HORST: Composing Optimizer Geometries for Sparse Transformer Training

Tom Jacobs TOM.JACOBS@CISPA.DE
CISPA Helmholtz Center for Information Security

Rohan Jain ROHAN.JAIN@CISPA.DE
CISPA Helmholtz Center for Information Security

Rebekka Burkholz BURKHOLZ@CISPA.DE
CISPA Helmholtz Center for Information Security

Abstract

Sparsifying transformers remains a fundamental challenge, as standard optimizers fail to simultaneously encourage sparsity and maintain training stability. Effective adaptive optimizers exhibit an implicit L_∞ bias favoring stability, yet, sparsity requires an L_1 bias. To integrate sparsity, we propose a composition of optimizer steps, which we cast as non-commutative operators to analyze and combine their optimization geometry in a principled way. This yields HORST (Hyperbolic Operator for Robust Sparse Training), a modular optimizer that inherits stability from adaptive methods while inducing L_1 sparsity bias through a hyperbolic mirror map. Our experiments demonstrate its utility for sparse training of transformers on both vision and language tasks. HORST significantly outperforms AdamW baselines across all sparsity levels, with large gains at higher sparsity.

1. Introduction

Transformers [71] have become a dominant architecture across deep learning, achieving state-of-the-art performance in areas such as natural language processing [8] and computer vision [16]. Yet, they incur a steep computational cost: modern transformer models require billions of parameters and enormous memory footprints [30, 38], making deployment at scale a practical challenge. Network sparsification, i.e. the removal of redundant weights, offers a promising route to model compression [28, 45], reducing storage requirements and potentially enabling faster inference on hardware that supports sparse operations [25, 52]. While pruning methods have achieved considerable success in convolutional architectures [17, 34, 60], they work less well for transformers: sparsification consistently leads to significant performance degradation especially at higher sparsity levels where compression would be most practically valuable. This contrast is not incidental. Convolutional networks are trained with standard SGD, while transformers rely on adaptive optimizers such as AdamW [49] to train more stably. We argue that it is precisely this difference in optimization that underlies the difference in sparsifiability. The implicit bias induced by the optimizer during training shapes the geometry of the learned weights, and adaptive optimizers impose a bias that is fundamentally at odds with the structure required for sparse solutions [35, 73]. Figure 1 illustrates this through the standardized weight distributions of a pretrained ResNet-50 [29] (Convolutional) and a DeiT-base [69] (Transformer).

Beyond minimizing the training loss, optimizers impose a preference over which solution is selected when many equivalent ones exist. In the overparameterized regime, where infinitely many

© T. Jacobs, R. Jain & R. Burkholz.

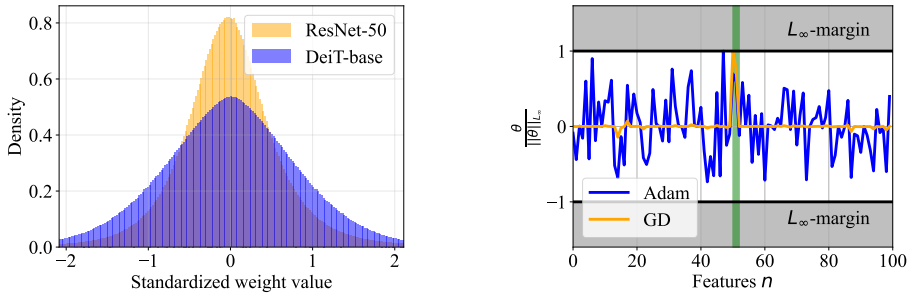

Figure 1: (Left) Standardized weight distributions for a pretrained ResNet-50 (with SGD) and a DeiT-base (with AdamW). The transformer weights are more spread out, which illustrates the different implicit bias (L_2 and L_∞). (Right) Linear separable binary classification with a deep diagonal linear network $\langle u \odot v, x \rangle$ where $\theta = u \odot v$. GD learns the sparse features (green) while Adam overfits on the data.

parameter configurations achieve zero training loss, this preference (i.e. the implicit bias of the optimizer) is influenced by the geometry of the update rule [27]. Gradient descent, for instance, finds minimum L_2-norm solutions in simple settings such as linear regression or the L_2-max margin in binary classification with homogeneous neural networks [51]. Adaptive optimizers such as Adam [39] are closely related to sign descent [6, 35, 70], a form of steepest descent with respect to the L_∞-norm, and accordingly exhibit an L_∞ implicit bias: weight magnitudes tend to equalize across parameters rather than concentrate. This is exactly the wrong bias for sparsity, which requires weight mass to concentrate on a small subset of parameters while the rest are driven to zero. However in the steepest descent family with respect to a general norm $||\cdot||$, inducing an L_1 bias would require taking steps with respect to the L_1-norm, which corresponds to coordinate descent, updating only the single largest gradient coordinate at each step, a procedure that is not practical for large deep networks as schematically illustrated in Figure 2.

A different optimizer geometry class, mirror descent, offers a different route: by replacing the Euclidean geometry of gradient descent with one induced by a mirror map $\nabla R : \mathbb{R}^n \to \mathbb{R}^n$, the parameters can also be made to favor sparse solutions [32, 67]. In particular, the hyperbolic entropy mirror map is known to induce an L_1 bias, promoting sparsity [58, 75]. This idea was recently exploited in the HAM optimizer [34], which alternates a standard optimizer step with a hyperbolic mirror step, and was shown to improve sparse training of convolutional networks. However, transformers present a fundamentally different challenge: they leverage a different geometric bias (L_∞) for stable training. This is incompatible with standard mirror descent which directly uses the gradient. Therefore, an explanation for the tension between modern optimizers and sparsity is a geometric one. In this work, we propose to combine them in a principled way, leveraging stability of adaptive methods and the L_1 sparsity bias of mirror descent.

We formalize this tension by treating optimization steps as functional operators [5] and identifying a structural dichotomy: steepest descent is especially well suited to the L_∞ bias that stabilizes adaptive training, while mirror descent is especially well suited to the L_1 bias needed for sparsity. Relying on either class alone therefore comes at a cost. We show that composing both operations in the right order resolves this conflict, yielding HORST (Hyperbolic Operator for Robust Sparse Training), a modular optimizer that integrates seamlessly with standard adaptive methods. We

validate HORST on vision transformers and show consistent and significant improvements over AdamW baselines across sparsity levels, with the largest gains in the high-sparsity regime where adaptive optimizers struggle most.

2. Composable optimizer operators and their structural properties

We treat optimization updates as functional operators, maps that take a parameter vector and a gradient (oracle) and return a step direction. This abstraction lets us reason about optimizers at the level of what their effect is on the gradients, which is precisely the level at which the dichotomy we identify becomes legible (Figure 2). The functional formulation is motivated by the modularity framework proposed in Bernstein and Newhouse [5]. The key idea of modularity is that an optimizer can be applied to a group of parameters: rather than a single operation acting on all parameters simultaneously, the operator acts independently on subsets of parameters, layers, heads, or individual weights. In similar spirit we can compose update rules: two operators can be applied in sequence, each acting on the output of the previous one, and the result is itself a valid optimizer.

	L_∞ **bias**	L_1 **bias**
Steepest descent	Sign descent ($\simeq$ Adam) *bounded, stable*	Coordinate descent *infeasible for large networks*
Mirror descent	cosh entropy *loses coercivity, stalls*	Hyperbolic entropy *unbounded, sparsity*

Figure 2: **Steepest-Mirror Descent Dichotomy:** Each geometric optimization class is effective at inducing the corresponding dual implicit bias. Both coordinate descent and cosh-entropy are infeasible due to slow convergence.

Optimizer operators. Since our overarching goal is parameter sparsity we consider a modular objective on the parameter level $f : \mathbb{R}^n \to \mathbb{R}$ with gradient oracle $g : \mathbb{R}^n \to \mathbb{R}^n$ and parameters $\theta \in \mathbb{R}^n$. The gradient oracle and parameters together with some geometric quantity, a norm or a metric tensor, form a triple that characterizes the operator. Given two operators A and B, their composition is defined by

$$AB(\theta, g) := (A \circ B)(\theta, g) := A(\theta,\ B(\theta, g))\,,$$

that is, B produces a modified gradient direction and A then acts on that direction instead of the gradient itself. Moreover, in general the update rule is given by: $\theta_{k+1} = \theta_k - \eta A(\theta_k, g_k)$ for $k \in [T]$, where $\eta > 0$ is the learning rate and (θ_k, g_k) are the parameter and gradient estimate at iteration k. In §C we recall basic properties of operators: boundedness, linearity and commutativity.

