Grassmannian Packings in Neural Networks: Learning with Maximal Subspace Packings for Diversity and Anti-Sparsity

Dian Ang Yap
Stanford University
Stanford, CA 94305
dayap@stanford.edu

Nicholas Roberts
Carnegie Mellon University
Pittsburgh, PA 15213
nick11roberts@cmu.edu

Vinay Uday Prabhu
UnifyID AI Labs
Redwood City, CA 94063
vinay@unify.id

Abstract

Kernel sparsity ("dying ReLUs") and lack of diversity are commonly observed in CNN kernels, which decreases model capacity. Drawing inspiration from information theory and wireless communications, we demonstrate the intersection of coding theory and deep learning through the Grassmannian subspace packing problem in CNNs. We propose Grassmannian packings for initial kernel layers to be initialized maximally far apart based on chordal or Fubini-Study distance. Convolutional kernels initialized with Grassmannian packings exhibit diverse features and obtain diverse representations. We show that Grassmannian packings, especially in the initial layers, address kernel sparsity and encourage diversity, while improving classification accuracy across shallow and deep CNNs with better convergence rates.

1 Introduction

Filter level sparsity, commonly known as "dying kernels" or "dying ReLUs" are common phenomena in Convolutional Neural Networks (CNNs). This phenomenon has been attributed to adaptive optimizers (Adam, Adagrad, Adadelta), high L2 or weight decay, and the usage of the ReLU activation function [14, 11, 18]. Attempts to mitigate the dying ReLU issue with Leaky ReLUs have not been entirely effective, with minor overall impact on the emergent sparsity [14].

Figure 1: *Left*: sparse filters in CNNs; *Right*: diverse, non-sparse Grassmannian kernels.

While kernel sparsity or pruning could benefit CNNs for faster inference and smaller models [6, 15, 9, 4], sparsity in the lower layers could be detrimental as the first few layers, especially the first layer, are "sensitive to initialization" are critical to forming good predictions [20]. Since the initial layers learn simple patterns and subsequent layers build upon initial layers to develop complex features, sparsity in initial layers would decrease model capacity.

Besides sparsity, kernel diversity has garnered much attention: Zeiler et. al. proposed a set of best practices to improve upon *Alexnet* with different kernel sizes, stride-lengths and feature-scale clipping [19] to prevent specific filters from "dominating". Capturing distinctive features, especially in lower level layers, ensures that each filter learns something different and improves computational capacity that the CNNs dispense at the classification problem[5].

Here, we seek to address two problems, namely *sparsity* and *lack of diversity* of kernels, especially for the most important first few layers. We propose Grassmannian subspace packings as an initialization to project channel information onto a space where the basis vectors are maximally distant to capture as much diversity as possible and increase model capacity. We refer to the latter as "channel expander"
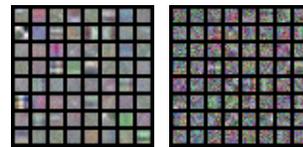
layers. We demonstrate that models with first-layer Grassmannian kernels achieve better accuracy and convergence rate with diverse, non-sparse kernels, which improves model capacity.

2 Grassmannian line and subspace packings

Grassmannian subspace packing [2] in experimental mathematics and information theory has been used in beamforming techniques for physical layer wireless communications, limited feedback codebook-based downlink beamforming schemes and sparse signal reconstruction [12, 10, 16].

Set theoretic definition. The real Grassmann manifold $G(m; k)$ is define as the set of all k-dimensional subspaces in $\mathbb{R}^m$, where $G(m; k) = \{W \subset \mathbb{R}^m \mid \dim(W) = k\}$. The Grassmannian N-subspace packing problem is the problem of finding a set of N k-dimensional subspaces in $G(m; k)$ that maximize the minimum pairwise distance between subspaces in the set under some metric [3].

