{T}^*$, and ordinary squared loss $\widetilde{\mathcal{L}}$, such that:*

1. *the internal edges of the reduced TTN are exactly the edges of C;*

2. *every internal edge of the reduced model $\widetilde{T}(\widetilde{\theta})$ has rank strictly less than r_e;*

3. *every internal edge of the reduced target $\widetilde{T}^*$ has rank r_e;*

4. $\widetilde{T}(\widetilde{\theta}) \neq \widetilde{T}^*$;

5. *if θ is a local minimum of $\mathcal{L}$, then $\widetilde{\theta}$ is a local minimum of $\widetilde{\mathcal{L}}$;*

6. *every strict local descent path for $\widetilde{\mathcal{L}}$ at $\widetilde{\theta}$ lifts to a strict local descent path for $\mathcal{L}$ at θ.*

Proof. If no internal edge is rank-deficient, then Theorem 5.2 implies $T(\theta) = T^*$, contrary to assumption. Hence at least one internal edge is rank-deficient. Choose a connected component C of the subgraph induced by the rank-deficient internal edges, and root the tree at a node of C. Let V_C be the set of vertices incident to edges of C. Every internal edge leaving V_C is full-rank, since C is a connected component.

We first describe a single boundary stripping step. Let e be a full-rank boundary edge, and let H_e be the factor produced by the frozen subtree outside V_C. Let F_e be the matricized contraction of the retained side across e, so that the represented tensor has edge matricization $F_e H_e^\top$. Since e is full-rank, Proposition D.2 gives $D_e = 0$, so H_e has full column rank. Set $G := H_e^\top H_e$. The change of variables

$$H_e \mapsto H_e G^{-1/2}, \qquad F_e \mapsto F_e G^{1/2}$$

leaves $F_e H_e^\top$ unchanged. Thus it is only an invertible reparametrization of the retained boundary factor, and we may compute in coordinates where $H_e^\top H_e = I_{r_e}$. The reduced parameter $\widetilde{\theta}$ below is defined in these coordinates; when we later lift reduced perturbations back to the original network, we undo this change of variables. In these coordinates the cross-Gram $M_e = H_e^{*\top} H_e$ differs from the original cross-Gram by right multiplication by $G^{-1/2}$, so it is still invertible.

Now apply the same change of variables to every maximal subtree attached to V_C. Let B be the set of boundary edges leaving V_C. After these changes, each frozen factor satisfies $H_e^\top H_e = I_{r_e}$ for $e \in B$.

For any choice of the retained core tensors, let X be the tensor obtained by contracting only the vertices in V_C, leaving open the original external modes incident to V_C and one additional mode of dimension r_e for each $e \in B$. Reattaching the frozen subtrees is the linear map

$$\Gamma X := \text{the tensor obtained from } X \text{ by contracting its } e\text{-mode with } H_e \text{ for every } e \in B.$$

Since the columns of each H_e are orthonormal, Γ is an isometry. Therefore, for every such open tensor X,

$$\frac{1}{2}\|\Gamma X - T^*\|_F^2 = \frac{1}{2}\|X - \Gamma^* T^*\|_F^2 + \frac{1}{2}\|(I - \Gamma\Gamma^*)T^*\|_F^2.$$

Observe the second term is a constant independent of X. We define the reduced TTN to be the network on the vertices V_C, with the boundary edges in B treated as external modes, and we set

$$\widetilde{T}^* := \Gamma^* T^*.$$

Concretely, this target reduction acts as follows on each stripped boundary edge. For a boundary edge $e \in B$, the adjoint Γ^* contracts the corresponding target factor with H_e. Thus, using the target cut factorization $T^{*(e)} = F_e^* H_e^{*\top}$, the target factor on the new boundary mode is

$$T^{*(e)} H_e = F_e^* H_e^{*\top} H_e = F_e^* M_e.$$

The reduced parameter $\widetilde{\theta}$ is the retained collection of core tensors in the chosen coordinates, and the reduced loss is

$$\widetilde{\mathcal{L}}(\widetilde{\phi}) := \frac{1}{2}\|\widetilde{T}(\widetilde{\phi}) - \widetilde{T}^*\|_F^2.$$

The internal edges of this reduced TTN are exactly the edges of C, proving (1). Since the construction only changes coordinates on boundary modes, every internal edge of $\widetilde{T}(\widetilde{\theta})$ remains rank-deficient, proving (2).

It remains to check the target ranks on the core edges. For a stripped boundary edge, M_e is invertible by Corollary D.5. Thus replacing F_e^* by $F_e^* M_e$ only multiplies the corresponding boundary mode of the core target by an invertible matrix. In any matricization across an internal core edge, this appears as multiplication on the left or right by an invertible matrix, and therefore does not change the rank. Hence, for each $e \in C$,

$$\operatorname{rank}(\widetilde{T}^{*(e)}) = \operatorname{rank}(T^{*(e)}) = r_e.$$

This proves (3). For any $e \in C$, the model has $\operatorname{rank}(\widetilde{T}(\widetilde{\theta})^{(e)}) < r_e$ while the target has $\operatorname{rank}(\widetilde{T}^{*(e)}) = r_e$, so $\widetilde{T}(\widetilde{\theta}) \neq \widetilde{T}^*$, proving (4).

It remains to record the local variational relation between the two losses. The stripping construction defines a local embedding

$$\jmath : \widetilde{\Theta} \to \Theta$$

near $\widetilde{\theta}$: given reduced core tensors, undo the boundary coordinate changes and reattach all frozen tensors outside V_C. This map sends $\widetilde{\theta}$ to θ. By the isometry identity above,

$$\mathcal{L}(\jmath(\widetilde{\phi})) = \widetilde{\mathcal{L}}(\widetilde{\phi}) + C$$

for all $\widetilde{\phi}$ sufficiently close to $\widetilde{\theta}$, where C is independent of $\widetilde{\phi}$. The map $\jmath$ is continuous, and the coordinate changes on the retained boundary modes are invertible. Therefore local minimality of θ implies local minimality of $\widetilde{\theta}$, proving (5). The same identity shows that any strict local descent path for $\widetilde{\mathcal{L}}$ lifts through $\jmath$ to a strict local descent path for $\mathcal{L}$, proving (6). □

D.5 Descent on the Rank-Deficient Core

We now construct a descent path on the rank-deficient core, using the reduced squared loss supplied by Lemma D.7.

For a core vertex v, let $\widetilde{n}_v$ be defined analogous to n_v (Appendix B), as the product of the dimensions of the external modes incident to v in the reduced TTN, with $\widetilde{n}_v = 1$ if there are none. Thus a rank-one tensor on the reduced external modes can be written as $\bigotimes_v u_v$, with $u_v \in \mathbb{R}^{\widetilde{n}_v}$.

By Lemma D.7, the reduced residual $\widetilde{R} := \widetilde{T}(\widetilde{\theta}) - \widetilde{T}^*$ is nonzero. Hence there is a rank-one tensor $U = \bigotimes_v u_v$, with $u_v \in \mathbb{R}^{\widetilde{n}_v}$, such that $\langle \widetilde{R}, U \rangle < 0$.

For each internal core edge e, choose a unit vector $\eta_e \in D_e$. For each core node v with incident core edges $e_1, \ldots, e_k$, perturb

$$\delta W_v := \eta_{e_1} \otimes \cdots \otimes \eta_{e_k} \otimes u_v.$$

Let $\widetilde{\theta}(t)$ be the reduced parameter obtained by replacing each core tensor W_v by $W_v + t\,\delta W_v$.

Lemma D.8. *Let N be the number of core nodes. Then*

$$\widetilde{T}(\widetilde{\theta}(t)) = \widetilde{T}(\widetilde{\theta}) + t^N U.$$

Proof. For $S \subseteq V_C$, define $\widetilde{\theta}^S$ by

$$W_v(\widetilde{\theta}^S) = \begin{cases} \delta W_v, & v \in S, \\ W_v(\widetilde{\theta}), & v \notin S. \end{cases}$$

By multilinearity of the reduced TTN contraction,

$$\widetilde{T}(\widetilde{\theta}(t)) = \sum_{S \subseteq V_C} t^{|S|} \widetilde{T}(\widetilde{\theta}^S). \tag{3}$$

We claim that $\widetilde{T}(\widetilde{\theta}^S) = 0$ for every proper nonempty subset $S \subsetneq V_C$. Indeed, connectedness of the core gives an internal core edge $e = \{x, y\}$ with $x \in S$ and $y \notin S$. The tensor δW_x has factor η_e on its e-mode, while the tensor at y is unperturbed. Since $D_e = \ker(\operatorname{mat}_e W_y)$ by Proposition D.2, contracting W_y against η_e along its e-mode yields zero. Hence the entire term $\widetilde{T}(\widetilde{\theta}^S)$ is zero.

Of the terms in Equation (3), only $S = \varnothing$ and $S = V_C$ remain. The first term is $\widetilde{T}(\widetilde{\theta})$. In the second, every internal core edge contracts two copies of η_e, giving $\langle \eta_e, \eta_e \rangle = 1$, and the remaining external factors give $\bigotimes_v u_v = U$. Since this term uses one perturbation at each of the N core vertices, its coefficient is t^N. □

Lemma D.9. *For all sufficiently small $t > 0$,*

$$\widetilde{\mathcal{L}}(\widetilde{\theta}(t)) < \widetilde{\mathcal{L}}(\widetilde{\theta}).$$

Proof. By Lemma D.8 and the definition of the loss function,

$$\widetilde{\mathcal{L}}(\widetilde{\theta}(t)) - \widetilde{\mathcal{L}}(\widetilde{\theta}) = t^N \langle \widetilde{R}, U \rangle + \frac{1}{2} t^{2N} \|U\|_F^2.$$

Since $\langle \widetilde{R}, U \rangle < 0$, this is negative for all sufficiently small $t > 0$. □

The path $\widetilde{\theta}(t)$ is continuous and satisfies $\widetilde{\theta}(0) = \widetilde{\theta}$, so this descent is arbitrarily local in parameter space.

We now prove the main theorem by merely lifting this descent path on the core to the entire network.

D.6 Main Theorem

Theorem 4.3. *Every local minimum of $\mathcal{L}$ that is a minimum-norm point is a global minimum with $\mathcal{L} = 0$.* ∃∀N

Proof. Suppose for contradiction that θ is a minimum-norm local minimum with $T(\theta) \neq T^*$. By Lemma D.7, there is a reduced core parameter $\widetilde{\theta}$ which is a local minimum of the reduced loss $\widetilde{\mathcal{L}}$. But Lemma D.9 gives a path $\widetilde{\theta}(t)$ with $\widetilde{\mathcal{L}}(\widetilde{\theta}(t)) < \widetilde{\mathcal{L}}(\widetilde{\theta})$ for all sufficiently small $t > 0$, a contradiction. □

Remark D.10. The descent direction has order N, the number of nodes in the rank-deficient core: $\widetilde{\mathcal{L}}(\widetilde{\theta}(t)) - \widetilde{\mathcal{L}}(\widetilde{\theta}) \sim t^N \langle \widetilde{R}, U \rangle$. Intuitively, this means that the more the network needs to learn "in one step," the higher the order of the descent direction and the harder it is to escape the saddle point. Abbe et al. (2023) see similar phenomenon in neural networks and hypothesize that this is a general mechanism for learning difficulty.

