Task complexity shapes internal representations and robustness in neural networks

Robert Jankowski[1,2] **Filippo Radicchi**[3] **M. Ángeles Serrano**[1,2,4]
Marián Boguñá[1,2] **Santo Fortunato**[3]
[1]Universitat de Barcelona [2]Universitat de Barcelona Institute of Complex Systems (UBICS)
[3]Center for Complex Networks and Systems Research (CNetS), Indiana University [4]ICREA
{robert.jankowski,marian.serrano,marian.boguna}@ub.edu
{f.radicchi,santo.fortunato}@gmail.com

Abstract

Neural networks excel across a wide range of tasks, yet remain "black boxes". In particular, how their internal representations are shaped by the complexity of the input data and the problems they solve remains obscure. In this work, we introduce a suite of five data-agnostic probes—pruning, binarization, noise injection, sign flipping, and bipartite network randomization—to quantify how task difficulty influences the topology and robustness of representations in multilayer perceptrons (MLPs). MLPs are represented as signed, weighted bipartite graphs from a network science perspective. We contrast easy and hard classification tasks on the MNIST and Fashion-MNIST datasets. We show that binarizing weights in hard-task models collapses accuracy to chance, whereas easy-task models remain robust. We also find that pruning low-magnitude edges in binarized hard-task models reveals a sharp phase-transition in performance. Moreover, moderate noise injection can enhance accuracy, resembling a stochastic-resonance effect linked to optimal sign flips of small-magnitude weights. Finally, preserving only the sign structure—instead of precise weight magnitudes—through bipartite network randomizations suffices to maintain high accuracy. These phenomena define a model- and modality-agnostic measure of task complexity: the performance gap between full-precision and binarized or shuffled neural network performance. Our findings highlight the crucial role of signed bipartite topology in learned representations and suggest practical strategies for model compression and interpretability that align with task complexity.

1 Introduction

Neural networks have achieved remarkable success across a wide range of applications and now underpin many aspects of our daily lives. However, their vast number of trainable parameters often renders them opaque "black boxes" that, despite their effectiveness, sacrifice interpretability [11, 21]. To address this, the emerging field of mechanistic interpretability (MI) seeks to reverse-engineer the parameters and algorithms of trained networks in order to understand precisely how and why they produce their outputs [55, 7]. A common first step in MI is to decompose a network into simpler, more analyzable components. In the case of one of the simplest architectures—the multilayer perceptron (MLP)—each layer can be viewed, from a network-science perspective, as a signed, weighted bipartite graph. Such graphs are a central object of study in network science, which analyzes complex systems ranging from telecommunications and computer networks to biological and social networks [48, 46, 45, 20]. Many real-world networks exhibit characteristic topological features—power-law degree distributions, the small-world property, community structure, and high clustering coefficient—that reflect underlying organizational principles. By treating each MLP layer

Preprint.

as a complex network, we can apply these same tools, such as network null models, laying the groundwork for a deeper understanding of how neural networks learn and generalize.

Neural network performance depends not only on its architecture and training procedure but also on the complexity of the tasks it must solve [3, 24]. Task difficulty shapes the representations a network learns-more challenging problems typically demand finer-grained or more abstract features [31]. For example, distinguishing between visually similar classes forces the network to encode subtle differences that are not required when classes are easily separable [36]. These differences should be observed by modeling each MLP layer as a signed bipartite graph—where positive and negative weights correspond to signed edges—and analyzing its structure. Understanding the difficulty of a task guides model selection, architecture design, and optimization strategy. Additionally, by understanding which parts of the task are more difficult to learn, we can gain insight into how the network processes information and identify potential biases or limitations.

In this work, we investigate the internal representations learned by a fully connected multilayer perceptron (MLP) through the lens of network science, contrasting an "easy" task with a "hard" task on MNIST and Fashion-MNIST datasets. To this end, we design five complementary experimental probes: pruning (progressively removing edges with the smallest absolute weights), binarization (reducing all weights to ± 1), noise injection (adding zero-mean noise of varying amplitude to the weights), flipping signs (changing the sign of the smallest-magnitude weights), and bipartite network randomization (shuffling connections while preserving given networks' properties). Our key findings are as follows.

- Binarizing an MLP trained on a hard task causes its accuracy to collapse to chance, whereas the easy-task model remains quite robust.
- As we prune low-magnitude edges, a binarized model trained on the hard task exhibit a sharp performance transition at a characteristic sparsity level.
- For the same model, injecting moderate noise can boost accuracy—a manifestation of stochastic resonance—while excessive noise degrades performance.
- The performance peak in the noise experiment arises from flipping the sign of the weights with the smallest absolute values.
- Randomizing the bipartite connectivity while preserving the sign of each weight leaves the network's accuracy on the easy task nearly unchanged, demonstrating that the learned representations depend more critically on the sign structure than on precise weight magnitudes.

These findings enable us to quantify task complexity in a data-agnostic manner. This means that our probes can be applied to any model and any modality as long as it contains an MLP layer. As a case study, we used these probes to evaluate the robustness of each layer in a DistilBERT model trained for Named Entity Recognition (NER). We discovered that the earliest layers are the least robust—but that simple pruning of the smallest-magnitude weights can improve their performance. In contrast, in the deeper layers, it is the sign of each weight that matters the most. Practically speaking, this means those deeper layers can be binarized at inference time without any loss in accuracy.

2 Related work

Network pruning and binary neural networks. Model compression via pruning and quantization has been extensively explored to reduce inference cost while retaining accuracy [44, 12, 35]. Early work on Optimal Brain Surgeon (OBS) uses a second-order Taylor approximation of the training loss to identify and remove weights with minimal impact on performance [22]. First-order, data-driven methods such as Taylor pruning estimate the change in loss induced by removing individual filters or channels, achieving significant FLOP reductions on large CNNs with minimal retraining [42, 23]. More recently, single-shot techniques like SNIP perform connection saliency scoring at initialization—eliminating the need for any gradient-based fine-tuning to attain high sparsity levels [33]. Parallel to pruning, Binary Neural Networks (BNNs) constrain weights and activations to $\{-1, +1\}$ to enable extreme compression and ultra-fast bitwise operations. BinaryConnect and BinaryNet introduced stochastic and deterministic binarization schemes, demonstrating that end-to-end training of 1-bit networks can achieve competitive accuracy on small benchmarks [13, 51]. Subsequent architectures such as MeliusNet employ dense feature propagation and learned scaling

factors to close the gap to full-precision models even on ImageNet-scale data [8]. However, these methods typically focus on worst-case accuracy drops and rarely analyze how task difficulty modulates robustness to quantization.

Similarity of neural network models. Advances in understanding how different networks or layers encode information have been driven by measures of representational and functional similarity [26]. Representational similarity measures assess how activations of intermediate layers differ, whereas functional similarity measures compare the outputs of neural networks with respect to their task. Representational similarity measures include SVCCA [49], which aligns the subspaces spanned by activations to compare layers or models, RSA [28], which assesses the geometry of activation patterns by correlating pairwise distance matrices and has been applied to assess the relationship between visual tasks and their task-specific models [16], and CKA [27], which uses kernel methods to produce robust, scale-invariant similarity scores. On the other hand, within functional similarity measures class, we can highlight types such as: performance, hard prediction [39, 40], soft-prediction [1], or gradient-based [34] measures. In this work, since we work with a simple MLP, we aim to compare representation through performance analysis and probe the models' internals differently, which can be classified as one of the functional measures.

Task complexity. The difficulty of learning a task has been studied from both neuroscience and machine learning perspectives. Recent empirical studies show that neural networks trained on tasks with high intra-class similarity or fine-grained distinctions tend to learn deeper or more distributed feature hierarchies [31]. Additionally, Mukherjee et al. [43] demonstrated that the modality of the output task plays a crucial role in shaping interpretable object representations. It has also been shown that to ensure better learning outcomes, representations may need to be tailored to both task and model to align with the implicit distribution of model and task [64]. When visually assessing images, one might intuitively conclude that datasets composed of grayscale images—such as MNIST [32] or Fashion-MNIST [59]—are generally easier to classify than RGB-valued datasets like CIFAR [29]. Metrics such as the Structural Similarity Index Measure (SSIM)[57] or Learned Perceptual Image Patch Similarity (LPIPS)[62] could serve as proxies to quantify the classification difficulty between image classes. However, in this work, we propose using neural network probes instead. These probes offer greater generalizability and can be extended beyond images to quantify task difficulty across various domains.