For an N packing problem of $G(m; k)$, let $_{m \times k}$ denote the set of subspaces in $\mathbb{R}^{m \times k}$. We find $_{m \times k}$ where the minimum distance between any $\mathbf{w}_i; \mathbf{w}_j \in _{m \times k} (i \neq j)$ is maximized. The two popular distance metrics being chordal distance and Fubini-Study distance, which are defined below. With the restriction that $k \leq m$, for two k-dimensional subspaces $\mathcal{S}$ and $\mathcal{T}$,

$$d_{\text{chord}}(\mathcal{S}; \mathcal{T}) := \sqrt{\sin^2{}_1 + \cdots + \sin^2{}_k} \quad (1)$$
$$= \left[k - \|\boldsymbol{S} * \boldsymbol{T}\|_F^2\right]^{\frac{1}{2}};$$

$$d_{\text{FS}}(\mathcal{S}; \mathcal{T}) := \arccos\left(\prod_k^{'} \cos {}_k^{'}\right) \quad (2)$$
$$= \arccos |\det \boldsymbol{S} * \boldsymbol{T}|;$$

For our experiments, we use generate a codebook matrix for subspace packing based on Fubini-Study distance which has better experimental results[1][13]. We first generate a *codebook matrix*, $\mathbf{W} = \{\mathbf{w}_1 | \mathbf{w}_2 | \ldots | \mathbf{w}_N\}; \mathbf{w}_i \in _{m \times k}$. $\mathbf{W}$ is a rank-k codebook characterized by the minimum distance of packing $(\mathbf{W})$,

$$(\mathbf{W}) := \min_{i \neq j} d(\mathbf{w}_i; \mathbf{w}_j) \quad (3)$$

We seek to find $\mathbf{W} = \{\mathbf{w}_i\}_{i=1}^N$ that maximizes $(\mathbf{W})$,

$$\max_{\{\mathbf{w}_i\}, \mathbf{w}_i \in G(m,k)} \min_{i \neq j} d(\mathbf{w}_i; \mathbf{w}_j) \quad (4)$$

The Rankin bound [1] provides the theoretical upper bound of the maximum of the minimum pairwise distances between subspaces and is given by,

$$(\mathbf{W}) \leq \sqrt{\frac{(m - k)N}{m(N - 1)}} \quad (5)$$

Grassmannian packing seeks to answer *"what is the best way to optimally pack N k-dim subspaces in an M-dim space?"*. For the first convolutional kernel with width = 3, height = 3 (9 parameters per channel) with input channels = 3, output channels = 32, we ask:

> *"How can we optimally pack 9 3-dim subspace in a 32-dim space,*
> *such that any two subspaces (kernels) are maximally separated?"*

3 Experiments

We first generate Grassmannian codebooks for subspace packings, and load the codebook as CNN kernels. We compare accuracy and rate of convergence of shallow CNNs and deep CNNs. For a shallow 4-conv CNN, we trained 3 models: Kaiming-initialized CNNs, the same CNN but with Grassmannian as first layer (frozen and trainable). These models had their first layer initialized as $G(32; 3; 9)$ for CIFAR10 and CIFAR100, and $G(32; 1; 9)$ for single-channel MNIST and KMNIST[2].

For deep CNNs, we used ResNet18 architecture and initialized the first layer with $G(64; 3; 49)$[3]. We trained ResNet with batch size of 64, SGD optimizer with learning rate of 0.01, momentum of 0.9, with a learning rate decay by a factor of $\sqrt{10}$ every 10 epochs [8]. Besides having Kaiming-initialized baseline, first-layer Grassmann and first-layer frozen Grassmann, we also scaled the Grassmann as per

---

[1] https://www.mathworks.com/matlabcentral/fileexchange/41652-grassmannian-design-package.

[2] The first conv layer has parameters (width 3, height 3, 1 or 3 input channels and 32 output channels).

[3] The first layer of ResNets are conv kernels (width 7, height 7, 3 input channels and 64 output channels).

Kaiming init, such that the activations are correctly scaled that facilitates learning without vanishing gradients [7]. We scale the Grassmannian basis such that the magnitude is of factor $\sqrt{2/d_{\text{in}}}$. Since the scaling factor is constant, the directions of Grassmannian subspace basis vectors are preserved.