Definition 1 *(Steepest operator) Given the triple $(g, \theta, ||\cdot||)$ the steepest descent operator $S_{||\cdot||} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ with respect to a norm $||\cdot||$ is defined as:*

$$S_{||\cdot||}(\theta, g) := \textit{arg max}_{||d||\leq 1}\langle g(\theta), d\rangle \qquad \textit{such that} \qquad \theta_{k+1} = \theta_k - \eta S_{||\cdot||}(\theta_k, g_k) \qquad \textit{for } k \in [T].$$

Definition 2 *(Mirror operator) Given the triple $(g, \theta, \nabla R)$ the mirror descent operator $M_R : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ with respect to the mirror map R is defined as:*

$$M_R(\theta, g) := \nabla^2 R^{-1}(\theta) g(\theta) \qquad \textit{such that} \qquad \theta_{k+1} = \theta_k - \eta M_R(\theta_k, g_k) \qquad \textit{for } k \in [T].$$

Lemma 3 (See also Lemma 17.) *The steepest operator is bounded but not linear.*

Lemma 4 (See also Lemma 18) *The mirror operator is linear and unbounded in general.*

Together, Lemmas 3 and 4 confirm the dichotomy summarized in Figure 2: each class has exactly the property the other lacks.

Implicit bias of composition. Corollary 12 motivates the bounded composed operator $M_R S_{\|\cdot\|}$. The ordering is essential because the two operators do not commute (Corollary 11): by Lemma 10, applying the mirror map *first* yields $S_{\|\cdot\|_\infty} M_R = S_{\|\cdot\|_\infty}$, so sign descent erases the mirror map and only the L_∞ bias remains.

$$M_R S_{\|\cdot\|}(\theta, g) = \nabla^2 R^{-1}(\theta) \operatorname*{arg\,max}_{\|d\|\leq 1} \langle g(\theta), d\rangle \qquad \theta_{k+1} = \theta_k - \eta M_R S_{\|\cdot\|}(\theta_k, g_k) \qquad \text{for } k \in [T].$$

Concretely, we focus here on the entropy mirror map $\nabla R(\theta) = \log\theta$ and steepest descent norm $||\cdot||_p$ with $p \in [2, \infty)$. These are closely related to the Hyperbolic entropy ($\gamma \to 0$) and sign descent ($p \to \infty$). We can characterize the implicit bias in the binary classification setting on linear separable data for the continuous-time flow ($\eta \to 0$), described by the steepest-mirror differential inclusion:

$$\frac{d\theta_t}{dt} \in -\nabla^2 R^{-1}(\theta_t)\,\mathrm{sign}(\nabla\mathcal{L}(\theta_t))|\nabla\mathcal{L}(\theta_t)|^{q-1}/||\nabla\mathcal{L}(\theta_t)||_q \quad \text{where } q \text{ is such that } \frac{1}{p}+\frac{1}{q}=1. \quad (1)$$

Theorem 5 *Consider steepest-mirror descent with respect to the L_p-norm, $p \in \mathbb{N}_{\geq 2}$, and mirror map $\nabla R(\theta) = \log(\theta)$. Then the iterates of Eq. (1) converge in direction to a KKT point of:*

$$\min\ \|\theta\|_1 \qquad \textit{such that} \qquad y_i\langle\theta, x_i\rangle \geq 1 \quad \textit{for all } i \in [K].$$

Proof This can be shown by using the steepest mirror flow connection to homogeneous reparameterization as developed in [35] and combining it with the steepest flow characterization of the max margin in [70]. For the full proof see Theorem 19. ■

Theorem 5 establishes that the entropy mirror map $\nabla R(\theta) = \log\theta$ can overwrite the implicit bias of steepest descent with respect to an L_p-norm for $p \in [2, \infty)$. Moreover, this suggests that the sparsity bias may extend to the limiting regime ($p \to \infty$, $\gamma \to 0$). However, establishing this limit rigorously remains an open problem.

3. HORST: optimizer for sparse transformers

We first recall the HAM optimizer scheme [34] in Algorithm 1. This alternates a base optimizer (AdamW) with an exponential step which corresponds to using the mirror map $\nabla R(\theta) = \log(\theta)$ as used in Theorem 5. Note that this entropy mirror map does not have full domain $\mathbb{R}^n$ and is not inversely coercive on its own motivating the alternating scheme. Next, the hyperparameter $\alpha \geq 0$ is used to determine the relative importance of the updates. Moreover, extra (decoupled) regularization is used to induce sparsity as a default captured by the hyperparameter $\beta \geq 0$. This motivates three design choices for HORST (Algorithm 2) to turn the composition into a practical optimizer: 1) We replace the steepest descent step with Adam in the exponential update. 2) We use decoupled weight decay with strength $\lambda > 0$. 3) We adopt an alternating scheme from [34] to recover inverse coercivity that applies the exponential update rule after the standard optimizer step, with a hyperparameter $\alpha \geq 0$ that controls the relative strength of the sparsity bias.

4. Experimental validation

Toy illustration. We first show the importance of composing on Adam with the entropy mirror map itself. We consider binary linear classification on linear separable data. The target solution is sparse and consists of two features. We describe the details of the setting in Appendix H. As we show in Figure 3 (Right), using an entropy mirror map after Adam (Exp-Adam) leads to more sparsity, recovering the target solution better highlighted by the green band. In contrast composing in the opposite way (Adam-Exp) leads to overfitting as was the case for Adam in Figure 1.

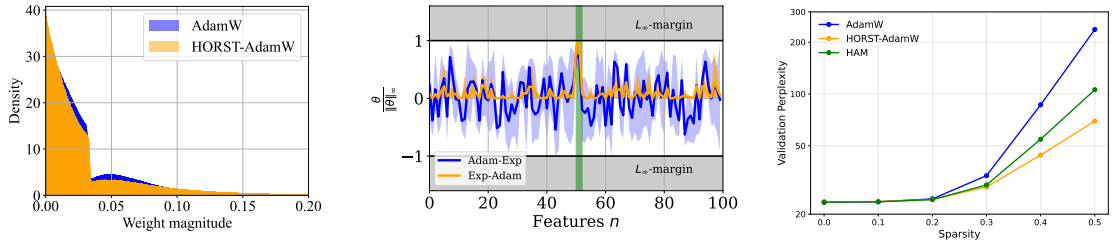

Figure 3: (Left) The final unmasked weight distribution of a DeiT-base trained with AC/DC to 70% sparsity. HORST successfully concentrates more weights at zero finding better generalizable sparse solutions. (Middle) Linear separable binary classification with a diagonal linear network $\langle \theta, x \rangle$, Exp-Adam learns a sparser representation while Adam-Exp does not, this is in line with Lemma 10. (Right) One-shot layerwise unstructured magnitude pruning of dense GPT-2 Small checkpoints trained on SlimPajama-6B with AdamW vs. HAM vs. HORST-AdamW; no fine-tuning. HORST-AdamW achieves lower validation perplexity than both. See Table 6 for details.

Vision and language transformer experiments. We evaluate HORST-AdamW on both vision and language transformer settings to test whether its sparsity-inducing optimizer geometry improves performance under pruning. Unless otherwise noted we use the hyperparameters $\alpha = 5, \beta = 0$ for HORST.[1] In the vision experiments, HORST-AdamW is combined with the AC/DC sparsification [60] pipeline and applied to DeiT-base and DeiT-small models on ImageNet [14] across high sparsity levels, for details see Appendix I. Compared with the AdamW baseline, HORST-AdamW consistently achieves higher validation accuracy, with the largest gains appearing at $80 - 90\%$ sparsity reported in Table 1. In the language experiments, dense GPT-2 Small [64] checkpoints trained on SlimPajama-6B [66] are evaluated using one-shot layerwise unstructured magnitude pruning without fine-tuning. HORST-AdamW yields lower validation perplexity after pruning than both AdamW and HAM (Figure 3), indicating that the proposed optimizer produces weights that are more robust to sparse compression, for further details see Appendix J. Overall, these experiments show that composing AdamW with a hyperbolic mirror-style update improves sparse transformer performance across both vision and language tasks while noticeably changing the global weight distribution as shown in Figure 3 and 7.

1. The * indicates that $\beta = 1e - 2$ was used instead of $\beta = 0$, which gave a slightly better improvement.

5. Conclusion

We have developed a compositional perspective on optimization by treating update rules as operators and analyzing their geometric and algebraic structure. This reveals a fundamental dichotomy: steepest descent operators are bounded but nonlinear and naturally associated with stability and L_∞-type implicit bias, while mirror descent operators are linear but generally unbounded and can induce sparsity-promoting L_1 structure through the (hyperbolic) entropy mirror map motivating HORST.

Acknowledgments We gratefully acknowledge the Gauss Centre for Supercomputing e.V. for funding this project by providing computing time on the GCS Supercomputer JUWELS at Jülich Supercomputing Centre (JSC) [37]. We are also grateful for funding from the European Research Council (ERC) under the Horizon Europe Framework Programme (HORIZON) for proposal number 101116395 SPARSE-ML.

References

[1] 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.

[2] 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 *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=o9JP7N8YO9.

