Connecting Parameter Magnitudes and Hessian Eigenspaces at Scale using Sketched Methods

Anonymous authors
Paper under double-blind review

Abstract

Recently, it has been observed that when training a deep neural net with SGD, the majority of the loss landscape's curvature quickly concentrates in a tiny *top* eigenspace of the loss Hessian, which remains largely stable thereafter. Independently, it has been shown that successful magnitude pruning masks for deep neural nets emerge early in training and remain stable thereafter. In this work, we study these two phenomena jointly and show that they are connected: We develop a methodology to measure the similarity between arbitrary parameter masks and Hessian eigenspaces via Grassmannian metrics. We identify overlap as the most useful such metric due to its interpretability and stability. To compute overlap, we develop a matrix-free algorithm based on sketched SVDs that allows us to compute over 1000 Hessian eigenpairs for nets with over 10M parameters—an unprecedented scale by several orders of magnitude. Our experiments reveal an overlap between magnitude parameter masks and top Hessian eigenspaces consistently higher than chance-level, and that this effect gets accentuated for larger network sizes. This result indicates that *top Hessian eigenvectors tend to be concentrated around larger parameters*, or equivalently, that *larger parameters tend to align with directions of larger loss curvature.* Our work provides a methodology to approximate and analyze deep learning Hessians at scale, as well as a novel insight on the structure of their eigenspace.

1 Introduction

Deep learning (DL) benefits from overparametrization; but not all parameters are equally important. Often, a substantial portion of parameters can be *pruned*, i.e. removed, without compromising the model's performance (see Blalock et al., 2020; Hoefler et al., 2021). One efficient and popular method to identify these subnetworks is via parameter magnitude (Han et al., 2015). Interestingly, these subnetworks materialize very early in training (Frankle & Carbin, 2019), and once they emerge, their topology stops changing significantly (Achille et al., 2019; You et al., 2020). In other words, competitive subnetworks *crystallize early* in training and remain *stable* thereafter (Section 2.2, Figures 12 and 13).

Concurrent research focuses on the loss Hessian, characterizing the loss landscape's curvature. Multiple studies found that empirically the Hessian spectrum separates into two parts: The *bulk* subspace, with many, near-zero eigenvalues, and the *top* subspace with a few eigenvalues of significantly larger magnitude (e.g. Dauphin et al., 2014; Sagun et al., 2018). Crucially, Gur-Ari et al. (2018) observed that, after initial training iterations, the gradient predominantly lies in this top subspace where it remains relatively stable throughout training. Analogous to the work on parameter masks, this indicates that the top Hessian eigenspace *crystallizes early* in training and tends to remain *stable* (Section 2.1, Figures 12 and 13).

In this work, we explore the connection between these largely independent lines of research—both reporting *early crystallization* of substructures. Our **contributions** include:

1. We establish a common theoretical ground for comparing binary parameter masks $\boldsymbol{m}_k$ (which select k parameters and discard the rest) and top-k eigenspaces of a Hessian. Both can be cast as rank-k orthogonal matrices belonging to the same Stiefel manifold (Section 3). This allows a direct comparison of their spans

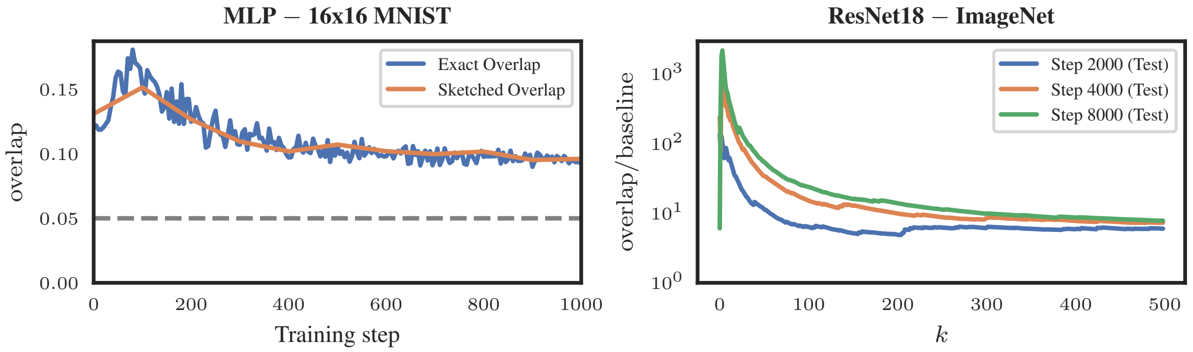

Figure 1: Overlap between top-k parameter magnitude masks and top-k Hessian eigenspaces is consistently and substantially above random chance. *(Left)* Measurements for a 7030-parameter network trained on 16×16 downsampled MNIST until convergence (see Section 6.1 and Figure 12): Exact overlap between top-k parameters and eigenvectors with $k = 350$ (—), approximate overlap via sketched eigendecomposition (—), and chance-level baseline (- -) (see Section 4.2). Note how overlap is larger than chance, and sketched overlap is a good approximation. *(Right)* Lines show the ratio *sketched* overlap *vs. chance-level baseline* for a model with $>$11M parameters trained on ImageNet (see Section 6.2), at three different points during training, and as a function of k. Note how overlap is always higher than baseline, up to a factor of 1000.

using *Grassmannian metrics*[1] (see Section 4.1). We review popular Grassmannian metrics and identify overlap (Eq. (2)) as the most meaningful metric (Section 4.2).

2. To efficiently compute overlap, we develop SEIGH (Section 5 and Alg. 2), a matrix-free eigendecomposition based on sketched SVDs (Tropp et al., 2019). Our open source implementation[2] allows to compute top-k Hessian eigendecompositions for $k > 10^3$ on neural networks with over 10M parameters, an unprecedented scale by orders of magnitude.

3. We provide empirical evidence that the similarities between the spaces induced by parameter magnitude masks and top-k Hessian eigenspaces are consistently and substantially higher than random chance (Figure 1). This suggests that, in DL, top Hessian eigenvectors tend to be concentrated around larger parameters throughout the training process (Section 6).

Seminal work on "Optimal Brain Damage" (LeCun et al., 1989) established a theoretical link between *individual parameter values* and loss curvature, but only *at convergence*. Recent work often focuses on loss curvature for *layer-wise* parameter groups (e.g. Sankar et al., 2021), or involves just a few ($k \approx 10$) Hessian eigenpairs without emphasis on parameters (e.g. Gur-Ari et al., 2018; Papyan, 2020; Dangel et al., 2022). In contrast, we connect Hessian *eigenspaces* to the *spans* of arbitrary parameter subsets, e.g. magnitude masks, throughout the *entire training*—including the relevant early stage—in an interpretable and scalable manner. We can thus jointly study the connection between these two phenomena and provide novel insights into a neural net's Hessian structure. This connection implies being able to approximate expensive Hessian quantities via cheap parameter inspection, which bears potential for downstream tasks where the Hessian plays a prominent role, such as optimization (Martens, 2016), pruning (LeCun et al., 1989), or uncertainty estimation (Kristiadi et al., 2021).

2 Hessians, Parameters and Early Crystallization

We consider a supervised classification/regression setup with train set $\mathcal{D}_{\text{train}} := \{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^N$ of labeled data $(\boldsymbol{x}_n, \boldsymbol{y}_n) \in \mathbb{X} \times \mathbb{Y}$, from an unknown data-generating distribution P. The neural net $f_{\boldsymbol{\theta}}(\boldsymbol{x}) : \mathbb{X} \to \mathbb{Y}$ maps inputs $\boldsymbol{x}$ to predictions $\hat{\boldsymbol{y}}$ via parameters $\boldsymbol{\theta} \in \mathbb{R}^D$. A loss function $\ell : \mathbb{Y} \times \mathbb{Y} \to \mathbb{R}_{\geqslant 0}$ penalizes differences between prediction

---

[1] *Grassmannians* are manifolds of low-dimensional subspaces of a given vector space.

[2] `https://anonymous.4open.science/r/hessian_overlap-3EB1`

$\hat{\boldsymbol{y}}$ and true label $\boldsymbol{y}$. The goal is to minimize the inaccessible *risk* $L_P(\boldsymbol{\theta}) := \int \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y}) dP$ via the proxy *empirical risk* $L_{\mathcal{D}_{\text{train}}}(\boldsymbol{\theta}) := \frac{1}{N} \sum_{n=1}^{N} \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n), \boldsymbol{y}_n)$. For large N, we approximate $L_{\mathcal{D}_{\text{train}}}(\boldsymbol{\theta})$ using *mini-batches* $\mathcal{B} \overset{\text{iid}}{\sim} \mathcal{D}_{\text{train}}$ of $B \ll N$ samples. For f, ℓ twice differentiable, we use the *gradient* $\boldsymbol{g}(\boldsymbol{\theta}) := \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) \in \mathbb{R}^D$ and the *Hessian* $\boldsymbol{H}(\boldsymbol{\theta}) := \nabla^2_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) \in \mathbb{R}^{D \times D}$. With $\boldsymbol{g}, \boldsymbol{H}$ we refer to *any* loss gradient or Hessian. To emphasize the data domain, we use a subindex, e.g. $\boldsymbol{H}_{\mathcal{B}}$ for the Hessian of the mini-batch loss.

2.1 The Hessian in Deep Learning

The Hessian plays a prominent role in DL applications. A useful characterization of $\boldsymbol{H}$ is its eigendecomposition $\boldsymbol{H} = \boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{U}^\top = \sum_{i=1}^{D} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^\top$. Here, $\boldsymbol{U}$ is orthogonal, with *eigenvectors* $\boldsymbol{u}_i$, and $\boldsymbol{\Lambda}$ is diagonal and real-valued with (ordered) *eigenvalues* $|\lambda_1| \geqslant \ldots \geqslant |\lambda_D|$. We call $\boldsymbol{U}^{(k)} := \{\boldsymbol{u}_i\}_{i=1}^{k}$ the *top-k eigenbasis* of $\boldsymbol{H}$, and span$(\boldsymbol{U}^{(k)})$ the *top-k eigenspace*. The top-k eigendecomposition $\boldsymbol{H}^{(k)} = \sum_{i=1}^{k} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^\top$ minimizes $\|\boldsymbol{H} - \boldsymbol{H}^{(k)}\|$ for all unitarily invariant norms (Golub & Van Loan, 2013, Th. 2.4.8).

Recent literature has extensively investigated the Hessian **spectrum** of neural nets, revealing that the eigenvalues are typically clustered into two parts: (1) The *bulk* of eigenvalues with near-zero magnitude and (2) a few *top* eigenvalues with significantly larger magnitude (e.g. Sagun et al., 2018; Papyan, 2019). Thus, $\boldsymbol{H}$ can be well-approximated by its top eigenpairs. For the Hessian **eigenspace**, Li et al. (2018a) showed that projecting the whole space onto a few random, fixed dimensions still allows Stochastic Gradient Descent (SGD) to perform competitively—provided enough dimensions are given—leading to the idea of an intrinsic dimensionality of problems. Gur-Ari et al. (2018) observed that this restriction to a lower-dimensional, fixed subspace seems to happen spontaneously anyway: After a brief initial training period, the gradient predominately lies within a small subspace spanned by the few top Hessian eigenvectors and this space changes little over the remaining training process. However, these phenomena might rely on optimizer and model choices (Li et al., 2018a; Ghorbani et al., 2019).

One fundamental issue greatly limiting the use of $\boldsymbol{H}$ for DL is its prohibitively large size, with D^2 entries. Consequently, most scalable methods are *matrix-free*, relying on Hessian-Vector Products (HVPs) to compute linear maps $\boldsymbol{w} = \boldsymbol{H}\boldsymbol{v}$ in just $\mathcal{O}(D)$ memory and time (Pearlmutter, 1994). Examples are the computation of spectral densities (Yao et al., 2020; Papyan, 2018) or top-k eigendecompositions (Gur-Ari et al., 2018; Dangel et al., 2022). To make Hessian properties more accessible, specialized DL libraries have been developed recently (e.g. Dangel et al., 2020; Yao et al., 2020; Elsayed & Mahmood, 2022), but efficiently accessing large portions of the Hessian remains a major challenge (see Section 5).

2.2 Parameter Masks and Early Crystallization

Binary parameter masks $\boldsymbol{m}_k \in \mathbb{B}^D$, with $\mathbb{B} = \{0, 1\}$, consisting of k ones and the rest zeros, can be used to define subsets of parameters from a given model. A mask is k-sparse if it has k non-zero elements. We measure mask sparsity using the ratio $\rho := {}^{k}\!/_{D}$. For non-binary vectors $\boldsymbol{v} \in \mathbb{R}^D$, we instead measure whether a small subset of indices ι contains a large proportion of the total norm of the vector. When ι is known, this can be directly expressed as the ratio: $\kappa(\boldsymbol{v}) := {}^{\|\boldsymbol{v}_\iota\|_2^2}\!/_{\|\boldsymbol{v}\|_2^2}$ (see Hurley & Rickard, 2009). A popular choice for $\boldsymbol{m}_k$ is to take the k-largest parameters by magnitude: it is computationally cheap, and has been shown to be effective for neural net pruning (Han et al., 2015), leading to smaller models that can often achieve competitive performances with $\rho \in [1\%, 10\%]$ (e.g. Gale et al., 2019; Blalock et al., 2020; Hoefler et al., 2021).

In this work, we do not perform any pruning, but rather focus on a key feature of parameter magnitude masks. Not only can they lead to competitive performance when used for pruning and be found very early in training (Frankle & Carbin, 2019; Frankle et al., 2020) but they also *stabilize soon afterwards*: You et al. (2020) compared Hamming distances between periodically extracted pruning masks and found they stop changing early in training, aligning with the loss of information plasticity reported in Achille et al. (2019). This *early crystallization* bears a striking parallel with observations made for the top Hessian subspace, and motivates our investigation: Are these two phenomena connected? How would one measure this connection? What would such a connection tell us about the parameters and the Hessian?

3 Parameter Masks as Orthogonal Projections

Quantifying the connection between parameter magnitude masks and top-k Hessian eigenspaces—both exhibiting early crystallization—requires a way to relate a mask to a subspace. In this section, we observe that both $\boldsymbol{U}^{(k)}$ (the top-k eigenbasis) and any k-sparse mask[3] $\boldsymbol{m}_k$ can be characterized as elements of the same compact Stiefel manifold $\mathbb{O}^{D\times k} = \{\boldsymbol{Q}{:}\boldsymbol{Q}\in\mathbb{R}^{D\times k},\ \boldsymbol{Q}^\top\boldsymbol{Q}=\boldsymbol{I}_k\}$ (Absil et al., 2004), where $\boldsymbol{I}_k\in\mathbb{R}^{k\times k}$ denotes the identity. Specifically for masks, we further consider the subset $\mathbb{M}^{D\times k}\subset\mathbb{O}^{D\times k}$, with columns having exactly one 1, each row at most one 1, and the rest is zeros.