Network science in deep learning Network science methods have been applied to neural network research in several ways. Custom loss functions based on graph-theoretic principles have been proposed for graph neural networks [10], and fully connected architectures have been analyzed in terms of classic centrality measures to link network structure with model performance [54]. Similarly, recurrent neural networks have been shown to exhibit universal patterns of signed motifs [63], and more generally neural networks have been studied as dynamical systems to characterize their learning trajectories [30, 25, 37]. Network science insights have also informed the design and initialization of neural architectures. Sparse connectivity patterns inspired by scale-free graphs have been used to improve the efficiency of training large networks [41], graph-based initialization schemes have been developed to accelerate convergence [53], and random wiring schemes drawn from network models have been explored [60]. Graph-theoretic metrics—such as average shortest-path length and clustering coefficient—have been applied to characterize deep architectures, linking connectivity patterns to generalization performance [61], comparing artificial networks with biological neural circuits [15], and assessing model robustness under perturbations [58]. A recent position paper surveys many additional opportunities for network-science approaches in deep learning [9]. Despite these advances, little work has applied signed, weighted bipartite graph analysis to understand how task complexity drives emergent topological transitions in the weight space. Our work fills this gap by systematically probing MLP layers under pruning, binarization, noise injection, and randomization experiments, revealing novel phase transition-like behavior dependent on task difficulty.

Mechanistic interpretability. Mechanistic interpretability has been applied primarily to large-language models, where circuit-level analyses reveal functional subnetworks and token-wise attributions [50]. Extensions to Graph Transformers [17] and to bilinear MLPs [47] uncover attention-based motifs and feature-interaction circuits. However, these efforts focus on local circuits or weight factorizations and overlook the global connectivity patterns that a network science perspective can reveal.

3 Probing the internal representation of neural networks

Our primary benchmarks are the MNIST and Fashion-MNIST datasets. MNIST contains 70,000 28×28 grayscale images of handwritten digits (0–9), and Fashion-MNIST contains 70,000 28×28 grayscale images of Zalando clothing items across 10 categories. Initially, to define *easy* and *hard* tasks, we calculate the Structural Similarity Index (SSIM) [57] distance, i.e., $1 - \mathrm{SSIM}$, between all pairs of classes. The larger the SSIM distance, the greater the difference between the two images. Two identical images have zero SSIM distance. We then select the class pair with the highest SSIM distance as the easy task and the pair with the lowest SSIM distance as the hard task. On MNIST, the easiest pair is $\{0, 7\}$ and the hardest is $\{7, 9\}$. On Fashion-MNIST, the easiest pair is {`Dress`, `Pullover`}, while the hardest is {`Dress`, `Trousers`}. In Figure 6 in the Appendix, we show the SSIM distance heatmaps for both datasets. Let us now define the **E-model** (**H-model**) to refer to a model trained on an easy (hard) task.

The input grayscale images are flattened, and we begin by training two multilayer perceptrons (MLPs), each with a single hidden layer of dimension $d = 64$, on binary classification tasks of differing difficulty. Optimization is performed using the Adam optimizer together with a cosine-annealing learning-rate schedule with a maximum of 10 epochs. At each step, we measure the test accuracy and stop training when this value is maximized. For completeness, in the Appendix, we report experiments using hidden-layer sizes $d = 32$ (Figures 10-11) and $d = 128$ (Figures 12-13).

All of our probing methods are applied to each trained network *without* any further fine-tuning or retraining. In the following sections, we provide a detailed description of each probe. The code for reproducing experiments is available at `https://anonymous.4open.science/r/probing-neural-networks/`.

3.1 Pruning and binarizing

There are many methods for pruning neural networks. In this work, we use a simple post-training strategy—iteratively removing the weights with the smallest absolute values, as in the Lottery Ticket Hypothesis [18]. After each pruning step, we measure test accuracy on both easy and hard tasks. We also evaluate binarized versions of these pruned models—where each remaining weight is replaced by its sign—and refer to them as the signed-E and signed-H models, respectively.

Figure 1a shows that the E-model retains higher accuracy as the smallest-magnitude weights are removed, whereas the H-model's performance drops much more rapidly. Interestingly, pruning can even increase the accuracy of the signed-E model. Most strikingly, the signed-H model, which initially performs at near random, exhibits a performance transition and even surpasses the H-model's accuracy for some sparsity levels. A similar behavior is present for the Fashion-MNIST dataset (see Figure 1b).

Even though the test accuracy of the E-model and H-model is high, their internal representation differ. One could argue that this distinction can be measured through the distribution of weights. However, as shown in Figure 7, the standard deviation of the weights, $\sigma(w)$, depends on the dataset. For MNIST, $\sigma(w)$ is narrower and smaller for the easy task, whereas it is wider and larger for Fashion-MNIST. Hence, weight statistics alone cannot serve as a reliable measure of task complexity.

3.2 Noise injection

As an additional probe, we inject noise into the network weights. Specifically, we perturb each weight w by adding a random variable drawn from the uniform distribution, $U(-a, a)$, where a controls the noise level.

For each noise magnitude, we evaluate test accuracy on both the E- and H-models, as well as their binarized ("signed") counterparts. Figures 1c,d show that the E-model remains substantially more robust under noise injection than the H-model. Moreover, adding a moderate amount of noise to the signed-E and signed-H models can actually improve their accuracy—a phenomenon akin to stochastic resonance [4, 38], which has been documented in a wide range of systems, including bistable ring lasers, semiconductor devices, chemical reactions, and climate dynamics [19, 5, 6].

In our context, this stochastic-resonance–like effect appears in the accuracy curves. When the noise standard deviation is much smaller than the average weight standard deviation $\bar{\sigma}(w)$ (indicated by

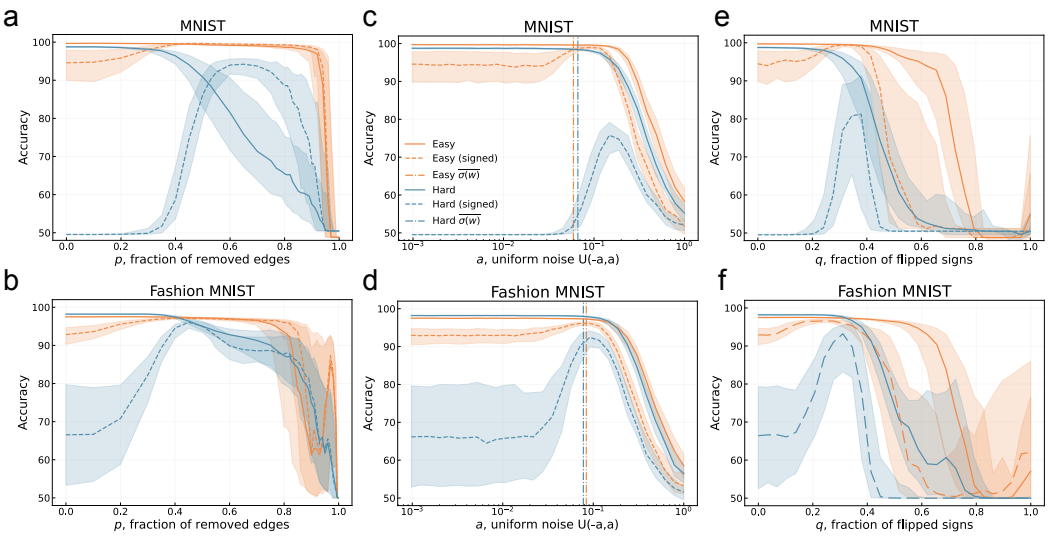

Figure 1: (**a, b**) Pruning experiment. The test accuracy as a function of the fraction of removed edges. (**c, d**) Noise injection experiment. The test accuracy as a function of the uniform noise level injected into the weights. The vertical lines show the average standard deviation of the weights. (**e, f**) Sign flipping experiment. The test accuracy as a function of the fraction of the smallest-magnitude sign flipped. All curves are averaged over 100 random initializations. Shaded regions denote the interquartile range (IQR), and the solid lines represent the median.

the vertical dotted lines), we observe no improvement in model performance. Conversely, when the noise level significantly exceeds $\bar{\sigma}(w)$, accuracy degrades to near-random levels. Thus, there exists an optimal noise level region for which performance is maximized.