We check for convergence rate by examining first-epoch accuracy of shallow CNNs on MNIST, KMNIST, CIFAR10 and CIFAR100, as well as accuracy, kernel sparsity and diversity. As adaptive optimizers and weight decay could affect sparsity, we test on different optimizers such as SGD, Adadelta and Adam, and with/without weight decay of 0.0001.

4   Results and Discussions

Grassmannian models achieve higher accuracy on the 10 class Imagenette problem with ResNets, even when Grassmannian kernels are frozen upon initialization. With weight decay, Grassmannian kernels achieve the highest accuracy out of the four, whereas Kaiming-scaled Grassmannian packings outperforms the rest without weight decay, as per Table 1.

Table 1: ResNet18 on 10-class ImageNette with SGD optimizer, with and without weight decay.

Model	Validation Accuracy (%)	
	Without Decay	With Decay
Baseline	90.7	90.4
Grassmannian	92.4	**91.8**
Grassmannian, Frozen	90.8	91.6
Grassmannian, Kaiming-Scaled	**92.8**	91.5

Grassmannian initialization also improves convergence for both shallow (Figure 6) and Deep CNNs (Figure 3), with both frozen and trainable Grassmannian models achieving higher first-epoch accuracy (Table 2). Grassmannian subspace initialization, where each convolutional filter was assigned the subspace basis vectors for their corresponding Grassmannian packing scheme, outperformed the baseline on every dataset.

Table 2: Faster Convergence of Shallow CNN, first epoch accuracy.

Model	Validation Accuracy (%)			
	MNIST	KMNIST	CIFAR10	CIFAR100
Baseline	90.9	59.7	33.6	2.57
Grassmannian, Frozen	91.9	66.2	40.5	**4.73**
Grassmannian	**92.6**	**66.4**	**41.1**	4.70

Interestingly, we see a marginal improvement over Grassmannian initialization by simply rendering the Grassmannian initialization untrainable. We postulate that in certain cases, simply projecting the input feature maps to be as far apart as possible as opposed to using another trainable convolutional layer is enough for later layers to make better use of their own capacity, at least in early training.
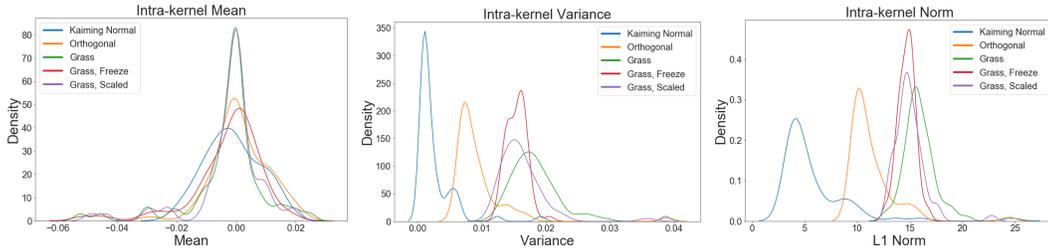

Figure 2: Distribution of mean, mariance and norm over the 64 kernels of first layer of ResNet.

Qualitative inspection of the kernels (Figure 7) reveal that Grassmannian kernels learn diverse features with a mixture of edge, color and pattern detectors in each kernel, without any kernel dominating that shows improved diversity and fewer sparsity. While baseline and Grassmannian packings gave mean close to 0, intra-kernel variance and norms are higher for Grassmannian packings, indicating diversity and fewer sparsity with the distribution across kernels shown in Figure 2.

Interestingly, trainable Grassmannian packings outperform baseline which outperforms frozen Grassmannian packings when adaptive optimizers are used (especially Adam, see Figure 5). Since Adam contributes to sparsity [14, 18], it is at odds with Grassmannian packings that are against sparsity and for diversity. Nonetheless, most models such as ResNet, RandWire, NASNet and DenseNet achieve SOTA results through SGD with a careful learning rate scheduler, which could further benefit from Grassmannian packings due to their demonstrated performance using SGD [17].

