NEURAL NETWORK COMPRESSION: THE FUNCTIONAL PERSPECTIVE

Israel F. Mason-Williams
University of Cambridge
ifm24@cam.ac.uk

ABSTRACT

Compression techniques, such as Knowledge distillation, Pruning, and Quantization reduce the computational costs of model inference and enable on-edge machine learning. The efficacy of compression methods is often evaluated through the proxy of accuracy and loss to understand similarity of the compressed model. This study aims to explore the functional divergence between compressed and uncompressed models. The results indicate that Quantization and Pruning create models that are functionally similar to the original model. In contrast, Knowledge distillation creates models that do not functionally approximate their teacher models. The compressed model resembles the dissimilarity of function observed in independently trained models. Therefore, it is verified, via a functional understanding, that Knowledge distillation is not a compression method. Thus, leading to the definition of Knowledge distillation as a training regulariser given that no *knowledge* is *distilled* from a teacher to a student.

1 INTRODUCTION

The recent growth in size and complexity of neural network models towards billion parameter models (Muhamed et al., 2021) has fueled unprecedented advancements in various domains such as computer vision (Dehghani et al., 2023) and natural language processing (Hoffmann et al., 2022). While these complex models boast remarkable performance, their deployment in resource-constrained environments remains challenging due to their high computational costs (Wu et al., 2019). Knowledge distillation (Hinton et al., 2015), Pruning (LeCun et al., 1989), and Quantization have emerged as solutions to bridge the gap between computational efficiency and complexity.

The current understanding of compression techniques is that compressed models approximate the original function of uncompressed models (Hinton et al., 2015). This assumption results from an accuracy and loss-based analysis of the compress model. For example, the underpinning notion of Knowledge distillation involves transferring the knowledge encapsulated in a sophisticated teacher to a more streamlined student. It is assumed that the student will approximate the teacher's function as the student can match or improve performance via Knowledge distillation (Hinton et al., 2015). However, functional analysis of independent models shows that models trained on the same dataset with similar accuracy and loss can form very different functional representations (Fort et al., 2019). As a result, using accuracy and loss alone to gauge functional similarity is unsound as it infers a symmetric relationship of functional similarity and accuracy, which is not apparent for similarly-performing independent models. As a result, this paper asks the following questions:

1. Does Knowledge distillation, Quantization or L1 Pruning result in a compressed model that is functionally similar to the original model?
2. Which compression methods, if any, are the most efficacious for function preservation?
3. How does compression and function preservation scale across multiple architectures and datasets?

The results show:

- Knowledge distillation cannot be considered a compression method as it results in a student model that is as functionally dissimilar to the teacher as independently trained models.

- Different α values do not impact the degree of functional similarity between a student and a teacher in Knowledge distillation.
- Post hoc Quantization is the best method for functional preservation across all explored architectures and datasets.
- L1 Pruning is an effective method for functional preservation across datasets and is partially constrained by the architecture and dataset.
- Compression of a model needs to be viewed from functional proximity rather than just accuracy as accuracy is a false similarity proxy.

2 BACKGROUND

2.1 KNOWLEDGE DISTILLATION

Knowledge distillation suggests that a student can successfully approximate accurate internal representations provided by a singular pre-trained model or an ensemble of the models (Hinton et al., 2015). In Knowledge distillation, a unidirectional backpropagation updates the student's weights while preserving the teacher's. Embodying a nuanced strategy distilling knowledge from a proficient teacher to a student (Hinton et al., 2015). Knowledge distillation leverages a distillation loss function and a *temperature parameter* ($\mathcal{T}$) to smooth teacher outputs; higher $\mathcal{T}$ are said to enable access to a teacher's dark knowledge. Additionally, the loss computation incorporates an *alpha factor (α)* ranging between zero and one that scales the influence of the student and teacher before backpropagating the loss. A low α value (close to 0) emphasises student-centric information, and a high α value (close to 1) tends toward a more substantial reliance on knowledge transfer from the teacher model. Self distillation has proven helpful in improving the accuracy of a student with the same architecture as a teacher (Allen-Zhu & Li, 2023). Literature suggests that in Self distillation, the teacher guides the model to an accuracy that exceeds the teacher's, by guiding the student to a flatter minima (Zhang et al., 2022).

