WE SHOULD UNDERSTAND SYMMETRY BREAKING IN SPARSE TRAINING TO UNMASK THE ROLE OF OVERPARAMETERIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Overparameterization is a cornerstone of deep learning. When combined with weight-space symmetries, it leads to many equivalent solutions in the loss landscape, making it easier for an optimizer to find a good solution. However, breaking these symmetries — as in Sparse Neural Networks (SNNs) — reveals fundamental limitations in our training algorithms, present in both sparse and dense settings. Although in the literature sparse training research has mostly been motivated by making training efficient, in this position paper, we argue that sparse training with a fixed mask provides a valuable testbed for studying symmetry, optimization, and their interplay in a principled manner. We explore novel examples of how breaking permutation symmetry and rescaling invariance jointly affect optimization, and can inspire algorithmic advancements. Sparse training research can provide valuable insights into the role of symmetries and overparameterization in training in general, not limited to sparse training. This position paper argues that understanding how symmetry breaking can improve or hinder training could uncover deeper theoretical insights and inform the design of more robust optimizers and architectures that respect symmetries.

1 INTRODUCTION

With the exponential growth in neural network models, the development of more efficient training methods is of paramount importance. Sparse training, in contrast to dense training, uses models with sparse connections or weight matrices, and is a promising direction towards reducing the computational cost of training. Recent work has shown that pre-trained models can be pruned post-training, empirically showing the learned solution resides in a low-dimensional sub-space. Dense over-parameterization is, however, crucial for optimization, making sparse neural networks difficult to train. This raises a fundamental question — what breaks sparse training? Understanding sparse training can not only lead to better sparse training methods but also help us better understand the role of overparameterization.

Early work on the Lottery Ticket Hypothesis (LTH) demonstrated and proved the existence of sparse "winning tickets" that can be trained to match dense model performance when initialized appropriately. However, these winning tickets are typically identified after training a dense model, using information unavailable at initialization. This motivated pruning-at-initialization (PaI) methods, which aim to identify winning tickets without any training. Despite many proposed heuristics—based on weight magnitude, gradient information, or higher-order statistics, PaI methods consistently fail to match dense performance and often perform no better than random pruning (Frankle et al., 2021a; Gadhikar et al., 2023). This persistent failure suggests that the difficulty of PaI is not merely algorithmic, but an inherent difficulty of the model structure and optimization.

Recent work has tried to better understand LTH, PaI and Sparse Training from the perspective of weight-space symmetries. Adnan et al. (2025) and Gadhikar & Burkholz (2024) observe that symmetries play a key role in sparse training; these works show that the alignment between the sparse mask and the solution basin is important to find a good solution without overparameterization. Adnan et al. (2025) enabled generalization of LTH masks to new random initialization using permutation symmetries to align masks with a known solution basin. Furthermore, Gadhikar & Burkholz (2024) show that aligning the parameter signs at initialization with the *orthant* (n-dimensional generalization of quadrant in 2D) of the trained dense model is sufficient to find the winning ticket. These two works, while proposing

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.

independent insights, highlight the need for near-perfect orthant alignment for sparse training, which can explain why PaI methods fail to find the winning tickets. Both Adnan et al. (2025) and Gadhikar & Burkholz (2024) study LTH, which is a special case where the weights during early training and the sparse mask are implicitly aligned with the same orthant as the good generalizable models.

In this position paper, we unify these insights and argue that the core difficulty of PaI lies not only in structure learning, but more fundamentally in the trainability of Sparse Networks with Fixed Masks (STFM). Rather than treating sparse training as an obstacle to be circumvented, we propose that the research community should use it as a diagnostic tool. By analyzing how sparse masks interact with overparameterization, symmetries, and optimization dynamics, we can gain deeper insights into why overparameterization works — and the role of symmetries in training. Next, we develop this position in detail, arguing that sparse training with fixed masks provides a uniquely informative lens on the role of symmetries in deep learning optimization.

1.1 Position: Sparse Training is a Valuable Testbed

> **Position.** Sparse Training with a Fixed Mask (STFM) should be viewed not only as a method for efficient training, but also as a controlled way to study why overparameterized neural networks optimize so effectively. Overparameterized networks exhibit extensive weight-space symmetries and many functionally equivalent solutions, while fixed sparse masks explicitly break these symmetries by restricting the accessible parameter space, making optimization challenges that are often hidden in dense networks easier to expose and analyze. We argue that insights from this perspective could guide the design of more trainable and efficient neural networks that rely less on massive overparameterization.

Sparse Training provides a valuable testbed to gain deeper insights into the role of overparameterization and, as a corollary, drive progress toward efficient deep learning.

Why this position paper? In many ways, pruning at initialization (PaI) defines the holy grail of neural network sparsification and comprises two main challenges: (a) efficient structure learning that identifies an initial sparse mask and (b) training a network with this fixed sparse mask (STFM).

(a) Structure learning is likely impossible if we aim to find a task-specific, highly sparse mask without the ability to extract the necessary information from the data (Kumar et al., 2024a). The challenge is exacerbated by the fact that, currently, we cannot solve the optimization part (b). As a consequence, we cannot evaluate the potential of different structures and, in consequence, most PaI methods achieve performance comparable to random pruning (Frankle et al., 2021b; Liu et al., 2021; Gadhikar et al., 2023). This perceived hardness of PaI has led the research community to shift their focus towards computationally more expensive sparsification techniques.

However, as we argue in this position paper, there are many compelling reasons to pay closer attention to the trainability aspect (b). The sparse training community has largely accepted its difficulty without deeply investigating the underlying causes. This position paper therefore focuses on analyzing these causes more carefully and outlines strategies and a research agenda for overcoming the associated challenges.

Well-known reasons to study sparse training with a fixed mask. Solving sparse training with a fixed mask (STFM) would create an immense practical impact across different domains. For instance, a) we could train sparse architectures that are efficiently designed, e.g., taking hardware constraints into account (Lie, 2023). b) Problem-specific constraints, i.e. in model architecture, that are defined by domain knowledge could be taken into account from the beginning, shaping interpretable learning dynamics. c) STFM could support the efficient search for unstructured masks by a bottom up approach (contrary to pruning a large network). d) STFM would boost sparse training that simultaneously learns masks (e.g. dynamic sparse training). e) Insights on what constitute a good mask would be more readily available and could contribute to solving early structure learning, if not PaI.

Beyond the reasons that make STFM practically relevant, more importantly, even if we cannot completely close the performance gap between STFM and dense training, we gain valuable insights into fundamental deep learning challenges, for instance, pertaining to the interplay between overparameterization, weight space symmetries, and optimization. Understanding the benefits of

overparameterization and finding ways to achieve similar effects more effectively, could pave the way towards more efficient deep learning. To make our line of argument more concrete, we discuss two exemplary symmetries, i.e. permutation and rescale symmetry, and how their interplay with overparameterization impacts learning dynamics.

The interplay between symmetries and trainability. Highly overparameterized deep neural networks have a high capacity and plasticity to easily learn complex functions. The permutation symmetry of neurons in a layer contributes to this high adaptivity, as the many different neurons could assume different and exchangeable roles. Already at initialization, the neurons live in diverse orthants, which are defined by the signs of the weight parameters. During training, weights rarely have to change signs, as the probability is high that some of the many neurons in a layer have weights with signs that largely align with a sparse solution. In contrast, flipping signs is often associated with large movements through parameter space and is generally hampered by a rescale symmetry that influences the learning speed of different layers, as we will discuss in detail. Overparameterized networks usually require less of this movement.