5 Conclusion

In this work, we showed that the use of Grassmannian packings for first layer initialization improved performance in the contexts of shallow networks and deep networks, even with untrainable Grassmannian kernels, across multiple datasets. Grassmannian kernels are best used with SGD optimization and works with weight decay or otherwise. Our results suggest improvement of the model's capacity to learn by simply initializing the first layer filters to be maximally distant and capture as many diverse features as possible. In the future, we hope to extend this analysis to explore the several other options for metric used in the Grassmannian packing, and search for ways to inform the choice in metric.

References

[1] Barg, A. and Nogin, D. Y. Bounds on packings of spheres in the grassmann manifold. *IEEE Transactions on Information Theory*, 48(9):2450–2454, 2002.

[2] Conway, J. H., Hardin, R. H., and Sloane, N. J. Packing lines, planes, etc.: Packings in grassmannian spaces. *Experimental mathematics*, 5(2):139–159, 1996.

[3] Dhillon, I. S., Heath, J. R., Strohmer, T., and Tropp, J. A. Constructing packings in grassmannian manifolds via alternating projection. *Experimental mathematics*, 17(1):9–35, 2008.

[4] Gale, T., Elsen, E., and Hooker, S. The state of sparsity in deep neural networks. *arXiv preprint arXiv:1902.09574*, 2019.

[5] Graff, C. A. and Ellen, J. Correlating filter diversity with convolutional neural network accuracy. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 75–80. IEEE, 2016.

[6] Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[7] He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

[8] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[9] Li, H., Kadav, A., Durdanovic, I., Samet, H., and Graf, H. P. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

[10] Love, D. J. and Heath, R. W. Limited feedback unitary precoding for spatial multiplexing systems. *IEEE Transactions on Information theory*, 51(8):2967–2976, 2005.

[11] Lu, L., Shin, Y., Su, Y., and Karniadakis, G. E. Dying relu and initialization: Theory and numerical examples. *arXiv preprint arXiv:1903.06733*, 2019.

[12] Malioutov, D., Cetin, M., and Willsky, A. S. A sparse signal reconstruction perspective for source localization with sensor arrays. *IEEE transactions on signal processing*, 53(8):3010–3022, 2005.

[13] Medra, A. and Davidson, T. N. Flexible codebook design for limited feedback systems via sequential smooth optimization on the grassmannian manifold. *IEEE Transactions on Signal Processing*, 62(5): 1305–1318, 2014.

[14] Mehta, D., Kim, K. I., and Theobalt, C. On implicit filter level sparsity in convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 520–528, 2019.

[15] Molchanov, P., Tyree, S., Karras, T., Aila, T., and Kautz, J. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440*, 2016.

[16] Prabhu, V. U., Karachontzitis, S., and Toumpakaris, D. Performance comparison of limited feedback codebook-based downlink beamforming schemes for distributed antenna systems. In *2009 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology*, pp. 171–176. IEEE, 2009.

[17] Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., and Recht, B. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems*, pp. 4148–4158, 2017.

[18] Yaguchi, A., Suzuki, T., Asano, W., Nitta, S., Sakata, Y., and Tanizawa, A. Adam induces implicit weight sparsity in rectifier neural networks. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 318–325. IEEE, 2018.

[19] Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

[20] Zhang, C., Bengio, S., and Singer, Y. Are all layers created equal? *arXiv preprint arXiv:1902.01996*, 2019.