2.2 QUANTIZATION

Quantization is a form of model compression that has gained significance in overcoming bottlenecks associated with deploying machine learning models on-edge devices (Gholami et al., 2022). It is primarily utilised to overcome overwhelming memory and computational requirements at inference. Post-hoc Quantization is applied after training to a conventional model trained with complete 32-bit floating-point precision (Banner et al., 2019). The full precision model weights and biases undergo Quantization, which involves a transformative shift towards lower-bit precision, ranging from 16 to 4-bit integers. Quantization promises many benefits, such as reduced model size and accelerated inference speed; however, it presents a trade-off as it can result in accuracy degradation (Gholami et al., 2022). Techniques such as Quantized Aware Training are employed to preserve crucial information and minimise the compromise on accuracy at lower bit regimes (Banner et al., 2018).

2.3 MODEL PRUNING

Model Pruning, formerly known as Optimal Brain Damage (ODB), follows the notion that network complexity can be reduced by removing parameters with low saliency, referring to parameters whose elimination minimally impacts the training error (LeCun et al., 1989). It is observed that parameters with small magnitudes exhibit lower saliency and, therefore, can be removed (pruned) from the overall network (LeCun et al., 1989). ODB developed into model Pruning by employing a magnitude-based weight Pruning method that is more computationally efficient and scales better to complex networks (Zhu & Gupta, 2018) such as L1 Pruning. In terms of model compression, structured Pruning (Anwar et al., 2017) is the most viable method to accelerate the hardware performances of deep neural networks. L1 unstructured Pruning does not adhere to a particular geometry constraint, and consequently, additional information is required to denote sparse locations, which can be a bottleneck for efficient computation (Anwar et al., 2017).

2.4 FUNCTIONAL SIMILARITY

Fort et al., (2019) explored functional diversity in ensembles by showing the functional landscape through softmax representations. They reduce the softmax predicted outputs on the test set to a two-dimensional representation using t-NSE (Van der Maaten & Hinton, 2008) and plot the resulting representation. The representation serves as a low dimensional visualization of the functional similarity between two or more models (Fort et al., 2019).

3 EXPERIMENTAL SETUP

To explore the compression methods functional similarity, three architectures, LeNet (LeCun et al., 1998), ResNet-50 (He et al., 2016) and MobileNet (Howard et al., 2017) (Table 4 in Appendix), were trained on CIFAR10 and CIFAR100 (Krizhevsky et al., 2009), with an initial experiment conducted using the LeNet architecture on the MNIST dataset (LeCun et al., 2010) in 4.1. The findings are evaluated for how compression methods and functional similarity scale with increasing dataset complexity (Table 5 in Appendix).

Three independent models of the same architecture were trained for each respective architecture and dataset. Each model was trained for 25 epochs with a batch size of 64, with the final model being saved. After each epoch, the model was evaluated against the test dataset, and the softmax outputs were saved. The method presented by Fort et al., 2019 described in 2.4 is employed to show the functional diversity of compression outcomes. Each compression method was applied independently to the base model, which is the first independently trained model. The other independent models are presented in figures for reference.

Self distillation is employed to explore functional similarity of student and teacher models. The base model is used as the teacher, and the student is the base model at initialisation. The student has the capacity to match the exact representation of the teacher, allowing for a comprehensive understanding of the functional relation elicited by distillation. In the Self distillation results, $\mathcal{T}$ = 3, and $\alpha = [0.1, 0.5, 0.9]$. The α values allow observation of functional similarity to the teacher depending on the proportion of the teacher's signal used during training. All Self distillation results were averaged across three random seeds, with each functional similarity shown in the Appendix in B.0.1 -B.0.2. Quantization was applied as a post hoc method at 16 and 8-bit regimes. For Pruning, L1 Pruning was employed as a post hoc method on the final model with no fine-tuning; functional similarity is compared when Pruning [10%, 30%,50%,70%,90%] of the original network.