Symmetry breaking by sparse masking. Introducing sparse masks breaks symmetries in a sense that the neurons become more specialized and less likely to match specific target neurons. Furthermore, the plasticity of sparse neurons suffers more from the rescale symmetry, constraining them to their initial orthant, as they hardly flip weight signs and thus can seldom assume different roles. In summary, a combination of reduced permutation symmetry and plasticity induced by an active rescale symmetry turns sparse training with a fixed mask into a hard problem.

Paths towards a solution. Both the analysis of symmetry breaking and the observation of hindered sign flipping has led to algorithmic innovations. The first one has inspired the utilization of known targets to find initializations of masks through activation alignment (Adnan et al., 2025). The second one has improved the ability to learn signs by training through novel parameterizations (Gadhikar et al., 2025) and optimizers (Jacobs et al., 2026a). Both ideas define major steps forward to tackle sparse training with a fixed mask and highlight opportunities for further improvements: We need better alignment between the mask and the orthants of a solution, which could be achieved with a combination of learning permutations and signs. The further study of the interplay of weight-space symmetries, overparameterization, and optimization might reveal additional roadblocks to the trainability of neural networks with sparse fixed masks, which demand our attention.

2 Related Work

Weight-space symmetries. Hecht-Nielsen (1990) demonstrated that neural networks are *permutation invariant*, where swapping any two neurons within a hidden layer does not alter the underlying function being learned. The permuted network remains functionally equivalent to its original configuration, i.e. neural networks are symmetric functions. The existence of permutation symmetries in weight space creates copies of global minima at different points in weight space (Entezari et al., 2022; Goodfellow et al., 2016; Simsek et al., 2021). Weight sparsity, achieved through pruning, can reduce the number of weight symmetries in a neural network. Pruning neurons reduces the number of permutation symmetries in a layer. Unstructured pruning, is heterogeneously removing individual weights, which breaks the weight symmetry of individual neurons in layers. Another symmetry in neural networks is rescale symmetry, which leads to a "balance equation" for gradient flow (Gunasekar et al., 2017; Marcotte et al., 2023). This equation determines which solutions are reachable (Gadhikar & Burkholz, 2024; Gadhikar et al., 2025). We highlight that for sparse training this prevents moving between orthants.

Implicit regularization. In dense training the consequences of the balance equation have been studied extensively and is also known as: implicit regularization (Gunasekar et al., 2017; Li et al., 2021; Pesme et al., 2021; Woodworth et al., 2020; Wang & Klabjan, 2024; Marcotte et al., 2023; Dominé et al., 2024; Kunin et al., 2024; Tsilivis et al., 2025; Warmuth et al., 2021). In other words, the tendency of the neural network training without explicit regularization. Here, the balance equation forces consecutive layers to stay small introducing a bias towards sparsity or low rank. These balance equations occur in various setups such as transformers, ResNets (Marcotte et al., 2025), different training algorithms (Marcotte et al., 2024; Wang et al., 2023; Jacobs et al., 2026c), and even weight decay enforces balance (Marcotte et al., 2025; Jacobs et al., 2025). Implicit regularization literature

provides a possible explanation for why dense models do not overfit the training data. On the flip side, we argue that it is detrimental for sparse training.

Lottery Ticket Hypothesis. The Lottery Ticket Hypothesis (LTH) proposes to solve the sparse training problem by re-using the same initialization as used to train the pruned models. For very small models, training from such an initialization maintains the generalization performance of the pruned model and demonstrates that training with a highly sparse mask is possible (Frankle & Carbin, 2019; Ramanujan et al., 2020; Hassibi et al., 1993; Morcos et al., 2019; Frankle et al., 2021d; Fischer & Burkholz, 2021). An even stronger hypothesis (Ramanujan et al., 2020) was proven succinctly (Malach et al., 2020; Pensia et al., 2020; Fischer et al., 2021; Burkholz, 2022b; da Cunha et al., 2022; Burkholz, 2022a; Burkholz et al., 2022; Ferbach et al., 2022; Burkholz, 2024; Gadhikar et al., 2023), showing that pruning a random source model suffices and that no further training is required. However, subsequent work has shown that obtaining winning tickets for modestly-sized models requires using *weight rewinding* (Frankle et al., 2020) — requiring significantly more compute than dense training alone, especially considering that LTH also requires Iterative Magnitude Pruning (IMP), i.e. training of iteratively sparsified models. Furthermore, recent work has shown that the LTH effectively re-learns the original pruned solution it is derived from (Evci et al., 2022).

Pruning at Initialization. In contrast to the LTH, Pruning at Initialization (PaI) methods aim to identify a sparse topology that can be trained to match dense performance without any training (Lee et al., 2019; Tanaka et al., 2020; Liu et al., 2021). In contrast to LTH, PaI aims to find the winning ticket at initialization without any training steps. However, PaI methods often exhibit a trainability gap compared to dense training or dynamic sparsity methods. Frankle et al. (2021c) showed that performance is largely determined by the layer-wise sparsity ratio rather than the specific pruning criterion, highlighting a key limitation of PaI methods.

3 Role of Symmetries in Training

Neural networks have weight-space symmetries, such as permutation symmetries in linear layers, rotation symmetries in the attention heads of transformers, and rescaling symmetries between layers. In this position paper, we focus on permutation and rescale symmetry, using a two-layer network as an example. One potential way to understand the role of symmetries is to study the implications of breaking and reducing them, as in Sparse Neural Networks (SNNs). We propose that the role of symmetries can be principally studied by first understanding the challenges of sparse training and how it works. In contrast to overparameterized models, SNNs contains fewer symmetries, therefore, fewer good solutions, making it difficult for gradient descent to converge to good solutions. We start by answering what is encoded in lottery tickets i.e. well trainable sparse neural networks.

3.1 What's encoded in a winning ticket?

The LTH showed the existence of a winning ticket, or a sparse mask, coupled to a specific initialization. Although training a SNN from a random initialization is difficult and fails to match dense model performance, winning tickets, when trained from scratch, match the dense model performance — sometimes even better. Understanding how winning tickets find good solution without overparameterization, can provide valuable insights in to training in general. Paul et al. (2023) observed that successive sparse models obtained during Iterative Magnitude Pruning are linearly mode connected, suggesting that pruning preserves the same solution basin. This was later extended by Adnan et al. (2025), who showed that dense and sparse models found using IMP are also linearly mode connected once normalization layers are properly accounted for (Jordan et al., 2023). Similarly, Evci et al. (2022) showed that solutions obtained by winning tickets lie in the same basin and are highly similar in weight and function space. Together, these results suggest that the winning ticket enables sparse training by restricting optimization to a favorable low-dimensional subspace of solutions: an *orthant*. More recent work has refined this view by explicitly accounting for weight-space symmetries. From a weight-space perspective, Adnan et al. (2025) showed that the LTH mask can be reused across random initializations if the mask is permuted to match the optimization basin of the new initialization. Furthermore, Gadhikar et al. (2025) further demonstrated that preserving only the *signs* of the corresponding solution (or rewind point) is sufficient to recover winning-ticket performance.