D.7 Tightness of Assumptions

This subsection details the necessity of the core assumptions underlying Theorem 4.3. Note that the core claim of this paper is an *existence* claim (that a model class with worst-case hardness can have a benign loss landscape) for which these assumptions are sufficient, but future generalizations may wish to weaken them.

For each condition, we outline explicit counterexamples or known results demonstrating that removing it permits the existence of spurious local minima. However, we believe that generalized results may still be possible by merely weakening these assumptions or the desired conclusions.

Minimum-norm is necessary. The minimum-norm condition in Theorem 4.3 is essential. Appendix G constructs an explicit spurious local minimum for a realizable 3-leaf Tucker decomposition at a parameter that is not minimum norm.

Realizability is necessary. Without realizability, Theorem 4.3 fails even in the simplest nontrivial TTN topology. Consider the star graph with all bond dimensions $r_e = 1$: the TTN output is $T(\theta) = \lambda u_1 \otimes \cdots \otimes u_m$, and fitting a non-rank-1 target T^* reduces to the best rank-1 tensor approximation problem. For $m \geq 3$ this problem is known to have spurious local minima for generic targets, in contrast to the matrix case ($m = 2$), where Baldi & Hornik (1989) rules them out. An explicit $3 \times 3 \times 3 \times 3$ supersymmetric target with a nonglobal local minimum is given in Kofidis & Regalia (2002, Example 4).

The square loss is necessary. Consider the case of a two-layer deep linear network $W_1 \in \mathbb{R}^{m \times n}, W_2 \in \mathbb{R}^{n \times m}$. If the hidden width m is less than the input/output width n, then Trager et al. (2019) shows one can have spurious critical points if one uses losses other than the square loss.[8] As tree tensor networks generalize deep linear networks, this shows that the square loss assumption is necessary.

[8] This does not appear to be prevented by our minimum-norm assumption.

However, note that this does not preclude the possibility of results like that of Trager et al. (2019), which show that the mere parameterization cannot create spurious critical points in deep linear networks if there are no such critical points in function space. We conjecture that a similar result holds in tree tensor networks under the minimum-norm assumption, but leave this to future work.

The tree topology is necessary. Chen et al. (2020) show that tensor networks with a ring topology can have spurious local minima. They exhibit, for any $d \geq 3$, a target of bond dimension $r + 1$ and external dimension $n \geq r^2 + 1$ together with a non-strict spurious local minimum in the ring class of bond dimension r^{d-1}, so the example persists under substantial over-parameterization. They use cycles to construct a mechanism similar to our counter-example in Appendix G, where local variations to match the target create cross-terms that dominate the loss. We suspect that this example does not violate the minimum-norm assumption, and thus it seems plausible that acyclicity is an independently necessary assumption.

E Escape Paths

As a corollary of the analysis in Appendix D, we establish absence of bad valleys without the minimum-norm assumption.

Corollary 4.4. *From any θ_0 with $T(\theta_0) \neq T^*$, there exists a continuous path $\gamma : [0, 1] \to \Theta$ with $\gamma(0) = \theta_0$ such that $t \mapsto \mathcal{L}(\gamma(t))$ is non-increasing and $\mathcal{L}(\gamma(1)) < \mathcal{L}(\gamma(0))$.*

Proof. The fiber $\mathcal{F} := \{\theta' : T(\theta') = T(\theta_0)\}$ is the preimage of a point under the polynomial map T, hence a real algebraic variety. Let C be the connected component of $\mathcal{F}$ containing θ_0. Since $\mathcal{F}$ is closed and the norm is proper, the norm attains its minimum on C at some $\theta_{\min} \in C$.

Semialgebraic sets have finitely many connected components, each semialgebraic (Bochnak et al., 2013). Hence C is open as a subset of $\mathcal{F}$, and so $\theta_{\min}$ is a local minimum of the norm over the full fiber $\mathcal{F}$. By Proposition F.10, $\theta_{\min}$ locally minimizing the norm over $\mathcal{F}$ implies the minimum-norm condition. Since connected semialgebraic sets are path-connected (Bochnak et al., 2013), there exists a continuous path from θ_0 to $\theta_{\min}$ within C. Along this path, T is constant, hence $\mathcal{L}$ is constant.

Since $T(\theta_{\min}) = T(\theta_0) \neq T^*$, the proof of Theorem 4.3 gives a continuous path from $\theta_{\min}$ along which $\mathcal{L}$ strictly decreases. Concatenating this path with the prior constant-loss path from θ_0 to $\theta_{\min}$ gives the desired non-increasing path from θ_0 to strictly lower loss. □

Remark E.1. This is weaker than Theorem 4.3: the constant-loss segment may leave any neighborhood of θ_0, so the corollary does not rule out spurious local minima in unrestricted parameter space.

F The Minimum-Norm Condition

Our core result, Theorem 4.3, applies only for points in parameter space satisfying the minimum-norm condition (Definition 4.1). We prove two main results to support the use of this assumption; we believe it is a significant contribution rather than a limitation. First, minimum norm is equivalent to a *balancedness* condition, an easily-checkable algebraic identity (Definition F.9 and Proposition F.10). This identity generalizes the well-known balancedness condition of the deep linear network literature, where it is conserved by gradient flow (Arora et al., 2018b; Du et al., 2018) and assumed in standard convergence analyses (Arora et al., 2018a; Bah et al., 2022). Second, gradient flow on any loss which depends only on the tensor $T(\theta)$ represented by the TTN (including the square loss we use) conserves balancedness, and ℓ_2 regularization drives the balancedness defect to zero at an exponential rate (Proposition F.12 and Corollary F.13).

The minimum-norm hypothesis is therefore equivalent to (a generalization of) a commonly-assumed and easily-checkable balancedness assumption, preserved by training dynamics, and produced by regularization. We believe this is a significant contribution of our work, and one that will provide a basis for future work on training dynamics of tree tensor networks.

Significantly, our proofs in this appendix make substantial use of (real) geometric invariant theory (GIT) (Mumford et al., 1994; Richardson & Slodowy, 1990). The gauge freedom of a TTN is an

action of the reductive group $G_{\mathcal{T}} = \prod_e \mathrm{GL}(r_e)$ on parameter space; minimum-norm parameters are equivalent to parameters which satisfy the *Kempf-Ness condition* (Kempf & Ness, 1979; Richardson & Slodowy, 1990); balancedness is the vanishing of its moment map; and the conservation law associated to balancedness is a Noether law for the gauge symmetry. The core result is that every fiber of the contraction map contains a unique closed gauge orbit (Proposition F.6). This can be viewed as a canonical-form statement: each representable tensor has a distinguished parameterization, unique up to the orthogonal gauge group, and the contraction map separates closed gauge orbits exactly as a GIT quotient map does. These conclusions require the tree topology and fail on networks with cycles (Remark F.7). The same machinery appears in Lindsey & Menon (2025), who apply the Kempf–Ness theorem to deep linear networks, and in Acuaviva et al. (2023), who define canonical forms of complex tensor networks by norm minimization over the gauge group.

F.1 The Fiber has a Unique Closed Gauge Orbit

This subsection proves that every fiber of the contraction map contains a unique closed gauge orbit (Proposition F.6). The argument has three steps. We first treat the two-node network, where contraction is just matrix multiplication, in Lemma F.2 and Corollary F.3. Second, we show that contracting an internal edge of the tree commutes with taking orbit closures (Lemma F.4). Third, we collapse the tree one edge at a time to show that any two parameters with the same represented tensor have intersecting orbit closures (Proposition F.5), which gives uniqueness, since distinct orbits are disjoint (Proposition F.6). Existence of a closed gauge orbit is general, but *uniqueness* requires the tree topology, as it fails on networks with cycles (Remark F.7).

F.1.1 The Two-Node Case: Minimal Factorizations

Throughout this subsection, fix dimensions m, n, r and let $\mathrm{GL}(r)$ act on pairs $(A, B) \in \mathbb{R}^{m\times r} \times \mathbb{R}^{r\times n}$ by

$$g \cdot (A, B) = (Ag^{-1}, gB), \tag{4}$$

leaving the product AB invariant.

Definition F.1 (Minimal factorization)**.** A pair (A, B) is a *minimal factorization* of $C := AB$ if

$$\operatorname{rank} A = \operatorname{rank} B = \operatorname{rank} C.$$

Lemma F.2. *Let $A \in \mathbb{R}^{m\times r}$, $B \in \mathbb{R}^{r\times n}$, and $C = AB$.*

(i) The pair (A, B) is a minimal factorization of C if and only if $\mathbb{R}^r = \operatorname{im} B \oplus \ker A$.

(ii) The orbit closure $\overline{\mathrm{GL}(r) \cdot (A, B)}$ contains a minimal factorization of C.

(iii) The minimal factorizations of C form a single $\mathrm{GL}(r)$-orbit.

Proof. (i) Since $A(\operatorname{im} B) = \operatorname{im}(AB) = \operatorname{im} C$, the restriction $A|_{\operatorname{im} B} : \operatorname{im} B \to \operatorname{im} C$ is surjective. Hence we have the equivalences

$$\operatorname{rank} B = \operatorname{rank} C \iff A|_{\operatorname{im} B} \text{ is injective} \iff \operatorname{im} B \cap \ker A = 0.$$

If this holds, then $\dim \operatorname{im} B = \operatorname{rank} C$. Under this condition, we have the further equivalences

$$\operatorname{rank} A = \operatorname{rank} C \iff \dim \ker A = r - \dim \operatorname{im} B \iff \dim(\operatorname{im} B + \ker A) = r.$$

Combining the two equivalences, we have that $\operatorname{rank} A = \operatorname{rank} B = \operatorname{rank} C$ is equivalent to $\operatorname{im} B \cap \ker A = 0$ and $\dim(\operatorname{im} B + \ker A) = r$. Notice that the two conditions $\operatorname{im} B \cap \ker A = 0$ and $\dim(\operatorname{im} B + \ker A) = r$ are together equivalent to the internal direct sum decomposition $\mathbb{R}^r = \operatorname{im} B \oplus \ker A$.

(ii) Let $K = \operatorname{im} B \cap \ker A$, and choose a complement L of K in $\operatorname{im} B$ and a complement M of $\operatorname{im} B$ in $\mathbb{R}^r$, so that $\mathbb{R}^r = L \oplus K \oplus M$. Let π_L, π_K, π_M be the associated projections. Note $\pi_M B = 0$ (as $\operatorname{im} B = L \oplus K$) and $A\pi_K = 0$ (as $K \subseteq \ker A$).