Appendix
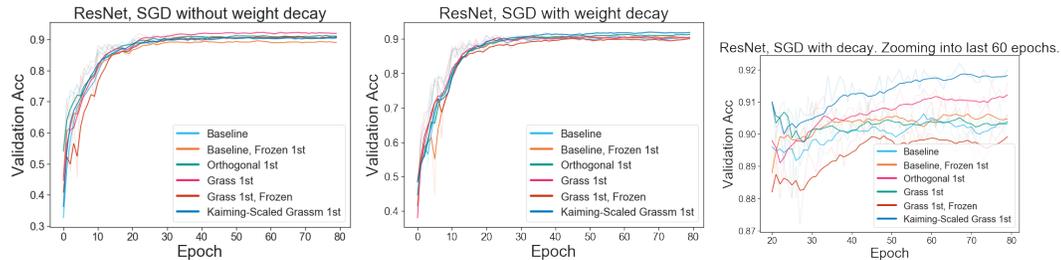
Figure 3: ResNet with and without weight decay, with Grassmannian achieving better convergence and better final accuracy with everything else kept constant. Last image: zooming into last few epochs for ResNet with weight decay.

Table 3: Final Accuracy of ResNet-56 on CIFAR-10 with first layer Xavier, Grassmannian trainable, and Grassmannian Frozen.

INITIALIZATION	TEST ACC
STANDARD, XAVIER	91.85
GRASSMANNIAN, FIXED	91.98
GRASSMANNIAN, TRAINABLE	**92.31**

$$\left\{ y_{i,j,k} = f\left(\mathbf{x}_{i,j} * \mathbf{w}_k + b_k \right) \right\}_{k=1}^{K} ; \mathbf{w}_k \in \mathbb{R}^m$$
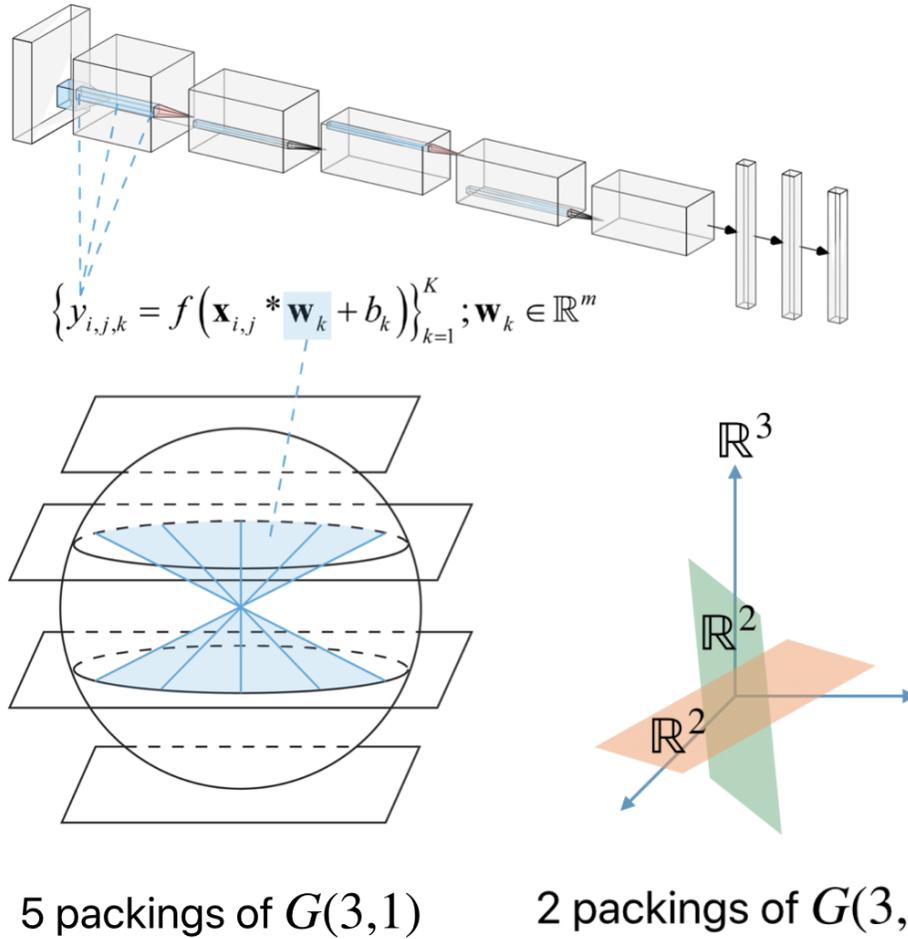
5 packings of $G(3,1)$ 2 packings of $G(3,2)$