These findings point to a unifying interpretation — the winning ticket is not merely a subset of important weights, but an initialization that is already aligned with the correct (broken) symmetry configuration. In particular, the winning ticket's mask together with the initialization signs places the sparse network in the *correct orthant* of the weight space. As a result, when winning tickets are trained from random initialization, training does not need to move between orthants to reach the solution, the difficulty of which is discussed in Gadhikar et al. (2025). This perspective explains both the initialization dependence of winning tickets and their apparent fragility to random initialization. Random sparse initializations are unlikely to lie in the correct orthant and must rely on difficult sign changes during training. In contrast, a winning ticket starts in an orthant that already contains the solution, making the optimization problem easier. Once symmetries are properly accounted for via sign alignment or permutation, the same mask generalizes across random initializations, as demonstrated by (Adnan et al., 2025).

Implication. Winning tickets succeed by preserving the symmetry-breaking configuration induced by overparameterization, not by identifying intrinsically important weights.

The necessity of orthant alignment for optimization. Recent works have studied the necessity of correct alignment for sparse training. Gadhikar & Burkholz (2024); Gadhikar et al. (2025) showed that a SNN can be trained from a random initialization to match dense accuracy when initialized in the correct orthant. To initialize the network in the correct orthant, Gadhikar et al. (2025) proposed using the signs from a trained model that has converged to a good solution. When the SNN is initialized in the correct orthant, i.e., using the mask and signs of the winning ticket, the model is able to recover the performance of LTH, AC/DC (Peste et al., 2021), STR (Kusupati et al., 2020), and RiGL (Evci et al., 2020b), as shown in table 1. Initializing with the correct signs and the mask of the winning ticket enables sparse training. These results highlight the importance of the correct orthant, with further evidence across optimizers and architectures in Section D.

Table 1: **Orthant alignment.** for a ResNet50 trained on ImageNet with 90% sparsity. The mask is first derived from the named sparse training method. Observe, baseline performance is recovered when the learned fixed mask is initialized with the learned parameter signs, but not with the learned parameter magnitude and random signs. (Results from (Gadhikar et al., 2025))

Method	AC/DC	STR	RiGL
Baseline	<u>74.68</u>	<u>73.65</u>	<u>73.75</u>
Correct orthant	**74.88**	**73.77**	**73.74**
Random orthant	70.6	68.38	71.89

Can we find a permutation to correct the alignment? To further demonstrate that aligning orthants of a randomly initialized model is important for the optimizer to find the solution, we highlight the observation from the Adnan et al. (2025), which empirically showed that given a sparse mask from a winning ticket and a new random initialization (different than the one used for obtaining the winning ticket), the mask can be permuted by a suitable permutation π. Permuting the mask with permutation π can align the orthant of the model initialized with the random initialization and the permuted mask with the correct orthant and helps the optimizer to find the solution easily and match the dense performance, as shown by the authors.

3.2 Improbability of the correct orthant at initialization

Orthant and subspace. In $\mathbb{R}^d$, the coordinate hyperplanes partition the space into 2^d *orthants*, i.e. the n-dimensional generalization of quadrants in 2D. Each orthant corresponds to a unique sign pattern of the coordinates. Formally, for a sign vector $\boldsymbol{s} \in \{-1,+1\}^d$, the associated orthant is defined as $\mathcal{O}_{\boldsymbol{orth}} := \{\boldsymbol{w} \in \mathbb{R}^d \,|\, s_i w_i \geq 0, \forall i \in \{1,\ldots,d\}\}$. This generalizes quadrants in two dimensions and octants in three dimensions to arbitrary dimension. Moreover, with a fixed sparse mask, we constrain the active parameters by applying a mask $\mathbf{m} \in \{0,1\}^d$. This leads to a restriction to a subspace: $\mathcal{O}_{\textbf{sub}} := \text{span}\{e_i \in \mathbb{R}^d \,|\, m_i = 1, i \in \{1,\ldots,d\}\}$. where e_i is the basis vector of the $i-$th parameter.

Problem setup. To understand how weight-space symmetries and parameter signs affect optimization, we consider a toy neural network $f : (\mathbb{R}^{d+1})^N \times \mathbb{R}^d \to \mathbb{R}$ with a single hidden layer of N neurons and input dimension d such that $f(a,\mathbf{w};\mathbf{x}) := \sum_{j\in[N]} a_j\, \sigma(\mathbf{w}_j^T \mathbf{x})$, where $\sigma : \mathbb{R} \to \mathbb{R}$ a homogeneous

activation function such as the ReLU. Assume the target solution is given by a single neuron $a^*\sigma(w^{*T}\cdot)$ with $(a^*,\mathbf{w}^*)\in\mathbb{R}^{d+1}$ and that the task lies in lower dimensional space, such that $\mathbf{w}^*\neq 0$ for only m dimensions with $m<d$. Moreover, $s:=\text{sign}(\mathbf{w}^*)\in\{-1,1,0\}^d$ determines the orthant of the inner layer. To induce sparsity we introduce a mask $\mathbf{m}_j\in\{0,1\}^d$ which is applied to the weight vector $\mathbf{w}_j$ denoted by $\mathbf{m}_j\odot\mathbf{w}_j$, where $\odot$ is the Hadamard product or pointwise multiplication.

Permutation Symmetries. In the dense model, all neurons are structurally identical and therefore fully exchangeable: any permutation of hidden neurons, together with the corresponding permutation of outgoing weights, leaves the model function unchanged. In contrast, sparse training with fixed masks (STFM) breaks this exchangeability. Since each neuron is associated with a mask $\mathbf{m}_j$, neurons generally have different sparsity patterns and different numbers of active incoming connections. As a result, they are no longer structurally identical and only permutations among neurons with identical masks preserve the parameterization. Consequently, the set of admissible permutations is strictly reduced, leading to fewer equivalent solutions and weaker/broken permutation symmetry.

To illustrate this, we show that the probability is extremely low that we initialize a sparse mask such that SGD is able to learn a solution. To reach the target, the subspace induced by the target should be covered by the neural network f. So all trainable weights in the mask of the target should also be trainable by f. In other words, the total subspace of the inner layer parts of all neurons $\mathbf{w}_j$ for $j\in[N]$ should contain the subspace of $\mathbf{w}^*$.

Theorem 3.1. *Let $\mathcal{O}^*_{\textbf{sub}}$ denote the subspace of the target and $\mathcal{O}_{\textbf{sub},\,j}$ for $j\in[N]$ the subspace of each neuron. The probability of a random global layer mask with $\ell\geq m$ non-zero entries to recover the target is given by: $\mathbb{P}\left(\mathcal{O}^*_{\textbf{sub}}\subset\bigcup_{j=1}^N\mathcal{O}_{\textbf{sub},\,j}\right)\approx\left(1-e^{-\ell/d}\right)^m$, where the approximation holds for $\ell<<Nd$.*

Proof. This follows from a counting argument and using an approximation of binomial coefficients. (See Theorem C.1)

Theorem 3.1 indicates that it is very unlikely that a randomly initialized sparse model is in the same subspace as a target for $m\approx\ell<<Nd$. This illustrates how hard it actually is in sparse training to initialize a network correctly so that it is able to reach a good solution. For visual support, we plot the probabilities for various ℓ and m in Figure 1c.