[3] Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/c0c783b5fc0d7d808f1d14a6e9c8280d-Paper.pdf.

[4] Shahar Azulay, Edward Moroshko, Mor Shpigel Nacson, Blake E Woodworth, Nathan Srebro, Amir Globerson, and Daniel Soudry. On the implicit bias of initialization shape: Beyond infinitesimal mirror descent. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 468–477. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/azulay21a.html.

[5] Jeremy Bernstein and Laker Newhouse. Modular duality in deep learning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=hErdffTsLu.

[6] Jeremy Bernstein, Jiawei Zhao, Kamyar Azizzadenesheli, and Anima Anandkumar. signSGD with majority vote is communication efficient and fault tolerant. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BJxhijAcY7.

[7] Jeremy Bernstein, Jiawei Zhao, Markus Meister, Ming-Yu Liu, Anima Anandkumar, and Yisong Yue. Learning compositional functions via multiplicative weight updates. *ArXiv*, abs/2006.14560, 2020. URL https://api.semanticscholar.org/CorpusID:220055863.

[8] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual*

Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

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

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

[11] Tianlong Chen, Yu Cheng, Zhe Gan, Lu Yuan, Lei Zhang, and Zhangyang Wang. Chasing sparsity in vision transformers: An end-to-end exploration. In *Neural Information Processing Systems*, 2021. URL https://api.semanticscholar.org/CorpusID:235367934.

[12] Hung-Hsu Chou, Holger Rauhut, and Rachel A. Ward. Robust implicit regularization via weight normalization. *ArXiv*, abs/2305.05448, 2023. URL https://api.semanticscholar.org/CorpusID:258564503.

[13] Hung-Hsu Chou, Johannes Maly, and Dominik Stöger. How to induce regularization in linear models: A guide to reparametrizing gradient flow, 2024.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[15] 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.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[17] 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*, pages 2943–2952. PMLR, 2020.

[18] Mathieu Even, Scott Pesme, Suriya Gunasekar, and Nicolas Flammarion. (s)gd over diagonal linear networks: Implicit bias, large stepsizes and edge of stability. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 29406–29448. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/5da6ce80e97671b70c01a2e703b868b3-Paper-Conference.pdf.

[19] Chen Fan, Mark Schmidt, and Christos Thrampoulidis. Implicit bias of spectral descent and muon on multiclass separable data. In *High-dimensional Learning Dynamics 2025*, 2025. URL https://openreview.net/forum?id=tirGweSx3a.

[20] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019.

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

[22] 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.

[23] 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.

[24] Advait Harshal Gadhikar, Sohom Mukherjee, and Rebekka Burkholz. Why random pruning is all we need to start sparse. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10542–10570. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/gadhikar23a.html.

[25] Trevor Gale, Matei Zaharia, Cliff Young, and Erich Elsen. Sparse GPU kernels for deep learning. In Christine Cuicchi, Irene Qualters, and William T. Kramer, editors, *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2020, Virtual Event / Atlanta, Georgia, USA, November 9-19, 2020*, page 17. IEEE/ACM, 2020. URL https://doi.org/10.1109/SC41405.2020.00021.

[26] Suriya Gunasekar, Blake E Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Implicit regularization in matrix factorization. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/58191d2a914c6dae66371c9dcdc91b41-Paper.pdf.

[27] Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1832–1841. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/gunasekar18a.html.

[28] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems*, volume 28, 2015.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. URL https://doi.org/10.1109/CVPR.2016.90.

[30] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and L. Sifre. Training compute-optimal large language models. In *Thirty-sixth Conference on Neural Information Processing Systems*, 2022. URL https://api.semanticscholar.org/CorpusID:247778764.

[31] Yuan Hua, Jilin Zhang, Yingtao Zhang, Wenqi Gu, Leyi You, Baobo Xiong, Carlo Vittorio Cannistraci, and Hong Ming Chen. Cannistraci-hebb training on ultra-sparse spiking neural networks. *ArXiv*, abs/2511.05581, 2025. URL https://api.semanticscholar.org/CorpusID:282912995.

[32] 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.

[33] 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.

[34] Tom Jacobs, Advait Gadhikar, Celia Rubio-Madrigal, and Rebekka Burkholz. Hyperbolic aware minimization: Implicit bias for sparsity. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=XKB5Hu0ACY.

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

[36] 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/.

[37] Jülich Supercomputing Centre. JUWELS Cluster and Booster: Exascale Pathfinder with Modular Supercomputing Architecture at Juelich Supercomputing Centre. *Journal of large-scale research facilities*, 7:A183, 2021. doi: 10.17815/jlsrf-7-183.

[38] Jared Kaplan, Sam McCandlish, Thomas Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeff Wu, and Dario Amodei. Scaling laws for neural language models. *ArXiv*, abs/2001.08361, 2020. URL https://api.semanticscholar.org/CorpusID:210861095.

[39] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

[40] 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*, 2025. URL https://openreview.net/forum?id=vNdOHr7mn5.

[41] Chris Kolb, Christian L. Müller, Bernd Bischl, and David Rügamer. Smoothing the edges: Smooth optimization for sparse regularization using hadamard overparametrization. *Machine Learning*, 115(4):87, 2026. ISSN 1573-0565. doi: 10.1007/s10994-026-06997-0. URL https://doi.org/10.1007/s10994-026-06997-0.

[42] 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.

[43] Tim Large, Yang Liu, Minyoung Huh, Hyojin Bahng, Phillip Isola, and Jeremy Bernstein. Scalable optimization in the modular norm. *CoRR*, abs/2405.14813, 2024. URL https://doi.org/10.48550/arXiv.2405.14813.

[44] Mike Lasby, Anna Golubeva, Utku Evci, Mihai Nica, and Yani Ioannou. Dynamic sparse training with structured sparsity. *arXiv preprint arXiv:2305.02299*, 2023.

[45] Yann LeCun, John S. Denker, and Sara A. Solla. Optimal brain damage. In David S. Touretzky, editor, *Advances in Neural Information Processing Systems 2, [NIPS Conference, Denver, Colorado, USA, November 27-30, 1989]*, pages 598–605. Morgan Kaufmann, 1989. URL http://papers.nips.cc/paper/250-optimal-brain-damage.

[46] Dongyeop Lee, Kwanhee Lee, Jinseok Chung, and Namhoon Lee. SAFE: Finding sparse and flat minima to improve pruning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=10l1pGeOcK.

[47] 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.

[48] Zhiyuan Li, Tianhao Wang, Jason D. Lee, and Sanjeev Arora. Implicit bias of gradient descent on reparametrized models: On equivalence to mirror descent. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=k4KHXS6_zOV.

[49] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017. URL https://api.semanticscholar.org/CorpusID:53592270.

[50] Yannick Lunk, Sebastian James Scott, and Leon Bungert. Sparse training of neural networks based on multilevel mirror descent. *ArXiv*, abs/2602.03535, 2026. URL https://api.semanticscholar.org/CorpusID:285276107.

[51] Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. *ArXiv*, abs/1906.05890, 2019. URL https://api.semanticscholar.org/CorpusID:189898036.

[52] Asit Mishra, Jorge Albericio Latorre, Jeff Pool, Darko Stosic, Dusan Stosic, Ganesh Venkatesh, Chong Yu, and Paulius Micikevicius. Accelerating sparse deep neural networks. *arXiv preprint arXiv:2104.08378*, 2021.

[53] Decebal Constantin Mocanu, Elena Mocanu, Peter Stone, Phuong H Nguyen, Madeleine Gibescu, and Antonio Liotta. Evolutionary training of sparse artificial neural networks: a network science perspective. 2017.

[54] Sree Harsha Nelaturu, Advait Gadhikar, and Rebekka Burkholz. TurboPrune: High-Speed Distributed Lottery Ticket Training, 2024. URL https://github.com/nelaturuharsha/TurboPrune.

[55] Keigo Nishida, Eren Mehmet Kıral, Kenichi Bannai, Mohammad Emtiyaz Khan, and Thomas Möllenhoff. Log-normal multiplicative dynamics for stable low-precision training of large networks, 2025. URL https://arxiv.org/abs/2506.17768.

[56] Antonio Orvieto and Robert Gower. In search of adam's secret sauce. *ArXiv*, abs/2505.21829, 2025. URL https://api.semanticscholar.org/CorpusID:278959876.

[57] Hristo Papazov, Scott Pesme, and Nicolas Flammarion. Leveraging continuous time to understand momentum when training diagonal linear networks. In *AISTATS*, pages 3556–3564, 2024. URL https://proceedings.mlr.press/v238/papazov24a.html.

[58] Scott Pesme, Loucas Pillaud-Vivien, and Nicolas Flammarion. Implicit bias of SGD for diagonal linear networks: a provable benefit of stochasticity. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=vvi7KqHQiA.