We will take the limit of a particular $\mathrm{GL}(r)$-action to eliminate K and M, leading to a minimal factorization. Precisely, for $t > 0$ define $g_t = \mathrm{id}_L \oplus t\,\mathrm{id}_K \oplus t^{-1}\,\mathrm{id}_M \in \mathrm{GL}(r)$, which acts on (A, B) according to Equation (4). In the limit as t goes to zero, this action yields

$$\lim_{t\to 0}\big(Ag_t^{-1},\, g_tB\big) = \lim_{t\to 0}\big(A\pi_L + tA\pi_M,\ \pi_LB + t\pi_KB\big) = \big(A\pi_L,\, \pi_LB\big).$$

Label this limit by $\big(A_0, B_0\big) := \big(A\pi_L,\, \pi_L B\big)$. By construction, the pair $\big(A_0, B_0\big)$ is a member of the orbit closure $\overline{\mathrm{GL}(r) \cdot (A, B)}$. Moreover, it is a factorization of C, since $(Ag_t^{-1})(g_t B) = AB = C$ for all finite t, and matrix multiplication is continuous.

It is a minimal factorization by (i): $\operatorname{im} B_0 = \pi_L(L \oplus K) = L$, and $\ker A_0 = K \oplus M$, since A is injective on L (indeed $L \cap \ker A \subseteq L \cap K = 0$); hence $\mathbb{R}^r = \operatorname{im} B_0 \oplus \ker A_0$.

(iii) Let (A_0, B_0) and (A_1, B_1) be minimal factorizations of C. To show they lie on a single $\mathrm{GL}(r)$-orbit, we will construct an explicit $g \in \mathrm{GL}(r)$ such that $g \cdot (A_0, B_0) = (A_1, B_1)$. By part (i), we have $\mathbb{R}^r = \operatorname{im} B_i \oplus \ker A_i$ for $i = 0, 1$. Observe that the restriction

$$A_i|_{\operatorname{im} B_i} : \operatorname{im} B_i \to \operatorname{im} C$$

is an isomorphism: it is surjective because $A_i(\operatorname{im} B_i) = \operatorname{im} C$, and injective because $\operatorname{im} B_i \cap \ker A_i = 0$. Define

$$h := \big(A_1|_{\operatorname{im} B_1}\big)^{-1} \circ \big(A_0|_{\operatorname{im} B_0}\big) : \operatorname{im} B_0 \to \operatorname{im} B_1$$

so that $A_1 h = A_0$ on $\operatorname{im} B_0$. Since $\ker A_0$ and $\ker A_1$ have the same dimension, extend h by any isomorphism $\ker A_0 \to \ker A_1$. This gives an invertible map $g \in \mathrm{GL}(r)$, using the two decompositions $\mathbb{R}^r = \operatorname{im} B_i \oplus \ker A_i$ from (i).

We conclude by showing that $g \cdot (A_0, B_0) = (A_1, B_1)$, which requires that $A_0 g^{-1} = A_1$ and $gB_0 = B_1$. First, we show $gB_0 = B_1$. Observe that

$$A_1(gB_0) = A_0 B_0 = C = A_1 B_1.$$

Both gB_0 and B_1 have columns in $\operatorname{im} B_1$, where A_1 is injective, so $gB_0 = B_1$. To show that $A_0 g^{-1} = A_1$, again use the decomposition $\mathbb{R}^r = \operatorname{im} B_1 \oplus \ker A_1$ from (i). On $\operatorname{im} B_1$, we have $g^{-1} = h^{-1}$ and $A_0 h^{-1} = A_1$, so obviously $A_0 g^{-1} = A_1$; on $\ker A_1$, g^{-1} maps into $\ker A_0$, so both sides vanish. Therefore $(A_1, B_1) = g \cdot (A_0, B_0)$. □

Corollary F.3. *Let $A \in \mathbb{R}^{m \times r}$, $B \in \mathbb{R}^{r \times n}$, and $C = AB$. Then $\overline{\mathrm{GL}(r) \cdot (A, B)}$ contains* every *minimal factorization of C, and in particular one with*

$$\|A_0\|_F^2 = \|B_0\|_F^2 = \|C\|_*, \tag{5}$$

the nuclear norm (sum of singular values) of C. Consequently, any two factorizations of the same matrix have intersecting orbit closures.

Proof. The orbit closure $\overline{\mathrm{GL}(r) \cdot (A, B)}$ is $\mathrm{GL}(r)$-invariant: for any $g \in \mathrm{GL}(r)$,

$$g \cdot \overline{\mathrm{GL}(r) \cdot (A, B)} = \overline{g \cdot (\mathrm{GL}(r) \cdot (A, B))} = \overline{\mathrm{GL}(r) \cdot (A, B)}.$$

Thus, if the orbit closure contains one point of a $\mathrm{GL}(r)$-orbit, it contains that entire orbit. By Lemma F.2(ii), the orbit closure $\overline{\mathrm{GL}(r) \cdot (A, B)}$ contains some minimal factorization of C. Lemma F.2(iii) tells us that the $\mathrm{GL}(r)$-orbit of any minimal factorization contains every minimal factorization of C. Together these three facts imply that the orbit closure $\overline{\mathrm{GL}(r) \cdot (A, B)}$ contains every minimal factorization of C.

To show a representative satisfying Equation (5) exists, let $C = \sum_{j=1}^{k} \sigma_j\, u_j w_j^\top$ be a singular value decomposition with $k = \operatorname{rank} C$ and $\sigma_j > 0$. Since $k \le \operatorname{rank} B \le r$, we may choose orthonormal $e_1, \ldots, e_k \in \mathbb{R}^r$ and set

$$A_0 = \sum_j \sqrt{\sigma_j}\; u_j e_j^\top, \qquad B_0 = \sum_j \sqrt{\sigma_j}\; e_j w_j^\top.$$

Then $A_0 B_0 = C$, $\operatorname{im} B_0 = \operatorname{span}(e_1, \ldots, e_k)$, and $\ker A_0 = \operatorname{span}(e_1, \ldots, e_k)^\perp$, so (A_0, B_0) is minimal by Lemma F.2(i), with $\|A_0\|_F^2 = \|B_0\|_F^2 = \sum_j \sigma_j = \|C\|_*$.

Finally, the orbit closures of any two factorizations of C both contain the (nonempty) set of minimal factorizations of C, so they intersect. □

F.1.2 The General Case

Fix an internal edge $e = \{u, v\}$. As $\mathcal{T}$ is a tree, every edge connects two distinct nodes $u \neq v$. Contracting e merges u and v into a single node and yields a network $\mathcal{T}/e$ on the same external legs with one fewer internal edge; since $\mathcal{T}$ is a tree, so is $\mathcal{T}/e$. Let $c_e : \Theta_{\mathcal{T}} \to \Theta_{\mathcal{T}/e}$ be the partial contraction that replaces W_u, W_v by their contraction across e and leaves all other node tensors unchanged. Matricize the two adjacent tensors along e, with the bond as column index on the u-side and as row index on the v-side:

$$A = \mathrm{mat}_e(W_u) \in \mathbb{R}^{m \times r_e}, \qquad B = \mathrm{mat}_e(W_v) \in \mathbb{R}^{r_e \times n},$$

where m and n are the products of the dimensions of the remaining modes of u and of v. The merged node tensor is then the matrix product AB, and the gauge factor $\mathrm{GL}(r_e)$ acts on (A, B) exactly as in the previous subsection, leaving all other node tensors fixed. The gauge group of the smaller network is $G_{\mathcal{T}/e} = \prod_{e' \in E_{\mathrm{int}}(\mathcal{T}) \setminus \{e\}} \mathrm{GL}(r_{e'})$; the map c_e is $G_{\mathcal{T}/e}$-equivariant, is invariant under the $\mathrm{GL}(r_e)$-factor, and satisfies $T_{\mathcal{T}} = T_{\mathcal{T}/e} \circ c_e$ by associativity of contraction.

Lemma F.4. *For every internal edge $e = \{u, v\}$ and every $\theta \in \Theta_{\mathcal{T}}$,*

$$c_e\big(\overline{G_{\mathcal{T}} \cdot \theta}\big) = \overline{G_{\mathcal{T}/e} \cdot c_e(\theta)}.$$

Proof. ($\subseteq$) Since c_e is invariant under the $\mathrm{GL}(r_e)$-factor and equivariant for the remaining gauge group, we have $c_e(G_{\mathcal{T}} \cdot \theta) = G_{\mathcal{T}/e} \cdot c_e(\theta)$ pointwise; continuity of c_e then gives the inclusion of orbit closures.

($\supseteq$) An element $\eta \in \overline{G_{\mathcal{T}/e} \cdot c_e(\theta)}$ is given by $\eta = \lim_i h_i \cdot c_e(\theta)$ for some sequence $h_i \in G_{\mathcal{T}/e}$. We aim to prove $\eta \in c_e(\overline{G_{\mathcal{T}} \cdot \theta})$; that is, that there exists $\tilde{\eta} \in \overline{G_{\mathcal{T}} \cdot \theta}$ such that $c_e(\tilde{\eta}) = \eta$. Lift each h_i to $\widetilde{h}_i \in G_{\mathcal{T}}$ by letting it act trivially on the bond at e, and set $\theta_i = \widetilde{h}_i \cdot \theta \in G_{\mathcal{T}} \cdot \theta$. Then $c_e(\theta_i) = h_i \cdot c_e(\theta) \to \eta$. If θ_i was bounded, one could pass to a subsequence of θ_i that converges to a point $\tilde{\eta} \in \overline{G_{\mathcal{T}} \cdot \theta}$, with $c_e(\tilde{\eta}) = \eta$, and conclude the proof.

However, even though $c_e(\theta_i)$ converges, the sequence θ_i itself need not be bounded: gauges on the other bonds of u and v can blow up the two tensors adjacent to e while their contraction remains bounded. We correct this by replacing the pair of tensors at e with a norm-controlled minimal factorization of the same merged tensor.

Precisely, let $(A_i, B_i) = (\mathrm{mat}_e\, W_u(\theta_i),\ \mathrm{mat}_e\, W_v(\theta_i))$ and let $C_i = A_i B_i$ be the matricized merged tensor of $c_e(\theta_i)$. Applying Corollary F.3 to the pair (A_i, B_i), while leaving all other node tensors fixed, gives a point $\widetilde{\theta}_i \in \overline{\mathrm{GL}(r_e) \cdot \theta_i}$ obtained from θ_i by replacing (A_i, B_i) with a minimal factorization of C_i whose two factors each have squared Frobenius norm equal to $\|C_i\|_*$. Then $c_e(\widetilde{\theta}_i) = c_e(\theta_i)$, since the merged tensor and all other node tensors are unchanged.

Moreover the sequence $\widetilde{\theta}_i$ is bounded. The node tensors away from u and v are unchanged when passing from θ_i to $\widetilde{\theta}_i$, and they appear as node tensors in the convergent sequence $c_e(\theta_i)$, so they are bounded. The two tensors at u and v have squared Frobenius norm equal to $\|C_i\|_*$, which is bounded since C_i converges. Finally,

$$\widetilde{\theta}_i \in \overline{\mathrm{GL}(r_e) \cdot \theta_i} \subseteq \overline{G_{\mathcal{T}} \cdot \theta_i} = \overline{G_{\mathcal{T}} \cdot \theta},$$

where the inclusion uses $\mathrm{GL}(r_e) \subseteq G_{\mathcal{T}}$, and the equality uses $\theta_i \in G_{\mathcal{T}} \cdot \theta$.