Orthant alignment. Going one step further, consider a random symmetric initialization, standard in deep learning. Then, the probability of hitting the correct orthant is captured by Corollary 3.2.

Corollary 3.2. *Let $\mathcal{O}^*_{\textbf{orth}}$ denote the orthant of the target and $\mathcal{O}_{\textbf{orth},j}$ for $j\in[N]$ the orthant of each neuron. For a symmetric initialization, the probability of being in the correct orthant is: $\mathbb{P}\left(\mathcal{O}^*_{\textbf{orth}}\subset\bigcup_{j=1}^N\mathcal{O}_{\textbf{orth},\,j}\right)\approx\frac{1}{2^m}\left(1-e^{-\ell/d}\right)^m$.*

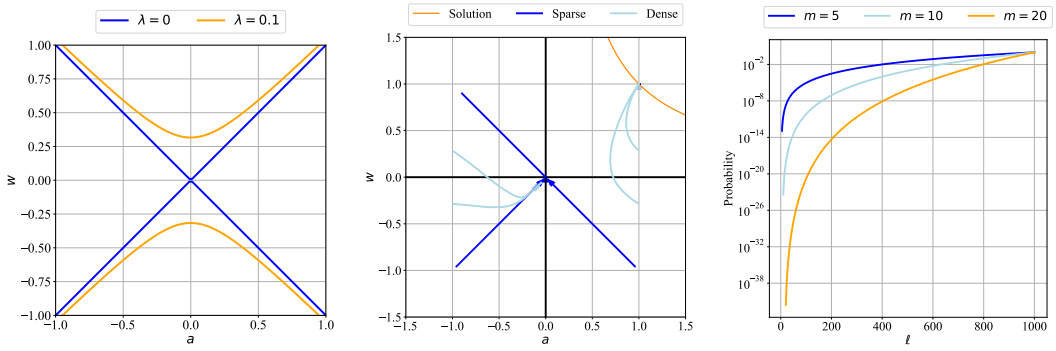
Proof. Condition on the probability to recover the target solution and apply Theorem 3.1. □

> **Implication.** Overparameterization exponentially increases the probability of starting in the correct orthant. Sparsity reduces this probability, which makes it difficult for the optimizer to find a good solution.

3.3 Moving orthants is inherently difficult

We previously discussed why orthant alignment is important for a randomly initialized model to find the solution. This raises an obvious question — can the optimizer move between orthants to reach the correct one? While this may seem intuitively possible, doing so is challenging in practice. If the weight parameters are not initialized in the correct orthant, reaching the solution requires sign flipping, i.e., the optimizer must change the sign of the relevant weights to enter the correct orthant. Even if we assume to be in the correct subspace, the probability of being in the same orthant remains small, as highlighted by Corollary 3.2.

To illustrate the difficulty of sign flipping and thus moving between orthants we consider the training dynamics under gradient flow (small learning rate) and mean squared error loss L and data set $Z:=\{(\mathbf{x}_i,y_i)\}_{i=1}^K$: $d(a_{j,t},\mathbf{w}_{j,t})=-\nabla_j L(a_t,\mathbf{w}_t,Z)dt$, where $(a_{j,0},\mathbf{w}_{j,0})=(a_{j,\text{init}},\mathbf{w}_{j,\text{init}})$ and



(a) Balance equation. (b) Dense vs sparse training. (c) Orthant alignment probability.

Figure 1: (a) The balance equation of a univariate one-neuron neural network. Balanced initialization ($\lambda = 0$) prevents moving through zero and thus switching orthants. (b) Training a single neuron ($n=5$, $m=1$) with gradient descent ($\eta=0.01$) starting from various orthants (quadrants). Observe that dense training can recover the target while sparse training cannot given that a has the correct sign (but not w). If a is initialized with the wrong sign, even dense training fails. (c) The probability of a random masked neuron ($N=1$) being in the same subspace as the target solution for $d=1000$.

∇_j denotes the gradient with respect to the $j-$th neuron. In this setting, the network parameters satisfy the following equation for all $t \geq 0$ (Gadhikar & Burkholz, 2024): $a_{j,t}^2 - ||\mathbf{w}_{j,t}||_2^2 = \lambda$, where $\lambda := a_{j,0}^2 - ||\mathbf{w}_{j,0}||_2^2$. This is also known as the "balance equation", moreover, a balanced initialization satisfies $\lambda = 0$. The balance equation is a direct result of the rescale symmetry of the neural network and the optimization algorithm. This equation also holds when replacing $\mathbf{w}_j$ with $\mathbf{m}_j \odot \mathbf{w}_j$ for all $j \in [N]$. Similar balance equations have been derived for deep architectures such as ResNets and transformer highlighting their generality (Marcotte et al., 2025).

As a consequence, if the second layer parameter of a neuron a_j has to flip its sign, the parameter has to move through zero, but this implies that $||\mathbf{w}_j||_2 = 0$ as well. This brings the training to a stop for both dense and sparse training. Furthermore, if most weights of $\mathbf{m}_j \odot \mathbf{w}_j \in \mathbb{R}^d$ need to flip their sign, which can occur when ($\ell \approx m$) we have that the same problem occurs in the first layer as all weights have to move through zero forcing a_j to zero as well. We illustrate this for the univariate case ($d=1$) in Figure 1. We observe that for a balanced initialization we have to move trough zero, thus moving between orthants is prevented. Note for slight imbalanced initialization ($\lambda = 0.1$) the parameter movement still slows down as the dynamics is forced to move close to zero as we will argue next.

Beyond gradient flow. The exact balance equation is derived in the gradient-flow setting, however, the underlying obstruction is more general: rescale symmetry couples the effective learning speed of adjacent layers, so sign changes require passing through low parameter norm regions where the relevant updates are small. This can be seen from the gradient magnitude of f for each neuron $j \in [N]$:

$$\nabla_j f(a, \mathbf{w}; \mathbf{x}) = \begin{bmatrix} \sigma(\mathbf{w}_j^T \mathbf{x}) \\ a_j \mathbf{x} \sigma'(\mathbf{w}_j^T \mathbf{x}) \end{bmatrix} = O\left(\begin{bmatrix} ||\mathbf{w}_j||_2 \\ |a_j| \end{bmatrix} \right).$$

The magnitude of the gradient for each layer is proportional to the other layer, slowing down training near zero as well as for imbalanced initializations. To illustrate this, we simulate gradient descent for a toy one-neuron neural network ($N=1$) with input dimension $d=5$ and $m=1$. For sparse training we mask everything except the contributing one ($\ell = 1$). In Figure 1, we plot the trajectories of the non-zero parameter initialized from a balanced initialization. In the case of sparse training, the target can only be recovered when we initialize in the correct orthant while dense training also recovers the target for $\text{sign}(\mathbf{w}_1) \neq \text{sign}(\mathbf{w}_1^*)$, where the index now refers to the first parameter of the neuron. Note that both fail to reach the target when $\text{sign}(a) \neq \text{sign}(a^*)$. For the experimental details see Appendix A.