4 RESULTS

4.1 MNIST

Table 1: Percentage deviation of base model test accuracy 99.15% on MNIST

	Self distillation with α			Quantization		L1 Pruning				
Architecture	**0.1**	**0.5**	**0.9**	**16-bit**	**8-bit**	**10%**	**30%**	**50%**	**70%**	**90%**
LeNet	-0.178 (±0.057)	-0.124 (±0.042)	-0.215 (±0.076)	-0.07	-0.05	-0.07	-0.08	-0.01	-1.04	-22.55

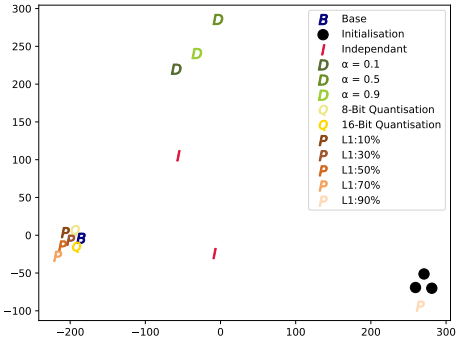


Figure 1: MNIST Functional Landscape for LeNet. B: Base model, P: Pruning, D: Self distillation, Q: Quantization; I: Independently trained model

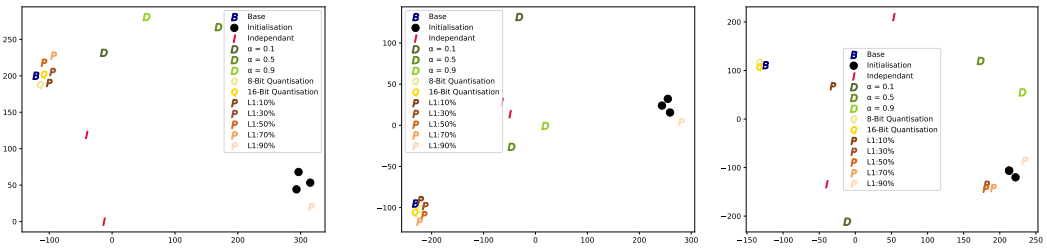
The test accuracy deviation from the base model given the application of each compression method is shown in Table 1. It is evident from the table that each compression method has a minute impact on the model's accuracy - showing that compression methods can be utilised without compromising accuracy provided by the original model. Only when the Pruning reaches 90% of the network there is reduction in accuracy above 2%. However, MNIST is a particularly trivial learning task, and, therefore, this is unsurprising. It can be observed, in Figure 1, that Quantization and Pruning (while accuracy deviation remains below 2%) remain the most similar to the base model functional representation. Self distillation, however, is as dissimilar to the base model as the independent models across all α values. The functional comparison suggests that distillation cannot be considered a compression method.

4.2 CIFAR10

Findings from MNIST are further explored on the CIFAR10 dataset. The base models for LeNet, ResNet-50 and MobileNet architectures achieved test accuracy's of 63.38%, 67.37% and 68.70 %, respectively.

Table 2: Percentage deviation of base model test accuracy on CIFAR10

	Self distillation with α			**Quantization**		**L1 Pruning**				
Architecture	**0.1**	**0.5**	**0.9**	**16-bit**	**8-bit**	**10%**	**30%**	**50%**	**70%**	**90%**
LeNet	8.74 ($\pm$0.65)	8.93 ($\pm$0.45)	8.57 ($\pm$0.72)	-0.62	-0.33	-0.25	-0.60	-2.95	-14.56	-78.466
ResNet-50	-0.74 ($\pm$1.16)	-3.41 ($\pm$1.35)	0.09 ($\pm$0.9)	0.00	-0.56	-0.08	0.48	-0.03	-0.04	-85.30
MobileNet	-18.25 ($\pm$0.56)	-20.08 ($\pm$0.77)	-21.15 ($\pm$1.48)	0.02	0.07	-16.84	-83.70	-85.43	-83.84	-85.40



(a) LeNet (b) ResNet-50 (c) MobileNet

Figure 2: CIFAR10 Functional Landscape. B: Base model, P: Pruning, D: Self distillation, Q: Quantization; I: Independently trained model

The CIFAR10 test accuracy percentage deviation results shown in Table 2 provide insights into which compression methods are best for each model. It is evident that all models see an equal fluctuation in accuracy when Quantization is applied. Interestingly, the L1 Pruning results show that different architectures respond uniquely to Pruning. This suggests that some architectures have more redundant information held than others, allowing for a harsher Pruning regime to be implemented.