Reordering parameters: Recall from Section 2.1 that the Hessian eigenvalues are sorted by descending magnitude, exposing a single cutting point between *top* and *bulk* eigenspace at dimension k. To simplify notation, we impose a similar cutting point to the parameters, by defining a permutation matrix $\boldsymbol{P}\in\mathbb{M}^{D\times D}$ for any given mask $\boldsymbol{m}_k$, such that the mask entries are grouped in *selected* (i.e. k entries with $m_i=1$) and *discarded* (i.e. $m_i=0$), i.e. $\tilde{\boldsymbol{m}} := \boldsymbol{P}^\top\boldsymbol{m} = (1,\dots,1,0,\dots,0)$. For a parameter magnitude mask, for example, we can define $\boldsymbol{P}$ to order the parameters by nonincreasing magnitude such that $i\leqslant j \Rightarrow |\boldsymbol{Pm}|_i \geqslant |\boldsymbol{Pm}|_j$. We can now permute the parameters $\tilde{\boldsymbol{\theta}}=\boldsymbol{P}^\top\boldsymbol{\theta}$, as well as the Hessian rows and columns $\tilde{\boldsymbol{H}} := \boldsymbol{P}^\top\boldsymbol{H}\boldsymbol{P} = \tilde{\boldsymbol{U}}\boldsymbol{\Lambda}\tilde{\boldsymbol{U}}^\top$. This has no loss of generality, since $(\boldsymbol{m},\boldsymbol{\theta},\boldsymbol{H})\cong(\tilde{\boldsymbol{m}},\tilde{\boldsymbol{\theta}},\tilde{\boldsymbol{H}})$ is an isomorphism, $\boldsymbol{H}$ and $\tilde{\boldsymbol{H}}$ are *similar*, and the loss curvature remains unaltered ($\tilde{\boldsymbol{\theta}}^\top\tilde{\boldsymbol{H}}\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}^\top\boldsymbol{H}\boldsymbol{\theta}$). Then, any permuted k-sparse masking operation can be expressed as:

$$\boldsymbol{P}^\top(\boldsymbol{m}_k\odot\boldsymbol{\theta}) = \tilde{\boldsymbol{m}}_k\odot\tilde{\boldsymbol{\theta}} = \underbrace{\begin{pmatrix}\boldsymbol{I}_k & 0\\ 0 & 0\end{pmatrix}}_{:=\tilde{\boldsymbol{\Phi}}}\tilde{\boldsymbol{\theta}} =: \boldsymbol{I}_{D,k}\boldsymbol{I}_{D,k}^\top\tilde{\boldsymbol{\theta}},$$

where $\boldsymbol{I}_{D,k} = \binom{\boldsymbol{I}_k}{\boldsymbol{0}}\in\mathbb{R}^{D\times k}$, which is clearly an element of the (binary) Stiefel manifold $\mathbb{M}^{D\times k}$. Also note that this holds for any $\boldsymbol{P}$, not only for those defined for parameter magnitude masks.

Partitioning $\tilde{\boldsymbol{H}}$: Consider now the following partition of the reordered Hessian with $\boldsymbol{V},\boldsymbol{D}\in\mathbb{R}^{k\times k}$, $\bar{\boldsymbol{W}},\bar{\boldsymbol{E}}\in\mathbb{R}^{(D-k)\times(D-k)}$, and $\bar{\boldsymbol{V}},\boldsymbol{W}^\top\in\mathbb{R}^{(D-k)\times(k)}$:

$$\tilde{\boldsymbol{H}} := \underbrace{\left(\begin{array}{c:c}\boldsymbol{V} & \boldsymbol{W}\\ \hdashline \bar{\boldsymbol{V}} & \bar{\boldsymbol{W}}\end{array}\right)}_{\tilde{\boldsymbol{U}}=\boldsymbol{P}^\top\boldsymbol{U}}\underbrace{\left(\begin{array}{c:c}\boldsymbol{D} & \\ \hdashline & \bar{\boldsymbol{E}}\end{array}\right)}_{\boldsymbol{\Lambda}}\underbrace{\left(\begin{array}{c:c}\boldsymbol{V}^\top & \bar{\boldsymbol{V}}^\top\\ \hdashline \boldsymbol{W}^\top & \bar{\boldsymbol{W}}^\top\end{array}\right)}_{\tilde{\boldsymbol{U}}^\top=\boldsymbol{U}^\top\boldsymbol{P}}. \tag{1}$$

With this partition, $\tilde{\boldsymbol{U}}^{(k)} =: \binom{\boldsymbol{V}}{\bar{\boldsymbol{V}}}$ is the *top-k* Hessian eigenbasis, and $\binom{\boldsymbol{W}}{\bar{\boldsymbol{W}}}$ the *bulk* eigenbasis. Conversely, the rows of $(\boldsymbol{V}|\boldsymbol{W})$ correspond to the *selected* parameters, and $(\bar{\boldsymbol{V}}|\bar{\boldsymbol{W}})$ to the *discarded* ones. Since $\tilde{\boldsymbol{U}}$ is orthogonal, we have $\tilde{\boldsymbol{U}}^{(k)}\in\mathbb{O}^{D\times k}$, i.e. an element of the same Stiefel manifold.

4 Measuring Subspace Similarity via Grassmannian Metrics

We set out now to quantify the similarity between a k-sparse parameter mask and the top-k Hessian eigenspace. Specifically, we are only interested in the similarity of their spans, since the *spaces* are the ones reported to undergo early crystallization. To connect parameter spaces to loss curvature, one may consider perturbing the parameters, and empirically measuring the impact on the loss. This turns out to be problematic due to the non-PSD nature of $\boldsymbol{H}$ (see Appendix A.1). Instead, Grassmannian metrics provide a natural and theoretically grounded way of achieving this, given that k-sparse parameter masks and the top-k Hessian eigenspace can both be cast as elements of the same Stiefel manifold (Section 3). We review popular Grassmannian metrics in Section 4.1, and analyze them in more depth in Section 4.2, finding that the overlap metric is both interpretable and stable.

4.1 Grassmann Manifolds and their Metrics

Grassmann manifolds are extensively studied (e.g. Witten, 1988; Absil et al., 2004; Bendokat et al., 2020) and have recently been applied to DL (e.g. Gur-Ari et al., 2018; Zhang et al., 2018; Dangel et al., 2022). A

---

[3] This can be generalized to masks and eigenspaces with different k using Schubert varieties (Ye & Lim, 2016).

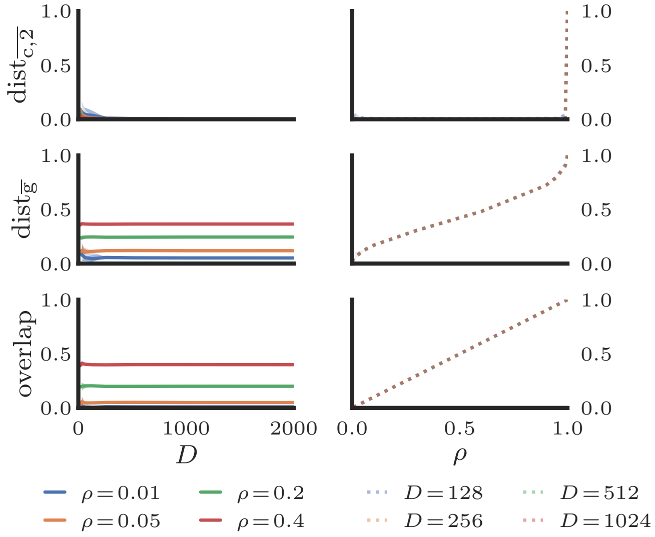

Figure 2: Behavior of different Grassmannian metrics for random pairs of matrices and masks. *(Left)* As a function of D, for different sparsity ratios $\rho := {}^{k}/_{D}$. *(Right)* As a function of ρ, for different ambient dimensions D. Lines show the median metric for 50 random pairs, the (almost imperceptible) shaded regions span the 5-95 percentiles (see Figure 8 for broader distributions). Each row shows a selected Grassmannian metric (see Appendix A.3 for more metrics): $\mathrm{dist}_{\overline{c,2}}$ is a representative example of a *collapsing* metric, being ≈ 0 almost everywhere. The overlap metric is non-collapsing, and its expectation equals ρ.

Grassmann manifold $\mathcal{G}_{k,D}$ is the set of all k-dimensional subspaces of a given D-dimensional Euclidean space. Two orthogonal matrices $\boldsymbol{Q}_1, \boldsymbol{Q}_2 \in \mathbb{O}^{D \times k}$ map to the same element $\mathfrak{g} \in \mathcal{G}_{k,D}$ if and only if their column span is identical. Thus, the subset of all matrices in $\mathbb{O}^{D \times k}$ that map to $\mathfrak{g}_i = \mathrm{span}(\boldsymbol{Q}_i)$ forms an *equivalence class* $\mathcal{S}_i^{\mathbb{O}} := \{\boldsymbol{Q}_j \colon \boldsymbol{Q}_j \boldsymbol{Z}_j = \boldsymbol{Q}_i,\ \boldsymbol{Z}_j^\top \boldsymbol{Z}_j = \boldsymbol{I}_k\}$ (e.g. Edelman et al., 1998).

The *geodesics* (i.e. shortest paths) between two elements in $\mathcal{G}_{k,D}$ are available in closed-form and can be characterized in terms of the *principal angles* $\boldsymbol{\sigma} \in [0, \frac{\pi}{2}]^k$, i.e. the "amount of rotation" required to transition from one space to another. The principal angles between two matrices in $\mathbb{O}^{D \times k}$ can be obtained via an SVD of their product, satisfying the invariance to changes within $\mathcal{S}_i^{\mathbb{O}}$ (see Appendix A.2). Based on this, there are several *Grassmannian metrics* capturing different notions of distance between subspaces (Qiu et al., 2005). Here we highlight overlap, defined as:

$$\mathrm{overlap}(\boldsymbol{Q}_1, \boldsymbol{Q}_2) = \frac{1}{k} \| \boldsymbol{Q}_1^\top \boldsymbol{Q}_2 \|_F^2 = \frac{1}{k} \| \cos(\boldsymbol{\sigma}) \|_F^2 \in [0,1] \, . \tag{2}$$

We see e.g., that if overlap is 0, then the spans of both matrices are orthogonal to each other, and an overlap of 1 indicates that both spaces are identical. We provide definitions and a more exhaustive review of overlap and other Grassmannian metrics in Appendix A.3.

4.2 Comparing Grassmannian Metrics

Our aim is now to identify the most useful Grassmannian metric for comparing neural net parameter masks and Hessian eigenspaces. To empirically compare how different Grassmannian metrics change as a function of ambient dimension D (for fixed ρ) and sparsity ratio $\rho = {}^{k}/_{D}$ (for fixed D), we conduct a synthetic experiment: We randomly draw matrices from $\mathbb{O}^{D \times k}$ and masks from $\mathbb{M}^{D \times k}$ and compute various Grassmannian metrics between them, normalized to be in $[0, 1]$, with higher metric indicating higher similarity. Results are highlighted in Figure 2 (experimental procedure and further experiments in Appendix A.4). This empirical analysis is complemented with a theoretical result showing that the expected overlap equals exactly ρ, as seen in Figure 2 (proof in Appendix A.5). The main insights from this analysis (Appendix A.6 extends the discussion) are:

I. For a fixed sparsity ratio ρ, the expectation of all measured metrics becomes *predictable* as D increases: Already in the $D \approx 10^3$ regime (much smaller than modern DL scenarios), expectations converge tightly to values that *seem to only depend on* ρ (Figure 2 left). We gathered empirical baseline values in Table 2.

II. Not all metrics are equally *informative*: Some metrics quickly collapse to 0 as D increases and ρ decreases (see e.g. Figure 2 (top) or Figures 9 and 10). Looking at the metric definitions in Appendix A.3, we can see the reason: Collapsing metrics (e.g. $\text{dist}_{\overline{c,2}}$, $\text{dist}_{\overline{p,2}}$ and $\text{dist}_{\overline{a}}$) are strongly influenced by individual angles, whereas non-collapsing metrics (e.g. $\text{dist}_{\overline{g}}$, $\text{dist}_{\overline{c,\text{F}}}$, $\text{dist}_{\overline{p,\text{F}}}$ and overlap) average over all angles.

With these insights, we identify overlap as the most suitable metric among the reviewed ones, since: (1) *It provides a stable and efficient baseline:* Among the non-collapsing metrics, it features the smoothest expectation, equaling exactly $\rho = {}^{k}/_{D}$. This provides us with a stable, simple, theoretically-grounded, and analytical baseline, establishing a **“chance-level overlap”** that we can compare against[4]. This is particularly beneficial for deep learning, bypassing the need to compute empirical baselines for large D over many ρ. (2) *It is related to other metrics:* overlap can be mapped to other popular metrics such as the Hamming distance, Intersection over Union (IoU), or $\text{dist}_{p,\text{F}}$ via bijections (see Appendix A.3). (3) *It has precedence in the literature:* The overlap metric has been used to measure eigenspace similarity in recent deep learning works (e.g. Gur-Ari et al., 2018; Dangel et al., 2022).

5 Sketched Hessian Eigendecompositions to Measure Overlap

To compute the overlap between parameter magnitude masks and top-k Hessian eigenspaces, we can now simply plug in the permutation from Section 3 into Eq. (2): $\boldsymbol{Q}_1$ becomes the permuted mask matrix $\boldsymbol{I}_{D,k}$, and $\boldsymbol{Q}_2$ the correspondingly permuted top eigenspace $\tilde{\boldsymbol{U}}^{(k)}$, yielding:

$$\text{overlap}(\boldsymbol{I}_{D,k}, \tilde{\boldsymbol{U}}^{(k)}) = \frac{1}{k} \|\boldsymbol{I}_{D,k}^{\top} \tilde{\boldsymbol{U}}^{(k)}\|_F^2 = \frac{1}{k} \|\boldsymbol{V}\|_F^2 \, , \tag{3}$$

i.e. our problem reduces to computing the Frobenius norm of $\boldsymbol{V}$, a $k \times k$ submatrix of $\tilde{\boldsymbol{U}}^{(k)}$ (Eq. (1)).

A case for eigendecompositions: We aim to estimate $\|\boldsymbol{V}\|_F^2$, where $\boldsymbol{V}$ is part of $\tilde{\boldsymbol{U}}^{(k)}$, a $D \times k$ orthogonal matrix spanning the top eigenspace. Since all standard algorithms to find such matrices (Gram–Schmidt, Householder transformation, Givens rotation) require to keep track of at least k full vectors (Golub & Van Loan, 2013), *a memory requirement of $\mathcal{O}(kD)$ seems inescapable in general.* Still, one may recognize that any orthogonal matrix $\hat{\boldsymbol{Q}}$ with the same span as $\tilde{\boldsymbol{U}}^{(k)}$ yields the same Frobenius norm (due to unitary invariance). However, we need to ensure that the span of said $\hat{\boldsymbol{Q}}$ does not overlap with the bulk eigenspace: It needs to be exclusively associated with the top-k eigenvalues. As a consequence, *knowledge of the eigenvalues also seems inescapable in general.* In conclusion, to measure the top-k overlap, we argue that a rank-k eigendecomposition is generally needed.