3.3 Flipping signs

To further investigate the stochastic resonance-like effect observed in the accuracy curves, we design a simple experiment in which we flip the signs of the smallest-magnitude weights. First, we sort all weights by their absolute values and then flip the sign of a fraction q of the smallest-magnitude weights. In Figures 1e,f, we plot the test accuracy as a function of q for the original models and their binarized counterparts. Consistent with our noise-injection findings, flipping a nonzero fraction of the smallest-magnitude weights in both the signed-E and signed-H models improves performance, yielding a more optimal accuracy peak. These results indicate that it is the signs of the weights—rather than their exact values—that are most critical to model performance. To test this hypothesis, we next apply a series of bipartite network randomizations.

3.4 Bipartite network randomization

Each MLP layer can be represented as a signed, weighted bipartite graph. The graph comprises two disjoint node sets—left L (inputs) and right R (outputs)—with edges only running between L and R. A forward signal propagates from L to R. In the unweighted case, each node $i \in L \cup R$ has two degree counts: k_i^+ (the number of positive-weight edges) and k_i^- (the number of negative-weight edges). In the weighted formulation, these become strengths—$s_i^+ = \sum_{j:w_{ij}>0} w_{ij}$ and $s_i^- = \sum_{j:w_{ij}<0} |w_{ij}|$—summing the magnitudes of the positive or negative edges incident on i. Finally, we denote the degree (or strength) distributions over all positive and negative edges by $P(k^+)$ and $P(k^-)$ (or $P(s^+)$ and $P(s^-)$ in the weighted case).

We introduce seven distinct randomization strategies, each of which preserves different structural properties of these bipartite graphs. Figure 2a illustrates these methods, and Table 1 summarizes their key characteristics.

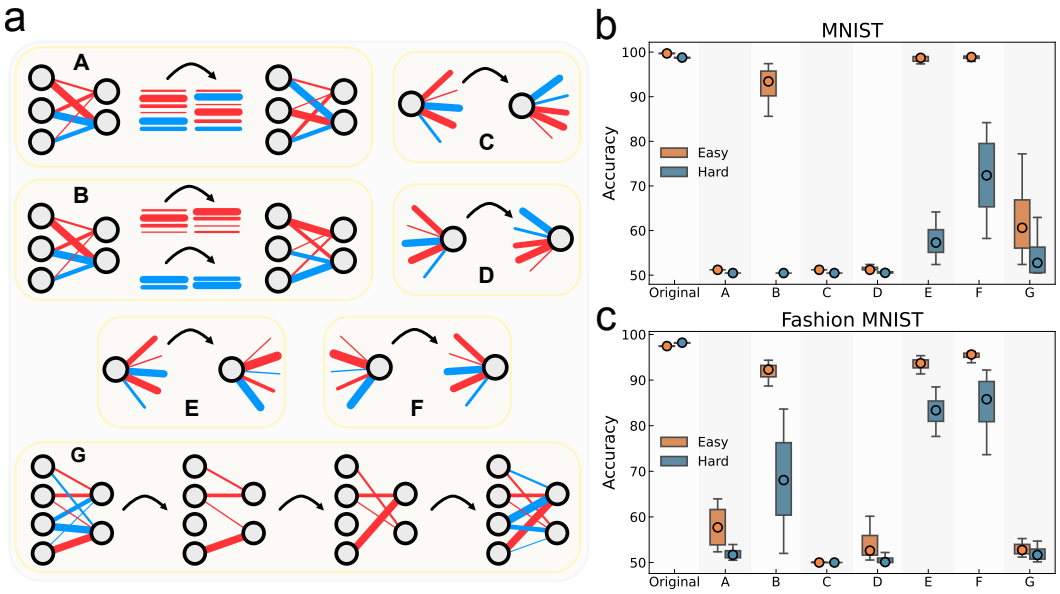

Figure 2: (**a**) Visualization of the seven types of bipartite randomizations. The accuracy of the neural network after applying each type of bipartite randomization for (**b**) MNIST and (**c**) Fashion MNIST. Boxplots show the distribution of test accuracies across 100 independent network trainings, whereas scatter markers denote the median accuracy.

Table 1: Types of bipartite randomizations and properties preserved after randomization where α is a fraction of the edges with positive sign. A ✓ in the *Keeps original sign* column indicates that each edge retains its original positive or negative sign under that randomization.

Type	α	k_L^-	k_L^+	s_L^-	s_L^+	k_R^-	k_R^+	s_R^-	s_R^+	Keeps original sign
A	✓	-	-	-	-	-	-	-	-	✓
B	✓	✓	✓	-	-	✓	✓	-	-	✓
C	✓	✓	✓	✓	✓	-	-	-	-	✓
D	✓	-	-	-	-	✓	✓	✓	✓	✓
E	✓	✓	✓	✓	✓	✓	✓	-	-	✓
F	✓	✓	✓	-	-	✓	✓	✓	✓	✓
G	✓	✓	✓	-	-	✓	✓	-	-	-

Using these randomization strategies, we evaluate the post-randomization accuracy of our trained MLPs. Figure 2b presents accuracy boxplots for the E- and H-models. We see that only those strategies that (1) preserve both the positive and negative degree distributions $P(k^+)$ and $P(k^-)$ and (2) retain each edge's original sign, maintain high performance. If either of these properties is altered, accuracy falls to chance. Notably, both randomizations B and G keep the degree distributions fixed, but only B preserves accuracy—demonstrating that the specific arrangement of positive versus negative weights is itself critical.

We further evaluate the randomization strategies that preserve high accuracy in the pruning experiment. As in Section 3.1, we measure both the original and randomized models' performance at each fraction of removed edges. Figure 3 shows that randomization initially causes an accuracy drop for both the E- and H-models. However, as sparsity increases, the randomized E-model's accuracy steadily recovers and ultimately matches that of the original E-model. By contrast, the randomized H-model undergoes an abrupt transition—similar to the signed-H-model. These results confirm that preserving the learned edge signs, rather than the precise weight values, is essential for maintaining high performance under heavy pruning.

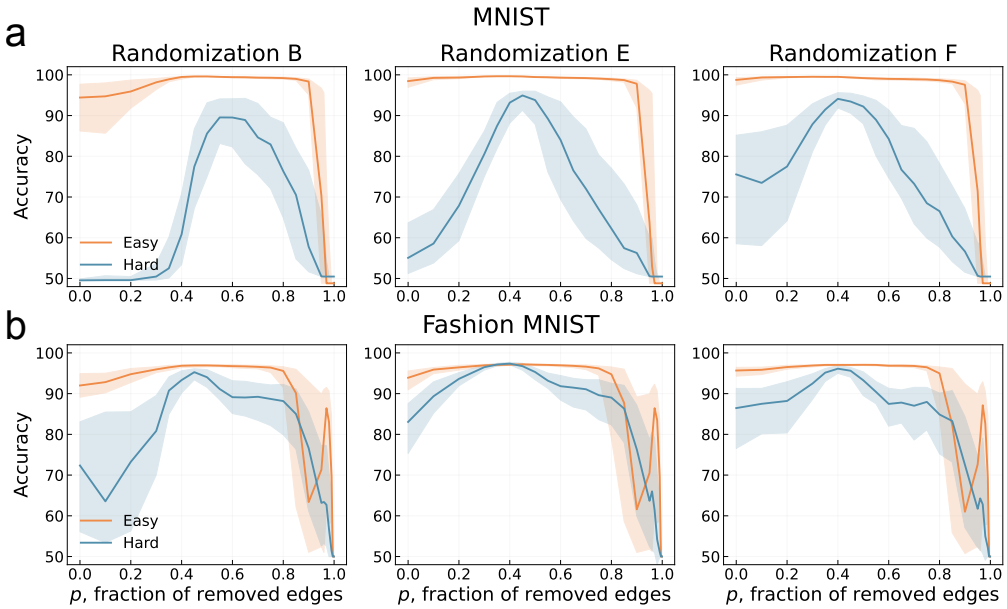

Figure 3: The test accuracy in a function of the fraction of removed edges after applying bipartite randomization B, E, and F for (**a**) MNIST and (**b**) Fashion-MNIST. All curves are averaged over 100 random initializations. Shaded regions denote the interquartile range (IQR), and the solid lines represent the median.