Passing to a subsequence, $\widetilde{\theta}_i \to \widetilde{\eta}$. Since $\overline{G_{\mathcal{T}} \cdot \theta}$ is closed, $\widetilde{\eta} \in \overline{G_{\mathcal{T}} \cdot \theta}$, and by continuity $c_e(\widetilde{\eta}) = \lim_i c_e(\widetilde{\theta}_i) = \lim_i c_e(\theta_i) = \eta$. Hence $\eta \in c_e\big(\overline{G_{\mathcal{T}} \cdot \theta}\big)$. □

Proposition F.5. *If $\theta, \theta' \in \Theta_{\mathcal{T}}$ satisfy $T(\theta) = T(\theta')$, then*

$$\overline{G_{\mathcal{T}} \cdot \theta} \cap \overline{G_{\mathcal{T}} \cdot \theta'} \neq \varnothing.$$

Proof. We induct on the number of internal edges. With no internal edges the network is a single node, T is the identity, and $G_{\mathcal{T}}$ is trivial, so $\theta = \theta'$.

Otherwise choose an internal edge e. Since $\mathcal{T}$ is a tree, $\mathcal{T}/e$ is again a tree and has one fewer internal edge, so it lies in the induction class. Moreover, since

$$T_{\mathcal{T}/e}(c_e(\theta)) = T(\theta) = T(\theta') = T_{\mathcal{T}/e}(c_e(\theta')),$$

the induction hypothesis applied to $\mathcal{T}/e$ yields a point

$$\xi \in \overline{G_{\mathcal{T}/e} \cdot c_e(\theta)} \cap \overline{G_{\mathcal{T}/e} \cdot c_e(\theta')}.$$

We now lift this common point back to the original tree. By Lemma F.4, the membership of ξ in the first orbit closure gives

$$\widehat{\theta} \in \overline{G_{\mathcal{T}} \cdot \theta} \qquad \text{with} \qquad c_e(\widehat{\theta}) = \xi.$$

The membership of ξ in the second orbit closure similarly gives

$$\widehat{\theta'} \in \overline{G_{\mathcal{T}} \cdot \theta'} \qquad \text{with} \qquad c_e(\widehat{\theta'}) = \xi.$$

Thus $c_e(\widehat{\theta}) = c_e(\widehat{\theta'})$. Equivalently, all node tensors of $\widehat{\theta}$ and $\widehat{\theta'}$ away from u and v agree, and their pairs at e are two factorizations of the same merged tensor. Applying Corollary F.3 to these two factorizations, while leaving the common node tensors away from u and v fixed, gives a point ζ lying in both $\mathrm{GL}(r_e)$-orbit closures.

It remains to check that ζ lies in the two original orbit closures. We show $\zeta \in \overline{G_{\mathcal{T}} \cdot \theta}$; the same argument gives $\zeta \in \overline{G_{\mathcal{T}} \cdot \theta'}$. To see this, observe the nested inclusions

$$\zeta \in \overline{\mathrm{GL}(r_e) \cdot \widehat{\theta}} \subseteq \overline{G_{\mathcal{T}} \cdot \widehat{\theta}} \subseteq \overline{G_{\mathcal{T}} \cdot \theta}.$$

The first inclusion uses $\mathrm{GL}(r_e) \subseteq G_{\mathcal{T}}$. For the second inclusion, $\widehat{\theta} \in \overline{G_{\mathcal{T}} \cdot \theta}$, and this orbit closure is $G_{\mathcal{T}}$-invariant. Hence $G_{\mathcal{T}} \cdot \widehat{\theta} \subseteq \overline{G_{\mathcal{T}} \cdot \theta}$; since the latter set is closed, it contains $\overline{G_{\mathcal{T}} \cdot \widehat{\theta}}$. □

Proposition F.6. *Every fiber of the contraction map $T : \Theta_{\mathcal{T}} \to T(\Theta_{\mathcal{T}})$ contains a minimizer of the norm. The gauge orbit of any such minimizer is the* unique *closed $G_{\mathcal{T}}$-orbit in that fiber.*

Proof. Existence. Fix a fiber

$$\mathcal{F}_\theta = \{\eta : T(\eta) = T(\theta)\}.$$

This set is closed by continuity of T and nonempty by definition, so the norm attains a minimum on it. (Indeed, choosing any $\theta_1 \in \mathcal{F}_\theta$, it suffices to minimize over the compact set $\mathcal{F}_\theta \cap \{\eta : \|\eta\| \le \|\theta_1\|\}$.)

Let $\theta_{\min}$ be any norm minimizer in $\mathcal{F}_\theta$. Gauge transformations preserve the represented tensor, hence $G_{\mathcal{T}} \cdot \theta_{\min} \subseteq \mathcal{F}_\theta$. Therefore

$$\|\theta_{\min}\| \le \|g \cdot \theta_{\min}\| \qquad \text{for all } g \in G_{\mathcal{T}}.$$

Thus $\theta_{\min}$ minimizes the norm on its gauge orbit, i.e. it is a *minimum vector* of the gauge action in the sense of Richardson & Slodowy (1990).

We apply Richardson's closed-orbit theorem to this action. Its hypotheses hold: $G_{\mathcal{T}}$ is a product of real general linear groups acting rationally and linearly on $\Theta_{\mathcal{T}}$, and the Frobenius inner product is compatible with the action. Indeed, the maximal compact subgroup $\prod_e O(r_e)$ acts by isometries, and the symmetric matrices in the Lie algebra act by symmetric operators (a direct trace computation). By Theorem 4.4 of Richardson & Slodowy (1990), the orbit of a minimum vector is closed. Hence $G_{\mathcal{T}} \cdot \theta_{\min}$ is a closed orbit in the fiber.

Uniqueness. If $G_{\mathcal{T}} \cdot \eta$ and $G_{\mathcal{T}} \cdot \eta'$ are closed orbits in the same fiber, then $T(\eta) = T(\eta')$, so by Proposition F.5 their closures intersect. Since the two orbits are closed, these closures are the orbits themselves. Distinct orbits are disjoint, so $G_{\mathcal{T}} \cdot \eta = G_{\mathcal{T}} \cdot \eta'$. □

Remark F.7. The tree topology is required for the uniqueness half of Proposition F.6. Consider the simplest cyclic network: a single node with one bond joining it to itself, so that $\theta \in \mathbb{R}^{r \times r}$, the gauge group $\mathrm{GL}(r)$ acts by conjugation, and the contraction is $T(\theta) = \operatorname{tr} \theta$. Every diagonal matrix D is a critical point of the squared norm on its orbit — for symmetric X, $\langle XD - DX, D\rangle_F = \operatorname{tr}\big(X(DD^\top - D^\top D)\big) = 0$ — hence a minimum vector by Theorem 4.3 of Richardson & Slodowy (1990), and its conjugation orbit is closed by Theorem 4.4. Conjugation preserves eigenvalues, so diagonal matrices with distinct spectra lie in distinct orbits; for $r \ge 2$, a single trace fiber therefore contains a continuum of closed gauge orbits. (Compare the ring-topology counterexamples of Chen et al. (2020) discussed in the main text.)

Remark F.8. Over the complex numbers, Acuaviva et al. (2023) apply the same construction — norm minimization over the gauge group, via the Kempf–Ness theorem (Kempf & Ness, 1979) — to tensor networks of arbitrary geometry, obtaining a *minimal canonical form* that identifies two tensor networks exactly when their gauge orbit closures intersect. In the case of tree tensor networks, this minimal canonical form can be seen as closely related to the well-known *Vidal gauge* (Vidal, 2003; Shi et al., 2006). We are not able to immediately apply the algebraic machinery of complex GIT in the real case (see Acosta (2020) for details), so it was necessary to prove this result ourselves.

F.2 Equivalent Conditions to Minimum-Norm

This subsection characterizes the minimum-norm parameterizations (Definition 4.1). As stated, the condition is global, and difficult to check. Proposition F.10 shows that minimum norm is equivalent to *balancedness*, an algebraic condition which can be verified edge by edge.

Definition F.9 (Balanced parameterization)**.** A parameterization θ is *balanced* if at every internal edge $e = \{u, v\}$,

$$(\mathrm{mat}_e\, W_u)^\top\ \mathrm{mat}_e\, W_u\ =\ (\mathrm{mat}_e\, W_v)^\top\ \mathrm{mat}_e\, W_v.$$

Both sides are $r_e \times r_e$ Gram matrices. Intuitively, balancedness says the two endpoint tensors make use of the bond space identically.

This condition is a generalization of the well-known balancedness condition from the study of deep linear networks. In this case, the adjacent-layer differences $W_{j+1}^\top W_{j+1} - W_j W_j^\top$ are conserved by gradient flow (Arora et al., 2018b; Du et al., 2018), and the balanced manifold is a standard setting for convergence analyses (Arora et al., 2018a; Bah et al., 2022). Lindsey & Menon (2025) interpret this condition through geometric invariant theory, showing via the Kempf–Ness theorem that the ℓ_2 regularizer of a deep linear network is minimized exactly on the balanced manifold; Proposition F.10 is the corresponding statement for tree tensor networks.

Proposition F.10. *For $\theta \in \Theta_\mathcal{T}$ with fiber $\mathcal{F}_\theta = \{\theta' \in \Theta_\mathcal{T} : T(\theta') = T(\theta)\}$, the following are equivalent:*

(i) θ minimizes $\|\cdot\|$ on $\mathcal{F}_\theta$, i.e. θ is minimum norm;

(ii) θ locally minimizes $\|\cdot\|$ on $\mathcal{F}_\theta$;

(iii) θ minimizes $\|\cdot\|$ on its gauge orbit $G_\mathcal{T} \cdot \theta$, i.e. θ is a minimum vector;

(iv) θ is balanced.

Moreover, the minimum-norm points of a fiber form a single orbit of the orthogonal gauge group $K_\mathcal{T} = \prod_{e \in E_{\mathrm{int}}(\mathcal{T})} O(r_e)$.

Proof. We prove $(i) \Rightarrow (ii) \Rightarrow (iii) \Rightarrow (i)$, then $(iii) \Leftrightarrow (iv)$, and finally identify the minimum-norm points within a fiber.

(i) ⇒ (ii) is immediate.

(ii) ⇒ (iii). For g near the identity, gauge invariance gives $T(g \cdot \theta) = T(\theta)$, so $g \cdot \theta \in \mathcal{F}_\theta$. Thus $g \cdot \theta$ is a point of $\mathcal{F}_\theta$ near θ. Since θ locally minimizes the norm on $\mathcal{F}_\theta$, we have $\|g \cdot \theta\| \geq \|\theta\|$ for all such g. Hence $\rho_\theta : g \mapsto \|g \cdot \theta\|^2$ has a local minimum, and therefore a critical point, at the identity. By Richardson & Slodowy (1990, Theorem 4.3), θ is a minimum vector.