A case for *sketched* decompositions: Among the methods that satisfy the memory requirement of $\mathcal{O}(kD)$, **orthogonal iterations** (Golub & Van Loan, 2013), an extension of Rayleigh's power method, is a popular and effective one, requiring $\mathcal{O}(k)$ measurements per iteration and τ iterations, to converge at a rate proportional to $|\lambda_{k+1}/\lambda_k|^{\tau}$ (Golub & Van Loan, 2013, Th. 7.3.1). This leads to large τ for smoothly decaying spectra, likely scenario for $\boldsymbol{H}$, thus $\mathcal{O}(\tau k)$ Hessian-Vector Products (HVPs) can become infeasible. Ritz accelerations provide better convergence but still suffer from this issue (Golub & Van Loan, 2013, 10.1). Alternatively, **Lanczos iterations** convergence faster (Saad, 1980), (Golub & Van Loan, 2013, 10.1.6), but measurements must be done in sequential form, which can get very slow for $\boldsymbol{H}$, and they involve matrix powers, numerically unstable for rank-defficient matrices like $\boldsymbol{H}$. Lanczos iterations have been successfully applied to estimate spectral densities of large-scale deep learning Hessians (Papyan, 2018), but not to obtain their eigenpairs. **Sketched methods** (e.g. Halko et al., 2011), based on random measurements, not only satisfy $\mathcal{O}(kD)$ memory, they also exhibit *good convergence for $\mathcal{O}(k)$ measurements* and are *parallelizable and numerically stable* (Halko et al., 2011, 1.4.2, 4.2, and 6.2). This, combined with the ability to perform matrix-free random measurements (Tropp et al., 2019, 3.2), makes it possible to compute Hessian eigendecompositions at unprecedented scales, since the main bottleneck is now the $\mathcal{O}(kD)$ memory requirement (Tropp et al., 2019, 7.1) (for example, storing 1000 eigenpairs for a ResNet-18 (He et al., 2016) with 13M float parameters requires roughly 50GB of memory, same as 100 eigenpairs for a 10× larger model). Last but not least, affordable *a posteriori* methods allow to measure the approximation error and rank of the recovered decomposition (Halko et al., 2011; Tropp et al., 2019).

---

[4] In Appendix A.7 we also compare this analytical baseline to the empirical alternative of sampling random masks, showing it yields similar results.

Algorithm 1: SSVD (from Tropp et al. (2019))

```
Input: $N_I \in \mathbb{N}$                          // No. of inner measurements
Input: $N_O \leqslant N_I$                           // No. of outer measurements
Input: $A \in \mathbb{C}^{D_L \times D_R}$           // Linear operator
Input: $\Upsilon_I \in \mathbb{C}^{D_L \times N_I}$  // Left inner random matrix
Input: $\Upsilon_O \in \mathbb{C}^{D_L \times N_O}$  // Left outer random matrix
Input: $\Omega_I \in \mathbb{C}^{D_R \times N_I}$    // Right inner random matrix
Input: $\Omega_O \in \mathbb{C}^{D_R \times N_O}$    // Right outer random matrix
Output: $(P, U, \Sigma, V^*, Q^*)$ with $PU\Sigma V^* Q^* \approx A$
  // Perform outer measurements
1 $M_{LO}^* \leftarrow \Upsilon_O^* A$               // $\mathbb{C}^{N_O \times D_R}$
2 $M_{RO} \leftarrow A\Omega_O$                      // $\mathbb{C}^{D_L \times N_O}$
  // Perform inner measurements
3 $M_I \leftarrow \Upsilon_I^* A \Omega_I$           // $\mathbb{C}^{N_I \times N_I}$
  // Orthogonalize outer measurements
4 $(P, \_) \leftarrow \mathrm{qr}(M_{LO})$           // $P$ has orthonormal columns
5 $(Q, \_) \leftarrow \mathrm{qr}(M_{RO})$           // $Q$ has orthonormal columns
  // Solve core matrix via least squares and svd
6 $C \leftarrow (\Upsilon_I^* P)^\dagger M_I$
7 $C \leftarrow \big[(\Omega_I^* Q)^\dagger C\big]^*$
8 $(U, \Sigma, V^*) \leftarrow \mathrm{svd}(C)$      // $\Sigma$ are the singular values
9 return $(P, U, \Sigma, V^*, Q^*)$
```

Algorithm 2: SEIGH

```
Input: $N_I \in \mathbb{N}$                          // No. of inner measurements
Input: $N_O \leqslant N_I$                           // No. of outer measurements
Input: $A \in \mathbb{C}^{D \times D}$               // Hermitian linear operator
Input: $(\Upsilon, \Omega) \in \mathbb{C}^{D \times N_I}$  // Random measurement matrices
Output: $(Q, U, \Lambda)$ with $QU\Lambda U^* Q^* \approx A$
  // Perform all measurements. Note that $A\Omega_{1 \ldots N_O}$ is recycled
1 $M_I \leftarrow \Upsilon^* A \Omega_{1 \ldots N_I}$   // $\mathbb{C}^{D \times N_I}$
2 $M_O \leftarrow A \Omega_{1 \ldots N_O}$               // $\mathbb{C}^{D \times N_O}$
  // Orthogonalize outer measurements
3 $(Q, \_) \leftarrow \mathrm{qr}(M_O)$              // $Q$ has orthonormal columns
  // Solve core matrix via least squares and eigh
4 $(\bar{U}, \bar{\Lambda}, \bar{V}^*) \leftarrow \mathrm{svd}(M_I)$
5 $C_L \leftarrow (\Upsilon^* Q)^\dagger \bar{U}$
6 $C_R \leftarrow (\Omega^* Q)^\dagger \bar{V}$
7 $C \leftarrow C_L \bar{\Lambda} C_R^*$             // $C$ is Hermitian
8 $(U, \Lambda) \leftarrow \mathrm{eigh}(C)$         // $\Lambda$ are the eigenvalues
9 return $(Q, U, \Lambda)$
```

The core idea behind the sketched SVD is that, given a linear operator $A \in \mathbb{C}^{D_L \times D_R}$ of numerical rank k, any orthogonal matrices $P \in \mathbb{C}^{D_L \times N_O}, Q \in \mathbb{C}^{D_R \times N_O}$ that approximately capture the column and row space of A, respectively, satisfy $A \approx PP^*AQQ^*$ and can be efficiently obtained from $N_O = \mathcal{O}(k)$ random measurements and QR orthogonalization (Halko et al., 2011). Then, the *core matrix* $C := P^*AQ$ is much smaller than A and can be decomposed via classical methods. A further development features an oversampled *inner matrix* $M_I := \Upsilon_I^* A \Omega_I$ for random measurements Υ_I, Ω_I of $N_I > N_O$ columns each (Boutsidis et al., 2016). The SSVD from (Tropp et al., 2019, Sec. 2), gathered in Alg. 1, follows this idea:

$$A \approx P(P^*AQ)Q^* \approx P \underbrace{(\Upsilon_I^* P)^\dagger \Upsilon_I^* A \Omega_I \big[(\Omega_I^* Q)^\dagger\big]^*}_{C = U\Sigma V^* \quad \text{(svd)}} Q^* = (PU)\Sigma(V^* Q^*), \tag{4}$$

This yields an SVD, since PU and QV are orthogonal. It requires $2N_O + N_I$ measurements (lines 1-3 in Alg. 1), followed by thin matrix operations only. The memory cost is dominated by storing P and Q, i.e. $\mathcal{O}(N_O(D_L + D_R))$. Arithmetic is dominated by the QR orthogonalizations needed to obtain P and Q, as well as the two least-squares problems needed to solve the pseudoinverses (see (Tropp et al., 2019)). Crucially, this approximation only requires a single pass over A, yields tight bounds (Tropp et al., 2019, Th. 5.1) and leads to superior performance (Tropp et al., 2019, Sec. 7), due to its numerical stability, oversampled M_I, as well as uncorrelated measurements between P, Q and M_I (Halko et al., 2011, 5.5) (Tropp et al., 2019, 2.8.1).

Leveraging Symmetry: While previous works studied Hermitian extensions (Halko et al., 2011; Clarkson & Woodruff, 2017; Tropp et al., 2017), none of them proposes a single-pass sketched eigendecomposition via oversampled and uncorrelated inner matrix. In this work, we propose SEIGH (Alg. 2), a variant of SSVD that retains core oversampling while leveraging conjugate symmetry to drastically reduce the required measurements and memory. Without loss of generality (Halko et al., 2011), we assume $Q = P$, so only Q needs to be computed via $A\Omega$ outer measurements. Then, C is also Hermitian and can be eigendecomposed:

$$A \approx Q(Q^*AQ)Q^* \approx Q(\Upsilon_I^* Q)^\dagger \underbrace{\Upsilon_I^* A \Omega_I}_{M_I = \bar{U}\bar{\Lambda}\bar{V}^*} \big[(\Omega_I^* Q)^\dagger\big]^* Q^* = Q \underbrace{(\Upsilon_I^* Q)^\dagger \bar{U}\bar{\Lambda}\bar{V}^* \big[(\Omega_I^* Q)^\dagger\big]^*}_{C = C^* = U\Lambda U^* \quad \text{(eigh)}} Q^*$$

Setting $\Upsilon_I = \Omega_I$ would further simplify the structure, saving one pseudoinverse, but there is one caveat: We also wish to recycle part of the $A\Omega_I$ inner measurements to obtain Q, since this is a major bottleneck for the

Hessian. But recycling measurements and setting $\boldsymbol{\Upsilon}_I = \boldsymbol{\Omega}_I$ would lead to correlated measurements between $\boldsymbol{Q}$ and $\boldsymbol{M}_I$ which, as mentioned, has negative impact in the quality of this procedure (Tropp et al., 2019, 2.8.1). We opt to recycle the measurements, while keeping the uncorrelated $\boldsymbol{\Upsilon}_I = \boldsymbol{\Omega}_I$ and the two pseudoinverses. This way, only M_I measurements are needed, memory is roughly halved, and arithmetic is reduced by one QR decomposition and increased by one inner SVD. With SEIGH, we are now able to approximate $\|\boldsymbol{V}\|_F^2$ for large Hessians, leading to the *sketched* overlap approximation.

6 Magnitude Masks and Hessian Eigenspaces Overlap Substantially

We now investigate the similarity between the space spanned by the parameter magnitude masks and the top Hessian eigenspaces over the course of neural network training. We first study a small-scale toy problem (Section 6.1), where all involved quantities can still be computed exactly. This allows us to verify the existence of early crystallization, as well as the quality of our sketched overlap approximation. We observe an overlap consistently above random chance. We then move onto larger problems (up to $D > 11\text{M}$ and $k = 1500$), where we also confirm that the similarity between parameter magnitude masks and top Hessian eigenspaces is orders of magnitude above random chance (Section 6.2) , and that this effect is accentuated with network size.

6.1 Exact Computations on 16×16 MNIST

Table 1: Overview of experimental settings, detailing number of model parameters (D), learning rate (η), batch size (B), steps per epoch (T), test accuracy (acc) at step t, number of train/test samples used to compute H_{train}/H_{test} (N_{train}/N_{test} respectively), and number of SEIGH outer measurements (N_O, see Alg. 2)

Problem	Model	D	η	B	T	acc	N_{train}/N_{test}	N_O
16×16 **MNIST**	tanh-MLP Martens & Grosse (2015)	7030	0.3	500	100	95.78% ($t = 1000$)	500/500	355
CIFAR-10	3C3D-CNN Schneider et al. (2019)	895,210	0.0226	128	312	74.52% ($t = 8000$)	500/500	1000
CIFAR-100	ALL-CNN-C Springenberg et al. (2015)	1,387,108	0.1658	256	156	40.50% ($t = 8000$)	n.a./1000	1000
ImageNet	RESNET-18 He et al. (2016)	11,689,512	0.1	150	8207	17.33% ($t = 8000$)	n.a./5000	1500

For our toy problem, we seek a setup that is overparametrized enough to exhibit mask crystallization, but small enough so that full Hessian eigendecompositions can be computed. This is achieved by the setup from Martens & Grosse (2015), which is able to achieve zero training loss on downsampled MNIST digit classification with only 7030 parameters. We train using SGD as detailed in Table 1, reaching a test accuracy of 95.78% after 10 epochs (Figure 12). To compute the training and test Hessians, we use a fixed set of 500 random samples (50 per class) from each respective data split. At every fifth training step t_i, we compute the sparsity ratio κ (defined in Section 2.2) for the parameters $\boldsymbol{\theta}$ and the Hessian spectrum $\boldsymbol{\Lambda}$, observing that *both subspaces experience early collapse* (Figure 12). Then, for pairs of steps (t_i, t_j), we compute the IoU between successive parameter magnitude masks, as well as the overlap between successive Hessian eigenspaces. We observe that *both subspaces also experience early crystallization* (Figure 13). We reiterate that we merely *inspect* the parameter magnitude masks, but we never *apply* them, i.e., we never prune the network. Once we established our desired scenario, we compute several Grassmannian metrics between the spans of parameter magnitude masks and top Hessian eigenspaces (Figure 14). All non-collapsing metrics report that *both subspaces show a substantial and consistent similarity.* In particular, our sketched overlap approximation closely tracks the exact overlap (Figure 1, left), which is largest shortly after initialization and then decays, but still stabilizes well above random chance.