4 Defining task complexity

So far, we have compared models trained on easy and hard tasks. Binarizing or randomizing the H-model causes a large drop in accuracy, whereas the E-model's accuracy declines much less. This suggests a link between task difficulty and post-binarization (or randomization) performance. We therefore quantify task difficulty by measuring, for each image class pair in the MNIST dataset, the change in accuracy before versus after binarizing or randomizing.

We first note that the test accuracy for each digit class exceeds 98% (see Figure 8). Next, we evaluate how much accuracy changes once we apply our probes. In Figure 4a, we plot the difference in accuracy between each original model and its signed version. A smaller gap means the signed model's performance remains close to the original, whereas a larger gap indicates a harder classification task. For example, digits `0` and `3` show very little change—these are easy to distinguish—while digits `1` and `7` fall to around 50% accuracy after binarization, producing a substantial drop compared to the original. Applying bipartite randomization yields similar patterns (Figure 4b): the harder the digits are to classify, the greater the loss in accuracy. We further quantify this relation in Figure 4c. The Spearman correlation between the accuracy changes is very high.

Initially, we defined *easy* and *hard* tasks using the Structural Similarity Index Measure (SSIM), which quantifies visual similarity between image pairs. As shown in Figure 4, SSIM-easy task (`0` vs `7`) exhibits only a small drop in accuracy, whereas SSIM-hard task (`7` vs `9`) suffers accuracy losses approaching 50%. However, SSIM requires image data. On the other hand, our approach is data-agnostic. This means that our probes can be applied to any model and any data modality, as long as it contains MLP components.

5 Measuring layer robustness in a language model

In this case study, we evaluated the robustness of individual layers in a pretrained DistilBERT model[1] [52], fine-tuned on the CoNLL-2003 NER dataset [56]. DistilBERT, a distilled variant of BERT [14], contains approximately 65 million parameters. We focus on the named entity recognition

[1] `https://huggingface.co/dslim/distilbert-NER`

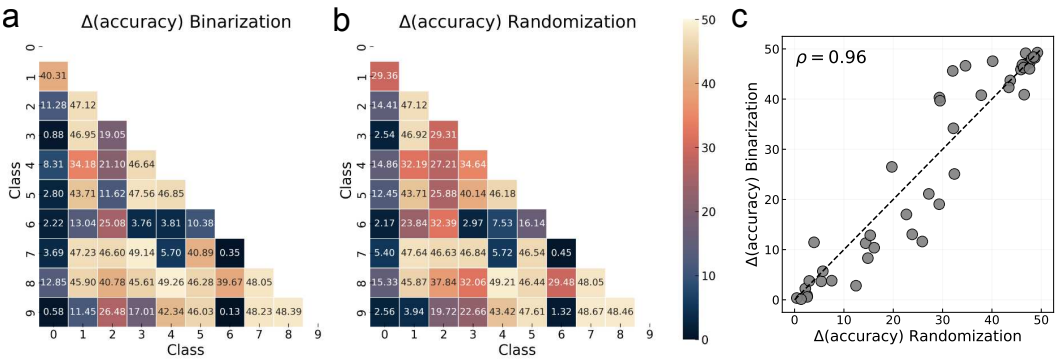

Figure 4: Difference in accuracy of the neural network for two-class discrimination under two modifications: (**a**) weight binarization, (**b**) application of bipartite randomization B. Each entry indicates the change in accuracy introduced by the modification, averaged over 10 realizations. (c) Scatter plot of the change in accuracy under randomization versus binarization. Each point represents one digit class pair. In the top left corner, the Spearman correlation coefficient is reported.

task, which aims to identify and categorize entities within text. We apply our diagnostic probes independently to six layers of the model: (1) the positional embedding layer, (2) the first linear layer of the first transformer block, (3) the second linear layer of the first transformer block, (4) the first linear layer of the final transformer block, (5) the second linear layer of the final transformer block, and (6) the token-classification (output) layer. For each probe, we report the test F1-score on the NER task.

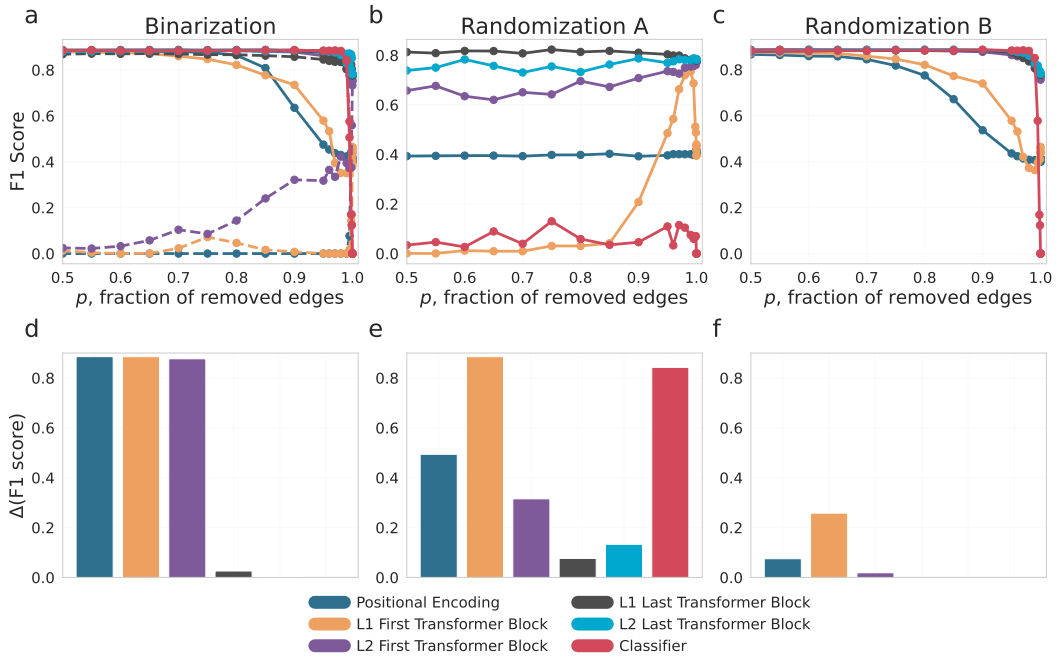

Figure 5: Case study on the DistilBERT model. F1 score as a function of the proportion of removed edges: (a) weight-binarization experiment—solid line: original weights; dotted line: binarized weights, (b) after applying randomization A, (c) after applying randomization B. In the bottom panel, for $p = 0$ (no edges removed), we plot the F1-score differences under (d) weight binarization, (e) randomization A, and (f) randomization B. Each color corresponds to a different probed layer of DistilBERT, evaluated on the NER task.

In Figures 5a-c, we plot the F1 score as a function of the fraction of removed edges for six transformer layers. First, consider the binarization experiment (Fig. 5a). The earliest layers suffer a rapid

decline in F1 score as the edges are removed, with the signed model's predictions becoming nearly indistinguishable from random. In contrast, the deepest layers remain remarkably robust: even under extreme edge removal, their binarized counterparts sustain high F1 scores. To make this comparison explicit, Figure 5d shows the difference between the original and binarized F1 curves for each layer. This pattern aligns with the findings of Bai et al. [2], who demonstrated that shallower transformer layers are more susceptible to quantization errors than deeper ones.