(iii) ⇒ (i). Since θ is a minimum vector, Richardson & Slodowy (1990, Theorem 4.4) implies that $G_\mathcal{T} \cdot \theta$ is closed. By Proposition F.6, there exists at least one norm minimizer $\theta_{\min}$ on $\mathcal{F}_\theta$, and its orbit is the unique closed gauge orbit in $\mathcal{F}_\theta$. Hence

$$G_\mathcal{T} \cdot \theta = G_\mathcal{T} \cdot \theta_{\min}.$$

Thus $\theta_{\min} \in G_\mathcal{T} \cdot \theta$. Because θ minimizes the norm on its gauge orbit,

$$\|\theta\| \leq \|\theta_{\min}\| = \min_{\eta \in \mathcal{F}_\theta} \|\eta\|.$$

The reverse inequality is automatic because $\theta \in \mathcal{F}_\theta$, so θ is minimum norm.

(iii) ⇔ (iv). By Richardson & Slodowy (1990, Theorem 4.3), θ is a minimum vector if and only if $\rho_\theta : g \mapsto \|g \cdot \theta\|^2$ is critical at the identity. We now compute this criticality condition. The Lie algebra of $G_\mathcal{T}$ is the direct sum of one copy of $\mathfrak{gl}(r_e)$ for each internal edge e. Every $X \in \mathfrak{gl}(r_e)$ decomposes as a sum of a skew-symmetric matrix and a symmetric matrix. The skew-symmetric directions generate the orthogonal gauge group $K_\mathcal{T}$, which preserves the Frobenius norm. Thus the derivative of ρ_θ vanishes automatically in those directions. Since directional derivatives at the identity form a linear functional on the Lie algebra, and every Lie-algebra direction is a sum of a skew-symmetric and a symmetric direction, ρ_θ is critical at the identity if and only if the derivative vanishes along every symmetric direction at every internal edge.

Fix $e = \{u, v\}$ and a symmetric matrix X. The one-parameter gauge $\exp(tX)$ at e acts by $\operatorname{mat}_e W_u \mapsto \operatorname{mat}_e W_u \exp(-tX)$ and $\operatorname{mat}_e W_v \mapsto \operatorname{mat}_e W_v \exp(tX)$, leaving all other node tensors fixed, so only these two terms in the total parameter norm vary. Thus

$$\frac{d}{dt}\Big|_{t=0} \Big(\big\|\operatorname{mat}_e W_u \exp(-tX)\big\|_F^2 + \big\|\operatorname{mat}_e W_v \exp(tX)\big\|_F^2 \Big) \\ = 2 \operatorname{tr}\Big[X \Big((\operatorname{mat}_e W_v)^\top \operatorname{mat}_e W_v - (\operatorname{mat}_e W_u)^\top \operatorname{mat}_e W_u \Big) \Big].$$

The bracketed matrix is symmetric, so this vanishes for every symmetric X if and only if that matrix is zero, i.e. if and only if θ is balanced at e. Since this holds independently at each internal edge, criticality of ρ_θ at the identity is equivalent to balancedness.

The $K_\mathcal{T}$-orbit. Let θ, θ' be minimum-norm points of the same fiber. By (i) ⇒ (iii), both are minimum vectors. Hence both gauge orbits are closed, and by Proposition F.6 they are the same orbit. In particular, $\theta' \in G_\mathcal{T} \cdot \theta$.

We now restrict from $G_\mathcal{T}$ to its maximal compact subgroup. By Richardson & Slodowy (1990, Theorem 4.3), the minimum vectors in the orbit of a minimum vector form a single $K_\mathcal{T}$-orbit. Since θ and θ' are both minimum vectors in the same $G_\mathcal{T}$-orbit, this gives $\theta' \in K_\mathcal{T} \cdot \theta$. Conversely, every point of $K_\mathcal{T} \cdot \theta$ is minimum norm, since $K_\mathcal{T}$ preserves both the norm and the fiber. □

Remark F.11. The tree topology of the network is necessary for Proposition F.10. However, only the implication (iii) ⇒ (i) uses the tree structure, via Proposition F.6. The chain (i) ⇒ (ii) ⇒ (iii) and the equivalence (iii) ⇔ (iv) hold for any tensor network whose internal edges join distinct nodes. In particular, minimum norm implies balancedness on any such network. The tree topology is needed for the converse direction: in the self-loop network of Remark F.7 with $r \geq 2$, $\theta = \operatorname{diag}(c, 0, \ldots, 0)$ is a minimum vector for $c \neq 0$, yet its trace fiber contains $(c/r)\, I_r$, of strictly smaller norm.

F.3 Minimum-Norm is Preserved by Gradient Flow

We show the minimum-norm condition is stable over training: it is preserved by gradient flow on the square loss, and more generally any loss that depends on the parameters only through the represented tensor (Proposition F.12). More specifically, Proposition F.12 shows that gauge invariance of the loss leads to a Noether-type conservation law for the failure of balancedness, generalizing the classical conserved quantities of deep linear networks (Arora et al., 2018b; Du et al., 2018). Further, in the presence of ℓ_2 regularization, the failure of balancedness is driven to zero at an exponential rate (Corollary F.13).

Let $\mathcal{L} = \ell \circ T$, where ℓ is any continuously differentiable function of the represented tensor — for instance the squared loss $\ell = \frac{1}{2|\mathcal{X}|}\| \cdot - T^*\|_F^2$ against a fixed target T^* — and consider the gradient flow $\dot\theta = -\nabla\mathcal{L}(\theta)$ with respect to the Frobenius inner product on $\Theta_\mathcal{T}$.

Proposition F.12. *Along every solution of the gradient flow $\dot\theta = -\nabla\mathcal{L}(\theta)$, the* balancedness defect

$$\Delta_e(\theta) := (\operatorname{mat}_e W_u)^\top \operatorname{mat}_e W_u - (\operatorname{mat}_e W_v)^\top \operatorname{mat}_e W_v \ \in \ \mathbb{R}^{r_e \times r_e}$$

is constant in time, at every internal edge $e = \{u, v\}$.

Proof. Fix an internal edge $e = \{u, v\}$. Write $A := \operatorname{mat}_e W_u, B := \operatorname{mat}_e W_v$, with the e-mode as the column index in both matricizations. Then $\Delta_e = A^\top A - B^\top B$. Matricization is a linear isometry, so gradient flow in these variables is obviously $\dot A = -\nabla_A \mathcal{L},\ \dot B = -\nabla_B \mathcal{L}$, where the gradients are

taken with all other parameters fixed. With this convention, contraction across e pairs the two column indices, so the local contraction has the form $AB^\top$. Hence the gauge change $A \mapsto AS$, $B \mapsto BS^{-\top}$ preserves the represented tensor for every $S \in \mathrm{GL}(r_e)$, and therefore preserves $\mathcal{L}$:

$$\mathcal{L}(AS, BS^{-\top}, \ldots) = \mathcal{L}(A, B, \ldots). \tag{6}$$

Differentiate Equation (6) along the one-parameter family $S = I + tX$, where $X \in \mathbb{R}^{r_e \times r_e}$ is arbitrary. Since

$$\frac{d}{dt}\Big|_{t=0} A(I + tX) = AX, \qquad \frac{d}{dt}\Big|_{t=0} B(I + tX)^{-\top} = -BX^\top,$$

we obtain

$$\langle \nabla_A \mathcal{L}, AX \rangle_F - \langle \nabla_B \mathcal{L}, BX^\top \rangle_F = 0.$$

Using Frobenius trace identities on both terms, we rewrite this as

$$\langle A^\top \nabla_A \mathcal{L}, X \rangle_F - \langle (\nabla_B \mathcal{L})^\top B, X \rangle_F = 0.$$

Since X is arbitrary, this forces

$$A^\top \nabla_A \mathcal{L} = (\nabla_B \mathcal{L})^\top B, \tag{7}$$

and taking transposes gives the equivalent identity

$$(\nabla_A \mathcal{L})^\top A = B^\top \nabla_B \mathcal{L}. \tag{8}$$

By the product rule,

$$\frac{d}{dt}\Delta_e = \frac{d}{dt}(A^\top A - B^\top B) = -(\nabla_A \mathcal{L})^\top A - A^\top \nabla_A \mathcal{L} + (\nabla_B \mathcal{L})^\top B + B^\top \nabla_B \mathcal{L}.$$

The first and fourth terms cancel by Equation (8), and the second and third terms cancel by Equation (7). Hence $\dot{\Delta}_e = 0$. □

Proposition 4.2. *Let $\theta(t)$ be a solution to the gradient flow equation $\dot{\theta}(t) = -\nabla \mathcal{L}(\theta(t))$. If $\theta(0)$ is a minimum-norm point, then $\theta(t)$ is a minimum-norm point for all t, and so is every limit point of $\theta(t)$.*

Proof. By Proposition F.10, minimum norm is equivalent to balancedness, so $\theta(0)$ is balanced, i.e. $\Delta_e(\theta(0)) = 0$ at every internal edge. By Proposition F.12, $\Delta_e(\theta(t)) = 0$ for all t, and by continuity of Δ_e the same holds at every limit point of $\theta(t)$. By Proposition F.10 again, all these balanced points are minimum norm. □

Thus, if the network is initialized to a minimum-norm point in parameter space, it will remain minimum norm under gradient flow. This is an instance of a broader phenomenon: conservation laws arising from parameterization symmetries have been studied systematically in deep learning (Kunin et al., 2020; Marcotte et al., 2023).

In fact, even if we do not initialize at a minimum-norm point, we show that we are driven exponentially toward one by the use of regularization (Corollary F.13). Note this does not contradict Proposition F.12, since regularization depends on the actual parameter values θ rather than just the contracted tensor $T(\theta)$.

Corollary F.13. *Fix $\lambda > 0$ and consider gradient flow of the regularized loss $\mathcal{L} + \frac{\lambda}{2}\|\cdot\|^2$, that is, $\dot{\theta} = -\nabla \mathcal{L}(\theta) - \lambda\theta$. Along every solution, each balancedness defect satisfies $\dot{\Delta}_e = -2\lambda\,\Delta_e$, so $\Delta_e(t) = e^{-2\lambda t}\,\Delta_e(0)$. In particular, every limit point of $\theta(t)$ as $t \to \infty$, if any, is balanced — hence minimum norm — regardless of initialization.*

Proof. Fix $e = \{u, v\}$ and write $A = \mathrm{mat}_e\, W_u$, $B = \mathrm{mat}_e\, W_v$, so $\Delta_e = A^\top A - B^\top B$. Under the regularized flow,

$$\dot{A} = -\nabla_A \mathcal{L} - \lambda A, \qquad \dot{B} = -\nabla_B \mathcal{L} - \lambda B. \tag{9}$$

Also, the product rule gives

$$\dot{\Delta}_e = \dot{A}^\top A + A^\top \dot{A} - \dot{B}^\top B - B^\top \dot{B}. \tag{10}$$

Substituting Equation (9) into Equation (10) and grouping the unregularized and regularization terms gives

$$\dot{\Delta}_e = \left[-(\nabla_A \mathcal{L})^\top A - A^\top \nabla_A \mathcal{L} + (\nabla_B \mathcal{L})^\top B + B^\top \nabla_B \mathcal{L}\right] + \left[(-\lambda A)^\top A + A^\top(-\lambda A) - (-\lambda B)^\top B - B^\top(-\lambda B)\right].$$

The first bracket is zero by Proposition F.12. The second bracket is

$$\left[(-\lambda A)^\top A + A^\top(-\lambda A) - (-\lambda B)^\top B - B^\top(-\lambda B)\right] = -2\lambda A^\top A + 2\lambda B^\top B = -2\lambda \Delta_e.$$

Therefore $\dot{\Delta}_e = -2\lambda \Delta_e$, and solving this linear equation gives $\Delta_e(t) = e^{-2\lambda t}\Delta_e(0)$. At any limit point as $t \to \infty$, every Δ_e vanishes by continuity, so the point is balanced. By Proposition F.10 balanced is equivalent to minimum norm, so we have the desired conclusion. □

Similar results are known in the literature. Kunin et al. (2020) show similar regularization effects for conserved quantities associated with scale symmetries in neural networks, and Lindsey & Menon (2025) show that regularization drives the flow to the balanced manifold in deep linear networks.