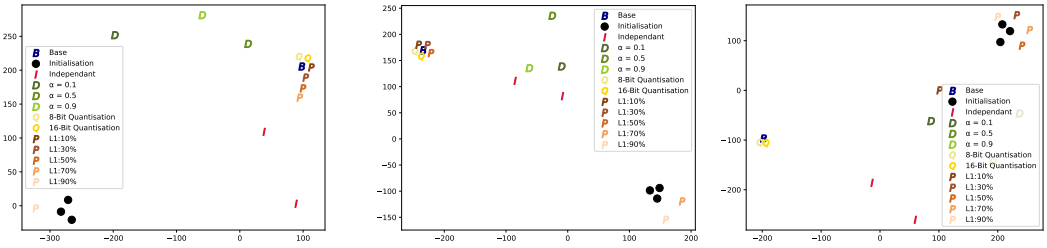
While the efficacy of the compression methods varies between the different architectures, the functional landscapes in Figure 2 indicate insights that mirror those witnessed on MNIST in Figure 1. The Quantized models remain close to the base models functional representation, and likewise for the pruned models of LeNet and ResNet-50. Notably, when 70% of the network is pruned for LeNet, and there is decrease of almost 15% in accuracy, the functional similarity remains close to the base model. Yet again, with Self distillation, the functional behaviour mimics that of the independent models - reiterating that the teacher does not pass any information that enables the student to approximate the function of the teacher.

4.3 CIFAR100

The final exploration was completed on CIFAR100, with the base test accuracy on the LeNet, ResNet-50 and MobileNet being 33%, 37% and 29.12% respectively.

Table 3: Percentage deviation of base model test accuracy on CIFAR100

Architecture	Self distillation with α			Quantization		L1 Pruning				
	0.1	**0.5**	**0.9**	**16-bit**	**8-bit**	**10%**	**30%**	**50%**	**70%**	**90%**
LeNet	15.64 ($\pm$1.56)	15.71 ($\pm$0.51)	15.63 ($\pm$1.56)	0.00	-0.16	-0.13	-0.7	-4	-32.31	-85.62
ResNet-50	-5.20 ($\pm$3.89)	0 ($\pm$2.55)	1.56 ($\pm$2.43)	-0.08	-0.16	0.05	-0.16	-1.45	-96.97	-97.34
MobileNet	-26.66 ($\pm$1.17)	-25.63 ($\pm$2.66)	-27.35 ($\pm$3.40)	0.17	-0.03	-36.26	-95.74	-96.01	-96.43	- 96.57



(a) LeNet (b) ResNet-50 (c) MobileNet

Figure 3: CIFAR100 Functional Landscape. B: Base model, P: Pruning, D: Self distillation, Q: Quantization; I: Independently trained model

Figure 3 consolidates indications provided by the MNIST and CIFAR10 results. Firstly, Quantization is the most universally applicable method for compressing a network whilst retaining functional similarity and accuracy across various architectures. Secondly, Pruning is an effective strategy for maintaining functional similarity when withstanding severe degradation in accuracy. Additionally, different architectures have varying sensitivity to L1 Pruning. Findings from Tables 2 and 3 offer a potentially more efficient pathway to Pruning by using less complex proxy datasets on specific architectures to gain insights into how sensitive an architecture is to Pruning. As a result, this will allow less resource expenditure to find the appropriate compression for an architecture at scale. Finally, while some architectures can realise accuracy gains due to distillation, Knowledge distillation cannot be considered a compression method as it does not approximate the function of the teacher. The student is repeatedly as functionally dissimilar to the base model as the independent models across all architectures and datasets.