Next, we examine two bipartite randomization schemes, A and B. Under randomization A (Fig. 5b), which, in isolation, previously degraded accuracy to chance, the model still retains reasonable performance when edges are removed. We attribute this resilience to the residual connections in each linear sublayer, which effectively bypass the randomized weights. Randomization B (Fig. 5c) has virtually no impact on the F1 score in the later layers, underscoring the inherent robustness of these models.

Finally, by comparing the performance drops induced by binarization versus those induced by randomization B (Figs. 5d and 5f), we see that binarization causes a substantially larger performance drop in the early layers. This is unsurprising: perturbations at the network's input propagate through all subsequent layers, amplifying their effect on overall performance. Randomization, on the contrary, produces a more modest decline. Yet, it follows the same relative layer-wise pattern, with early layers more affected than later ones.

The same overall trends persist in the noise-injection and sign-flip experiments (Fig. 9). However, under sign flipping, the positional-encoding layer's performance curve becomes non-monotonic. We believe this arises from the interplay of residual connections and LayerNorm, which together render the network invariant to a global sign inversion. Specifically, when $q = 1$, we invert the entire positional-encoding vector, as the model contains six transformer blocks—an even number—each successive sign inversion is counteracted by the next, so that by the time the representation reaches the final classification head, the original encoding is effectively restored.

6 Conclusions

Understanding task complexity is essential for designing robust neural networks, guiding model selection, and optimizing training. This work investigated the internal representations of MLP layers, contrasting models trained on *easy* versus *hard* tasks using five experimental probes: pruning, binarization, noise injection, flipping signs, and bipartite network randomization. Our findings demonstrate that task complexity fundamentally shapes the robustness of learned representations to perturbations. Critically, binarizing a model trained on a hard task causes its accuracy to collapse, while an easy-task model remains robust. Pruning the binarized hard-task model showed a sharp performance transition, unlike the easy-task model. Adding noise to binarized models can boost accuracy (stochastic resonance), and we found that this effect is linked to flipping the signs of the smallest weights, indicating the importance of weight signs. Bipartite network randomization experiments confirmed that the sign structure of weights is more critical than their precise magnitudes for maintaining performance. Only randomizations preserving both positive/negative degree distributions and original edge signs maintained high accuracy. These probes suggest a data-agnostic method to quantify task difficulty, where the magnitude of accuracy loss after binarization or randomization correlates with the task's difficulty or class distinction. Applying these probes to DistilBERT on a Named Entity Recognition task revealed that early layers are less robust than later layers to binarization, randomization, and pruning. The resilience of later layers, even under randomization, may be partly due to the presence of residual connections. In summary, our study highlights the impact of task complexity on learned representations, emphasizing the crucial role of weight signs and connectivity. Practically, layers where weight signs dominate performance could be candidates for binarization during inference. Future work could explore the link between these probe-based robustness measures and representational similarity metrics, such as CKA [27] or RSA [26].

Acknowledgments and Disclosure of Funding

We acknowledge the support of the AccelNet-MultiNet program, a project of the National Science Foundation (Award #1927425 and #1927418). R. J. acknowledges support from the fellowship FI-SDUR funded by Generalitat de Catalunya. F. R. acknowledges support from the Air Force Office

of Scientific Research (Grant No. FA9550-24-1-0039). M. Á. S. and M. B. acknowledge support from Grant No. TED2021-129791B-I00 funded by MCIN/AEI/10.13039/501100011033 and by "European Union NextGenerationEU/PRTR", and Grant No. PID2022-137505NB-C22 funded by MCIN/AEI/10.13039/501100011033 and by "ERDF A way of making Europe".

References

[1] L. J. Ba and R. Caruana. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/b0c355a9dedccb50e5537e8f2e3f0810-Paper.pdf.

[2] H. Bai, W. Zhang, L. Hou, L. Shang, J. Jin, X. Jiang, Q. Liu, M. Lyu, and I. King. BinaryBERT: Pushing the limit of BERT quantization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4334–4348, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.334. URL https://aclanthology.org/2021.acl-long.334/.

[3] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[4] R. Benzi, A. Sutera, and A. Vulpiani. The mechanism of stochastic resonance. *Journal of Physics A: Mathematical and General*, 14(11):L453, nov 1981. doi: 10.1088/0305-4470/14/11/006. URL https://dx.doi.org/10.1088/0305-4470/14/11/006.

[5] R. Benzi, G. Parisi, A. Sutera, and A. Vulpiani. Stochastic resonance in climatic change. *Tellus A: Dynamic Meteorology and Oceanography*, 34(1):10, Jan. 1982. ISSN 1600-0870. doi: 10.3402/tellusa.v34i1.10782. URL http://dx.doi.org/10.3402/tellusa.v34i1.10782.

[6] R. Benzi, G. Parisi, A. Sutera, and A. Vulpiani. A theory of stochastic resonance in climatic change. *SIAM Journal on Applied Mathematics*, 43(3):565–578, June 1983. ISSN 1095-712X. doi: 10.1137/0143037. URL http://dx.doi.org/10.1137/0143037.

[7] L. Bereska and E. Gavves. Mechanistic interpretability for ai safety–a review. *arXiv preprint arXiv:2404.14082*, 2024.

[8] J. Bethge, C. Bartz, H. Yang, Y. Chen, and C. Meinel. MeliusNet: Can Binary Neural Networks Achieve MobileNet-level Accuracy? *arXiv preprint arXiv:2001.05936*, 2020.

[9] C. Blöcker, M. Rosvall, I. Scholtes, and J. D. West. Insights from network science can advance deep graph learning. *arXiv preprint arXiv:2502.01177*, 2025.

[10] G. Bonifazi, F. Cauteruccio, E. Corradini, M. Marchetti, D. Ursino, and L. Virgili. A network analysis-based framework to understand the representation dynamics of graph neural networks. *Neural Computing and Applications*, 36(4):1875–1897, 2024.

[11] D. Castelvecchi. Can we open the black box of ai? *Nature News*, 538(7623):20, 2016.

[12] H. Cheng, M. Zhang, and J. Q. Shi. A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10558–10578, 2024. doi: 10.1109/TPAMI.2024.3447085.

[13] M. Courbariaux, Y. Bengio, and J. David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in Neural Information Processing Systems*, volume 28, pages 3123–3131, 2015.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

[15] Y. Du, L. Wang, L. Guo, J. Han, T. Liu, and X. Hu. Topological similarity between artificial and biological neural networks. In *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*, pages 1–5. IEEE, 2023.

[16] K. Dwivedi and G. Roig. Representation similarity analysis for efficient task taxonomy and transfer learning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12379–12388, 2019. doi: 10.1109/CVPR.2019.01267.

[17] B. El, D. Choudhury, P. Liò, and C. K. Joshi. Towards mechanistic interpretability of graph transformers via attention graphs. *arXiv preprint arXiv:2502.12352*, 2025.

[18] J. Frankle and M. Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=rJl-b3RcF7`.

[19] L. Gammaitoni, P. Hänggi, P. Jung, and F. Marchesoni. Stochastic resonance. *Reviews of Modern Physics*, 70(1):223–287, Jan. 1998. ISSN 1539-0756. doi: 10.1103/revmodphys.70.223. URL `http://dx.doi.org/10.1103/RevModPhys.70.223`.

[20] M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[21] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

[22] B. Hassibi, D. G. Stork, and G. J. Wolff. Optimal Brain Surgeon and General Network Pruning. In *Proceedings of the IEEE International Conference on Neural Networks*, 1993.

[23] Y. He, X. Zhang, and J. Sun. Channel Pruning for Accelerating Very Deep Neural Networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1389–1397, 2017.

[24] T. K. Ho. Complexity of representations in deep learning. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 2657–2663. IEEE, 2022.

[25] C. Jiang, Z. Huang, T. Pedapati, P.-Y. Chen, Y. Sun, and J. Gao. Network properties determine neural network performance. *Nature Communications*, 15(1):5718, 2024.