6.2 Sketched overlap on larger problems (CIFAR-10, CIFAR-100 and ImageNet)

We aim now to verify our discovery of substantial overlap in larger setups. Using the DEEPOBS framework (Schneider et al., 2019) and SGD, we train larger models on CIFAR-10/100 Krizhevsky & Hinton (2009) and on IMAGENET Deng et al. (2009) (see Table 1 for details). At steps $\{2000, 4000, 8000\}$, we gather the parameters and compute the top Hessian eigenspaces using SEIGH, which allows us to compute the sketched

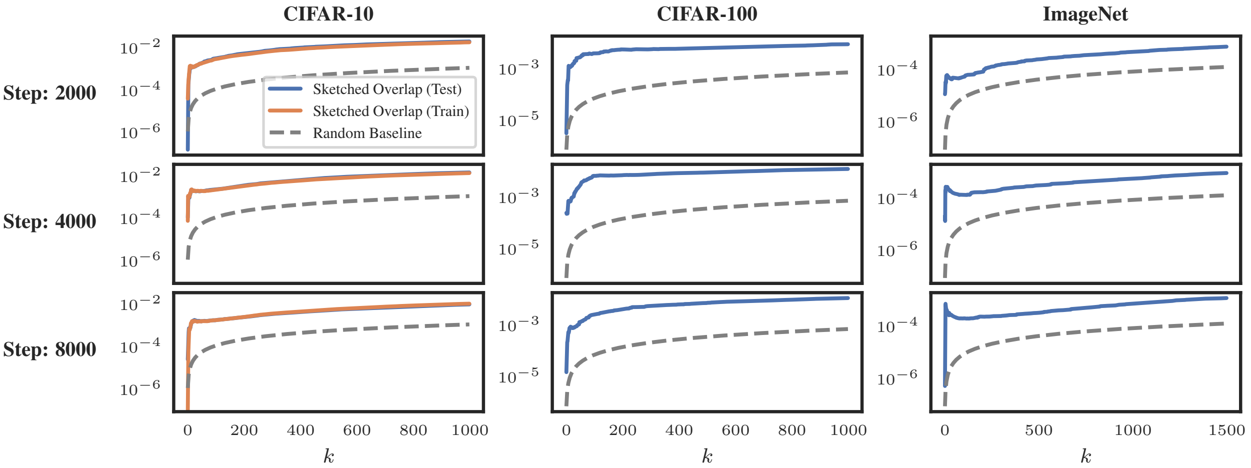

Figure 3: Overlap between parameter magnitude masks and top Hessian eigenspaces as a function of k at three different points during the training process. Shown is the sketched overlap between the mask of the largest k parameters by magnitude and the top-k Hessian eigenspace. Consistently across the training process and for all problems and values of k, we observe an overlap substantially larger than the random chance baseline (- -) introduced in Section 4.2.

overlap for every $k \in \{1, \dots, N_O\}$. To compute the Hessians, we choose N_{train}/N_{test} to have several (balanced) samples per class, and N_O to be substantially larger than the number of classes, which has been shown to be linked to the top eigenspace dimensionality (e.g. Gur-Ari et al., 2018; Papyan, 2020). We elaborate on these choices and report runtimes in Appendix B.2. Note that training CIFAR-100 and ImageNet relies on noisy data augmentations, which lead to nondeterministic behavior in $\boldsymbol{H}_{train}$. For this reason, we compute their overlap for $\boldsymbol{H}_{test}$ only. This choice is supported by the CIFAR-10 results, showing minimal difference in overlap for either $\boldsymbol{H}_{train}$ or $\boldsymbol{H}_{test}$.

We observe that the measured overlap is *significant for all problems, steps and dimensions* (Figure 3 and Figure 4, left), confirming the findings from our 16×16 MNIST toy problem. Furthermore, *this effect is accentuated by model size, surpassing a factor of* 10^3 *times the baseline* for the ResNet-18 on ImageNet (Figure 4, right). The observed sketched overlap also *raises in the very early steps*, perhaps linked to the crystallization covered in Section 2. We also note that, while the precision of sketched decompositions is reported to decay sharply for eigenpairs close to N_O (Tropp et al., 2019, 7.9), this does not affect our observation, since the obtained overlap is *particularly high for the lower regimes of* k, far from N_O and where the method is most reliable.

7 Conclusion

We started with the observation that, at the early stages of neural network training, both parameter magnitude masks and loss Hessian eigenspaces collapse and crystallize. To connect these two phenomena throughout the training process, we proposed a principled methodology that compares these two fundamental deep learning quantities using Grassmannian metrics, and identified overlap as a particularly advantageous one. We further developed SEIGH, a matrix-free sketched eigendecomposition that works for Hessians of over 10^7 parameters, allowing us to approximate overlap. Our experiments reveal an overlap between parameter magnitude masks and top Hessian eigenspaces well above chance level for all observed problems, training steps and dimensionalities, indicating that, in DL, *top Hessian eigenvectors tend to be concentrated around larger parameters*, or equivalently, that *large parameters tend to align with directions of large loss curvature.*

While the obtained overlap may be considered small in absolute terms, the fact that it is orders of magnitude above chance could play a crucial role at large scales, analogously to how mini-batch gradients produce very noisy fluctuations in the short term, but lead to qualitative performance and generalization over larger

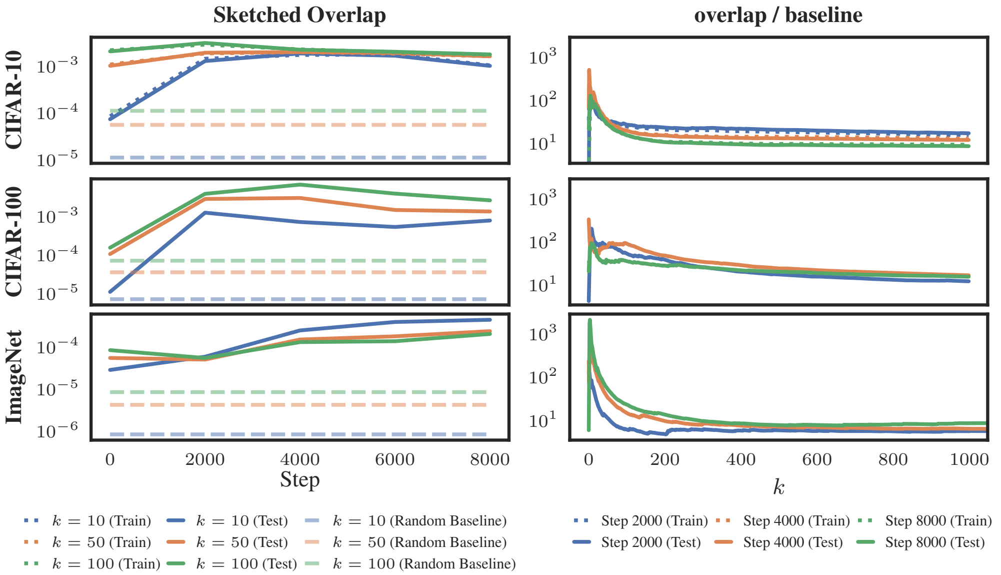

Figure 4: *(Left)* **Overlap between parameter magnitude masks and top Hessian eigenspaces as a function of training step.** Shown is the sketched overlap between masks of k-largest parameters by magnitude and top-k Hessian eigenspaces, for different values of k at different training steps. The observed overlap raises early on, and is consistently and substantially above random chance baseline (- -, introduced in Section 4.2). *(Right)* **Factor by which the measured sketched** overlap **is greater than the random baseline.** For most of the measured training process and choices of k, the observed overlap is at least 10× larger than random chance baseline. This multiple factor over the random baseline is largest for small k and increases with network size, surpassing 10^3.

training spans. The relevance of this connection could also bear potential for downstream tasks where the Hessian plays a prominent role (e.g. by approximating expensive Hessian quantities via cheap parameter inspection).

References

P. A. Absil, Robert Mahony, and Rodolphe Sepulchre. Riemannian Geometry of Grassmann Manifolds with a View on Algorithmic Computation. *Acta Applicandae Mathematicae*, 2004.

Alessandro Achille, Matteo Rovere, and Stefano Soatto. Critical Learning Periods in Deep Networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Thomas Bendokat, Ralf Zimmermann, and Pierre-Antoine Absil. A Grassmann Manifold Handbook: Basic Geometry and Computational Aspects, 2020.

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the State of Neural Network Pruning? In *Proceedings of Machine Learning and Systems (MLSys)*, 2020.

Christos Boutsidis, David P. Woodruff, and Peilin Zhong. Optimal Principal Component Analysis in Distributed and Streaming Models. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, 2016.

Kenneth L. Clarkson and David P. Woodruff. Low-rank psd approximation in input-sparsity time. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2017.

Felix Dangel, Frederik Kunstner, and Philipp Hennig. BackPACK: Packing more into Backprop. In *International Conference on Learning Representations (ICLR)*, 2020.

Felix Dangel, Lukas Tatzel, and Philipp Hennig. ViViT: Curvature Access Through The Generalized Gauss-Newton's Low-Rank Structure. *Transactions on Machine Learning Research (TMLR)*, 2022.

Yann N. Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and Attacking the Saddle Point Problem in High-Dimensional Non-Convex Optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Persi Diaconis and Mehrdad Shahshahani. The Subgroup Algorithm for Generating Uniform Random Variables. *Probability in the Engineering and Informational Sciences*, 1987.

Alan Edelman, Tomás A. Arias, and Steven T. Smith. The Geometry of Algorithms with Orthogonality Constraints. *SIAM Journal on Matrix Analysis and Applications*, 1998.

Mohamed Elsayed and A. Rupam Mahmood. HesScale: Scalable Computation of Hessian Diagonals, 2022.

Jonathan Frankle and Michael Carbin. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear Mode Connectivity and the Lottery Ticket Hypothesis. In *International Conference on Machine Learning (ICML)*, 2020.

Trevor Gale, Erich Elsen, and Sara Hooker. The State of Sparsity in Deep Neural Networks, 2019.

Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An Investigation into Neural Net Optimization via Hessian Eigenvalue Density. In *International Conference on Machine Learning (ICML)*, 2019.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 2013.

Guy Gur-Ari, Daniel A. Roberts, and Ethan Dyer. Gradient Descent Happens in a Tiny Subspace, 2018.

N. Halko, P. G. Martinsson, and J. A. Tropp. Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. *SIAM Review*, 2011.

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 (NeurIPS)*, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks. *Journal of Machine Learning Research*, 2021.

Niall Hurley and Scott Rickard. Comparing Measures of Sparsity. *IEEE Transactions on Information Theory*, 2009.

Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Learnable uncertainty under Laplace approximations. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2021.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report, 2009.

Yann LeCun, John Denker, and Sara Solla. Optimal Brain Damage. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1989.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the Intrinsic Dimension of Objective Landscapes. In *International Conference on Learning Representations (ICLR)*, 2018a.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018b.

James Martens. *Second-order Optimization for Neural Networks*. PhD dissertation, University of Toronto, 2016.

James Martens and Roger Grosse. Optimizing Neural Networks with Kronecker-Factored Approximate Curvature. In *International Conference on Machine Learning (ICML)*, 2015.

Yuri Neretin. On Jordan Angles and the Triangle Inequality in Grassmann Manifolds. *Geometriae Dedicata*, 2001.

Vardan Papyan. The Full Spectrum of Deepnet Hessians at Scale: Dynamics with SGD Training and Sample Size. *arXiv: Learning*, 2018.

Vardan Papyan. Measurements of Three-Level Hierarchical Structure in the Outliers in the Spectrum of Deepnet Hessians. In *International Conference on Machine Learning (ICML)*, 2019.

Vardan Papyan. Traces of Class/Cross-Class Structure Pervade Deep Learning Spectra. *Journal of Machine Learning Research*, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

Barak A. Pearlmutter. Fast Exact Multiplication by the Hessian. *Neural Computation*, 1994.

Li Qiu, Y. X. Zhang, and Chi-Kwong Li. Unitarily Invariant Metrics on the Grassmann Space. *SIAM Journal on Matrix Analysis and Applications*, 2005.

Yousef Saad. On the rates of convergence of the lanczos and the block-lanczos methods. *SIAM Journal on Numerical Analysis*, 1980.

Levent Sagun, Utku Evci, V. Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical Analysis of the Hessian of Over-Parametrized Neural Networks, 2018. ICLR Workshop.

Adepu Ravi Sankar, Yash Khasbage, Rahul Vigneswaran, and Vineeth N Balasubramanian. A Deeper Look at the Hessian Eigenspectrum of Deep Neural Networks and its Applications to Regularization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

Frank Schneider, Lukas Balles, and Philipp Hennig. DeepOBS: A Deep Learning Optimizer Benchmark Suite. In *International Conference on Learning Representations (ICLR)*, 2019.

Jost T. Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. In *ICLR Workshop*, 2015.

Joel A. Tropp, Alp Yurtsever, Madeleine Udell, and Volkan Cevher. Fixed-rank approximation of a positive-semidefinite matrix from streaming data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Joel A. Tropp, Alp Yurtsever, Madeleine Udell, and Volkan Cevher. Streaming low-rank matrix approximation with an application to scientific simulation. *SIAM Journal on Scientific Computing*, 2019.

Edward Witten. Quantum field theory, Grassmannians, and algebraic curves. *Communications in Mathematical Physics*, 1988.

Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W. Mahoney. PyHessian: Neural Networks Through the Lens of the Hessian. In *IEEE International Conference on Big Data (Big Data)*, 2020.

Ke Ye and Lek-Heng Lim. Schubert Varieties and Distances between Subspaces of Different Dimensions. *SIAM Journal on Matrix Analysis and Applications*, 2016.

Haoran You, Chaojian Li, Pengfei Xu, Yonggan Fu, Yue Wang, Xiaohan Chen, Richard G. Baraniuk, Zhangyang Wang, and Yingyan Lin. Drawing Early-Bird Tickets: Toward More Efficient Training of Deep Networks. In *International Conference on Learning Representations (ICLR)*, 2020.

Jiayao Zhang, Guangxu Zhu, Robert W. Heath, and Kaibin Huang. Grassmannian Learning: Embedding Geometry Awareness in Shallow and Deep Learning, 2018.

A Supplementary Material for Grassmannian Metrics

A.1 Perturbing the Parameters to Measure their Importance

In order to prove that a certain subset of parameters is “sensitive” whithout using the theoretical framework we propose, one may consider a perturbation study where large parameters are perturbed, and the resulting loss variation is measured. More variation would then relate to more sensitivity. Here we show that, in general, such an experiment does not yield an informative quantity towards this goal (unlike our proposed Grassmannians), due to the non-PSD nature of $\boldsymbol{H}$, (i.e. negative and positive directions of curvature may cancel out for arbitrary sets of parameters) and the lack of details about the loss landscape regularity.

At any parameter vector $\boldsymbol{\theta}\in\mathbb{R}^D$, and under additive perturbations $\boldsymbol{\delta}\in\mathbb{R}^D$, the loss can be expressed via its Taylor expansion (for gradient $\boldsymbol{g}$, Hessian $\boldsymbol{H}$ and Lagrange remainder R_3):

$$L(\boldsymbol{\theta}+\boldsymbol{\delta}) = L(\boldsymbol{\theta}) + \boldsymbol{g}^\top\boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^\top\boldsymbol{H}\boldsymbol{\delta} + R_3(\boldsymbol{\delta})$$

Then, given a parameter mask $\boldsymbol{m}\in\mathbb{B}^D$, the proposed perturbation study would consist in drawing random perturbations $\boldsymbol{\delta}$ from any i.i.d. distribution $\mathcal{N}$ with zero mean μ and unit variance, and then mask said perturbations via $\boldsymbol{\delta_m} = (\boldsymbol{m}\odot\boldsymbol{\delta})$ and add them to $\boldsymbol{\theta}$ in order to estimate the expectation:

$$\mathbb{E}_{\mathcal{N}}\left[L(\boldsymbol{\theta}+\boldsymbol{\delta_m})\right] = L(\boldsymbol{\theta}) + \underbrace{\boldsymbol{g}^\top\mathbb{E}_{\mathcal{N}}\left[\boldsymbol{\delta_m}\right] + \frac{1}{2}\langle\boldsymbol{H}, \mathbb{E}_{\mathcal{N}}\left[\boldsymbol{\delta_m}\boldsymbol{\delta_m}^\top\right]\rangle + \mathbb{E}_{\mathcal{N}}\left[R_3(\boldsymbol{\delta_m})\right]}_{\varepsilon}$$

This way, we are perturbing only the selected parameters in $\boldsymbol{m}$, and measuring their impact on the (known) loss $L(\boldsymbol{\theta})$: If $|\varepsilon|$ is larger, this can be interpreted as the subset ξ being more “sensitive".