Implication. Overparameterization increases symmetries and number of solutions. Therefore, more orthants contain solutions and training requires fewer sign flips.

3.4 Discussion

In section 3, we connect insights from LTH, sparse training, and weight-space symmetries. SNNs break permutation symmetries, making orthant alignment necessary during training. Furthermore, moving between orthants is hard due to the rescale symmetry. Furthermore, insights from sparse training suggest the need of being in the correct orthant for successful training. LTH enables sparse training from scratch by finding a winning ticket that couples the sparse mask and initialization, such that the weights are in the correct orthant. These insights highlight the role of overparameterization in training from a weight-space perspective. Overparameterization increases the number of orthants containing good solutions due to permutation symmetries, making it more likely for a randomly initialized network to find a good solution. This could be one of the reasons why Deep Neural Networks (DNNs), when trained independently from two different random initializations, often converge to good minima within a similar number of training steps and exhibit similar training accuracy evolution. These insights can not only be useful for improving the state of sparse training but can even help us design better optimizers for dense training. We discuss a few interesting research directions and an agenda for the community in section 4 based on these insights.

4 From heuristics to principles

In this work, we posit that the true value of studying sparse training lies not only in improving sparse methods, but in using it as a lens to understand the core challenges of optimization and generalization in neural networks. By isolating and amplifying these challenges, sparse training provides a unique opportunity to derive principles that extend to dense training. In this spirit, we outline research directions aimed at leveraging insights from sparsity to advance neural network training more broadly.

4.1 In-time-over-parameterization

Dynamic Sparse Training (DST), unlike PaI, updates the sparse topology/mask periodically, enabling exploration of different orthants during training (Evci et al., 2020b; Mocanu et al., 2018a). Each update prunes weights (typically by magnitude) and regrows new connections, allowing DST to explore many orthants and eventually converge to a winning ticket in the correct orthant. Although DST can match dense performance, it converges more slowly, with total training FLOPs comparable to dense models. This ability to overcome the limitations of PaI (section 3) makes DST a more promising approach for sparse training from random initialization. However, current methods rely on simple, inefficient heuristics—random topology exploration (Mocanu et al., 2018a) or noisy gradient signals, especially early in training (Evci et al., 2020b). A more principled approach is therefore needed: leveraging weight-space symmetries could guide dynamic sparse training more systematically, improving convergence and generalization by enabling efficient, structured exploration of orthants.

4.2 Symmetry-aware training methodologies

Symmetry-aware optimizers. Weight-space symmetries provides key insights into sparse training; introducing sparsity to a model restricts its symmetries in the weight-space, which makes training and finding a good solution from a random initialization difficult, as discussed in section 3. (Jaiswal et al., 2022) also showed that current sparse training baselines can be improved by using different training/model components, such as activation functions, and argued that training components that work well for dense training do not work well for sparse training. Since finding the correct orthant during training is needed for sparse training, developing new optimizers that encourage and accelerate sign flips could be the key to improve the state of sparse training. A recently proposed optimizer for sparse training, HAM (Jacobs et al., 2026a), uses these insights and accelerates sign flips, which both improves sparse and dense training generalization.

Unifying dense and sparse training. Rather than being ad hoc solutions specific to sparse layers, both the sparse initialization of Evci et al. (2022) and the sparse normalization of Adnan et al. (2026) arise from a more general formulation that accounts for heterogeneous neuron connectivity within a layer, and reduce to standard methods in the dense, homogeneous case. This establishes dense training as a special case of sparse training, where symmetry is preserved rather than broken. Consistent with this view, optimizers such as HAM also improve dense training, indicating that addressing the

challenges introduced by symmetry breaking in sparse settings reveals underlying principles of neural network optimization more broadly. However, existing approaches remain fragmented, and a unified, principled framework for sparsity-aware optimization is still lacking. Developing such a framework is crucial, as it can reveal the fundamental role of symmetry and connectivity in optimization (Ziyin, 2023), and translate these insights into more efficient and robust dense training methods.

Rethinking sparse models via symmetries. Weight-space symmetries depends on the architecture and the activation functions used in a model. For example, the ReLU activation function breaks rotation symmetries and exhibits permutation and rescale symmetries. Attention layers, in transformers, exhibit rotation symmetries due to the dot product. Since the probability of finding the correct orthant is proportional to the number of symmetries, designing new sparse model architectures and activation functions that increase symmetries in the context of sparse training can potentially improve the state of sparse training and improve on current baselines.

4.3 Understanding the role of symmetries in Transformers

Transformers exhibit multiple weight-space symmetries, including permutation, scaling, and attention-specific rotation invariances (Marcotte et al., 2025). Understanding the role of these symmetries remains an open challenge. Sparse training provides a controlled testbed to study them, since masks can break or reduce symmetries in ways that are otherwise difficult to isolate in dense models. This perspective is particularly relevant for large language models, where sparse training is challenging despite strong post-training methods such as SparseGPT (Frantar & Alistarh, 2023). At the same time, recent work on rotation-based quantization shows that Transformer rotation invariances can be exploited productively for compression (Liu et al., 2024). Sparse Transformers could therefore provide insights into the role and interplay of different symmetries in the transformers.

5 Alternative view

Information-theoretical perspective. In Kumar et al. (2024b), an information-theoretic perspective is taken to explain why PaI is unsuccessful, suggesting that there is an insufficient mutual information between the mask and the data at initialization to interpolate noisy training data. This suggests that overparameterization is needed to compensate for the lack of information at the start of training. While this reveals the failure mode it does not provide a mechanistic understanding of the role of overparameterization. Our position argues for understanding these mechanisms through the use of symmetry breaking. Overparameterization ensures a high probability that a subset of neurons is initialized in the correct orthant, allowing the optimizer to leverage weight-space symmetries to navigate the landscape. Thus, while information geometry may highlight the absence of an initially accessible solution and maybe even the masks, our perspective identifies the mechanical advantage, specifically the preservation of permutation and rescaling symmetries, that allows overparameterized models to find a solution if it exists.

Breaking symmetries might not break training. The symmetries of neural networks can be broken in ways other than inducing sparsity. For example, permutation symmetry can be broken by making activation functions dependent on the full layer (Lim et al., 2024). In contrast, breaking the symmetry in this particular way does not lead to performance degradation. This suggests that the failure of STFM is not caused by the loss of permutation symmetry in isolation, but rather by the specific way sparsity restricts the neural network to a subspace or orthant.

6 Conclusion

In this position paper, we highlighted the connection between sparse training, overparameterization and weight-space symmetries. These findings suggest that sparse training is not merely a tool for efficiency, but a controlled setting for exposing the role of symmetry in deep learning. By understanding how symmetry breaking hinders optimization and how alignment restores it, we can design algorithms that explicitly respect weight-space structure or breaks them deliberately if their effects are harmful. We believe this perspective opens new directions and discussion for designing better optimizer and understanding the role of symmetries in training.

AI USE STATEMENT

In this work, we used generative AI tools to edit a research paper to improve readability, propose a title or keywords for a research paper. We have **not** used generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results. We have reviewed all AI-assisted work: original meaning of the text was not changed when AI tools were used for editing some part of the manuscript. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Mohammed Adnan, Rohan Jain, Ekansh Sharma, Rahul Krishnan, and Yani Ioannou. Sparse training from random initialization: Aligning lottery ticket masks using weight symmetry. In *Forty-second International Conference on Machine Learning (ICML)*, July 2025. URL https://icml.cc/virtual/2025/poster/46094.