[59] Scott Pesme, Radu-Alexandru Dragomir, and Nicolas Flammarion. Implicit bias of mirror flow on separable data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=wiMaws0FWB.

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

[61] Thomas Pethick, Wanyun Xie, Kimon Antonakopoulos, Zhenyu Zhu, Antonio Silveti-Falls, and Volkan Cevher. Training deep learning models with norm-constrained lmos. *arXiv preprint arXiv:2502.07529*, 2025.

[62] Hoang Pham, The-Anh Ta, Shiwei Liu, Lichuan Xiang, Dung D. Le, Hongkai Wen, and Long Tran-Thanh. Towards data-agnostic pruning at initialization: What makes a good sparse mask? In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=xdOoCWCYaY.

[63] 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*, 2026. URL https://openreview.net/forum?id=EEZLBhyer1.

[64] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI*, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

[65] Andrei Semenov, Matteo Pagliardini, and Martin Jaggi. Benchmarking Optimizers for Large Language Model Pretraining. *arXiv preprint arXiv:2509.01440*, 2025. URL https://arxiv.org/abs/2509.01440.

[66] Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama. https://cerebras.ai/blog/slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama, 2023. URL https://huggingface.co/datasets/cerebras/SlimPajama-627B.

[67] Haoyuan Sun, Kwangjun Ahn, Christos Thrampoulidis, and Navid Azizan. Mirror descent maximizes generalized margin and can be implemented efficiently. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=0SVOleKNRAU.

[68] Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models. *ArXiv*, abs/2306.11695, 2023. URL https://api.semanticscholar.org/CorpusID:259203115.

[69] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, Proceedings of Machine Learning Research, pages 10347–10357. PMLR, 2021. URL http://proceedings.mlr.press/v139/touvron21a.html.

[70] 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.

[71] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[72] Tomas Vaškevičius, Varun Kanade, and Patrick Rebeschini. Implicit regularization for optimal sparse recovery, 2019. URL https://arxiv.org/abs/1909.05122.

[73] Manfred K Warmuth, Wojciech Kotlowski, Matt Jones, and Ehsan Amid. How rotation invariant algorithms are fooled by noise on sparse targets. In *36th International Conference on Algorithmic Learning Theory*, 2025. URL https://openreview.net/forum?id=GSgcnyQzpz.

[74] Stephan Wojtowytsch. Stochastic gradient descent with noise of machine learning type. part i: Discrete time analysis, 2021.

[75] Blake Woodworth, Suriya Gunasekar, Jason D. Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. In Jacob Abernethy and Shivani Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 3635–3673. PMLR, 09–12 Jul 2020. URL https://proceedings.mlr.press/v125/woodworth20a.html.

[76] Shuo Xie and Zhiyuan Li. Implicit bias of adamw: ℓ_∞-norm constrained optimization. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=CmXkdlO6JJ.

[77] Lu Yin, You Wu, Zhenyu Zhang, Cheng-Yu Hsieh, Yaqing Wang, Yiling Jia, Mykola Pechenizkiy, Yi Liang, Zhangyang Wang, and Shiwei Liu. Outlier weighed layerwise sparsity (owl): A missing secret sauce for pruning llms to high sparsity. *arXiv preprint arXiv:2310.05175*, 2023.

[78] Chenyang Zhang, Difan Zou, and Yuan Cao. The implicit bias of adam on separable data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=xRQxan3WkM.

[79] Peng Zhao, Yun Yang, and Qiao-Chu He. High-dimensional linear regression via implicit regularization. *Biometrika*, 109(4):1033–1046, February 2022. ISSN 1464-3510. doi: 10.1093/biomet/asac010. URL http://dx.doi.org/10.1093/biomet/asac010.

[80] Liu Ziyin and Zihao Wang. spred: Solving l1 penalty with sgd. In *International Conference on Machine Learning*, 2022. URL https://api.semanticscholar.org/CorpusID:259075663.

Appendix A. Experimental results

Here we present the most important experimental validations on scale for our proposed algorithm. In Table 1 we show how HORST improves the sparse training backbone AC/DC. Moreover, we include an additional experiment comparing with HAM in Table 3. On DeiT-small at 90% sparsity, HAM provides a modest improvement over the AdamW baseline, whereas HORST substantially outperforms both HAM and the baseline. This supports the importance of composing the exponential mirror-style update with the AdamW update itself, rather than applying the gradient-based exponential step used in HAM. In Figure 3 we show that HORST improves validation perplexity over HAM and AdamW baselines for GPT2 Small training.

Table 1: Performance in terms of validation accuracy for a DeiT model with different optimizers and the AC/DC sparsification method [60].

Model	Optimizer	70%	80%	90%
DeiT-base	AdamW	76.15 ± 0.71	74.18 ± 0.29	68.86 ± 2.05
DeiT-base	HORST-AdamW	$\mathbf{78.89 \pm 0.79}$	$\mathbf{78.63 \pm 1.26}$	$\mathbf{76.83 \pm 0.92}$
DeiT-small	AdamW	71.59 ± 1.12	67.07 ± 0.31	59.64 ± 1.52
DeiT-small	HORST-AdamW	$\mathbf{78.42 \pm 0.39}$	$\mathbf{75.83 \pm 0.47}$*	$\mathbf{69.41 \pm 2.75}$*

Appendix B. Algorithms

Here we provide pseudo code for our proposed algorithm HORST and for completeness also the algorithm for HAM. Note that setting $\alpha = 0$ recovers the base optimizer (AdamW), while increasing α strengthens the sparsity bias. Similar as HAM, HORST requires no additional gradient evaluations; its extra cost is a per-parameter exponential update. Moreover, η is the learning rate, λ the decoupled weight decay strength, and for completeness $\beta \geq 0$ the extra regularization parameter as in HAM. Observe that β for HORST is doing the same work as the decoupled weight decay $\lambda > 0$.

Algorithm 1 HAM

Input: steps T, schedule η, initialization θ_{init}, extra regularization $\beta \geq 0$, weight decay $\lambda \geq 0$, relative lr $\alpha \geq 0$.

for $k = 0, \ldots, T-1$ **do**

$$\theta_{k+\frac{1}{2}} \leftarrow \theta_k - \mathrm{Adam}(g(\theta_k), \eta) - \eta\lambda\theta_k$$

$$\theta_{k+1} \leftarrow \theta_{k+\frac{1}{2}} \exp\Big(-\eta \left(\alpha \,\mathrm{sign}(\theta_{k+\frac{1}{2}})\, g(\theta_k) + \beta\right)\Big)$$

end

return Model weights θ_T

Algorithm 2 HORST

Input: steps T, schedule η, initialization θ_{init}, extra regularization $\beta \geq 0$, weight decay $\lambda \geq 0$, relative lr $\alpha \geq 0$.

for $k = 0, \ldots, T-1$ **do**

$$\theta_{k+\frac{1}{2}} \leftarrow \theta_k - \mathrm{Adam}(g(\theta_k), \eta) - \lambda\eta\,\theta_k$$

$$\theta_{k+1} \leftarrow \theta_{k+\frac{1}{2}} \exp\Big(-\alpha \,\mathrm{sign}(\theta_{k+\frac{1}{2}})\, \mathrm{Adam}(g(\theta_k), \eta) - \eta\beta\Big) \qquad \texttt{// Composition}$$

end

return Model weights θ_T

Appendix C. Properties and examples

Operator properties. For studying the two operator classes, we define the structural properties we will use to characterize them. These properties are not desiderata we impose; rather, we will show in § D that each class naturally satisfies one and violates the other, and it is this asymmetry that makes composition the right tool.

Definition 6 *An operator A is linear if and only if it satisfies the following properties:*

- *for any $g, h \in \mathbb{R}^n$ and $\theta \in \mathbb{R}^n$ we have $A(\theta, g + h) = A(\theta, g) + A(\theta, h)$.*
- *for any $g \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}$ we have $A(\theta, \lambda g) = \lambda A(\theta, g)$.*

Definition 7 *An operator A is bounded if and only if there is a constant $C \geq 0$ independent of (θ, g) such that for all $\theta \in \mathbb{R}^n$ and $g \in \mathbb{R}^n$:*

$$||A(\theta, g)|| \leq C(1 + ||\theta||)$$

In Definition 7 we do not specify the norm; since all norms on $\mathbb{R}^n$ are equivalent. Intuitively, boundedness is a stability condition: it says that the optimizer's step size stays controlled, instead of becoming arbitrarily large when the gradient is large.

Definition 8 *Two operators A, B are commutative if their Lie Bracket $[A, B] := AB - BA = 0$.*

Commutativity will matter because, as we show in § D, the two operator classes do not commute. The order of composition therefore determines what is inherited from each class, and the specific ordering we propose is what allows HORST to simultaneously realize stability and a sparsity bias.