Since we assumed $\boldsymbol{\mu_m} \coloneqq \mathbb{E}_{\mathcal{N}}\left[\boldsymbol{\delta_m}\right] = 0$ (otherwise reparametrize $(\boldsymbol{\theta}+\boldsymbol{\delta_m}) \eqqcolon (\boldsymbol{\theta}+\boldsymbol{\mu_m}) + (\boldsymbol{\delta_m}-\boldsymbol{\mu_m})$ such that the reparametrized mean of the perturbations will be 0), and we also assumed a (masked) unit covariance $\mathrm{Cov}\left[\boldsymbol{\delta_m}\right] = \boldsymbol{I_m}$, we have:

$$\mathrm{Cov}\left[\boldsymbol{\delta_m}\right] \coloneqq \mathbb{E}_{\mathcal{N}}\left[(\boldsymbol{\delta_m}-\boldsymbol{\mu_m})(\boldsymbol{\delta_m}-\boldsymbol{\mu_m})^\top\right] = \mathbb{E}_{\mathcal{N}}\left[\boldsymbol{\delta_m}\boldsymbol{\delta_m}^\top\right] = \boldsymbol{I_m}$$

Then, the expectation of the perturbed loss simplifies:

$$\begin{aligned}\mathbb{E}_{\mathcal{N}}\left[L(\boldsymbol{\theta}+\boldsymbol{\delta_m})\right] &= L(\boldsymbol{\theta}) + \underbrace{\boldsymbol{g}^\top\mathbb{E}_{\mathcal{N}}\left[\boldsymbol{\delta_m}\right]}_{0} + \frac{1}{2}\langle\boldsymbol{H}, \underbrace{\mathbb{E}_{\mathcal{N}}\left[\boldsymbol{\delta_m}\boldsymbol{\delta_m}^\top\right]}_{\boldsymbol{I_m}}\rangle + \mathbb{E}_{\mathcal{N}}\left[R_3(\boldsymbol{\delta_m})\right] \\ &= L(\boldsymbol{\theta}) + \underbrace{\frac{1}{2}\,\mathrm{Tr}_{\boldsymbol{m}}(\boldsymbol{H}) + \mathbb{E}_{\mathcal{N}}\left[R_3(\boldsymbol{\delta_m})\right]}_{\varepsilon}\end{aligned}$$

We see now how, in general, this technique cannot be relied upon: since $\boldsymbol{H}$ is non-PSD in general, any of its subtraces could cancel R_3, leading to $\varepsilon = 0$. Therefore, this procedure is not guaranteed to be informative in measuring the impact of parameter subsets on the loss curvature. In contrast, our proposed method is guaranteed to yield larger similarities whenever the selected parameters in $\boldsymbol{m}$ tend to align with directions of larger loss curvature, as shown in the paper.

Alternatively, one could consider to sample perturbations for parameters *one by one*, i.e. by setting $\boldsymbol{m}$ to equal 1 at exactly m_i, and 0 otherwise. Then, the expectation of the perturbed loss becomes:

$$\mathbb{E}_{\mathcal{N}}\left[L(\boldsymbol{\theta} + \boldsymbol{\delta_m})\right] = L(\boldsymbol{\theta}) + \underbrace{\frac{1}{2}H_{ii} + \mathbb{E}_{\mathcal{N}}\left[R_3(\boldsymbol{\delta_m})\right]}_{\varepsilon_i}$$

The idea here is that the aggregated perturbation $\varepsilon' = \sum_i |\varepsilon_i|$ is now more likely to be informative, since positive and negative diagonal entries don't cancel anymore. This still entails a problematic trade-off that favours our Grassmannian characterization, for the following reasons: (1) In DL, the loss landscape may be far from convex or regular (e.g. Li et al., 2018b, Fig. 1), and to interpret ε' meaningfully we would still require some characterization of R_3 similar in spirit to the theoretically founded Grasmannian that we use. (2) This method requires to estimate one expectation per scalar parameter. For a set of k parameters and n samples per expectation, this means performing $\mathcal{O}(n \cdot k)$ forward propagations over a given dataset. In contrast, our method is theoretically grounded and our proposed sketched method requires just $\mathcal{O}(k)$ HVPs (see Section 5 and (Halko et al., 2011, 4.2) for more discussion).

A.2 Main Properties of Grassmannian Metrics

A desirable property of Grassmann manifolds is the availability of closed-form expressions for their *geodesics* (i.e. the shortest paths between any two elements $\mathfrak{g}_i, \mathfrak{g}_j \in \mathcal{G}$) and metric functions. We can thus measure distances between subspaces in an interpretable and computationally amenable manner: geodesics from $\mathfrak{g}_i$ to $\mathfrak{g}_j$ follow circular trajectories, and therefore their *distance* can be interpreted as the "amount of rotation" needed to go from one space to another. Such rotations can be succinctly expressed in terms of *principal angles* $\boldsymbol{\sigma}_{i\to j} \in [0, \frac{\pi}{2}]^k$, and they can be efficiently obtained from $\mathbb{O}^{D\times k}$ matrices via their Singular Value Decomposition (SVD) (e.g. Edelman et al., 1998, s. 4.3):

$$\boldsymbol{Q}_i^\top \boldsymbol{Q}_j =: \boldsymbol{L}_{i\to j}\, \mathrm{diag}\left(\cos(\boldsymbol{\sigma}_{i\to j})\right) \boldsymbol{R}_{i\to j}^\top, \quad \boldsymbol{L}, \boldsymbol{R} \text{ orthogonal.} \tag{5}$$

Since $\boldsymbol{Q}_i$,$\boldsymbol{Q}_j$ have orthonormal columns, $\cos(\boldsymbol{\sigma}_{i\to j}) \in [0{,}1]^k$ (e.g. Neretin, 2001), and more similar spans will yield larger singular values, which translate to smaller rotations. Importantly, singular values are invariant under similarity:

$$\begin{align} \mathrm{diag}\left(\cos(\boldsymbol{\sigma}_{i\to j})\right) &= \boldsymbol{L}_{i\to j}^\top \boldsymbol{Q}_i^\top \boldsymbol{Q}_j \boldsymbol{R}_{i\to j} \tag{6}\\ &= \boldsymbol{L}'^{\top}_{i\to j}(\boldsymbol{Z}_i^\top \boldsymbol{Q}_i^\top)(\boldsymbol{Q}_j \boldsymbol{Z}_j)\boldsymbol{R}'_{i\to j}. \tag{7} \end{align}$$

In other words, they are invariant under the action of any orthogonal $\boldsymbol{Z}$, which means that $\boldsymbol{\sigma}$ does not change if we replace an input matrix with any other matrix from the same equivalence class (see definition for $\mathcal{S}_i^{\mathbb{O}}$ in Section 4.1). The family of functions that satisfy this invariance, plus the axioms of metric spaces, form the family of *Grassmannian metrics* (e.g. Qiu et al., 2005), each capturing a different notion of distance between subspaces (largest principal angle, sum of principal angles…). See Appendix A.3 for more details on specific metrics.

A.3 Popular Grassmannian Metrics

In the following, we highlight popular metrics from the literature (e.g. Edelman et al., 1998). We abbreviate $\mathrm{dist}_*(\mathfrak{g}_i, \mathfrak{g}_j) = f(\boldsymbol{\sigma}_{i\to j})$ as $\mathrm{dist}_* = f(\boldsymbol{\sigma})$, where $*$ here parametrizes any unitarily invariant norm:

a) **Geodesic distance:** This is the *arc length* of the geodesic between the respective spaces in $\mathcal{G}$, defined as $\mathrm{dist}_g = \|\boldsymbol{\sigma}\|_2 \in [0,\ \frac{\pi}{2}\sqrt{k}]$.

b) **Chordal norm:** $\mathrm{dist}_{c,*}$, obtained by minimizing $\|\boldsymbol{Q}_i\boldsymbol{Z}_i - \boldsymbol{Q}_j\boldsymbol{Z}_j\|_*$ over orthogonal matrices $(\boldsymbol{Z}_1, \boldsymbol{Z}_2)$ (for that reason it is also called *Hausdorff distance*). The ℓ_2 and ℓ_F norms admit a closed-form solution in terms of principal angles: $\mathrm{dist}_{c,2} = \|2\sin\left(\frac{1}{2}\boldsymbol{\sigma}\right)\|_\infty \in [0, \sqrt{2}]$ and $\mathrm{dist}_{c,\mathrm{F}} = \|2\sin\left(\frac{1}{2}\boldsymbol{\sigma}\right)\|_2 \in [0, \sqrt{2k}]$.

c) Projection norm: Also called the *gap metric*, it uses the unique orthogonal projector representation of a given subspace, i.e. $\boldsymbol{\Psi}_i = \boldsymbol{Q}_i \boldsymbol{Q}_i^\top$, as follows: $\text{dist}_{p,*} = \|\boldsymbol{\Psi}_i - \boldsymbol{\Psi}_j\|_*$. Here, we also have closed-form expressions for the ℓ_2 and ℓ_F norms: $\text{dist}_{p,2} = \|\sin(\boldsymbol{\sigma})\|_\infty \in [0,1]$ and $\text{dist}_{p,\text{F}} = \|\sin(\boldsymbol{\sigma})\|_2 \in [0, \sqrt{k}]$.

d) Fubini-Study: This quantity is a measure of the *acute angle* between both spaces, generalized to higher dimensions: $\text{dist}_a = \arccos\left(|\det(\boldsymbol{Q}_i^\top \boldsymbol{Q}_j)|\right) = \arccos\left(\prod_i \cos(\sigma_i)\right) \in [0, \frac{\pi}{2}]$.

e) Overlap: The $\text{overlap} = \frac{1}{k}\|\boldsymbol{\Psi}_i \boldsymbol{Q}_j\|_F^2 \in [0,1]$ quantity was used in Gur-Ari et al. (2018) to measure subspace *similarity*. It is not a metric *per se*, since it is highest for equivalent subspaces and decreases with their distance, but it is a bijection of $\text{dist}_{p,\text{F}}$, as follows: $\frac{1}{k}\|\boldsymbol{\Psi}_i \boldsymbol{Q}_j\|_F^2 = \frac{1}{k}\|\boldsymbol{Q}_i^\top \boldsymbol{Q}_j\|_F^2 = \frac{1}{k}\|\cos(\boldsymbol{\sigma})\|_F^2 = 1 - \|\cos(\boldsymbol{\sigma})\|_F^2 = 1 - \frac{1}{k}\,\text{dist}_{p,F}^2$.

While the above metrics apply to any pair of matrices from $\mathbb{O}^{D\times k}$, there are also relevant metrics specific to $\mathbb{M}^{D\times k}$ (boolean subset of $\mathbb{O}^{D\times k}$ defined in Section 3), that can be characterized in a similar manner. Consider an arbitrary pair of k-sparse masks $(\boldsymbol{m}_i, \boldsymbol{m}_j)$, and their corresponding orthogonal projectors $\boldsymbol{\Phi} := \text{diag}(\boldsymbol{m})$. Then we have:

i) IoU: Typically used as an evaluation metric, it is defined as the relative number of entries present in *both* masks, i.e. $\text{IoU} := \frac{\boldsymbol{m}_i \cap \boldsymbol{m}_j}{\boldsymbol{m}_i \cup \boldsymbol{m}_j} \in [0,1]$. Then we have $\boldsymbol{m}_i \cap \boldsymbol{m}_j = \|\boldsymbol{\Phi}_i \boldsymbol{\Phi}_j\|_F^2 = k \cdot \text{overlap}$, and if both masks are k-sparse, we also have $\boldsymbol{m}_i \cup \boldsymbol{m}_j = 2k - (\boldsymbol{m}_i \cap \boldsymbol{m}_j) = k(2 - overlap)$, yielding the bijection $\text{overlap} = \frac{2\,\text{IoU}}{1+\text{IoU}}$.

ii) Hamming distance: This quantity, defined as the minimum number of bit-flips needed to pass from one mask to another, was used in You et al. (2020) to measure distances between pruning masks. It is in fact a Grassmannian metric: $\text{dist}_\text{H} := \|\boldsymbol{m}_i - \boldsymbol{m}_j\|_2^2 = \|\Phi_i - \Phi_j\|_F^2 = \text{dist}_{p,\text{F}}^2 \in [0, k]$, which means that the bijection $\text{overlap} = 1 - \frac{1}{k}\,\text{dist}_\text{H}$ also holds.

A.4 Synthetic Experiment on Grassmannian Metrics

Here we provide details about the synthetic experiment discussed in Section 4.2. We compute the reviewed Grassmannian metrics **a)** to **e)** from Appendix A.3 between randomly drawn matrices from $\mathbb{O}^{D\times k}$ and masks from $\mathbb{M}^{D\times k}$. We use the subgroup algorithm (Diaconis & Shahshahani, 1987) to sample matrices uniformly from $\mathbb{O}^{D\times k}$ and column permutations of $\boldsymbol{I}_{D,k}$ to sample uniformly from $\mathbb{M}^{D\times k}$. To determine if the metrics behave differently for masks than for general orthogonal matrices, we inspect three different *modalities*: pairs of matrices ($\mathbb{O}$-to-$\mathbb{O}$, Figures 5 and 6), masks ($\mathbb{M}$-to-$\mathbb{M}$, Figures 7 and 8), and matrix-mask pairs ($\mathbb{O}$-to-$\mathbb{M}$, Figures 9 and 10). We normalize all metrics, denoted by $\text{dist}_{\overline{*}}$, to be in $[0,1]$, with 1 indicating highest similarity (i.e. smallest distance). The overall procedure is gathered in Alg. 3.