Mohammed Adnan, Rohan Jain, Tom Jacobs, Ekansh Sharma, Rahul G. Krishnan, Rebekka Burkholz, and Yani Ioannou. SparseOpt: Addressing Normalization-Induced Gradient Skew in Sparse Training. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, July 2026.

Rebekka Burkholz. Convolutional and residual networks provably contain lottery tickets. In *International Conference on Machine Learning*, 2022a.

Rebekka Burkholz. Most activation functions can win the lottery without excessive depth. In *Advances in Neural Information Processing Systems*, 2022b.

Rebekka Burkholz. Batch normalization is sufficient for universal function approximation in CNNs. In *International Conference on Learning Representations*, 2024.

Rebekka Burkholz, Nilanjana Laha, Rajarshi Mukherjee, and Alkis Gotovos. On the existence of universal lottery tickets. In *International Conference on Learning Representations*, 2022.

Arthur da Cunha, Emanuele Natale, and Laurent Viennot. Proving the lottery ticket hypothesis for convolutional neural networks. In *International Conference on Learning Representations*, 2022.

Clémentine Carla Juliette Dominé, Nicolas Anguita, Alexandra Maria Proca, Lukas Braun, Daniel Kunin, Pedro A. M. Mediano, and Andrew M Saxe. From lazy to rich: Exact learning dynamics in deep linear networks. In *UniReps: 2nd Edition of the Workshop on Unifying Representations in Neural Models*, 2024. URL https://openreview.net/forum?id=yAE3LOjgA4.

Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Utku Evci, Erich Elsen, Pablo Castro, and Trevor Gale. Rigging the lottery: Making all tickets winners, 2020a. URL https://openreview.net/forum?id=ryg7vA4tPB.

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In *International Conference on Machine Learning*, pp. 2943–2952. PMLR, 2020b.

Utku Evci, Yani A. Ioannou, Cem Keskin, and Yann Dauphin. Gradient flow in sparse neural networks and how lottery tickets win. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, February 2022. doi: 10.1609/aaai.v36i6.20611.

Damien Ferbach, Christos Tsirigotis, Gauthier Gidel, and Bose Avishek. A general framework for proving the equivariant strong lottery ticket hypothesis, 2022.

Jonas Fischer and Rebekka Burkholz. Plant 'n' seek: Can you find the winning ticket?, 2021.

Jonas Fischer, Advait Harshal Gadhikar, and Rebekka Burkholz. Towards strong pruning for lottery tickets with non-zero biases, 2021.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Pruning neural networks at initialization: Why are we missing the mark? In *International Conference on Learning Representations*, 2021a.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Pruning neural networks at initialization: Why are we missing the mark? In *International Conference on Learning Representations*, 2021b.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Pruning neural networks at initialization: Why are we missing the mark? In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021c.

Jonathan Frankle, David J. Schwab, and Ari S. Morcos. Training BatchNorm and only BatchNorm: On the expressive power of random features in CNNs. In *International Conference on Learning Representations*, 2021d.

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, pp. 10323–10337. PMLR, 2023.

Advait Gadhikar and Rebekka Burkholz. Masks, signs, and learning rate rewinding. In *Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=qODvxQ8TXW.

Advait Gadhikar, Tom Jacobs, Chao Zhou, and Rebekka Burkholz. Sign-in to the lottery: Reparameterizing sparse training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=iwKT7MEZZw.

Advait Harshal Gadhikar, Sohom Mukherjee, and Rebekka Burkholz. Why random pruning is all we need to start sparse. In *International Conference on Machine Learning*, 2023.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT Press, 2016.

Suriya Gunasekar, Blake Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nathan Srebro. Implicit regularization in matrix factorization, 2017.

B. Hassibi, D.G. Stork, and G.J. Wolff. Optimal brain surgeon and general network pruning. In *IEEE International Conference on Neural Networks*, pp. 293–299 vol.1, 1993. doi: 10.1109/ICNN.1993.298572.

Robert Hecht-Nielsen. *On The Algebraic Structure of Feedforward Network Weight Spaces*. North-Holland, Amsterdam, 1990. ISBN 978-0-444-88400-8. doi: https://doi.org/10.1016/B978-0-444-88400-8.50019-4.

Tom Jacobs and Rebekka Burkholz. Mask in the mirror: Implicit sparsification. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=U47ymTS3ut.

Tom Jacobs, Chao Zhou, and Rebekka Burkholz. Mirror, mirror of the flow: How does regularization shape implicit bias? In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=MLiR9LS5PW.

Tom Jacobs, Advait Gadhikar, Celia Rubio-Madrigal, and Rebekka Burkholz. Ham: A hyperbolic step to regulate implicit bias. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://arxiv.org/abs/2506.02630.

Tom Jacobs, Rohan Jain, and Rebekka Burkholz. Horst: Composing optimizer geometries for sparse transformer training. *ArXiv*, abs/2605.21104, 2026b. URL https://api.semanticscholar.org/CorpusID:288655914.

Tom Jacobs, Chao Zhou, and Rebekka Burkholz. Never saddle: Reparameterized steepest descent as mirror flow. In *The Fourteenth International Conference on Learning Representations*, 2026c. URL https://openreview.net/forum?id=YgudIlQ9nC.

Ajay Kumar Jaiswal, Haoyu Ma, Tianlong Chen, Ying Ding, and Zhangyang Wang. Training your sparse neural network better with any mask. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9833–9844. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/jaiswal22a.html.

Keller Jordan, Hanie Sedghi, Olga Saukh, Rahim Entezari, and Behnam Neyshabur. REPAIR: renormalizing permuted activations for interpolation repair. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

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

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL https://api.semanticscholar.org/CorpusID:6628106.

Chris Kolb, Laetitia Frost, Bernd Bischl, and David Rügamer. Differentiable sparsity via d-gating: Simple and versatile structured penalization, 2025a. URL https://arxiv.org/abs/2509.23898.

Chris Kolb, Tobias Weber, Bernd Bischl, and David Rügamer. Deep weight factorization: Sparse learning through the lens of artificial symmetries. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=vNdOHr7mn5.

Tanishq Kumar, Kevin Luo, and Mark Sellke. No free prune: information-theoretic barriers to pruning at initialization. In *International Conference on Machine Learning*, 2024a.

Tanishq Kumar, Kevin Luo, and Mark Sellke. No free prune: Information-theoretic barriers to pruning at initialization. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024b.

Daniel Kunin, Allan Ravent'os, Cl'ementine Carla Juliette Domin'e, Feng Chen, David Klindt, Andrew Saxe, and Surya Ganguli. Get rich quick: exact solutions reveal how unbalanced initializations promote rapid feature learning. *ArXiv*, abs/2406.06158, 2024. URL https://api.semanticscholar.org/CorpusID:270371640.

Aditya Kusupati, Vivek Ramanujan, Raghav Somani, Mitchell Wortsman, Prateek Jain, Sham Kakade, and Ali Farhadi. Soft threshold weight reparameterization for learnable sparsity. In *Proceedings of the International Conference on Machine Learning*, July 2020.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Snip: single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*, 2019.