Steepest and mirror descent operators examples. We now recall examples of the two operator classes. Each is characterized by its own geometric quantity in the triple $(g, \theta, \cdot)$: a norm $|| \cdot || : \mathbb{R}^n \to \mathbb{R}$ for steepest descent, and a mirror potential $\nabla R : \mathbb{R}^n \to \mathbb{R}^n$ for mirror descent.

Example 1 *(Sign descent) Consider the triple $(g, \theta, || \cdot ||_\infty)$ then $S_{||\cdot||}(\theta, g) = sign\,(g(\theta))$.*

Example 2 *(Coordinate descent) Consider the triple $(g, \theta, \| \cdot \|_1)$; then $S_{\|\cdot\|}(\theta, g) = s_i\, e_i$ where $i = \arg\max_j |g_j(\theta)|$ and e_i denotes the $i-$th basis vector and $s_i := sign(g_i(\theta))$ the sign of the chosen gradient.*

Observe that the steepest operator is scale-invariant: only the direction of $g(\theta)$ matters, not its magnitude, so doubling the gradient leaves the step unchanged. Similarly Adam also normalizes the update [35, 70], which is precisely what makes it stable for transformer training. As we show in § D, scale-invariance implies the operator being bounded but not linear.

A second way to introduce geometry into the update is via a mirror potential, a convex function $R : \mathbb{R}^n \to \mathbb{R}$ whose Hessian $\nabla^2 R(\theta)$ defines a local metric at each point. Rather than projecting the gradient onto a norm ball, the mirror operator rescales it by the inverse metric. For this to be well-defined and well-behaved, we need to make assumptions on R we highlight the most important for our goal of contrasting the different implicit biases.

Definition 9 *(Inverse coercive) Let $R : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ be twice differentiable convex function. We say that R is inversely $\mu-$coercive iff there exists a constant $\mu > 0$, the coercivity constant, such that for all $\theta, \xi \in \mathbb{R}^n$:*

$$\xi^T \left(\nabla^2 R(\theta)\right)^{-1} \xi \geq \mu ||\xi||_{L_2}^2.$$

In Definition 9 inverse coercivity is the condition that controls how aggressively the mirror map rescales the gradient, moreover if the objective f satisfies the PL-inequality (Definition 16) it recovers linear convergence in the gradient flow setting [32]. For completeness, in Definition 15 (Legendre and Bregman) we recall two characterizations that make the mirror operator well-behaved. As the examples below illustrate, gradient descent and Hyperbolic entropy mirror maps that induce an L_2 and L_1 bias are inverse coercive with $\mu = 1$ and $\mu = \gamma$ respectively, while the $\cosh$-entropy that induces an $L_\infty-$bias is not.

Example 3 *(Gradient descent, L_2-bias) Consider $R(\theta) = \frac{1}{2}||\theta||_2^2$ then $M_R(\theta, g) = g(\theta)$ and satisfies all properties in Definition 9.*

Example 4 *(Hyperbolic entropy, L_1-bias) Consider $R_\gamma(\theta) = \sum_{i=1}^n \theta_i arcsinh(\frac{\theta_i}{\gamma}) - \sqrt{\theta_i^2 + \gamma^2}$ then $M_R(\theta, g) = \sqrt{\theta^2 + \gamma^2} g(\theta)$ and satisfies all properties in Definition 9 when $\gamma > 0$. Moreover this is related to the entropy mirror map $\nabla R(\theta) = \log \theta$ by taking $\gamma \to 0$.*

Example 5 *($\cosh$-entropy, L_∞-bias [59]) Consider $R(\theta) = \sum_{i=1}^n \cosh(\theta_i)$, then $M_R(\theta, g) = \frac{1}{\cosh(\theta)} g(\theta)$. This is a Legendre function and is admissible as a mirror map, but fails inverse μ-coercivity: the rescaling $1/\cosh(\theta)$ vanishes as $|\theta| \to \infty$, which progressively suppresses gradient updates for large weights and can lead to slow convergence.*

Appendix D. Intuition behind operator characterization

We provide more intuition behind the two operator classes defined in § 2. The steepest operator is well-behaved in terms of step size, it always returns a bounded direction, but it achieves this by throwing away gradient magnitude entirely, which is a nonlinear operation. The mirror operator, by contrast, rescales the gradient smoothly and proportionally, which is linear, but has no built-in ceiling on how large the step can be. It explains why the $\cosh$-entropy fails as a route to L_∞ stability: to equalize gradient coordinates via a linear rescaling, the inverse Hessian of R must damp large components and amplify small ones, which forces it to decay for large weights and lose coercivity, and with that linear convergence [32]. The steepest operator achieves the same bias through scale-invariance rather than linear rescaling, and is bounded by construction. Stability is therefore better achieved by steepest descent, not mirror descent.

Non-commutative and inheritance. Since the two classes have complementary properties, a natural question is whether composing them yields an operator that inherits both. The answer depends critically on the order of composition: one ordering leads to steepest operator alone, while the other, used in HORST, leverages both geometries. This is due to the non-commutative nature between the operator classes.

Lemma 10 *Consider $S_{||\cdot||_\infty}$ and M_R with R any separable mirror map then $S_{||\cdot||_\infty} M_R = S_{||\cdot||_\infty}$.*

Proof. This follows from a direct calculation. We know that R is separable and its Hessian is positive definite and $S_{||\cdot||_\infty}$ corresponds to sign descent this gives for any $\theta \in \mathbb{R}^n$ and $g \in \mathbb{R}^n$:

$$S_{||\cdot||_\infty} M_R(\theta, g) = \text{sign}(\nabla^2 R^{-1}(\theta) g(\theta)) = \text{sign}(g(\theta)) = S_{||\cdot||_\infty}(\theta, g). \qquad \square$$

Corollary 11 *The mirror and steepest operator do not commute i.e. $[S_{||\cdot||}, M_R] \neq 0$.*

Proof. Consider the the setting of Lemma 10. Then we know that $S_{||\cdot||_\infty} M_R = S_{||\cdot||_\infty}$. But now consider a mirror map that is not $\frac{1}{2}||\theta||_2^2$ we have that there is a $\theta \in \mathbb{R}^n$ such that:

$$M_R S_{||\cdot||_\infty}(\theta, g) = \nabla^2 R^{-1}(\theta)\ \text{sign}(g(\theta)) \neq \text{sign}(g(\theta)).$$

Therefore we do not have $[S_{||\cdot||_\infty}, M_R] = 0$ leading to a contradiction. $\square$

Lemma 10 and Corollary 11 show that the order of composition matters: if we apply the mirror map first, sign descent removes the effect of the mirror map. If we apply it second, the mirror map gets to act on the sign step and can potentially introduce the L_1 sparsity bias.

Thus, we have shown that the steepest operator is bounded but not linear, and the mirror operator is linear but not bounded. Each class has exactly what the other lacks, and composing them in the right order is what allows HORST to get the best of both worlds as we will show next in Corollary 12.

Corollary 12 *For any norm $\|\cdot\|$ and any separable mirror map R such that $||\nabla^2 R^{-1}(\theta)|| \leq C(1 + ||\theta||)$ for all $\theta \in domR$, the composed operator $M_R S_{||\cdot||}$ is bounded.*

Proof. This follows from boundedness of the steepest descent operator and linearity of the mirror operator i.e. $\|M_R S_{||\cdot||}(\theta, g)\| = \|\nabla^2 R^{-1}(\theta)\, S_{||\cdot||}(\theta, g)\| \leq C_1 \|\nabla^2 R^{-1}(\theta)\|\ \|S_{||\cdot||}(\theta, g)\| \leq C_1 \|\nabla^2 R^{-1}(\theta)\| \leq C_2(1 + ||\theta||)$. $\square$

Remark 13 *The growth condition in Corollary 12 on the inverse hessian $\nabla^2 R^{-1}$ is satisfied by all considered example mirror maps: gradient descent, hyperbolic entropy and* $\cosh$ *entropy.*

Appendix E. Related work

Steepest descent and modern optimization. Recent work views optimizers as modular operations on groups of parameters [5]. We build on this and focus on another basic operation: composition. This is motivated by the role of implicit bias in modern optimization, as Adaptive methods can overfit rotationally invariant sparse targets [73], and many large-scale optimizers can be viewed as steepest descent under suitable norms or geometries [19, 35, 43, 70]. This includes recent matrix and spectral methods such as Muon [36], and Scion[61]. Adam and AdamW related to sign descent based methods [56] also induce distinctive implicit biases: both exhibit L_∞ type of bias [76, 78]. This motivates composing a stable adaptive update with a sparsity-promoting mirror update.

Moreover, related multiplicative updates appear in log-normal training dynamics [55] and in MADAM, whose exponential update is tied to the entropy mirror map $\nabla R(\theta) = \log(\theta)$ and preserves the initial parameter signs [7]. The closest related optimizer is HAM [34], which alternates a base optimizer step with the entropy mirror step to obtain the sparsity bias of this reparameterization improving state-of-the-art sparse training methods. HORST follows the same geometric principle, but in the transformer setting where the base optimizer relies on AdamW: by composing an adaptive steepest-type update with a hyperbolic mirror update.