We want to inspect how the distribution of the metrics changes as a function of sparsity k and sparsity ratio $\rho = \frac{k}{D}$. First, given fixed values of ρ, we study the distribution of all Grassmannian metrics as a function of D (Figures 5, 7 and 9). Additionally, given fixed values of D we study how the Grassmannian metrics change as a function of ρ (Figures 6, 8 and 10). We used the following values:

- Number of random (matrix or mask) samples: $T = 50$
- For Figures 5, 7 and 9 we investigate four different (fixed) ratios $\rho \in \{0.4, 0.2, 0.05, 0.01\}$ at several increasing dimensions $d \in \{16, 32, 64, 128, 256, 512, 1024, 2048\}$.
- For Figures 6, 8 and 10 we investigated four (fixed) dimensions $d \in \{128, 256, 512, 1024\}$ at several increasing ratios $\rho \in \{0.005, 0.01, 0.05, 0.1, 0.33, 0.66, 0.9, 0.95, 0.99, 1\}$.

Note how, in the second case, we concentrate the sparsity ratios around the extremes. This is to better capture the behavior of collapsing metrics, as discussed in Section 4.2.

Algorithm 3: Synthetic experiment on Grassmannian metrics (see Section 4.2 for details).

```
Input: $\{D_1, D_2, ...\}$                                   // Matrix height ($D_i \in \mathbb{N}$)
Input: $\{r_1, r_2, ...\}$                                   // Width-to-height ratio ($r_i \in [0, 1]$)
Input: $\{(\mathbb{O}, \mathbb{O}), (\mathbb{O}, \mathbb{M}), (\mathbb{M}, \mathbb{M})\}$                  // Modality
Input: $\{\mathrm{dist}_{\overline{g}}, \mathrm{dist}_{\overline{c,2}}, \mathrm{dist}_{\overline{c,F}}, \mathrm{dist}_{\overline{p,2}}, \mathrm{dist}_{\overline{p,F}}, \mathrm{dist}_{\overline{a}}, \mathrm{overlap}\}$   // Metric (normalized)
Input: $T$                                                   // Number of random samples
$\mathcal{R} \leftarrow \varnothing$                                         // Result (a dictionary)
for $d \in \{D_1, D_2, ...\}$ do
    for $r \in \{r_1, r_2, ...\}$ do
        $k \leftarrow \max(1, \mathrm{round}(r \cdot d))$
        for $\mathrm{dist} \in \{\mathrm{dist}_{\overline{g}}, \mathrm{dist}_{\overline{c,2}}, \mathrm{dist}_{\overline{c,F}}, \mathrm{dist}_{\overline{p,2}}, \mathrm{dist}_{\overline{p,F}}, \mathrm{dist}_{\overline{a}}, \mathrm{overlap}\}$ do
            for $(\mathcal{M}_1, \mathcal{M}_2) \in \{(\mathbb{O}, \mathbb{O}), (\mathbb{O}, \mathbb{M}), (\mathbb{M}, \mathbb{M})\}$ do
                $\mathcal{H} \leftarrow \varnothing$                             // Collection of samples
                for $\{1, \ldots, T\}$ do
                    $(\boldsymbol{Q}_1, \boldsymbol{Q}_2) \overset{\text{unif.}}{\sim} (\mathcal{M}_1^{d \times k}, \mathcal{M}_2^{d \times k})$ $\mathcal{H} \leftarrow \mathcal{H} \cup \mathrm{dist}\left(\mathrm{span}(\boldsymbol{Q}_1), \mathrm{span}(\boldsymbol{Q}_2)\right)$
                end
                $\mathcal{R}_{[d, r, \mathrm{dist}, \mathcal{M}_1, \mathcal{M}_2]} \leftarrow \mathcal{H}$                 // Gather samples into result
            end
        end
    end
end
return $\mathcal{R}$
```

A.5 Expectation of overlap for Uniformly Random Matrices

In Section 4.2 we empirically observed that the expectation under uniformly distributed random matrices becomes *predictable* for all reviewed metrics, converging to a baseline value. Here we show that, for overlap, such expectation equals exactly $\rho = \frac{k}{D}$. This lemma depends on a standard calculation that was communicated to us by [redacted for anonymity].

Lemma A.1
Let $\boldsymbol{Q}_1, \boldsymbol{Q}_2$ be random matrices drawn uniformly from the Stiefel manifold $\mathbb{O}^{D \times k} := \{\boldsymbol{Q} : \boldsymbol{Q} \in \mathbb{R}^{D \times k},\ \boldsymbol{Q}^\top \boldsymbol{Q} = \boldsymbol{I}_k\}$. Then,

$$\mathbb{E}\left[\mathrm{overlap}(\mathrm{span}(Q_1), \mathrm{span}(Q_2))\right] = \frac{k}{D} \tag{8}$$

Proof. We start by rewriting the definition of overlap, presented in Section 4.1, in terms of the trace of orthogonal projectors $\boldsymbol{\Psi} = \boldsymbol{Q}\boldsymbol{Q}^\top \in \mathbb{R}^{D \times D}$. We make use of the idempotence of $\boldsymbol{\Psi}$ and the unitary invariance of the Frobenius norm:

$$\begin{aligned} \mathrm{overlap}(\mathrm{span}(\boldsymbol{Q}_1), \mathrm{span}(\boldsymbol{Q}_2)) &= \frac{1}{k}\|\boldsymbol{Q}_1^\top \boldsymbol{Q}_2\|_F^2 = \frac{1}{k}\|\boldsymbol{\Psi}_1 \boldsymbol{\Psi}_2\|_F^2 = \frac{1}{k}\,\mathrm{Tr}(\boldsymbol{\Psi}_1 \boldsymbol{\Psi}_2^2 \boldsymbol{\Psi}_1) && (9) \\ &= \frac{1}{k}\,\mathrm{Tr}(\boldsymbol{\Psi}_1 \boldsymbol{\Psi}_2 \boldsymbol{\Psi}_1) && (10) \end{aligned}$$

We further observe that, if $\boldsymbol{Q}$ is drawn uniformly from $\mathbb{O}^{D \times k}$, the marginal distribution of every column $\boldsymbol{q}$ is the uniform distribution over the Euclidean unit sphere, hence it is isotropic:

$$\mathbb{E}\left[\boldsymbol{q}\boldsymbol{q}^\top\right] = \frac{1}{D}\boldsymbol{I}_D \tag{11}$$

Where I_D is the rank-D identity matrix. Then, leveraging linearity of expectation, the orthogonal projector $\boldsymbol{\Psi} = \boldsymbol{Q}\boldsymbol{Q}^\top$ can be expressed as follows:

$$\mathbb{E}\left[\boldsymbol{\Psi}\right] = \sum_{i=1}^{k} \mathbb{E}\left[\boldsymbol{q}_i \boldsymbol{q}_i^\top\right] = \frac{k}{D}\boldsymbol{I}_D \tag{12}$$

Now, given two independent realizations $\boldsymbol{Q}_1, \boldsymbol{Q}_2$, we form the associated orthogonal projectors $\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2$. Write $\mathbb{E}_1, \mathbb{E}_2$ for the expectations of the respective distributions of $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$. Then, leveraging independence of $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$, idempotence of $\boldsymbol{\Psi}$, linearity of expectation, and $\mathrm{Tr}(\boldsymbol{\Psi}) = k$, we have:

$$\mathbb{E}\left[\mathrm{Tr}(\boldsymbol{\Psi}_1\boldsymbol{\Psi}_2\boldsymbol{\Psi}_1)\right] = \mathbb{E}_1\left[\mathrm{Tr}(\boldsymbol{\Psi}_1\mathbb{E}_2\left[\boldsymbol{\Psi}_2\right]\boldsymbol{\Psi}_1)\right] = \frac{k}{D}\mathbb{E}_1\left[\mathrm{Tr}(\boldsymbol{\Psi}_1\boldsymbol{I}_D\boldsymbol{\Psi}_1)\right] = \frac{k}{D}\,\mathrm{Tr}(\boldsymbol{\Psi}_1) = \frac{k^2}{D} \tag{13}$$

Replacing in the definition of overlap concludes the proof. □

Table 2: Empirical baselines for Grassmannian metrics. Shown are the measured expectations (averaged over 50 samples at $D = 2048$) of different Grassmannian metrics between two uniformly random matrices in $\mathbb{O}$ for different values of $\rho = \frac{k}{D}$. Note how overlap converges to exactly ρ (see Appendix A.5).

Metric	ρ				
	0.005	0.01	0.05	0.2	0.4
$\mathrm{dist}_{\overline{g}}$	0.03711	0.05191	0.11909	0.24534	0.36536
$\mathrm{dist}_{\overline{c,2}}$	0.00197	0.00161	0.00048	0.00046	0.00026
$\mathrm{dist}_{\overline{c,\mathrm{F}}}$	0.02983	0.04202	0.09984	0.21754	0.33783
$\mathrm{dist}_{\overline{p,2}}$	10^{-5}	10^{-5}	0	0	0
$\mathrm{dist}_{\overline{p,\mathrm{F}}}$	0.00248	0.00479	0.02513	0.1057	0.22527
$\mathrm{dist}_{\overline{a}}$	0	0	0	0	0
overlap	0.00495	0.00955	0.04962	0.20022	0.39980

A.6 Extended Discussion on Grassmannian Metrics

We extend here the main insights presented in Section 4.2:

Mask-vs-mask metrics have larger variance, and in some cases lower expectations: All distributions for the mask-vs-mask modality have higher variance compared to the other modalities (see Appendix A.4). Furthermore, the distributions for the $\mathrm{dist}_{\overline{g}}$ and $\mathrm{dist}_{\overline{c,\mathrm{F}}}$ metrics seem to follow lower trajectories for the mask-vs-mask modality, compared to the other modalities. This is not the case for $\mathrm{dist}_{\overline{p,\mathrm{F}}}$ and overlap, whose expectation does not seem to be affected by the modality.

Extremal values of ρ lead to saturation, except for overlap: Most metrics exhibit a nonlinear behavior near the extremes (Figure 6). This is particularly so for collapsing metrics, but saturation can also be observed in the non-collapsing ones. The only exception is overlap, whose expectation is linear as shown in Appendix A.5.

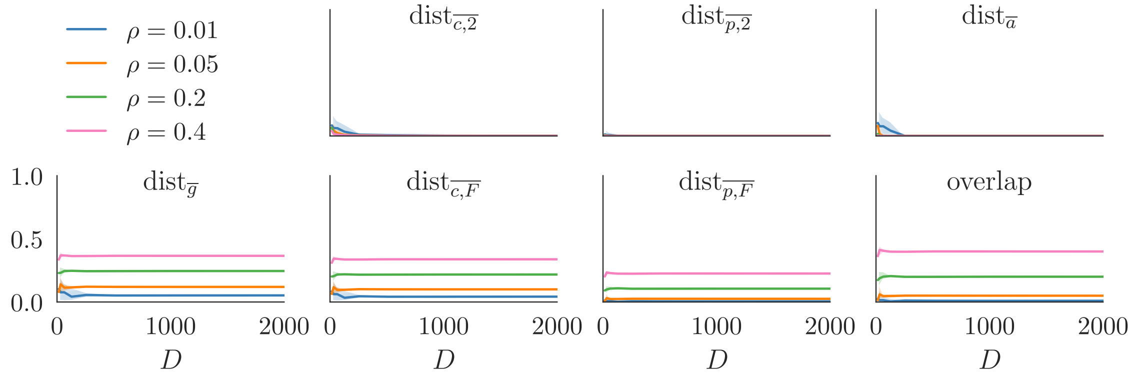

Figure 5: Grassmannian metrics for random pairs of matrices $(\boldsymbol{Q}_1, \boldsymbol{Q}_2) \overset{\text{unif.}}{\sim} (\mathbb{O}^{D\times k}, \mathbb{O}^{D\times k})$ as a function of D. Each subplot shows a different metric, and each color corresponds to a different ratio $r = \frac{k}{D}$. For each (D, k), we sample 50 random pairs and report the median of the resulting distribution (line plot) as well as the 5-95 percentiles (shaded regions).

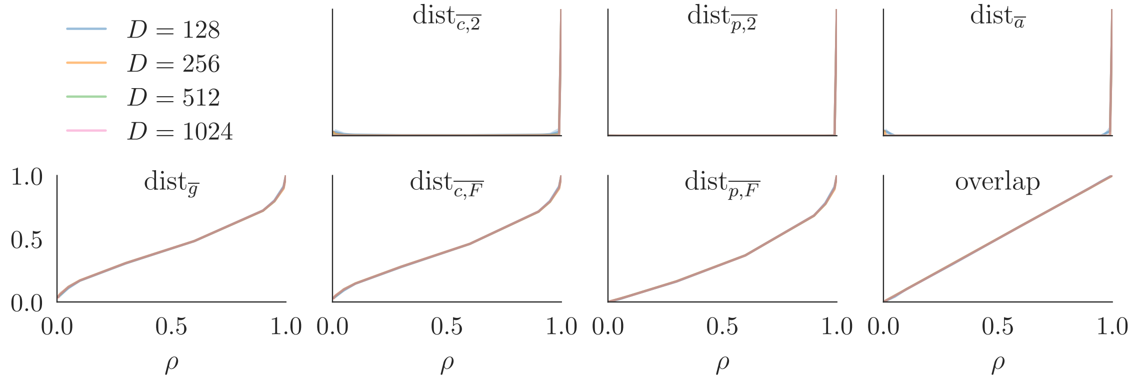

Figure 6: Grassmannian metrics for random pairs of matrices $(\boldsymbol{Q}_1, \boldsymbol{Q}_2) \overset{\text{unif.}}{\sim} (\mathbb{O}^{D\times k}, \mathbb{O}^{D\times k})$ as a function of $\rho = \frac{k}{D}$. Each subplot shows a different metric, and each color corresponds to a different dimension D. For each (D, k), we sample 50 random pairs and report the median of the resulting distribution (line plot) as well as the 5-95 percentiles (shaded regions).

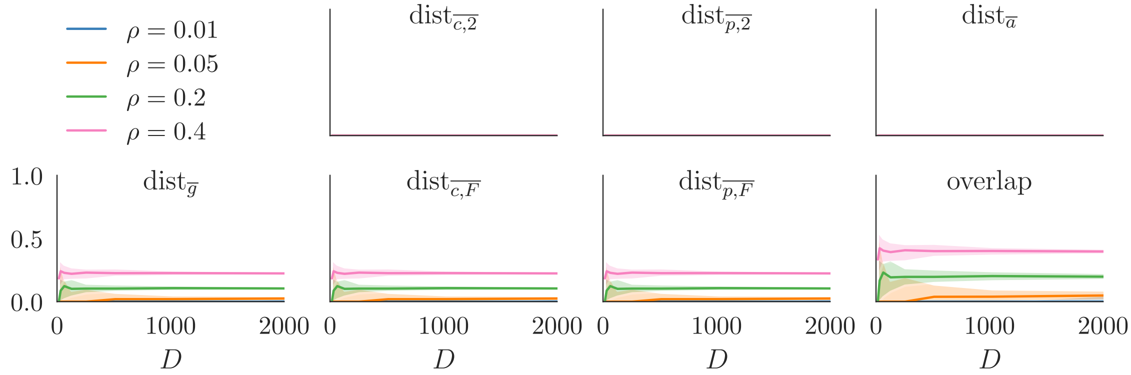

Figure 7: Grassmannian metrics for random pairs of masks $(\boldsymbol{Q}_1, \boldsymbol{Q}_2) \overset{\text{unif.}}{\sim} (\mathbb{M}^{D\times k}, \mathbb{M}^{D\times k})$ as a function of D and for fixed ρ. Each subplot shows a different Grassmannian metric with the different lines indicating four different ratios ρ. For each value of D, we report the median metric over 50 random pairs with the shaded regions showing the 5-95 percentiles.