Remark F.14. The proof of Proposition F.12 uses only the gauge invariance of $\mathcal{L}$ at the edge e, making no reference to the tree structure, so the conservation law holds on tensor networks of arbitrary topology. The tree structure enters only through Proposition F.10 in the proof of Proposition 4.2: on a general network, a balanced initialization still remains balanced under gradient flow, but balancedness is no longer necessarily equivalent to minimum norm. See also Remark F.11.

G A Spurious Local Minimum Without the Minimum-Norm Assumption

This appendix establishes that the minimum-norm assumption in Theorem 4.3 is necessary: without it, realizable TTNs can have spurious local minima. We construct an explicit example on a 3-leaf star, equivalently a Tucker decomposition, with bond dimensions equal to the target edge ranks. The example is not minimum norm, and it is a local minimum of the squared loss.[9] A related counterexample appears in Frandsen & Ge (2022). The example necessarily uses a multilinear interaction of order at least three: when all tensors have order two, one obtains deep linear networks, which have no spurious local minima for the square loss (Baldi & Hornik, 1989; Kawaguchi, 2016; Lu & Kawaguchi, 2017).

G.1 Setup

Consider a 3-leaf star TTN with core tensor $G \in \mathbb{R}^{3\times3\times3}$ and leaf matrices $A, B, C \in \mathbb{R}^{3\times3}$. The represented tensor is

$$T = \sum_{i,j,k=0}^{2} G_{ijk}\, A_{:i} \otimes B_{:j} \otimes C_{:k} \in \mathbb{R}^3 \otimes \mathbb{R}^3 \otimes \mathbb{R}^3,$$

where $A_{:i}$ denotes the i-th column of A, and similarly for B, C. The target is the diagonal tensor

$$T^* = e_{000} + e_{111} + e_{222},$$

where e_0, e_1, e_2 is the standard basis of $\mathbb{R}^3$ and $e_{abc} := e_a \otimes e_b \otimes e_c$. Every matricization of T^* has rank 3, so T^* is realizable at tight capacity $r_e = 3$ on every edge.

Consider the parameter point

$$A^0 = B^0 = C^0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

and

$$G^0 = e_{000} + e_{111} + e_{012} + e_{021} + e_{102} + e_{120} + e_{201} + e_{210}.$$

[9] Recall from Appendix B that throughout the appendices $\mathcal{L}$ denotes the un-normalized loss $\frac{1}{2}\|T(\theta) - T^*\|_F^2$; the explicit loss values below use this convention.

Since the third columns of A^0, B^0, C^0 vanish, every core term with an index equal to 2 drops out of the contraction, and therefore

$$T(\theta_0) = T_0 = e_{000} + e_{111}, \qquad T_0 - T^* = -e_{222}, \qquad \mathcal{L}(\theta_0) = \tfrac{1}{2}.$$

This point is not minimum norm. Indeed, the same output T_0 is represented by

$$A' = B' = C' = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad G' = e_{000} + e_{111},$$

for which $\|\theta'\|^2 = 8 < 14 = \|\theta_0\|^2$.

Theorem G.1. *The point $\theta_0 = (G^0, A^0, B^0, C^0)$ is a local minimum of $\mathcal{L}$: there exists a neighborhood of θ_0 on which $\mathcal{L} \geq \mathcal{L}(\theta_0) = \frac{1}{2}$. Since $T(\theta_0) \neq T^*$, this local minimum is spurious.*

Proof. Let $T = T(\theta)$ and write $\Delta T := T - T_0$. The loss difference is

$$\mathcal{L}(\theta) - \mathcal{L}(\theta_0) = \langle \Delta T, T_0 - T^* \rangle + \tfrac{1}{2}\|\Delta T\|_F^2.$$

Since $T_0 - T^* = -e_{222}$, the alignment term $\langle \Delta T, T_0 - T^* \rangle$ is exactly $-T_{222}$. This is the only term in the expansion that can be negative; the norm term $\|\Delta T\|_F^2$ is nonnegative. Thus any loss decrease must come from making T_{222} positive. However, we will show that attempting to make T_{222} positive forces an increase in the norm term, which dominates any possible decrease from the alignment term.

Precisely, let

$$S := \sum_{(i,j,k) \notin \{(0,0,0),(1,1,1),(2,2,2)\}} T_{ijk}^2,$$

the squared norm of the target-zero coefficients. It suffices to prove that, after restricting to a sufficiently small neighborhood of θ_0,

$$|T_{222}| \leq C S^{3/2}$$

for some constant $C > 0$, since then $\|\Delta T\|_F^2 \geq S$ and

$$\mathcal{L}(\theta) - \mathcal{L}(\theta_0) \geq \tfrac{1}{2} S - C S^{3/2}.$$

For a 3×3 matrix A, write $u^{(a)} \in \mathbb{R}^3$ for its a-th row, so $u_i^{(a)} = A_{ai}$, and define $v^{(b)}, w^{(c)}$ analogously for B, C. Let

$$F(x, y, z) := \sum_{i,j,k=0}^{2} G_{ijk} x_i y_j z_k.$$

Then $T_{abc} = F(u^{(a)}, v^{(b)}, w^{(c)})$. At the base point,

$$u^{(0)} = v^{(0)} = w^{(0)} = e_0, \qquad u^{(1)} = v^{(1)} = w^{(1)} = e_1, \qquad u^{(2)} = v^{(2)} = w^{(2)} = 0.$$

For a nearby parameter point, set

$$a := u^{(2)}, \qquad b := v^{(2)}, \qquad c := w^{(2)}.$$

We first prove that S controls the dormant rows a, b, c. Consider

$$\Phi_A(a) := \big(F(a, v^{(0)}, w^{(0)}), F(a, v^{(1)}, w^{(1)}), F(a, v^{(0)}, w^{(1)}), F(a, v^{(1)}, w^{(0)})\big).$$

These four components are $T_{200}, T_{211}, T_{201}, T_{210}$. At θ_0,

$$\Phi_A(a_0, a_1, a_2) = (a_0, a_1, a_2, a_2),$$

so $\|\Phi_A(a)\|^2 \geq \|a\|^2$ at the base point. Since Φ_A depends continuously on (G, B, C), after shrinking to a neighborhood $\mathcal{N}$ of θ_0 there is $\sigma > 0$ such that

$$T_{200}^2 + T_{211}^2 + T_{201}^2 + T_{210}^2 \geq \sigma^2 \|a\|^2$$

on $\mathcal{N}$. The same argument with the roles of A, B, C permuted gives analogous bounds for $\|b\|$ and $\|c\|$, using coefficients also included in S. Thus, shrinking $\mathcal{N}$ if necessary,

$$S \geq \sigma^2(\|a\|^2 + \|b\|^2 + \|c\|^2)$$

for all $\theta \in \mathcal{N}$.

Next, $T_{222} = F(a, b, c)$ is trilinear. On a sufficiently small neighborhood there is a constant $M > 0$ such that

$$|F(x, y, z)| \leq M\|x\|\|y\|\|z\| \qquad \text{for all } x, y, z \in \mathbb{R}^3.$$

Let $m^2 := \|a\|^2 + \|b\|^2 + \|c\|^2$. By AM–GM,

$$\|a\|\|b\|\|c\| \leq \left(\frac{m^2}{3}\right)^{3/2}.$$

Since $m^2 \leq \sigma^{-2}S$, there is a constant $C > 0$ such that

$$|T_{222}| \leq CS^{3/2}$$

throughout $\mathcal{N}$.

Shrinking $\mathcal{N}$ once more, we may assume $CS^{1/2} \leq \frac{1}{4}$ on $\mathcal{N}$. Hence

$$\mathcal{L} - \tfrac{1}{2} \geq \tfrac{1}{4}S \geq 0.$$

Thus θ_0 is a local minimum. The minimum is spurious because $T(\theta_0) = T_0 \neq T^*$. □

H THE PARITY EXAMPLE

In this appendix we prove Proposition 5.3.

Recall the setup of Section 5.2, including the parity target $T^* = \mathbf{1}[\, y = x_1 \oplus \cdots \oplus x_n \,]$, the binary tree tensor network architecture $T(\theta)$, and the parameter point $\theta_0 \in \Theta$. We assume $n \geq 4$ throughout. Also recall from the notation in Appendix B that throughout the appendices $\mathcal{L}$ denotes the un-normalized loss $\frac{1}{2}\|T(\theta) - T^*\|_F^2$.

Let $u_+ = (1, 1)$ and $u_- = (1, -1)$, the orthogonal Hadamard basis of $\mathbb{R}^2$. The following Lemma H.1 shows that the model and target are diagonal in Hadamard coordinates.

Lemma H.1. *It holds that:*

1. $W_v(\theta_0) = \frac{1}{2}u_+^{\otimes 3}$ *for every node* $v \in V$*;*
2. $T(\theta_0) = \frac{1}{2}u_+^{\otimes(n+1)}$*;*
3. $T^* = \frac{1}{2}\big(u_+^{\otimes(n+1)} + u_-^{\otimes(n+1)}\big)$.

Consequently,

$$R_0 := T(\theta_0) - T^* = -\tfrac{1}{2}u_-^{\otimes(n+1)}.$$

Proof. The tensor $W_v(\theta_0)$ has every entry equal to $1/2$, while $u_+^{\otimes 3}$ has every entry equal to 1, proving (1). Since the tree has $n - 1$ nodes and $n - 2$ internal edges, a direct calculation gives

$$T(\theta_0) = \big(\tfrac{1}{2}\big)^{n-1}2^{n-2}u_+^{\otimes(n+1)} = \tfrac{1}{2}u_+^{\otimes(n+1)},$$

proving (2).