Mirror descent and reparameterizations. Reparameterized gradient flows can be viewed as mirror flows [48], building on classical implicit-regularization results for matrix factorization and overparameterized models [3, 4, 26, 47, 72, 75, 79]. For sparsity, the central example is the diagonal-linear reparameterization [32], where the induced hyperbolic-entropy geometry interpolates between an L_2 and an L_1 bias, with weight decay and stochasticity driving the implicit bias towards L_1 [33, 58]. This mirror-flow perspective has been extended to large learning rates, momentum, explicit regularization, and weight normalization [12, 13, 18, 32, 33, 57], and to implicit-bias characterizations for mirror descent and exact dynamics of deep linear networks [15, 42, 59, 67]. Our proof uses this line of work in the following way: in the steepest mirror flow connection of Jacobs et al. [35], we interpret the composed steepest mirror dynamics as steepest descent under a multiplicative reparameterization, which yields the L_1 sparse-bias characterization in Theorem 5. HORST then turns this continuous-time mechanism into a discrete optimizer by composing an adaptive steepest-type step with an entropy mirror map update.

Classic sparse training. Sparse training has been studied extensively in vision tasks [1, 2, 24, 31, 41, 50, 62, 63], where dynamic sparse training methods such as SET [53] and RigL [17, 44] alternate magnitude pruning with gradient-informed regrowth strategies, and the lottery ticket hypothesis [9, 10, 20] established that there are trainable sparse subnetworks found at initialization. Among dense-sparse pipelines, AC/DC [60] alternates dense and sparse phases through iterative hard thresholding and is the method we adopt as the sparsification backbone for our empirical evaluation. [22, 23] further showed that the success of dense-sparse methods over pruning at initialization (PaI) traces back to their ability to flip parameter signs that PaI cannot. This is a secondary motivation we adopt for the alternating scheme in HORST as the multiplicative update on its own would prevent sign flips.

A complementary line of sparse-training works uses the mirror-descent/reparameterization mechanism described above to induce sparsity through optimization dynamics rather than through pruning alone. First, Ziyin and Wang [80] showed that L_1-type sparse solutions can be obtained by running vanilla SGD on a Hadamard factorization, while Kolb et al. [40] study how increasing

the depth of such pointwise factorizations can drive extreme sparsity, sometimes at the cost of performance. Moreover, Jacobs and Burkholz [32] show that continuous sparsification with the reparameterization $m \odot w$ undergoes an implicit-bias transition from L_2 to L_1.

Sparsity in transformers. Transformer sparsification has been studied through several complementary approaches. Post-training pruning methods such as SparseGPT [21], WANDA [68], and SAFE [46] remove weights after pretraining while attempting to preserve the layerwise activation statistics or local curvature structure. Other work studies layerwise sparsity allocation and weight redistribution, such as OWL [77]. For vision transformers, SVITE [11] explores end-to-end sparse training strategies. These approaches primarily modify the sparsification procedure, allocation rule, or pruning criterion. HORST instead modifies the optimizer used within a standard sparse training pipeline. Our results suggest that the implicit bias of AdamW is itself a bottleneck for sparse transformers, and that composing AdamW with a entropy mirror map update can improve the sparse solutions found by existing dense-to-sparse training methods.

Appendix F. Preliminaries implicit bias

Convex optimization We first recall the Bregman divergence, which naturally generalizes the Euclidean distance metric.

Definition 14 (Bregman divergence) *Given a differentiable convex function $R : \mathbb{R}^n \to \mathbb{R}$, the Bregman divergence associated to R is: $D_R(\theta, \xi) := R(\theta) - R(\xi) - \langle \nabla R(\xi), \theta - \xi \rangle$.*

Moreover we give two basic desired characterizations of the mirror maps in Definition 15.

Definition 15 *(Legendre and Bregman function) We say R is a Legendre function when the following holds:*

- *R is strictly convex on the interior of its domain int$(domR)$.*
- *For any sequence $\{\theta_i\}_{i=1}^{\infty}$ going to the boundary of $domR$, the gradient diverges, i.e. $\lim_{i\to\infty} ||\nabla R(\theta_i)|| = \infty$.*

We say R is a Bregman function if

- *For any $\theta \in domR$ and $\gamma \in \mathbb{R}$, $\{\xi \in domR | D_R(\theta, \xi) \leq \gamma\}$ is bounded.*
- *For any $\theta \in domR$ and sequence $\{\theta_i\}_{i=1}^{\infty} \subset$ int$(domR)$ such that $\lim_{i\to\infty} \theta_i = \theta$, it holds that $\lim_{i\to\infty} D_R(\theta, \theta_i) \to 0$.*

For standard convergence proofs [74] the PL-inequality can be used.

Definition 16 *(Polyak-Łojasiewicz inequality) Let $\Lambda > 0$ be a constant and $\mathcal{L} \in C^1(\mathbb{R}^n, \mathbb{R})$ then we say $\mathcal{L}$ satisfies the PL-inequality if and only if:*

$$||\nabla \mathcal{L}(\theta)||_2^2 \geq 2\Lambda(\mathcal{L}(\theta) - \mathcal{L}(\theta^*)) \text{ for all } \theta \in \mathbb{R}^n,$$

where θ^* is a minimizer of $\mathcal{L}$, which is assumed to exist for $\mathcal{L}$.

Assuming Definition 16 linear convergence for mirror flow:

$$d\theta_t = -\nabla^2 R(\theta_t)^{-1} \nabla \mathcal{L}(\theta_t) dt, \qquad \theta_0 = \theta_{\text{init}},$$

under the inversely coercive mirror map Assumption 15 can be shown:

$$d\mathcal{L}(\theta_t) = -\nabla \mathcal{L}(\theta_t) \nabla^2 R(\theta_t)^{-1} \nabla \mathcal{L}(\theta_t) dt \leq \nabla - \mu ||\mathcal{L}(\theta_t)||_2^2 dt \leq \mu 2 \mathcal{L} \Lambda(\theta_t) dt$$

where we assumed the minimum is attained at zero for simplicity. Then the result directly follows from applying Gronwall's Lemma as used in [32].

Implicit bias Now, we provide more details on the implicit of both operator classes. Recall that the setting is binary classification on linear separable data with K points $\{x_i, y_i\}_{i=1}^K$, $(x_i, y_i) \in \mathbb{R}^n \times \{\pm 1\}$, exponential loss $\ell_i(\theta) := \exp(-y_i \langle \theta, x_i \rangle)$, and total objective $\mathcal{L}(\theta) := \sum_{i \in [K]} \ell_i(\theta)$. We note that [70] result is more general and covers also homogeneous neural networks, which we actually use in the proof as we rely on a homogeneous reparameterization. Moreover, we let $\eta \to 0$ and consider the flow setting. On linearly separable data, the resulting iterates θ_t diverge along a fixed limiting direction $\bar{\theta} := \lim_t \theta_t / \|\theta_t\|$ [27].

For steepest descent with respect to $\| \cdot \|$, this direction solves the max-margin problem [70]:

$$\min \|\theta\| \qquad \text{such that} \qquad y_i \langle \theta, x_i \rangle \geq 1 \quad \text{for all } i \in [K].$$

For mirror descent with a separable coercive map R, it solves the analogous problem in the geometry of R [59, 67]:

$$\min \phi_\infty(\theta) \qquad \text{such that} \qquad y_i \langle \theta, x_i \rangle \geq 1 \quad \text{for all } i \in [K],$$

where $\phi_\infty(\theta) := \lambda \lim_{\eta \to 0} \eta \, r^{-1}(R(\theta/\eta))$, where $\lambda > 0$ is a fixed constant and r is the mirror potential of a single coordinate. This is the so-called horizon function associated to R. Sign descent and the hyperbolic entropy therefore induce L_∞ and L_1 biases respectively.

Appendix G. Main results

Here we provide full proofs of the statements in the main text.

Boundedness and linearity. First we show the basic properties of the operators in Lemmas 18 and 17.

Lemma 17 *The steepest operator is bounded but not linear.*

Proof. **Linearity:** A concrete counter example for linearity is the steepest descent operator with respect to $|| \cdot ||_\infty$ giving for any $g \in \mathbb{R}^n$ and positive $\lambda > 0$:

$$S_{||\cdot||_\infty}(\theta, \lambda g(\theta)) = \text{sign}(\lambda g(\theta)) = \text{sign}(g(\theta)) \neq \lambda \text{sign}(g(\theta)),$$

which is unequal except when $\lambda = 1$. In general, by definition the steepest descent operator is scale invariant violating linearity.