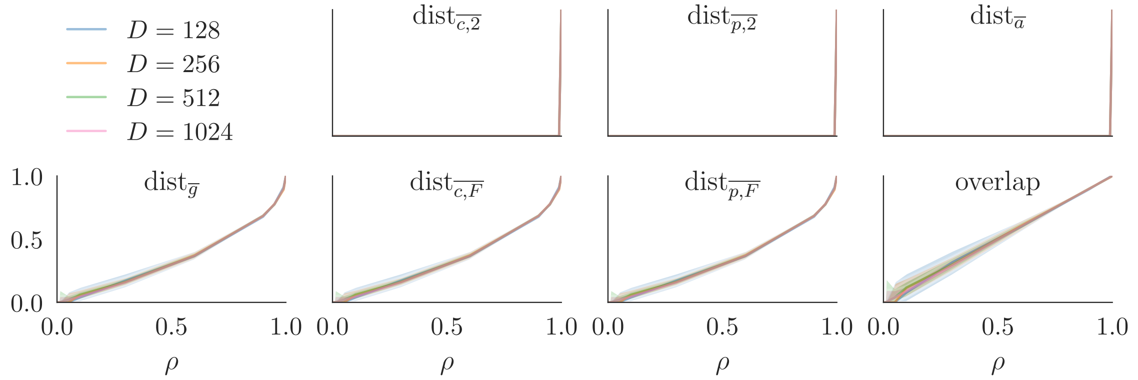

Figure 8: Grassmannian metrics for random pairs of masks $(\boldsymbol{Q}_1, \boldsymbol{Q}_2) \overset{\text{unif.}}{\sim} (\mathbb{M}^{D\times k}, \mathbb{M}^{D\times k})$ as a function of ρ and for fixed D. Each subplot shows a different Grassmannian metric with the different lines indicating four different dimensions D. For each value of ρ, we report the median metric over 50 random pairs with the shaded regions showing the 5-95 percentiles.

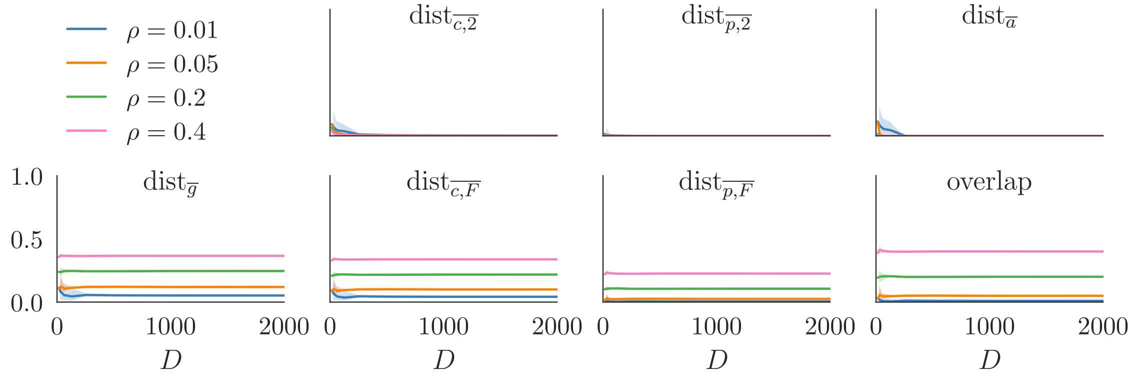

Figure 9: Grassmannian metrics for random pairs of matrices and masks $(\boldsymbol{Q}_1, \boldsymbol{Q}_2) \overset{\text{unif.}}{\sim} (\mathbb{O}^{D\times k}, \mathbb{M}^{D\times k})$ as a function of D and for fixed ρ. Each subplot shows a different Grassmannian metric with the different lines indicating four different ratios ρ. For each value of D, we report the median metric over 50 random pairs with the shaded regions showing the 5-95 percentiles.

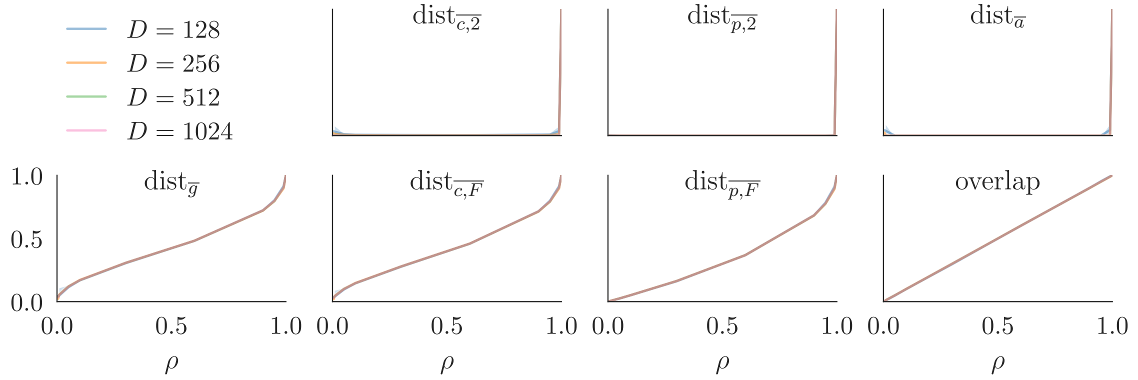

Figure 10: Grassmannian metrics for random pairs of matrices and masks $(\boldsymbol{Q}_1, \boldsymbol{Q}_2) \overset{\text{unif.}}{\sim} (\mathbb{O}^{D\times k}, \mathbb{M}^{D\times k})$ as a function of ρ and for fixed D. Each subplot shows a different Grassmannian metric with the different lines indicating four different dimensions D. For each value of ρ, we report the median metric over 50 random pairs with the shaded regions showing the 5-95 percentiles.

A.7 Analytical overlap Baseline vs. Sampling Random DL Masks

In Section 4.2 we justify the use of $\rho := {}^{k}\!/_{D}$ as an analytical, “chance-level” baseline for overlap, that we use in Section 6 to observe that the overlap between spaces spanned by magnitude prunning masks and top Hessian eigenspaces in DL is substantially large (see e.g. Figures 3 and 4).

In this context, one natural question one may ask is: *Do random DL masks also lead to higher-than-baseline overlaps?* To answer this question, we trained the 16×16 MNIST setup following settings described in Table 1, and for steps $t \in \{0, 100, 1000\}$, we computed the overlap between the top-k Hessian eigenspace and 10 randomly chosen masks of k nonzeros. Results are shown in Figure 11: We observe that, indeed, *random DL masks tend to behave like our analytical baseline*, which supports the use of the analytical baseline in our experiments.

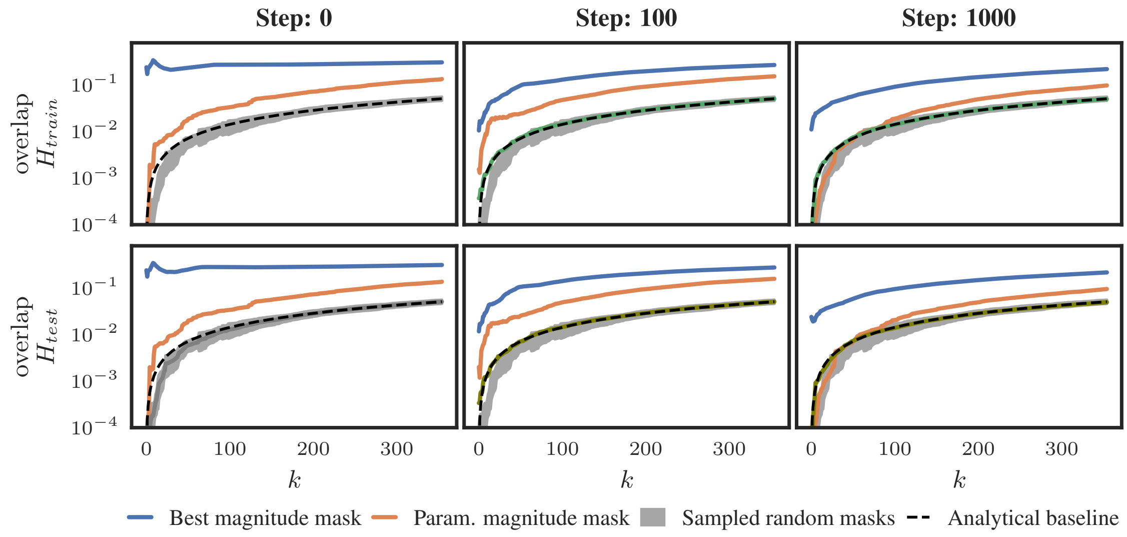

Figure 11: overlap **measurements for different masking criteria against $\boldsymbol{H}_{train}$ (top row) and $\boldsymbol{H}_{test}$ (bottom row).** *Best magnitude mask* represents the overlap obtained if we pick the k parameters with largest overlap with the top-k eigenspace. *Param. magnitude mask* corresponds to the k-largest parameters by magnitude. The *random mask* grey area covers ± one standard deviation for the overlap between $\boldsymbol{H}$ and 10 randomly sampled masks

.

Another natural follow-up question is: *How special are the parameter magnitude masks? Would some other non-random mask also achieve a high overlap?* To tackle this question, we also plot in Figure 11 the overlap for the top-k parameter magnitude masks, and the maximal overlap achievable by the best possible k-mask, serving as upper bound. We observe that, *while the magnitude pruning mask is substantially larger than baseline, it is still far from optimal.*

Still, there is a limited number of masks that can yield high overlap: Due to the orthogonality of the Hessian eigenbasis, each column of $\tilde{\boldsymbol{U}}^{(k)}$ must have an ℓ_2 norm of 1. This means that if a given set of parameters has an overlap of α, the overlap for any non-intersecting set of parameters is at most $1 - \alpha$. Intuitively, the overlap is a limited resource, and masks with an overlap significantly above chance-level are therefore bound to be scarce.

B Supplementary Material for the Experiments in Section 6

B.1 Results for 16×16 MNIST

This section includes supplementary material as part of the exhaustive experiments done for the MNIST toy problem. Figure 12 and Figure 13 verify the existence of early collapse and stabilization in both Hessian top space and parameter masks. They also show that SGD was able to successfully train the model under this setting. Figure 14 corroborates that the behavior of the Grassmannian metrics analyzed in Section 4 is also present for DL problems.

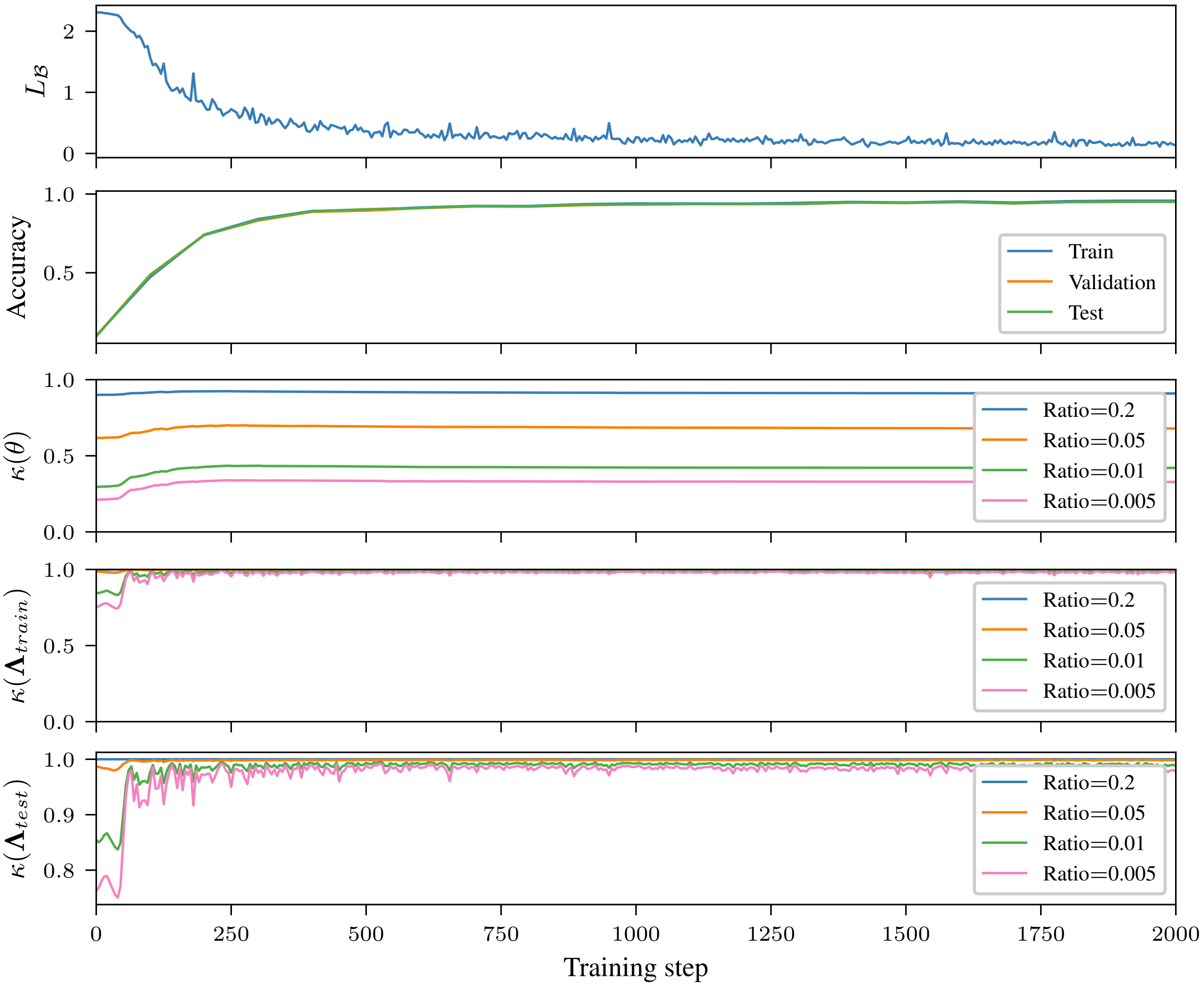

Figure 12: Both the parameters of the a neural network as well as their Hessian spectrum collapse early during training. *(top)* The top two subplots show the mini-batch loss $L_{\mathcal{B}}$ as well as the train/validation/test accuracy of the 7030-parameter model trained on 16×16 downsampled MNIST (see Section 6.1). *(middle)* Looking at the top $20\%, 5\%, 1\%, 0.5\%$ of parameters by magnitude, we can see that very early during training, most of the energy is concentrated on a small subset of the parameters (see Section 2.2 for a definition of κ). For example, shortly after initialization, the top 0.5% largest parameters by magnitude have roughly $1/4$ of the total ℓ_2 norm of all parameters. *(bottom)* We can observe a similar behavior for the Hessian spectrum on both the training set (fourth subplot) and the test set (fifth subplot). Only a few steps after training, most of the energy is concentrated in only 0.5% of the eigenvalues.