Since $(u_+)_b = 1$ and $(u_-)_b = (-1)^b$ for $b \in \{0, 1\}$,

$$\begin{aligned}\Big[\tfrac{1}{2}\big(u_+^{\otimes(n+1)} + u_-^{\otimes(n+1)}\big)\Big]_{x_1,\ldots,x_n,y} &= \tfrac{1}{2}\left(\prod_{i=1}^{n}(u_+)_{x_i}(u_+)_y + \prod_{i=1}^{n}(u_-)_{x_i}(u_-)_y\right)\\ &= \tfrac{1}{2}\left(1 + (-1)^{x_1+\cdots+x_n+y}\right)\\ &= \mathbf{1}[\, y = x_1 \oplus \cdots \oplus x_n \,].\end{aligned}$$

This is the definition of the parity target T^*, proving (3). □

Lemma H.2. *There exist constants $C, \rho > 0$ such that, for all $\theta \in \Theta$ with $\|\theta - \theta_0\| \le \rho$,*

$$\big|\langle R_0,\ T(\theta) - T(\theta_0)\rangle\big| \ \le\ C\|\theta - \theta_0\|^{n/2+1}.$$

Proof. Set $\delta W_v := W_v(\theta) - W_v(\theta_0)$. For $S \subseteq V$, define θ^S by

$$W_v(\theta^S) = \begin{cases} \delta W_v, & v \in S, \\ W_v(\theta_0), & v \notin S. \end{cases}$$

By multilinearity of the TTN contraction,

$$T(\theta) - T(\theta_0) \ =\ \sum_{S \subseteq V, S \neq \emptyset} T(\theta^S).$$

Then by Lemma H.1, we have

$$\langle R_0, T(\theta) - T(\theta_0)\rangle \ =\ -\tfrac{1}{2} \sum_{S \subseteq V, S \neq \emptyset} \langle u_-^{\otimes(n+1)}, T(\theta^S)\rangle. \tag{11}$$

We will show that the only nonzero terms in this sum are those where S includes every leaf and root tensor. Suppose first that an input leaf $\ell \in V$ lies outside S. Then by first performing all contractions away from ℓ, there exists a vector $a \in \mathbb{R}^2$ such that

$$\langle u_-^{\otimes(n+1)}, T(\theta^S)\rangle = \big\langle W_\ell(\theta_0), u_- \otimes u_- \otimes a\big\rangle = \tfrac{1}{2}\langle u_+, u_-\rangle^2 \langle u_+, a\rangle = 0.$$

Similarly, if the root $r \in V$ lies outside S, there exists $a, b \in \mathbb{R}^2$ such that

$$\langle u_-^{\otimes(n+1)}, T(\theta^S)\rangle = \big\langle W_r(\theta_0), a \otimes b \otimes u_-\big\rangle = \tfrac{1}{2}\langle u_+, a\rangle\langle u_+, b\rangle\langle u_+, u_-\rangle = 0.$$

Thus $\langle u_-^{\otimes(n+1)}, T(\theta^S)\rangle = 0$ for any term of Equation (11) where S does not contain every input leaf and the root.

There are $n/2$ input leaves and one root, so every nonvanishing subset has size $|S| \ge n/2 + 1$. For each such S, the scalar $\big\langle R_0, T(\theta^S)\big\rangle$ is multilinear in the tensors $\delta W_v : v \in S$, with all other node tensors fixed. Since there are only finitely many subsets S in Equation (11), there is a constant C' independent of S such that, for all θ sufficiently close to θ_0,

$$\big|\big\langle R_0, T(\theta^S)\big\rangle\big| \le C' \prod_{v \in S} \|\delta W_v\|_F \le C'\|\theta - \theta_0\|^{|S|}.$$

Shrinking ρ if necessary so that $\|\theta - \theta_0\| \le 1$, each surviving term in Equation (11) is bounded by $C'\|\theta - \theta_0\|^{n/2+1}$. Summing over the finitely many surviving subsets gives the claim. □

Proposition 5.3. *The point θ_0 is a critical point of $\mathcal{L}$, and a saddle of order at least $n/2$. In particular, there exist constants $C, \rho > 0$ such that*

$$\mathcal{L}(\theta) - \mathcal{L}(\theta_0) \ \ge\ -C\,\|\theta - \theta_0\|^{n/2+1} \qquad \textit{for all } \theta \in \Theta \textit{ such that } \|\theta - \theta_0\| \le \rho.$$

Proof. Write $\Delta T := T(\theta) - T(\theta_0)$. Since

$$\mathcal{L}(\theta) - \mathcal{L}(\theta_0) = \langle R_0, \Delta T\rangle + \tfrac{1}{2}\|\Delta T\|_F^2,$$

Lemma H.2 gives, for θ sufficiently close to θ_0,

$$\mathcal{L}(\theta) - \mathcal{L}(\theta_0) \ge -C\|\theta - \theta_0\|^{n/2+1}, \tag{12}$$

after possibly increasing C. This is the asserted order bound.

Since $n/2 + 1 > 1$, criticality follows from Equation (12) and the differentiability of $\mathcal{L}(\theta)$. □

Remark H.3. The bound is likely not tight. We conjecture the saddle order is in fact $n - 2$, which we have checked numerically at $n \in \{4, 8\}$. The exact order is not important for the present paper, and we leave this to future work.

I Implications for Learning With Stochastic Gradient Methods

The loss landscape results in the main body concern the geometry of the *population* loss $\mathcal{L}(\theta) = \frac{1}{2|\mathcal{X}|}\|T(\theta) - T^*\|_F^2$. However, the population loss is a theoretical quantity. In practice, one does not know the target T^*, and instead only has access to a finite number of samples (x_i, y_i) from the target. Instead of population gradient descent, one typically learns via a form of stochastic gradient descent using the per-sample losses $\ell(x_i, y_i; \theta)$.

The relevance of the population loss to stochastic gradient methods is standard and well-known (Robbins & Monro, 1951; Bottou et al., 2018), but we explain this perspective here in our setting for the interested reader. We show how minibatch SGD can be viewed, under standard approximations, as noisy gradient descent on this population loss (Appendix I.1). We then sketch how local minima (Appendix I.2) and saddle points (Appendix I.3) can affect SGD.

I.1 Setup

Recall that

$$\mathbb{E}_{(x,y)}[\ell(x, y; \theta)] \;=\; \mathcal{L}(\theta), \qquad \mathbb{E}_{(x,y)}[\nabla \ell(x, y; \theta)] \;=\; \nabla \mathcal{L}(\theta),$$

so the per-sample losses and gradients are unbiased noisy estimates of $\mathcal{L}(\theta)$ and $\nabla \mathcal{L}(\theta)$.

Stochastic gradient descent with step size τ and minibatches of size m can then be written as the discrete-time stochastic process

$$\theta_{t+1} \;=\; \theta_t \;-\; \tau\, \nabla \mathcal{L}(\theta_t) \;-\; \tau\, \xi_t(\theta_t),$$

where $\xi_t(\theta) \;=\; \frac{1}{m}\sum_i \nabla_\theta \ell(x_i, y_i; \theta) \;-\; \nabla \mathcal{L}(\theta)$ is a mean-zero noise term, independent across t conditional on θ_t, with covariance scaling as $1/m$.

Note that this is a sum of a population gradient term $\tau\, \nabla \mathcal{L}(\theta_t)$ and a noise term $\tau\, \xi_t(\theta_t)$; the only difference between SGD and population gradient descent on $\mathcal{L}$ is the noise term. Thus SGD acts like noisy gradient descent on the population loss $\mathcal{L}$.

Up to this point we have not made any approximations or new assumptions; we now proceed to make several standard assumptions to make analysis tractable. First, it is useful to take a continuous-time limit, which approximates SGD via the stochastic differential equation:

$$d\theta_t = -\nabla \mathcal{L}(\theta_t) dt + (\tau\, \Sigma(\theta_t))^{\frac{1}{2}}\, dW_t$$

where dW_t is the Wiener process (Brownian motion) and $\Sigma : \Theta \to \Theta \times \Theta$ is a position-dependent covariance matrix (Li et al., 2017). Note that we have assumed that the noise process is Brownian, which follows from the central limit theorem if the batch size is large and gradient noise is finite variance, but in some regimes this noise process can be more heavy-tailed (Şimşekli et al., 2019; Hodgkinson & Mahoney, 2021), a complication we temporarily put aside for ease of discussion.

Second, we make the assumption that the noise covariance is isotropic and homogeneous: $\Sigma(\theta_t) = \sigma I$ for some $\sigma \in \mathbb{R}$. This assumption is unrealistic, but it significantly eases discussion without fundamentally changing the qualitative conclusions.

Under these assumptions, SGD is modeled by the overdamped Langevin equation

$$d\theta_t = -\nabla \mathcal{L}(\theta_t)\, dt + \sqrt{2\varepsilon}\, dW_t,$$

where the effective noise scale is $\varepsilon = \tau\sigma/2$. Thus the stochastic process still "sees" the same potential function $\mathcal{L}$: the population gradient determines the deterministic drift, while the Brownian term allows occasional moves against the gradient.

I.2 Benign Loss Landscapes Help SGD

This viewpoint makes clear why suboptimal local minima are a serious obstruction for SGD, even though SGD is noisy. Suppose θ_0 is a strict local minimum of $\mathcal{L}$, but not a global minimum, and let B denote its basin of attraction under gradient flow. To leave this basin, the process must cross the

boundary of B. The relevant quantity is therefore the loss barrier[10].

$$\Delta_B := \inf_{\theta \in \partial B} \mathcal{L}(\theta) - \mathcal{L}(\theta_0).$$

If $\Delta_B > 0$, then escaping the basin requires the stochastic dynamics to move through points of higher loss. In the small-noise regime, this is a rare event: the classical Eyring–Kramers law (Eyring, 1935; Kramers, 1940) predicts exit times which scale, to leading exponential order, like

$$\mathbb{E}[t_{\text{exit}}] \asymp \exp\left(\frac{\Delta_B}{\varepsilon}\right),$$

with problem-dependent prefactors determined by the local geometry near the minimum and the geometry of the basin boundary (Berglund & Gentz, 2008). In terms of SGD, this means that a suboptimal basin whose loss barrier is large relative to the effective noise scale can trap the algorithm for extremely long times. Increasing the batch size or decreasing the step size reduces the noise scale and makes the dynamics closer to population gradient flow, so such barriers become even harder to cross.

This is the basic sense in which a benign population loss landscape helps SGD. If the population loss has a genuine suboptimal local minimum, then deterministic gradient flow is trapped, and SGD can escape only by a sufficiently large noise fluctuation, which can become vanishingly unlikely if the loss barrier is large. But if the population loss landscape has no suboptimal local minima, or at least admits non-increasing escape paths to lower loss, then this particular failure mode is absent.

Of course, one can instead increase stochasticity in SGD, for instance by increasing the step size, decreasing the batch size, or considering noise models beyond the Brownian approximation. Heavy-tailed gradient noise, in particular, can lead to qualitatively different escape behavior from local minima (Şimşekli et al., 2019; Nguyen et al., 2019). But this is not a substitute for a benign population landscape. Large stochastic fluctuations can also move the parameters to regions of higher loss, and increasing the noise scale weakens the relative influence of the population gradient. Thus noise can help escape a bad basin, but it is a costly and indirect mechanism; we may need it sometimes nevertheless[11], but it is preferable not to.

Thus it is much better, from the perspective of optimization, for the population landscape not to contain suboptimal wells in the first place. This is why our benign loss-landscape results for $\mathcal{L}$ are not just pure geometry statements but actually relevant for the behavior and efficiency of SGD in TTNs.