[26] M. Klabunde, T. Schumacher, M. Strohmaier, and F. Lemmerich. Similarity of neural network models: A survey of functional and representational measures. *ACM Computing Surveys*, 2023.

[27] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*, pages 3519–3529, 2019.

[28] N. Kriegeskorte, M. Mur, and P. A. Bandettini. Representational similarity analysis—connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2:4, 2008.

[29] A. Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012.

[30] E. La Malfa, G. La Malfa, G. Nicosia, and V. Latora. Characterizing learning dynamics of deep neural networks via complex networks. In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 344–351. IEEE, 2021.

[31] A. K. Lampinen, S. C. Chan, and K. Hermann. Learned feature representations are biased by complexity, learning order, position, and more. *arXiv preprint arXiv:2405.05847*, 2024.

[32] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

[33] N. Lee, T. Ajanthan, and P. H. S. Torr. SNIP: Single-Shot Network Pruning Based on Connection Sensitivity. In *International Conference on Learning Representations*, 2019.

[34] Y. Li, Z. Zhang, B. Liu, Z. Yang, and Y. Liu. Modeldiff: testing-based dnn similarity comparison for model reuse detection. In *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA '21, page 139–151. ACM, July 2021. doi: 10.1145/3460319.3464816. URL `http://dx.doi.org/10.1145/3460319.3464816`.

[35] T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang. Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing*, 461:370–403, 2021.

[36] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 1449–1457, 2015.

[37] Y. Lu, W. Yang, Y. Zhang, Z. Chen, J. Chen, Q. Xuan, Z. Wang, and X. Yang. Understanding the dynamics of dnns using graph modularity. In *European Conference on Computer Vision*, pages 225–242. Springer, 2022.

[38] S. Ludwig. Stochastic resonance improves the detection of low contrast images in deep learning models. *arXiv preprint arXiv:2502.14442*, 2025.

[39] O. Madani, D. Pennock, and G. Flake. Co-validation: Using model disagreement on unlabeled data to validate classification algorithms. In *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2004. URL `https://proceedings.neurips.cc/paper_files/paper/2004/file/92f54963fc39a9d87c2253186808ea61-Paper.pdf`.

[40] C. Marx, F. Calmon, and B. Ustun. Predictive multiplicity in classification. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6765–6774. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/marx20a.html`.

[41] D. C. Mocanu, E. Mocanu, P. Stone, P. H. Nguyen, M. Gibescu, and A. Liotta. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature communications*, 9(1):2383, 2018.

[42] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz. Pruning convolutional neural networks for resource efficient inference. In *International Conference on Learning Representations*, 2017.

[43] K. Mukherjee and T. T. Rogers. How does task structure shape representations in deep neural networks? In *NeurIPS 2020 Workshop SVRHM*.

[44] J. O. Neill. An overview of neural network compression. *arXiv preprint arXiv:2006.03669*, 2020.

[45] M. E. Newman. The structure of scientific collaboration networks. *Proceedings of the national academy of sciences*, 98(2):404–409, 2001.

[46] M. E. Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[47] M. T. Pearce, T. Dooms, A. Rigg, J. M. Oramas, and L. Sharkey. Bilinear mlps enable weight-based mechanistic interpretability. *arXiv preprint arXiv:2410.08417*, 2024.

[48] M. Pósfai and A.-L. Barabási. *Network science*, volume 3. Citeseer, 2016.

[49] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep representations. In *Advances in Neural Information Processing Systems*, volume 30, pages 6076–6085, 2017.

[50] D. Rai, Y. Zhou, S. Feng, A. Saparov, and Z. Yao. A practical review of mechanistic interpretability for transformer-based language models. *arXiv preprint arXiv:2407.02646*, 2024.

[51] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks. In *European Conference on Computer Vision*, pages 525–542, 2016.

[52] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[53] L. Scabini, B. De Baets, and O. M. Bruno. Improving deep neural network random initialization through neuronal rewiring. *Neurocomputing*, 599:128130, 2024.

[54] L. F. Scabini and O. M. Bruno. Structure and performance of fully connected neural networks: Emerging complex network properties. *Physica A: Statistical Mechanics and its Applications*, 615:128585, 2023.

[55] L. Sharkey, B. Chughtai, J. Batson, J. Lindsey, J. Wu, L. Bushnaq, N. Goldowsky-Dill, S. Heimersheim, A. Ortega, J. Bloom, et al. Open problems in mechanistic interpretability. *arXiv preprint arXiv:2501.16496*, 2025.

[56] E. F. Tjong Kim Sang and F. De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003. URL `https://www.aclweb.org/anthology/W03-0419`.

[57] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

[58] A. Waqas, H. Farooq, N. C. Bouaynaya, and G. Rasool. Exploring robust architectures for deep artificial neural networks. *Communications Engineering*, 1(1), Dec. 2022. ISSN 2731-3395. doi: 10.1038/s44172-022-00043-2. URL `http://dx.doi.org/10.1038/s44172-022-00043-2`.

[59] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[60] S. Xie, A. Kirillov, R. Girshick, and K. He. Exploring randomly wired neural networks for image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1284–1293, 2019.

[61] J. You, J. Leskovec, K. He, and S. Xie. Graph structure of neural networks. In *International Conference on Machine Learning*, pages 10881–10891. PMLR, 2020.

[62] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. doi: 10.1109/CVPR.2018.00068.

[63] X.-J. Zhang, J. M. Moore, G. Yan, and X. Li. Universal structural patterns in sparse recurrent neural networks. *Communications Physics*, 6(1):243, 2023.

[64] J. Zilly, L. Hetzel, A. Censi, and E. Frazzoli. Quantifying the effect of representations on task complexity, 2019. URL `https://arxiv.org/abs/1912.09399`.

A Appendix

A.1 Structural Similarity Index distance between pairs of classes

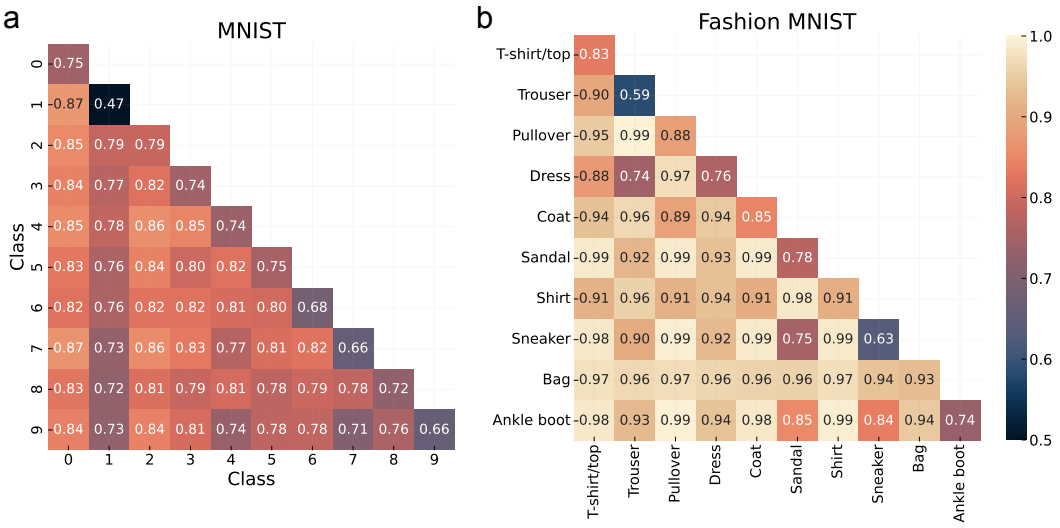

Figure 6: The Structural Similarity Index (SSIM) distance between all pairs of classes for (a) MNIST and (b) Fashion MNIST. The lower the value, the more similar the two pairs of classes are.