Figure 4: Illustration of how the first conv layer is initialized with Grassmannians, with visualizations of Grassmannian packings. *Left*: 5-packing of $G(3,1)$, which answers "how should 5 laser beams passing through a single point in $\mathbb{R}^3$ be arranged so as to make the angle between any two of the beams as large as possibe". *Right*: 2-packing of $G(3,2)$. These concepts can be extended to higher dimensions.

(a) SGD

(b) Adadelta

(c) Adam
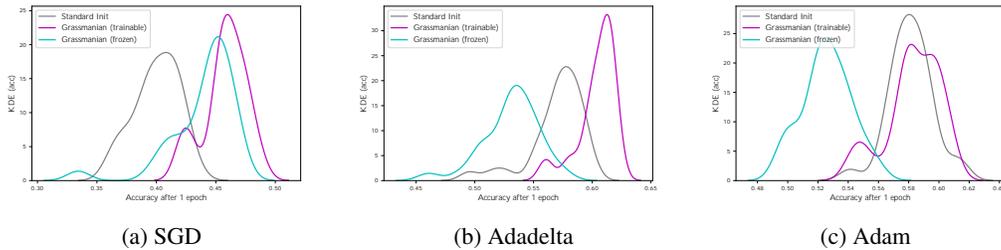
Figure 5: Distributions of 30 runs of first-epoch test accuracy at the first epoch **with different optimizers** on the same shallow CNN architecture, comparing initialization of first layer using standard Xavier initialization, frozen Grassmannian packings, and trainable Grassmannian packings.
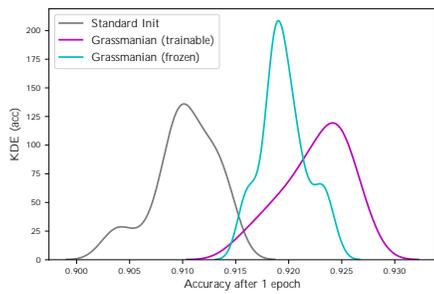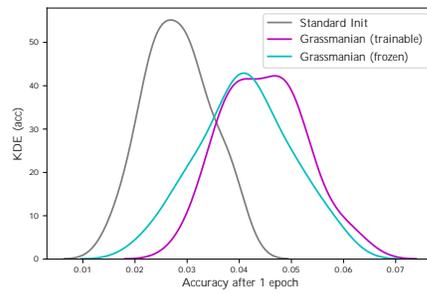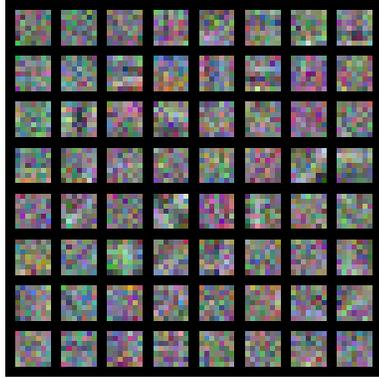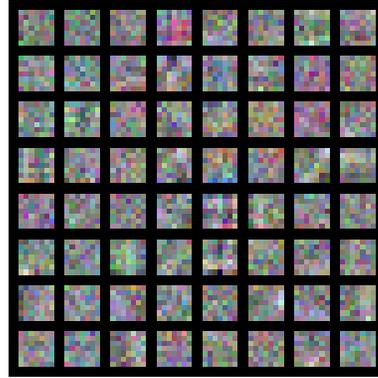
(a) MNIST
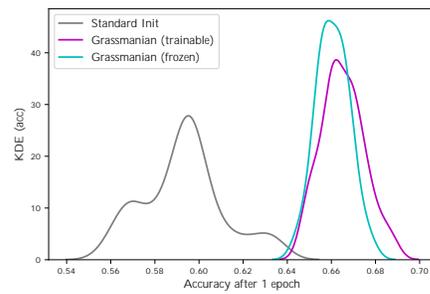
(b) KMNIST
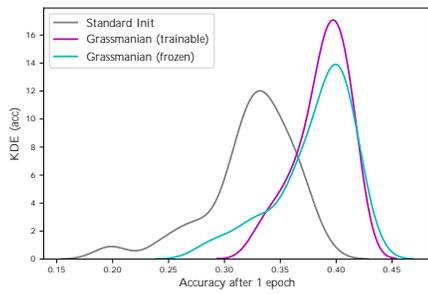
(c) CIFAR10