Jiangyuan Li, Thanh V. Nguyen, Chinmay Hegde, and Raymond K. W. Wong. Implicit sparse regularization: The impact of depth and early stopping, 2021. URL https://arxiv.org/abs/2108.05574.

Sean Lie. Cerebras architecture deep dive: First look inside the hardware/software co-design for deep learning. *IEEE Micro*, 43(3):18–30, 2023. doi: 10.1109/MM.2023.3256384.

Derek Lim, Theo Putterman, Robin Walters, Haggai Maron, and Stefanie Jegelka. The empirical impact of neural parameter symmetries, or lack thereof. In *Advances in Neural Information Processing Systems*, volume 37, pp. 28322–28358, 2024.

Shiwei Liu, Tianlong Chen, Xiaohan Chen, Li Shen, Decebal Constantin Mocanu, Zhangyang Wang, and Mykola Pechenizkiy. The unreasonable effectiveness of random pruning: Return of the most naive baseline for sparse training. In *International Conference on Learning Representations*, 2021.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: Llm quantization with learned rotations. *ArXiv*, abs/2405.16406, 2024. URL https://api.semanticscholar.org/CorpusID:270062819.

Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *ArXiv*, abs/1711.05101, 2017. URL https://api.semanticscholar.org/CorpusID:3312944.

Eran Malach, Gilad Yehudai, Shai Shalev-Shwartz, and Ohad Shamir. Proving the lottery ticket hypothesis: Pruning is all you need. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6682–6691. PMLR, 2020.

Sibylle Marcotte, Rémi Gribonval, and Gabriel Peyr'e. Abide by the law and follow the flow: Conservation laws for gradient flows. *ArXiv*, abs/2307.00144, 2023. URL https://api.semanticscholar.org/CorpusID:259316537.

Sibylle Marcotte, Rémi Gribonval, and Gabriel Peyré. Keep the momentum: Conservation laws beyond euclidean gradient flows, 2024. URL https://arxiv.org/abs/2405.12888.

Sibylle Marcotte, Rémi Gribonval, and Gabriel Peyré. Transformative or conservative? conservation laws for resnets and transformers, 2025. URL https://arxiv.org/abs/2506.06194.

Decebal Constantin Mocanu, Elena Mocanu, Peter Stone, Phuong H. Nguyen, Madeleine Gibescu, and Antonio Liotta. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature Communications*, 9(1), jun 2018a. doi: 10.1038/s41467-018-04316-3. URL https://doi.org/10.1038%2Fs41467-018-04316-3.

Decebal Constantin Mocanu, Elena Mocanu, Peter Stone, Phuong H Nguyen, Madeleine Gibescu, and Antonio Liotta. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature communications*, 9(1):2383, 2018b.

Ari S. Morcos, Haonan Yu, Michela Paganini, and Yuandong Tian. One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers, 2019.

Mansheej Paul, Feng Chen, Brett W. Larsen, Jonathan Frankle, Surya Ganguli, and Gintare Karolina Dziugaite. Unmasking the lottery ticket hypothesis: What's encoded in a winning ticket's mask? In *The Eleventh International Conference on Learning Representations*, 2023.

Ankit Pensia, Shashank Rajput, Alliot Nagle, Harit Vishwakarma, and Dimitris Papailiopoulos. Optimal lottery tickets via subset sum: Logarithmic over-parameterization is sufficient. In *Advances in Neural Information Processing Systems*, volume 33, pp. 2599–2610, 2020.

Scott Pesme, Loucas Pillaud-Vivien, and Nicolas Flammarion. Implicit bias of sgd for diagonal linear networks: a provable benefit of stochasticity, 2021.

Alexandra Peste, Eugenia Iofinova, Adrian Vladu, and Dan Alistarh. Ac/dc: Alternating compressed/decompressed training of deep neural networks, 2021.

Hoang Pham, The-Anh Ta, Tom Jacobs, Rebekka Burkholz, and Long Tran-Thanh. The graphon limit hypothesis: Understanding neural network pruning via infinite width analysis. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=EEZLBhyer1.

Vivek Ramanujan, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari. What's hidden in a randomly weighted neural network? In *Computer Vision and Pattern Recognition*, pp. 11893–11902, 2020.

Berfin Simsek, François Ged, Arthur Jacot, Francesco Spadaro, Clément Hongler, Wulfram Gerstner, and Johanni Brea. Geometry of the loss landscape in overparameterized neural networks: Symmetries and invariances. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9722–9732. PMLR, 2021.

Hidenori Tanaka, Daniel Kunin, Daniel L. Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. In *Advances in Neural Information Processing Systems*, 2020.

Nikolaos Tsilivis, Gal Vardi, and Julia Kempe. Flavors of margin: Implicit bias of steepest descent in homogeneous neural networks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=BEpaPHDl9r.

Kun Wang, Yuxuan Liang, Pengkun Wang, Xu Wang, Pengfei Gu, Junfeng Fang, and Yang Wang. Searching lottery tickets in graph neural networks: A dual perspective. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Dvs-a3aymPe.

Shuyang Wang and Diego Klabjan. A mirror descent perspective of smoothed sign descent. In *Conference on Uncertainty in Artificial Intelligence*, 2024. URL https://api.semanticscholar.org/CorpusID:273482612.

Manfred K. Warmuth, Wojciech Kotłowski, and Ehsan Amid. A case where a spindly two-layer linear network decisively outperforms any neural network with a fully connected input layer. In Vitaly Feldman, Katrina Ligett, and Sivan Sabato (eds.), *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, volume 132 of *Proceedings of Machine Learning Research*, pp. 1214–1236. PMLR, 16–19 Mar 2021. URL https://proceedings.mlr.press/v132/warmuth21a.html.

Blake Woodworth, Suriya Gunasekar, Jason D. Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models, 2020.

Liu Ziyin. Symmetry induces structure and constraint of learning. 2023. URL https://api.semanticscholar.org/CorpusID:263310669.

Liu Ziyin and Zihao Wang. spred: Solving l_1 penalty with sgd, 2023. URL https://arxiv.org/abs/2210.01212.

A EXPERIMENTAL DETAILS.

We provide details for the simulation of Figure 1b. We want to learn a teacher neuron that has $a^* = 1$ and $w^* = (1,0,0,0,0)$. So only the first coordinate matters. We initialize $\mathbf{w}_0 \in \mathbb{R}^5$ such that $\lambda = 0$ for the dense and the sparse case i.e. balanced. Moreover for the sparse configuration we apply a mask to the redundant coordinates that become not trainable. For the data set Z we sample $\mathbf{x}_j \sim N(0, I_5)$ for $j \in [500]$ and y_j follows from the teacher setup. We train for 20000 epochs with learning rate 0.01.

B DETAILS ON SPARSE TRAINING METHODS

Identifying effective sparse masks at initialization presents a significant challenge. Consequently, PaI methods often fail to achieve dense model performance, even at moderate sparsity levels. Recent theoretical studies underscore this limitation, indicating that PaI methods cannot match the performance of dense models (Kumar et al., 2024b) and trainability issues due connectivity patterns (Pham et al., 2025). In contrast to PaI, DST methods address this limitation by iteratively updating the sparse mask during training. This dynamic adjustment allows DST methods to explore diverse sparse topologies, enabling performance comparable to that of dense models.