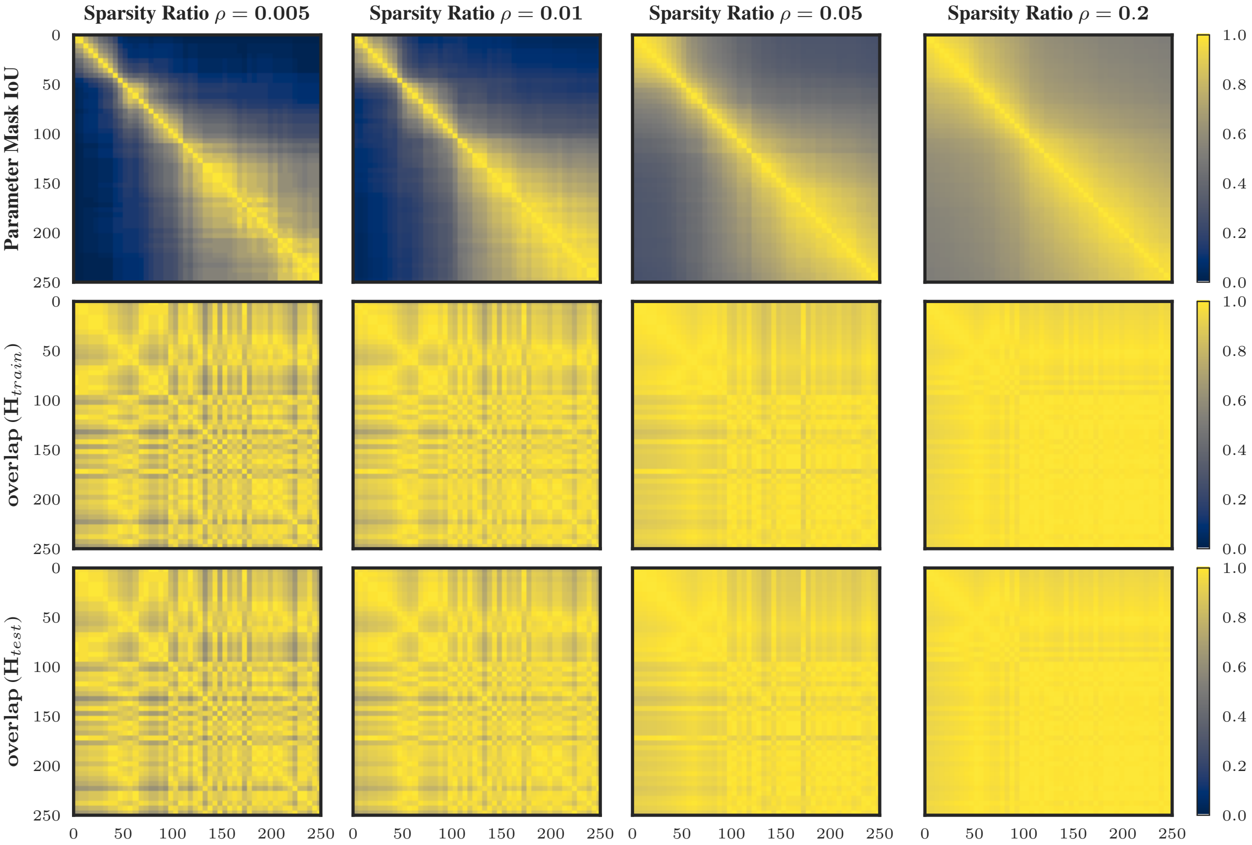

Figure 13: Both parameter pruning masks (top row) and Hessians (bottom two rows) remain relatively stable after an initial training phase. Following You et al. (2020), the depicted matrices represent all pairwise similarities (i.e. higher is better) from the beginning of training (top left corners) until step 250 (bottom right corners), for the 7030-parameter model trained on 16×16 downsampled MNIST (see Section 6.1). For this reason, all matrices are symmetric and have unit diagonals. *(top)* Even when selecting only 0.5% of the parameters (left column), masks collected at different training steps show a remarkable similarity after an initial phase of training. *(middle and bottom)* The overlap metric for top Hessian eigenspaces on the train (middle) and test set (bottom) extracted at different training steps and for different subspace sizes (columns). Starting from initialization, the Hessian eigenspaces do not change significantly over the course of training. No substantial differences in behavior between $\boldsymbol{H}_{test}$ and $\boldsymbol{H}_{train}$ can be observed.

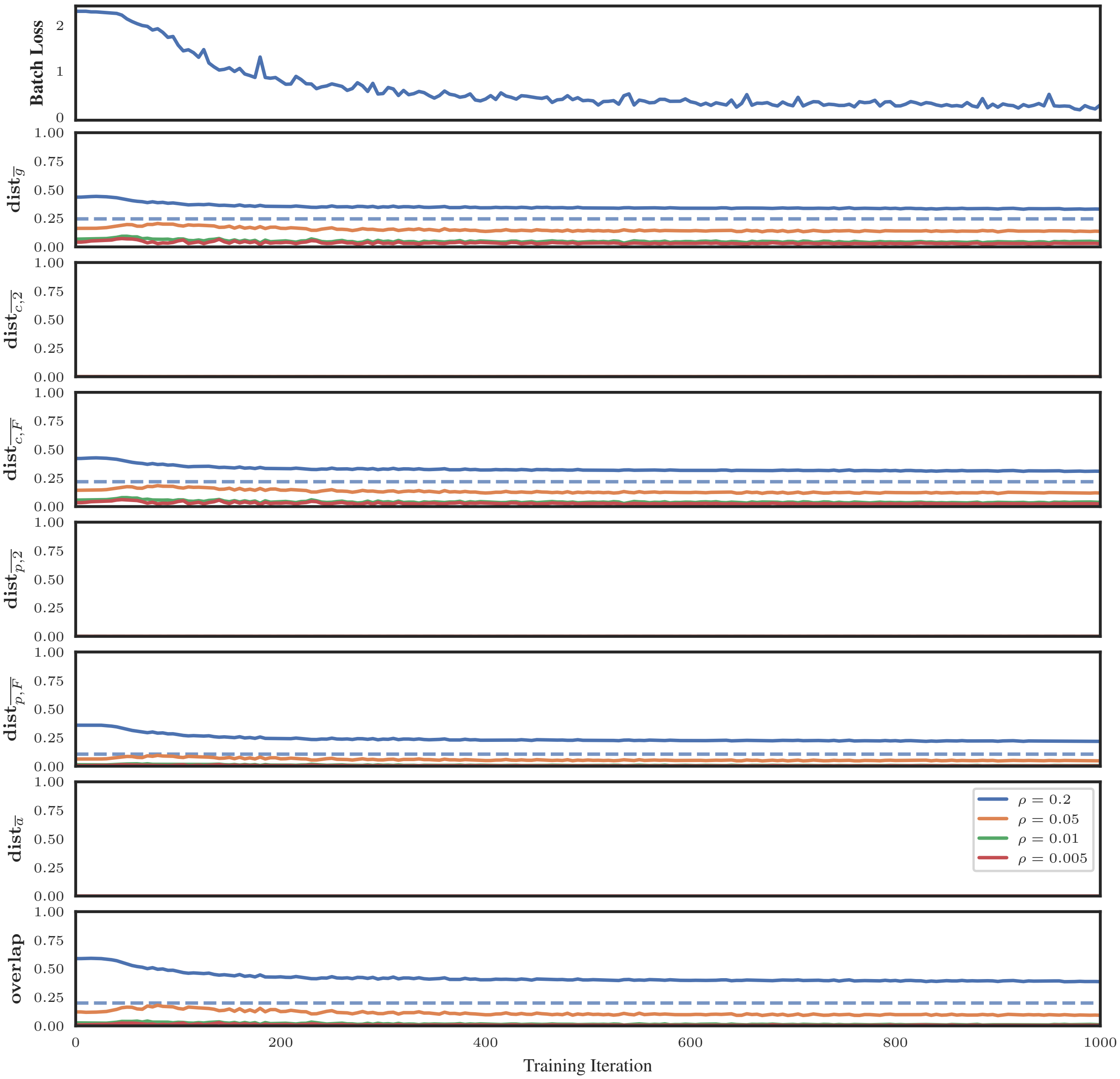

Figure 14: Different Grassmannian metrics between pruning masks and top Hessian subspaces. Each line shows a particular sparsity level ρ, i.e.. the ratio of unpruned parameters or the size of the top Hessian subspace relative to the full Hessian, as a function of training progress for the 7030-parameter model trained on 16×16 downsampled MNIST (see Section 6.1). All non-collapsing metrics reveal a consistent and substantial similarity between spaces spanned by pruning masks and top Hessian subspaces well above random chance (random baselines gathered from our synthetic experiments are shown in dashed lines for $\rho = 0.2$), while collapsing metrics are effectively zero due to the large D. For better visibility, we limit the plot to the first 1000 steps but all metrics remain stable afterwards.

B.2 Discussion on Computational Resources

We now discuss the resources involved in the computation of the Hessian eigendecompositions, as well as the experimental design choices sourrounding them. The goal is to reflect the scale of computations involved in our experiments, and to convey an idea of the overall resources needed to reproduce them. For more rigorous asymptotics involving memory, arithmetic and measurement costs, readers are invited to review the references exhaustively provided in Section 5, particularly (Tropp et al., 2019, Sec. 4).

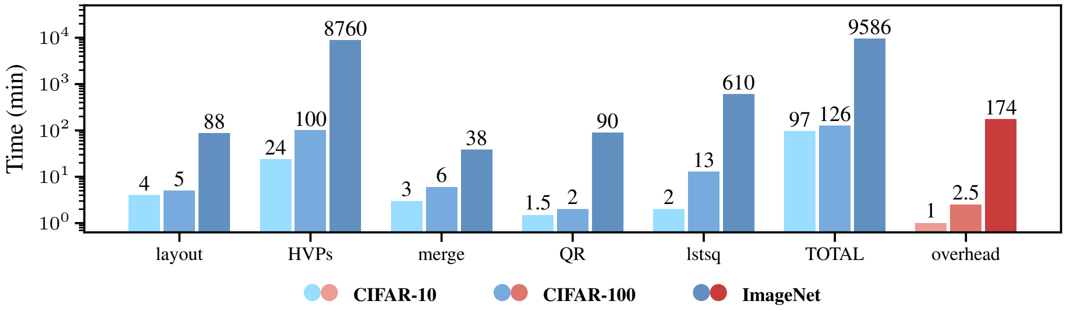

Figure 15: Runtimes of main SEIGH operations to compute a single Hessian eigendecomposition, assuming a single computer with 400GB RAM equipped with an NVIDIA A100 (40GB) graphics card. Note that the three involved eigendecompositions feature different networks, datasets, and rank, so a blind one-to-one comparison of runtimes is not possible: refer to Appendix B.2 for an interpretation of this figure.

Figure 15 gathers the average times it took to compute a single sketched Eigendecomposition, for each one of the Hessians covered in Section 6.2. Specifically:

- Resources needed to tune or train the model are here ignored. We used `PyTorch` Paszke et al. (2019).
- The *layout* step allocates the $\mathcal{O}(Dk)$ storage memory in the form of virtual `HDF5` datasets[5]. This allows to efficiently write numerical data in parallel, which is instrumental for the scalability of the procedure.
- The HVPs, computed with the help of `CurvLinOps`[6], correspond to the $\boldsymbol{A\Omega}_{1 \dots N_I}$ measurements in line 1 of Alg. 2. As we can see, they take the bulk of the runtime: each HVP requires two forward and backward passes over the Hessian dataset (consisting of 500, 1000 and 5000 samples for the three respective problems, see N_{train}/N_{test} in Table 1). While this can be done in batched fashion, the HVP runtime is linearly affected by dataset size. Another linear factor is the number of measurements (see N_O in Table 1). Crucially, those can be fully parallelized, so the runtimes reported under the *HVPs* column can be divided by the number of available machines. This property of sketched methods, combined with the HDF5 technology, is the main enabler for scalabiltiy here.
- The *merge* step stems from a technicality: most operative systems don't allow a process to open thousands of files at the same time. Hence, we need to merge the virtual HDF5 dataset into a monolithic file. This can be done with only $\mathcal{O}(D)$ memory overhead.
- Although (Tropp et al., 2019, Sec. 4) points out that the *QR decomposition* (line 3 of Alg. 2) dominates the (asymptotic) arithmetic cost for the whole procedure, interestingly this a relatively fast step. We also note that the main reason why we needed so much RAM memory was the need to load $\mathcal{O}(Dk)$ entries in-memory to perform this step. Out-of-core QR orthogonalizations would greatly help reducing RAM requirements.
- The remaining operations (lines 4-8 in Alg. 2) are here gathered under the *lstsq* label, which takes a considerable portion of the overall runtime. To improve this, lines 5 and 6 could be parallelized, roughly halving the runtime, but still faster solutions would greatly help towards more scalable Hessian eigendecompositions.

---

[5] `https://github.com/HDFGroup/hdf5`

[6] `https://github.com/f-dangel/curvlinops`

Figure 15 also reports the *TOTAL* runtime of the steps purely involved with the eigendecomposition. Further *overheads*, related e.g. to the use of a computational cluster, are reported separately. In practice, computing e.g. a single ResNet-18 Hessian eigendecomposition with up to fifteen A100s took a bit over a day. While this may seem like a heavyweight computation, it is still a substantial improvement, enabling previously intractable computations. In this section we have also outlined a few ways in which this could be further improved.

Due to the scale of the experiments, we had to be conservative with the choice of hyperparameters. We chose dataset sizes N_{train}/N_{test} such that all classes are balanced, and each class has at least 5 examples. While the chosen number of measurements N_O is much smaller than the ambient dimension D, the tremendously rank-defficient structure of $\boldsymbol{H}$ (see Section 2.1) allows for such a disparity. Still, it has been noted that the numerical rank of $\boldsymbol{H}$ may be linked to the number of classes (eg Gur-Ari et al., 2018; Papyan, 2020; Dangel et al., 2022). For this reason we chose N_O values well above this limit.