5 Conclusion

The results presented in this paper create a new paradigm for considering compression methods. Through this paradigm, Quantization and L1 Pruning can be defined as compression methods, as the resulting compressed model strongly resembles the function of the base model, notwithstanding significant fluctuations in the accuracy from the base model. Not only do Quantization and L1 Pruning cluster around the base model, they are much closer than independent models are to one another, further validating that the function of the base model is maintained during the application of these compression methods. In instances where a robust model is going to be compressed, it would be apt to first employ Quantization followed by Pruning instead of a Knowledge distillation method. The same results show that Knowledge distillation produces students as functionally dissimilar to their teacher as independently trained models. The implication is that to consider Knowledge distillation, a compression method is inaccurate from a functional perspective. Consequently, Knowledge distillation should be perceived as a form of regularised training. Additionally, it could also be argued that the name Knowledge distillation is misleading and, therefore, should be revisited and explored to understand what is truly happening in the distillation process. The findings ultimately raise questions over the feasibility of transfer knowledge between neural networks as this work suggests that it is not possible with current Knowledge distillation methods.

References

Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. In *The Eleventh International Conference on Learning Repre-*

sentations, 2023. URL https://openreview.net/forum?id=Uuf2q9TfXGA.

Sajid Anwar, Kyuyeon Hwang, and Wonyong Sung. Structured pruning of deep convolutional neural networks. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 13(3):1–18, 2017. URL https://dl.acm.org/doi/10.1145/3005348.

Ron Banner, Itay Hubara, Elad Hoffer, and Daniel Soudry. Scalable methods for 8-bit training of neural networks. *Advances in neural information processing systems*, 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/e82c4b19b8151ddc25d4d93baf7b908f-Abstract.html.

Ron Banner, Yury Nahshan, and Daniel Soudry. Post training 4-bit quantization of convolutional networks for rapid-deployment. *Advances in Neural Information Processing Systems*, 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/c0a62e133894cdce435bcb4a5df1db2d-Abstract.html.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, and . Padlewski, et al. Scaling vision transformers to 22 billion parameters. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 7480–7512. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/dehghani23a.html.

Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019. URL https://arxiv.org/pdf/1912.02757.pdf.

Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. In *Low-Power Computer Vision*, pp. 291–326. Chapman and Hall/CRC, 2022. URL https://arxiv.org/abs/2103.13630.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. URL https://ieeexplore.ieee.org/document/7780459.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. URL https://arxiv.org/abs/1503.02531.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. URL https://arxiv.org/abs/2203.15556.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. URL https://arxiv.org/abs/1704.04861.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989. URL https://proceedings.neurips.cc/paper_files/paper/1989/file/6c9882bbac1c7093bd25041881277658-Paper.pdf.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. URL https://ieeexplore.ieee.org/abstract/document/726791.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]*, 2, 2010. URL http://yann.lecun.com/exdb/mnist.

Aashiq Muhamed, Iman Keivanloo, Sujan Perera, James Mracek, Yi Xu, Qingjun Cui, Santosh Rajagopalan, Belinda Zeng, and Trishul Chilimbi. Ctr-bert: Cost-effective knowledge distillation for billion-parameter teacher models. In *NeurIPS Efficient Natural Language and Speech Processing Workshop*, 2021. URL https://neurips2021-nlp.github.io/papers/20/CameraReady/camera_ready_final.pdf.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. URL https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf.

Carole-Jean Wu, David Brooks, Kevin Chen, and . Chen, et al. Machine learning at facebook: Understanding inference at the edge. In *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 331–344, 2019. doi: 10.1109/HPCA.2019.00048. URL https://ieeexplore.ieee.org/abstract/document/8675201.

Linfeng Zhang, Chenglong Bao, and Kaisheng Ma. Self-distillation: Towards efficient and compact neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4388–4403, 2022. doi: 10.1109/TPAMI.2021.3067100. URL https://ieeexplore.ieee.org/document/9381661.