Bounded: By definition the steepest descent operator is maximum argument on bounded norm ball therefore it is bounded. □

Lemma 18 *The mirror operator is linear and unbounded in general.*

Proof. **Linearity:** Using the definition consider two gradient estimates $g, h \in \mathbb{R}^n$ evaluated at $\theta \in \mathbb{R}^n$ and a coefficient $\lambda \in \mathbb{R}$ then first we have:

$$M_R(\theta, g+h) = \nabla^2 R^{-1}(\theta)\left(g(\theta) + h(\theta)\right) = \nabla^2 R^{-1}(\theta) g(\theta) + \nabla^2 R^{-1}(\theta) h(\theta) = M_R(\theta, g) + M_R(\theta, h).$$

Moreover, we have

$$M_R(\theta, \lambda g) = \nabla^2 R^{-1}(\theta)\left(\lambda g(\theta)\right) = \lambda \nabla^2 R^{-1}(\theta)\left(g(\theta)\right) = \lambda M_R(\theta, g). \qquad \square$$

Unbounded: Consider gradient descent. We can choose a function $f(\theta) = \log(\theta)$ which has gradient $f'(\theta) = 1/\theta$ for which we have $\lim_{\theta \to 0^+} f'(\theta) = \infty$ and thus unbounded. $\square$

Implicit bias. We can now characterize the implicit bias of the composed operator.

Theorem 19 *Consider steepest-mirror descent with respect to the L_p-norm, $p \in \mathbb{N}_{\geq 2}$, and mirror map $\nabla R(\theta) = \log(\theta)$. Then the iterates of Eq. (1) converge in direction to a KKT point of:*

$$\min \|\theta\|_1 \qquad \textit{such that} \qquad y_i \langle \theta, x_i \rangle \geq 1 \quad \textit{for all } i \in [K].$$

Proof. This can be shown by using the steepest mirror flow connection to homogeneous reparameterization as developed in [35] and combining it with the steepest flow characterization of the max-margin in [70]. Using a time reparameterization $dt = ||\nabla \mathcal{L}(\theta_s)||_q ds$ such that $\frac{1}{p} + \frac{1}{q} = 1$ the flow can be written as:

$$\frac{d\theta_t}{dt} \in -|\theta| \text{sign}\left(\partial \mathcal{L}(\theta_t)\right) |\partial \mathcal{L}(\theta_t)|^{q-1}. \tag{2}$$

We now show equivalence of this flow to a re-parameterized standard steepest descent flow. Consider the reparameterization $\theta = \Pi_{i \in [p]} w_i$ and we train with steepest descent with respect to L_p-norm (with time rescaled as above). Moreover, we initialize such that $|w_{i,0}| = |\theta_0|^{1/p}$ i.e. balanced. Then we have the following flow equation for each $i \in [p]$:

$$\frac{dw_{i,t}}{dt} \in -\text{sign}\left(\partial_{w_i} \mathcal{L}(\theta_t)\right) |\partial_{w_i} \mathcal{L}(\theta_t)|^{q-1}.$$

Using the balance equation as in [35] ($\forall t \geq 0$, we have almost everywhere $|w_{i,t}| = |\theta_t|^{1/p}$). The evolution of $\theta_t = \Pi_{i \in [p]} w_i$ is equal to Eq.(2). It follows from Theorem 3.4 in [70] that the iterates $(w_1, \ldots, w_p)$ converge in direction to the following optimization problem:

$$\frac{1}{2} ||(w_1, \ldots, w_p)||^2_{L_p} \qquad \text{such that} \qquad y_i \langle \theta, x_i \rangle \geq 1 \text{ for all } i \in [K].$$

As the map $z \to z^{2/p}$ for $p > 0$ is strictly increasing for $z \geq 0$ the optimization is equivalent to:

$$\frac{1}{2} \sum_{i \in [p]} ||w_i||^p_{L_p} \qquad \text{such that} \qquad y_i \langle \theta, x_i \rangle \geq 1 \text{ for all } i \in [K].$$

Combining this with the balance equation (which also holds for the normalized iterates) the optimization problem above is equivalent to:

$$\frac{p}{2} ||\theta||_{L_1} \qquad \text{such that} \qquad y_i \langle \theta, x_i \rangle \geq 1 \text{ for all } i \in [K],$$

which concludes the result as positive multiplicative constant does not alter the solution of the objective. $\square$

Appendix H. Implicit bias of operator classes on sparse linear classification

As specified in the main text we consider binary classification with linear separable data. More specifically, the data can be separated by a sparse support vector. Let $\mathcal{S} = \{\mathbf{x_i}, y_i\}_{i=1}^N$ be a dataset of i.i.d. samples with $\mathbf{x_i} \in \mathbb{R}^D$ drawn from $\mathcal{N}(0, I_D)$ and labels $y_i \in \{\pm 1\}$ for all $i \in [N]$ generated by a sparse teacher $\theta_\star \in \mathbb{R}^D$ via $y_i = \text{sign}\langle \theta_\star, x_i \rangle$, where $\theta_\star$ is supported only on its first two coordinates. The learner is the linear model $f(x;\theta) = \langle \theta, x \rangle$ with $\theta \in \mathbb{R}^D$, trained under the empirical exponential margin loss $\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^N e^{-y_i\langle \theta, x_i \rangle}$. We fix $D = 100, N = 80$, placing the problem in the over-parametrized regime where infinitely many θ achieve zero training loss and the implicit bias of the optimizer determines which one is selected. We use this setting to illustrate the dichotomy: each optimizer in our framework is ran for a fixed budget of full-batch steps and we compare the directions $\bar{\theta}$ they select to the sparse teacher $\theta_\star$. Optimizer with an L_1 sparsity bias should recover $\theta_\star$. In contrast, an optimizer with an L_∞ bias will spread mass across all D coordinates, saturating at the $\pm||\theta||_\infty$ margin.

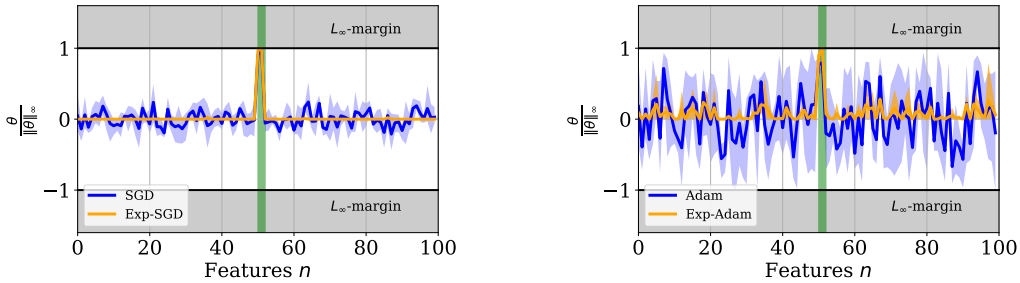

Figure 4: Implicit bias of additive vs. multiplicative steepest descent on sparse linear classification. Each panel plots the final classifier θ, normalized by $||\theta||_\infty$ across the D features. The two informative coordinates of the sparse teacher $\theta_\star$ are highlighted by the green vertical strip. Each optimizer is averaged across 3 seeds and the shaded bands show the variance across them. The grey strips at ± 1 mark the L_∞-margin region. On the (left), SGD (blue) shows slight spikes on the informative coordinates but spreads non-trivial mass across the spurious ones, consistent with its L_2 max-margin implicit bias; Exp-SGD (orange), the multiplicative mirror-descent counterpart drives spurious features into a tight band near zero and spikes only on the informative ones, corresponding to known L_1 implicit bias. On the (right), the same comparison with Adam (blue) and its multiplicative composition Exp-Adam (orange): Adam saturates at ± 1 across nearly all features in line with the L_∞ implicit bias whereas Exp-Adam concentrates mass on the two informative coordinates and drives spurious coordinates near zero, exhibiting an L_1 sparsity bias.

Optimization details and hyperparameter selection: All optimizers are ran in the full-batch regime (batch size $= N$) for $T = 10^4$ epochs across 3 data seeds. We use $\eta = 10^{-2}$ as the baseline learning rate for SGD and Exp-SGD, and $\eta = 10^{-1}$ for SignSGD; these settings give comparable per-step displacement of θ since SignSGD's update has unit per-coordinate magnitude regardless of the gradient norm, whereas SGD/Exp-SGD scale with $|\nabla\mathcal{L}|$. Note that on the exponential loss, $|\nabla\mathcal{L}(\theta)| \to 0$ exponentially fast as the margin grows, so any optimizer whose step is proportional to $|\nabla\mathcal{L}|$ including SGD, Exp-SGD, and $\cosh$ has a vanishing effective step size in the last phase, while SignSGD does not. Adam, Exp-Adam, and Adam-Exp use the same $\eta = 10^{-2}$ together

with the standard momentum and bias-correction parameters $(\beta_1, \beta_2, \varepsilon) = (0.9, 0.999, 10^{-8})$. For $\cosh$-entropy, the choice of learning is more delicate than for other optimizers, We perform a grid sweep $\eta \in \{0.001, 0.005, 0.009, 0.01, 0.03, 0.05, 0.1, 0.2, 0.3, 0.5, 0.7, 0.9\}$ across 3 data seeds and select the value that minimizes the final exponential loss $\mathcal{L}(\theta_T)$ while keeping $\eta < 1$ to avoid instability. The best performing setting is $\eta = 0.9$ as shown in 6.