Continuous sparsification. Other methods such as continuous sparsification have also been proposed exploiting reparameterizations leading to more gradual sparsification process (Jacobs & Burkholz, 2025; Kolb et al., 2025b; Ziyin & Wang, 2023; Kusupati et al., 2020; Kolb et al., 2025a). These methods do rely on dense training to reach high performance at least at the start of training. The connection between symmetry breaking and sparsity reveals therefore an underlying reason for these methods to outperform methods such as PaI

Sparse Evolutionary Training (SET). To address the limitations of training with a static sparse mask, (Mocanu et al., 2018b) proposed SET, which initiates training with a randomly generated sparse mask. Since a random sparse mask cannot capture the relationship between the data and the model, a fraction of weights with the lowest magnitudes are removed after a few epochs or steps to maintain the sparsity level. Simultaneously, an equal number of new connections are randomly activated. This dynamic pruning and regrowth of connections allows SET to maintain stochasticity during optimization, achieving better performance than PaI methods. SET is inspired by biological neural networks, particularly the synaptic pruning that occurs during sleep, where weaker synapses are eliminated.

Rigging the Lottery Ticket (RigL). Inspired by SET, RigL, proposed by (Evci et al., 2020a), does not start with a random mask that has uniform layer-wise sparsity. Instead, it uses the Erdős-Rényi-Kernel (ERK) distribution, which considers the channel dimensions and layer widths to generate the initial random sparse mask. Similar to SET, RigL updates the mask during training. However, unlike SET, which grows connections randomly, RigL grows connections based on the gradient norm. Specifically, pruned or masked parameters with the highest gradient values are reactivated. Although RigL achieves performance comparable to dense models, its slower convergence leads to a training FLOP count comparable to that of dense training. The criterion for updating masks in RigL is as follows:

1. Update Schedule: RigL uses a cosine decay schedule to update the mask. Let ΔT denote the number of iterations between each update step, T_{end} denote the iteration at which to stop updating the mask, and α the initial fraction of mask parameters to be updated, then the cosine schedule can be given by:

$$f_{decay}(t;\alpha,T_{end}) = \frac{\alpha}{2}\left(1+\cos\left(\frac{t\pi}{T_{end}}\right)\right) \tag{1}$$

2. Drop Criterion: At each mask update step, RigL removes an f fraction of the weights based on their magnitudes.
3. Grow Criterion: Unlike SET, which reactivates connections randomly, RigL uses dense gradient information to activate connections with the highest gradient norm. Consequently, RigL requires access to dense gradients during each mask update.

C PROOF THEOREM 3.1.

We provide here the proof of the probability of being in the correct subspace. The proof is an application of the inclusion-exclusion principle.

Theorem C.1. *Let $\mathcal{O}^*_{\textbf{\textit{sub}}}$ denote the subspace of the target and $\mathcal{O}_{\textbf{\textit{sub}},\,j}$ for $j \in [N]$ the subspace of each neuron. The probability of a random mask with $\ell \geq m$ non-zero entries to recover the target solution is given by:*

$$\mathbb{P}\left(\mathcal{O}^*_{\textbf{\textit{sub}}} \subset \bigcup_{j=1}^{N} \mathcal{O}_{\textbf{\textit{sub}},\,j}\right) \approx \left(1 - e^{-\ell/d}\right)^m$$

where the approximation holds for $\ell << Nd$.

Proof. Denote the probability $P := \mathbb{P}\left(\mathcal{O}^*_{\textbf{sub}} \subset \bigcup_{j=1}^{N} \mathcal{O}_{\textbf{sub},\,j}\right)$ and define B_i for $i \in [m]$ to be the event that none of the neurons has that feature. Then we can write the probability P as:

$$P = 1 - \mathbb{P}(\cup_{i\in[m]} B_i)$$

By the inclusion-exclusion principle we have:

$$\mathbb{P}(\cup_{i\in[m]} B_i) = \sum_{i=1}^{m} (-1)^{i+1} \sum_{\text{for all sets } |I|=i} \mathbb{P}(\cap_{j\in[I]}\, B_j)$$

The number of sets of size i is $\binom{m}{i}$ and the probability can now be calculated exactly by using that we can choose $\binom{Nd}{\ell}$ items to keep in general and we need to exclude Ni items so $\binom{Nd-Ni}{\ell}$. This then gives the formula:

$$P = \sum_{i=0}^{m} (-1)^i \binom{m}{i} \frac{\binom{Nd-Ni}{\ell}}{\binom{Nd}{\ell}}.$$

We can furthermore approximate the ratio of binomials by an exponential under the assumption that $\ell << Nd$:

$$\frac{\binom{Nd-Ni}{\ell}}{\binom{Nd}{\ell}} \approx e^{-i\ell/d}$$

Then after it follows from using the binomial formula that we have:

$$P \approx \left(1 - e^{-\ell/d}\right)^m$$

□

D CHANGING OPTIMIZERS AND ARCHITECTURE

We consider here two ablations supporting that startomg from the correct orthant is important in two different settings. First, we show that changing the optimization algorithm to AdamW or Muon still leads to a stiking gap between a random orthant and the correct orthant. Furthermore, we consider the training of a DeiT-small Transformer on Imagenet to illustrate that orthant movement remains difficult in different architectures.

Optimizer choice. We conduct a similar experiment as in Gadhikar et al. (2025) with the optimizers AdamW Kingma & Ba (2014); Loshchilov & Hutter (2017) and Muon Jordan et al. (2024). We first train SGD and momentum together with the same sparse training methods on a ResNet-50 and retrain the sparse mask with and without the correct orthant initialization. We use the default optimization hyperparameters of both optimizers. We report the the results in Table 2 and find that also here correct orthant initialization leads to substantially closer target recovery. Note however reaching the target is harder for these optimizers compared to SGD as these are not the default optimizers for the ResNet-50 Imagenet setting.

Method	Optimizer	ACDC	RiGL	STR
Baseline	SGD+M	73.19	73.718	72.5
Random Orthant	AdamW	71.34	71.78	67.18
Correct Orthant	AdamW	72.34	73.26	73.13
Random Orthant	Muon	71.23	72.2	65.29
Correct Orthant	Muon	71.99	72.54	72.08

Table 2: Retraining with different optimizers AdamW and Muon starting from a random and correct orthant.

Transformer architecture. We now consider a similar experiment for a DeiT small. To isolate the studied symmetries in the main text we freeze the attention blocks with the correct weights and obtained mask as these contain the additional rotation symmetry. We also provide the obtained LayerNorm parameters for both the random and correct orthant. The starting mask (90% sparsity) and weights are obtained using AC/DC method and the HORST-AdamW optimizer Jacobs et al. (2026b). The retraining for both orthant configurations is done with the AdamW optimizer and is done for 100 epochs. We report the performance in Table 3. Observe that the correct orthant leads to closer target recovery.

Orthant	ACDC
Baseline	76.73
Correct orthant	74.91
Random orthant	66.24

Table 3: Validation accuracies $(\%)$ for a DeiT-small with mask (90% sparsity) obtained by the ACDC method. Retraining keeps the attention blocks frozen.