A.2 Standard deviations of learned weights

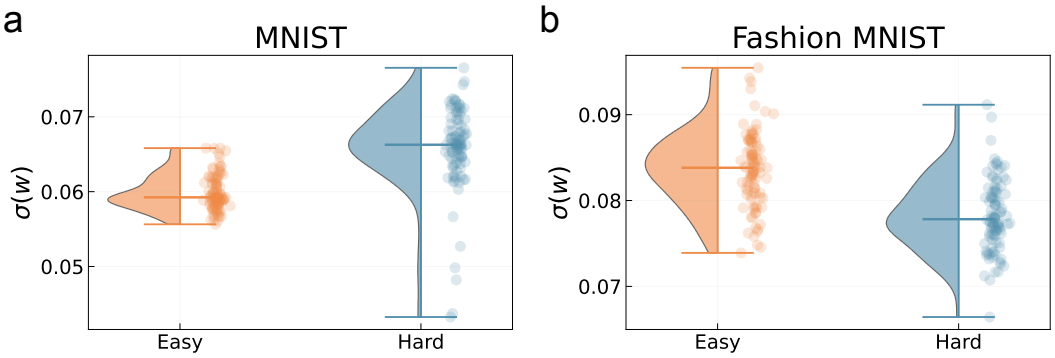

Figure 7: The distribution of weight standard deviations for (a) MNIST and (b) Fashion MNIST. Each point corresponds to a single trained neural network.

A.3 Accuracy heatmap for MNIST

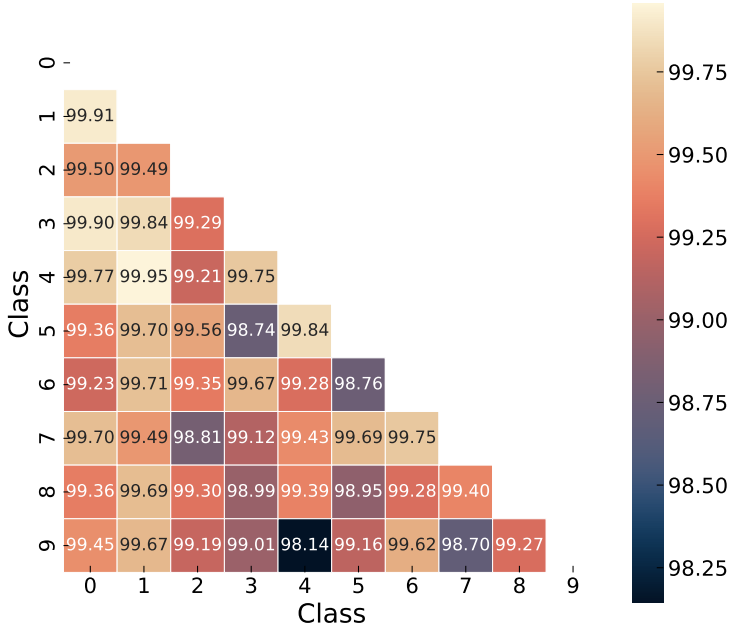

Figure 8: The accuracy heatmap for MNIST. Each entry shows the accuracy of the neural network trained to distinguish between two classes. The results are averaged over 10 realizations.

A.4 Additional experiments for DistilBERT

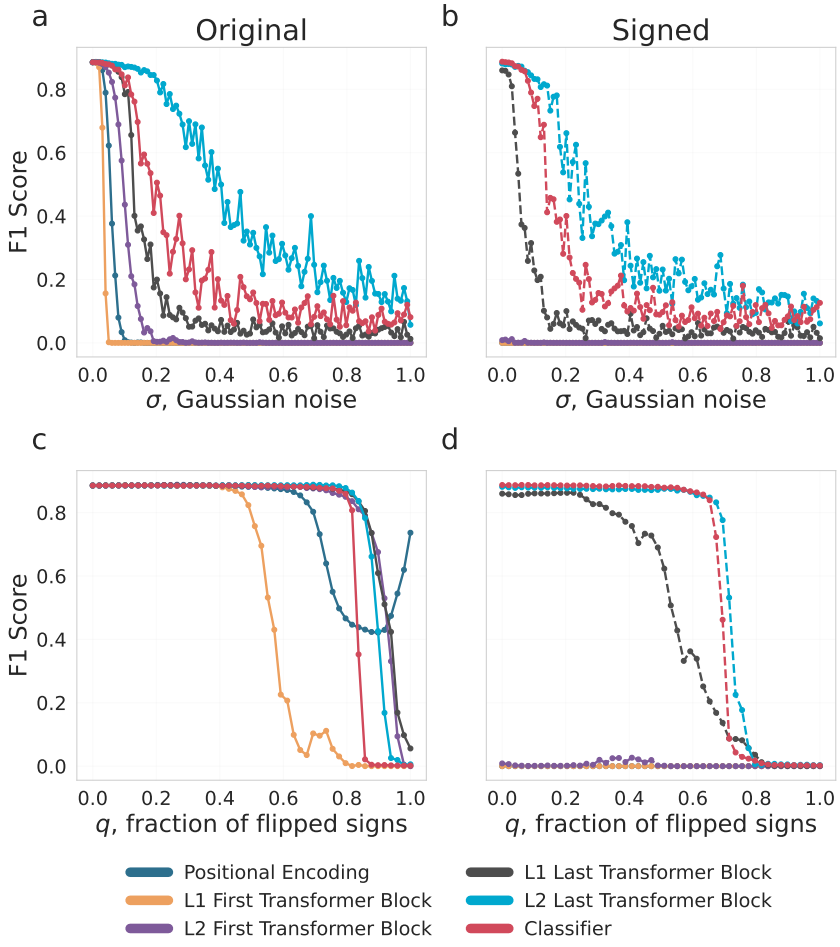

Figure 9: (a,b) The noise injection experiment. The F1 score as a function of the Gaussian noise injected into the weights. (c,d) Sign flip experiment. The F1 score as a function of the fraction of the smallest-magnitude sign flipped. Each color corresponds to a different probed layer of DistilBERT, evaluated on the NER task.

A.5 Additional experiments for MNIST and Fashion-MNIST datasets

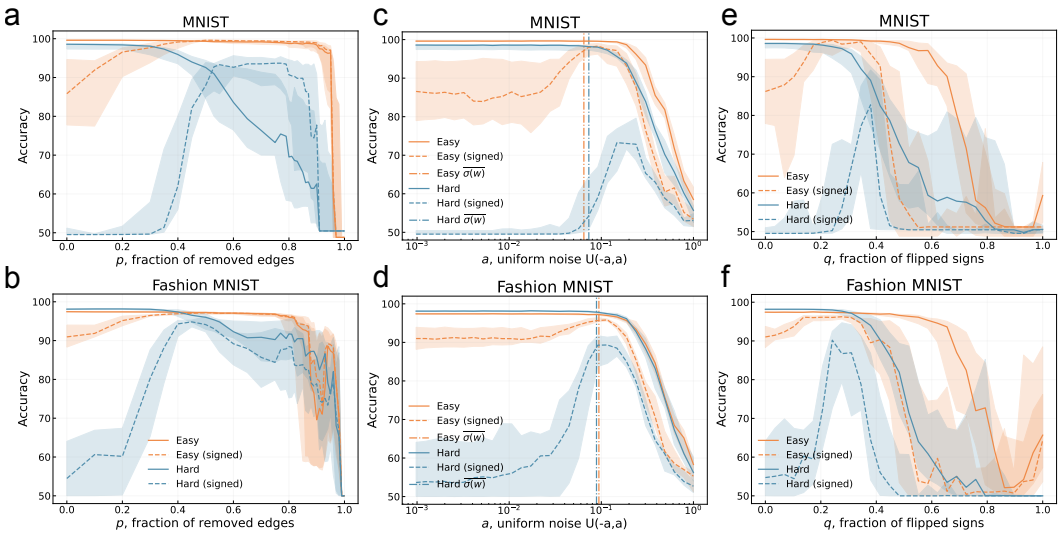

Figure 10: (**a, b**) Pruning experiment. The test accuracy as a function of the fraction of removed edges. (**c, d**) Noise injection experiment. The test accuracy as a function of the uniform noise level injected into the weights. The vertical lines show the average standard deviation of the weights. (**e, f**) Sign flipping experiment. The test accuracy as a function of the fraction of the smallest-magnitude sign flipped. All curves are averaged over 20 random initializations **for hidden layer size** $d = 32$. Shaded regions denote the interquartile range (IQR), and the solid lines represent the median.