(d) CIFAR100

Figure 6: Distributions of 30 runs of first-epoch test accuracy at the first epoch with SGD with different datasets, comparing initialization of first layer using standard Xavier initialization, frozen Grassmannian packings, and trainable Grassmannian packings.
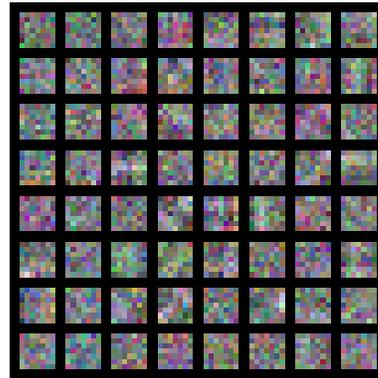
(a) Grassmannian, no weight decay.

(b) Kaiming-scaled Grassmannian, no weight decay.
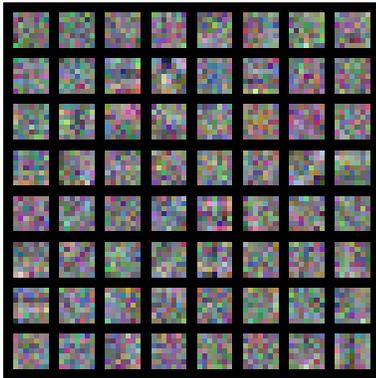
(c) Grassmannian, weight decay.

(d) Kaiming-scaled Grassmannian, weight decay.

Figure 7: Visualizations of Trainable Grassmannian Kernels of ResNet trained using SGD, with and without weight decay after training for 80 epochs on ImageNette.
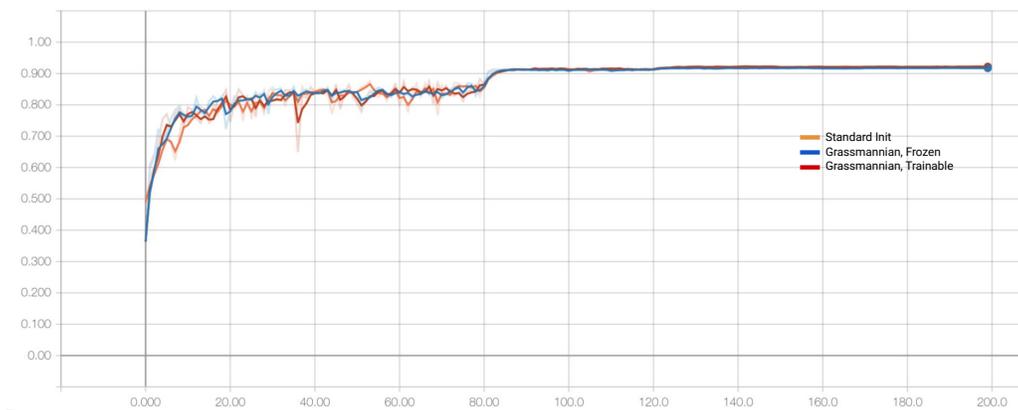

Figure 8: Comparison of standard initialization and Grassmannian initialization of first layer as both trainable and untrainable parameters on ResNet56 trained on CIFAR-10. Grassmannian approaches have a faster convergence with marginally better test accuracy with Adam optimizer used in all 3 cases.