I.3 Degenerate Saddle Points Can Still Slow SGD

The preceding discussion explains why the absence of suboptimal local minima is useful for SGD: it removes energetic barriers. However, it does not imply that SGD makes rapid progress. A landscape can have no bad local minima while still containing saddle points whose escaping directions are difficult to find or exploit.

The important distinction is between nondegenerate and degenerate saddles. At a strict saddle, the Hessian has a negative eigenvalue. Equivalently, in suitable local coordinates there is a direction v such that

$$\mathcal{L}(\theta_0 + sv) - \mathcal{L}(\theta_0) = -\frac{\lambda}{2}s^2 + O(s^3)$$

with $\lambda > 0$. Along this direction, the linearization of gradient flow is unstable: a small displacement in the v-direction grows approximately like $e^{\lambda t}$. Thus, once the dynamics acquire even a small component in the unstable direction, the saddle can be escaped in time logarithmic in the inverse size of that component. This is the regime captured by much of the saddle-escape literature: explicit perturbations or stochastic gradients can seed a component in a negative-curvature direction, after which the deterministic drift amplifies it (Jin et al., 2017; Daneshmand et al., 2018).

[10] In the case where the noise covariance is not isotropic and homogeneous, the relevant quantity becomes the Freidlin-Wentzell quasipotential rather than the loss (Freidlin & Wentzell, 1998)

[11] For instance, our results rely on several idealized assumptions (Appendix D.7) - if in practice the loss landscape is not exactly benign but instead approximately benign, stochasticity may be able to make up the difference.

This means that ordinary strict saddles are not a plausible mechanism for worst-case hardness, at least when the negative curvature is bounded below uniformly. A degenerate saddle (such as that exhibited by parity in Section 5.2) is different, however. At such a point, there may be no useful negative-curvature direction at quadratic order. Descent may appear only through higher-order coordinated perturbations. A local form such as

$$\mathcal{L}(\theta_0 + sv) - \mathcal{L}(\theta_0) \approx -cs^k, \qquad k > 2,$$

illustrates the change. The corresponding gradient-flow equation along this coordinate has the form

$$\dot{s} \approx cks^{k-1}.$$

Starting from a small seed $s_0 > 0$, the time needed to reach an $O(1)$ distance is no longer logarithmic in $1/s_0$; for $k > 2$, it scales like a power of $1/s_0$. The loss improvement $-cs^k$ is small, and the vector field itself is weak near the saddle. Higher order means that small perturbations are amplified much more slowly.

This is the sense in which degenerate saddles can reconcile benign landscapes with computational hardness. A benign landscape can rule out suboptimal wells: there may always exist a non-increasing path to lower loss. But the beginning of that path may require a coordinated higher-order move, and the gradient signal toward that move may vanish to high order. SGD is then not blocked by an energetic barrier; rather, it can be slowed by a need for coordination and the resulting loss flatness.

Note that in the presence of stochastic noise, high-order degeneracy may be necessary but not sufficient for optimization difficulty, and one may also need the escape path to be sufficiently unlikely to be found by chance. One must return to the classical Eyring–Kramers picture discussed in Appendix I.2. For metastable Langevin dynamics near a local minimum, the leading exponential contribution to the exit time comes from the loss barrier. When the minimum is instead a saddle point there is no loss barrier, and this exponential obstruction is absent. However, the local geometry near the saddle still affects the prefactor, and nonquadratic saddles require modified Eyring–Kramers laws (Berglund & Gentz, 2008).

Thus the absence of a loss barrier does not by itself imply that escape is geometrically easy; degenerate saddles can still change the relevant time scale through the prefactor term. We believe a rigorous analysis of the Eyring–Kramers prefactor term in such saddle points, extending Berglund & Gentz (2008), to be an interesting direction for future work.

J AUTOFORMALIZATION IN LEAN

Formalizing mathematics in proof assistants such as LEAN 4 (de Moura & Ullrich, 2021; mathlib Community, 2020) has traditionally required substantial expertise both in the underlying mathematics and in the proof assistant itself. Recent advances in large language model capabilities have made formal verification of research-level mathematics increasingly practical (see e.g. (Wu et al., 2022; Ilin, 2026)). In this appendix, we propose that suitable theoretical results in machine learning, particularly self-contained results about tractable toy models, should increasingly be accompanied by a machine-checked LEAN formalization.

We formalized Theorem 4.3 and Theorem 5.2, the two results most central to the paper's loss landscape claims. We selected these results both for their mathematical importance and because their dependencies could be developed using just the existing Mathlib infrastructure. A clickable LEAN logo in the right margin appears beside formalized statements. Appendix J.1 describes the autoformalisation and verification workflow; Appendix J.2 explains the paper-to-LEAN correspondence and differences in proof organisation; and Appendix J.3 argues more broadly for the adoption of LEAN certificates for suitable theoretical research in machine learning. The supplementary library is available in the companion GitHub repository[12].

J.1 DEVELOPMENT AND VERIFICATION

The authors specified the target results, and the formalization was generated and reviewed using Anthropic's Claude Opus 4.8 and Claude Fable 5, and OpenAI's GPT-5.5 via subscriptions. We do

[12] https://anonymous.4open.science/r/ttn-loss-landscape-EC57

```
theorem minNorm_isLocalMin_isGlobalMin
    {Tstar : a.Ext → ℝ}
    (hreal : a.Realizable Tstar)
    {θ : a.Param}
    (hmn : a.MinNorm θ)
    (hloc : IsLocalMin (a.loss Tstar) θ) :
    a.IsGlobalMin Tstar θ ∧
      a.loss Tstar θ = 0
```

```
theorem not_fullRank_of_critical_of_not_isGlobalMin
    {Tstar : a.Ext → ℝ}
    (hreal : a.Realizable Tstar)
    {θ : a.Param}
    (hcrit : a.Critical Tstar θ)
    (hng : ¬ a.IsGlobalMin Tstar θ) :
    ¬ a.FullTuckerRank θ
```

Figure 5: Formal statements of the two main results, with proof bodies omitted. Top: Theorem 4.3, stating that a minimum-norm local minimum for a realizable target is global and has zero loss. Bottom: Theorem 5.2, stating that a critical point that is not a global minimum cannot have full Tucker rank.

not report an API-equivalent cost because complete records of token usage, elapsed time, and cost were not retained.

The library uses LEAN 4.31.0 and Mathlib 4.31.0. The files contain no `sorry`, `admit`, `native_decide`, or project `axiom` declarations. The proofs of Theorems 4.3 and 5.2 are checked with Comparator (Lean FRO, 2026). A separate `Challenge.lean` file states the target theorems without importing the modules that prove them. Comparator checks that these statements agree with the production statements, including the definitions they reference, verifies that the proofs use only explicitly permitted axioms, and replays the proof terms through the LEAN kernel. These checks guard against the substitution of weaker statements and the introduction of hidden axioms.

Still, Comparator cannot establish whether the formal statements faithfully express the manuscript's claims. The authors therefore reviewed the statements of the two main results, including the relevant definitions and hypotheses, and affirm their correspondence to the manuscript. This review was at the theorem level: it did not systematically match intermediate results or proof steps to LEAN declarations, nor did it include a line-by-line audit of the generated proof scripts.

J.2 PAPER-TO-LEAN CORRESPONDENCE AND PROOF ORGANIZATION

The formalization focuses on Theorems 4.3 and 5.2, whose LEAN statements appear in Figure 5, with margin markers linking to the corresponding code. It preserves the manuscript's definitions and assumptions, including realizability and minimum norm, but differs in the following representations and proof steps:

- **Contraction coordinates.** The formalization defines the represented tensor directly as a sum over internal bond indices of products of node-tensor entries, implementing the undirected contraction description of Definition A.5. The correspondence with the directed multilinear-map definition of Definition 3.1 is explained mathematically in Appendix A; this correspondence is not itself formalized in LEAN.
- **External modes.** LEAN combines the external modes at each node into a single finite index type. The theorem `representedLit_extEquiv` proves that contraction with individual external modes agrees with this aggregated representation under reindexing.
- **Criticality.** The predicate `Critical` is defined using Mathlib's `HasFDerivAt`, requiring the loss to have zero Fréchet derivative at the parameter point. The theorem `critical_iff_nodewiseCritical` proves that this is equivalent to the vanishing of the first-order loss variation with respect to each node tensor, the characterization used in the algebraic proofs.

- **The lemma used in Theorem 5.2.** The manuscript's Lemma C.1 establishes full column rank for both contraction factors at any internal edge. The LEAN proof uses only the leaf-edge case: `Hfun_full_col_rank` proves that the factor obtained by contracting the rest of the tree has full column rank. This suffices for the induction by leaf removal.

Not all statements were chosen to be included in the formalization. For example, we did not include Corollary 4.4 in the formalization. Its proof relies on results from real algebraic geometry and geometric invariant theory (see Appendices E and F) whose formalization lies outside our scope. We instead focused on the two main results, whose proofs mostly use linear algebra and some graph theory that can be formalized using existing Mathlib infrastructure.

The formalization process did not uncover any major flaws with the proof, but it did help expose and fix one particular edge case. Specifically, the case where $T^* = 0$ presented an issue for the reduction argument of Appendix D.3, which replaces each bond dimension r_e with $\mathrm{rank}(T^{*(e)})$. When $T^* = 0$, these ranks are zero, so the compressed network would violate the requirement of Definition 3.1 that bond dimensions be positive. The argument in Appendix D.3 now includes an explicit argument to handle the $T^* = 0$ case.[13]

J.3 Formalization as a Verification Check for Toy Models

Theoretical machine learning often uses simplified models to isolate mechanisms that are difficult to study in full neural networks. Results about these models inform our understanding of optimization, generalization, and the effects of architectural assumptions. Several such results have been formally verified, including an expressiveness theorem for convolutional arithmetic circuits (Bentkamp et al., 2019), convergence results for Hopfield networks and Boltzmann machines (Cipollina et al., 2025), and a conversion of networks with piecewise-affine activations into piecewise-affine functions (Aleksandrov & Völlinger, 2023). These projects also provide infrastructure for further work. For example, Bentkamp et al. (2019) developed a reusable tensor library and simplified and generalized the expressiveness proof during formalization.

For research built around a toy model, this reuse is particularly valuable. Results about expressiveness, optimization, and learning dynamics can share formal definitions of the architecture and its parameters, allowing assumptions to be compared directly and proofs to build on previously checked lemmas. In the present library, the two main theorems share definitions of the TTN architecture, contraction, loss, and rank conditions. These definitions remain available for further analysis of the model.

Furthermore, recent advances in language models and agent workflows support increasingly autonomous formalization (Zhang et al., 2026b; Milikic et al., 2026). Authors can specify the target results and delegate the implementation to agents, while retaining responsibility for checking that the formal statements express the intended mathematical claims. This was the division of work in our project. For results whose dependencies are available in existing libraries, we therefore propose including machine-checked proofs as supplementary material.

Of course, verification can only establish the mathematical consequences of a toy model's assumptions; whether those assumptions meaningfully capture phenomena of interest remains an empirical question.

[13] In the LEAN code, this case is treated directly in the proof of Theorem 4.3.