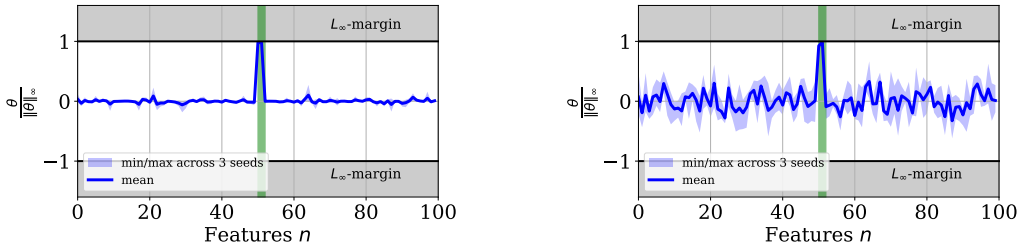

Figure 5: Sparse linear classification with mirror maps. On the (left), the learned features by the hyperbolic entropy, similarly as the exponential update corresponding the entropy mirror map gives rise to an L_1 implicit bias. On the (right), in contrast, the $\cosh$-entropy spreads more mass along the non-informative coordinates.

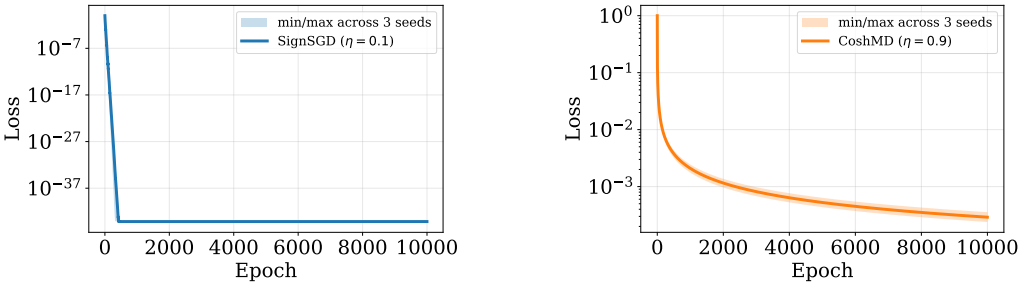

Figure 6: The evolution of the loss, for signSGD (left) and the $\cosh$-entropy (right). Observe that $\cosh$-entropy has much slower convergence.

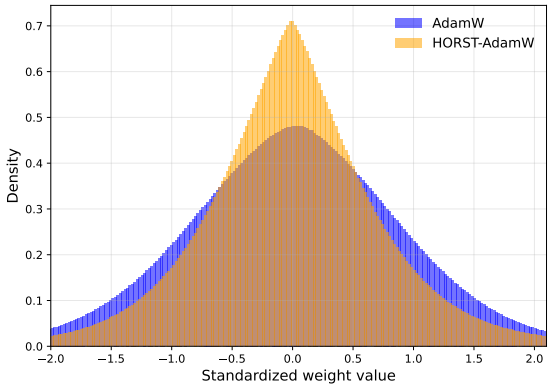

Figure 7: **HORST-AdamW induces a sparser weight distribution.** Standardized weight distributions at end of training for a dense GPT-2 Small model trained on SlimPajama-6B for 25K iterations with HORST-AdamW vs. AdamW. We observe that HORST-AdamW concentrates weights sharply around zero with lighter tails, while AdamW retains a broader, near-Gaussian profile. This indicates the presence of an implicit L_1 bias.

Appendix I. Details on vision transformer experiment

For the DeiT training setup we use standard data augmentations such as label smoothing, mixup and cutmix. However, we do not use random augmentation. We also use the default hyperparameters of AdamW as given in Table 2. Moreover, we have ablated over the additional hyperparameters $\alpha \in \{5, 10, 200\}$ and $\beta \in \{0,\ 1e-3,\ 1e-2\}$ on DeiT-base for 70% sparsity level and found that $\alpha = 5, \beta = 0$ works the best. Note $\alpha = 200$ was tried first as it was used in [34] for HAM, however we found that $\alpha = 200$ is highly unstable and crashes, as this is an effective multiplication of the learning rate. For all other settings we adopt these hyperparameters. Each experimental run was run on 4 NVIDIA A100 GPUs with 40GB of memory. The code used is based on the repository by [54].

Table 2: Hyperparameters of the sparse training for vision transformers experiments.

Model	Learning Rate	Weight Decay	Epochs	Batch Size	Optimizer	Scheduler
DeiT-base	0.005	$1e{-}1$	300	1024	AdamW	Triangular
DeiT-small	0.005	$1e{-}1$	300	1024	AdamW	Triangular

Comparison with HAM We also compare with the state-of-the-art sparsity aware optimizer HAM [34], which is the closest optimizer to HORST. We use the default parameter $\alpha = 200$. Moreover, as for AdamW, β plays the same role as the decoupled weight decay. We consider multiple values $\beta \in \{1e-3, 1e-2, 0\}$ and report the best accuracy values in Table 3. Note that this level of improvement is in line with the reported gains in [34] in Table 8 and 9.

Long run To emphasize that HORST's benefits do not become negligible with longer training we provide a single seed long run (600 Epochs) for 70% sparsity. We report the validation accuracy in Table 4. We observe that both optimizers have now improved accuracy and furthermore the gap between them has decreased a little, however, it remains significant.

Table 3: Comparison with the method HAM on a DeiT-small at 90% sparsity trained on ImageNet for seed 1. The geometric mismatch between gradient based exponential and the AdamW update makes it harder for HAM to improve over the baseline.

Method	Val. Acc.
HAM	60.06
HORST	**70.5**
Baseline	59.23

Table 4: Performance comparison of AdamW and HORST-AdamW together with AC/DC on a DeiT-small for 70% sparsity for longer training time of 600 epochs.

Optimizer	Value
AdamW	73.76
HORST-AdamW	$\mathbf{78.96}$

Appendix J. Details on GPT-2 experiment

We train a GPT-2 Small model ($n_{\text{layer}} = 12$, `n_head`$= 12$, `n_embd`$= 768$, context length 512, ~ 124M parameters with tied input/output embeddings) on the SlimPajama-6B subset using the GPT-2 BPE tokenizer (vocabulary size 50304). All runs share the hyperparameters in Table 5, use 3000 warmup steps, run in `bfloat16` precision, and use $\beta_1 = 0.9$, $\beta_2 = 0.95$ for AdamW. For the HORST-AdamW variant we additionally set $\alpha = 5$ and $\beta = 0$. Experiments were conducted using a single NVIDIA A100 GPU with 80 GB of memory. The code used is based mainly on the repository by [65].

Table 5: Hyperparameters of the dense pretraining for GPT-2 Small experiments. AdamW and HORST-AdamW use the same configuration.

Model	Learning Rate	Weight Decay	Iterations	Batch Size	Scheduler
GPT-2 Small	$1\text{e}{-3}$	$1\text{e}{-1}$	$25,000$	128	Cosine

For the one-shot pruning experiments we apply per-tensor (layerwise) unstructured magnitude pruning at sparsity levels $s \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$ to the dense AdamW and HORST-AdamW checkpoints. For each prunable weight tensor W and target sparsity s we zero the $\lfloor s \cdot |W| \rfloor$ entries of smallest absolute magnitude and leave the rest untouched, computing the threshold independently per tensor. Pruning is applied only to the `nn.Linear` weight tensors inside transformer blocks $1, \ldots, n_{\text{layer}} - 2$ (four tensors per block: the packed $Q/K/V$ projection, the attention output projection, and the two MLP projections). The token and positional embeddings, the tied LM head, all LayerNorm parameters and biases, and the first & last transformer blocks are kept dense throughout. No fine-tuning, weight recovery, or calibration data is used. Validation perplexity and accuracy are evaluated on the held-out SlimPajama validation split using the same evaluation pipeline used during training.

Table 6: Validation perplexity (lower is better) on SlimPajama-6B after one-shot layerwise unstructured magnitude pruning of dense GPT-2 Small checkpoints; no fine-tuning. HORST-AdamW achieves the lowest perplexity at every non-trivial sparsity level, with the gap to baseline AdamW significantly widening at higher sparsities.

Optimizer	Dense	10%	20%	30%	40%	50%
AdamW	23.46	23.61	24.54	33.49	86.50	237.17
HAM	23.48	23.60	24.31	29.58	54.58	105.76
HORST-AdamW	23.61	23.70	**24.32**	**28.78**	**44.16**	**69.58**