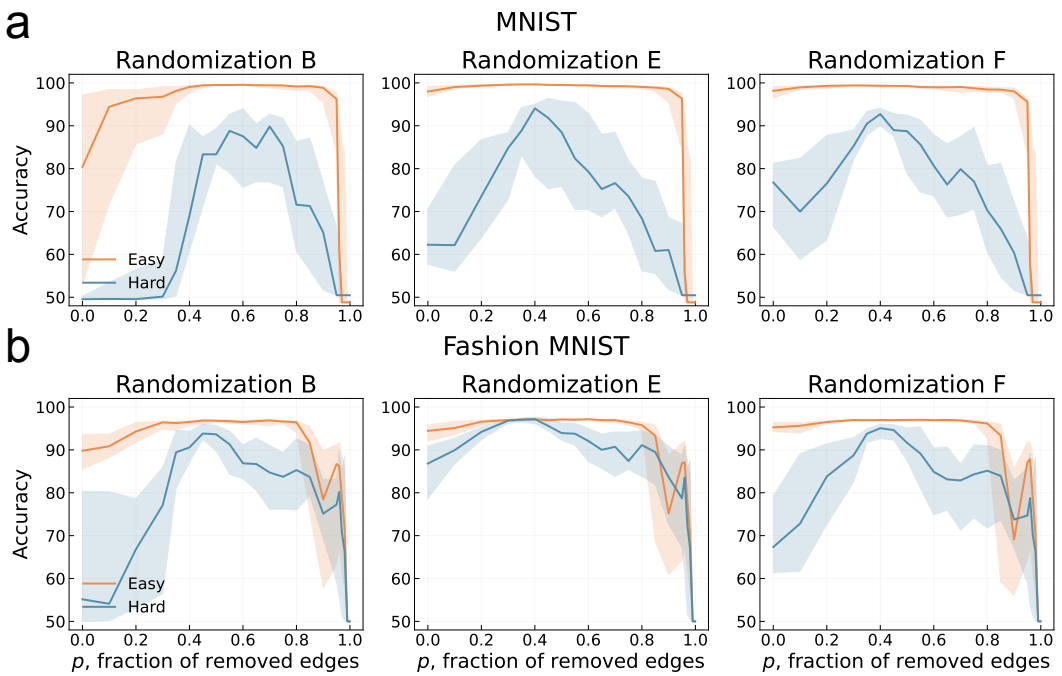

Figure 11: The test accuracy in a function of the fraction of removed edges after applying bipartite randomization B, E, and F for (**a**) MNIST and (**b**) Fashion-MNIST **for hidden layer size** $d = 32$. All curves are averaged over 100 random initializations. Shaded regions denote the interquartile range (IQR), and the solid lines represent the median.

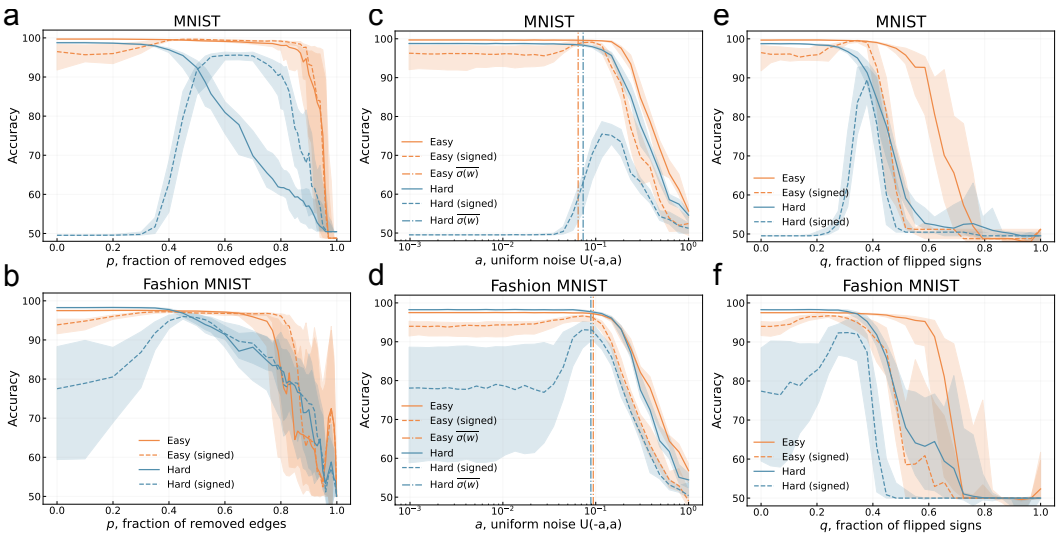

Figure 12: (**a, b**) Pruning experiment. The test accuracy as a function of the fraction of removed edges. (**c, d**) Noise injection experiment. The test accuracy as a function of the uniform noise level injected into the weights. The vertical lines show the average standard deviation of the weights. (**e, f**) Sign flipping experiment. The test accuracy as a function of the fraction of the smallest-magnitude sign flipped. All curves are averaged over 20 random initializations **for hidden layer size** $d = 128$. Shaded regions denote the interquartile range (IQR), and the solid lines represent the median.

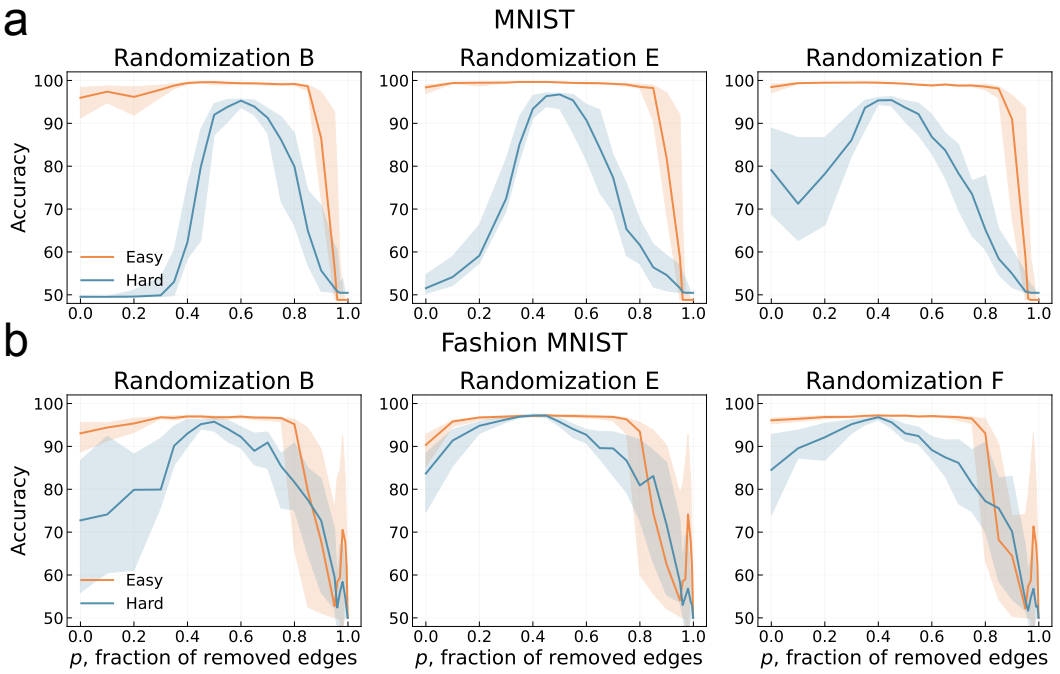

Figure 13: The test accuracy in a function of the fraction of removed edges after applying bipartite randomization B, E, and F for (**a**) MNIST and (**b**) Fashion-MNIST **for hidden layer size** $d = 128$. All curves are averaged over 100 random initializations. Shaded regions denote the interquartile range (IQR), and the solid lines represent the median.