Michael H. Zhu and Suyog Gupta. To prune, or not to prune: Exploring the efficacy of pruning for model compression, 2018. URL https://openreview.net/forum?id=S1lN69AT-.

A Appendix - Architecture and Dataset Complexity

Table 4: Architecture Features

MODEL	TRAINABLE PARAMETERS	SIZE(MB)
LeNet	225,034	0.879
MobileNet V1	3,736,906	14.34
ResNet-50	24,634,980	94.18

Table 5: Dataset Features

DATASET	INPUT DIMENSIONS	INSTANCES	NUMBER OF CLASSES
MNIST	1 x 28 x 28	70,000	10
CIFAR10	3 x 32 x 32	60,000	10
CIFAR100	3 x 32 x 32	60,000	100

B Appendix - Self distillation Functional Pathways

B.0.1 CIFAR 10

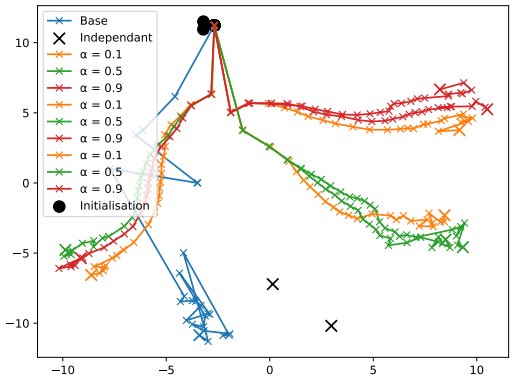


Figure 4: MNIST Functional Landscape for Self distillation on LeNet Architecture Across Random Seeds 2,24 and 42

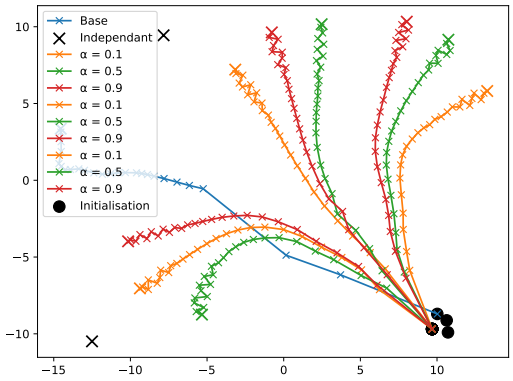


Figure 5: CIFAR10 Functional Landscape for Self distillation on ResNet Architecture Across Random Seeds 2,24 and 42

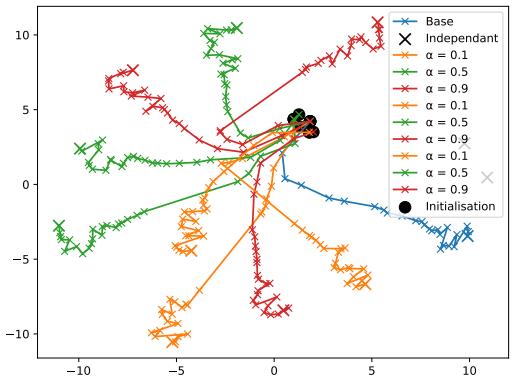


Figure 6: CIFAR10 Functional Landscape for Self distillation on MobileNet Architecture Across Random Seeds 2,24 and 42

B.0.2 CIFAR 100

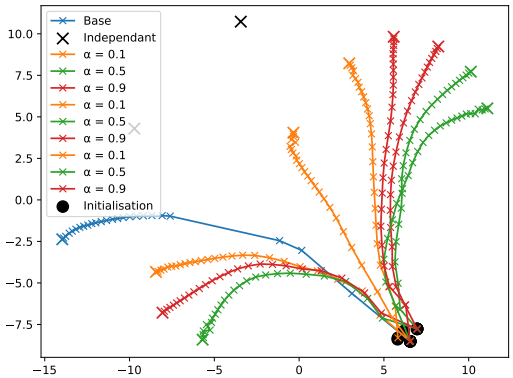


Figure 7: CIFAR100 Functional Landscape for Self distillation on LeNet Architecture Across Random Seeds 2,24 and 42

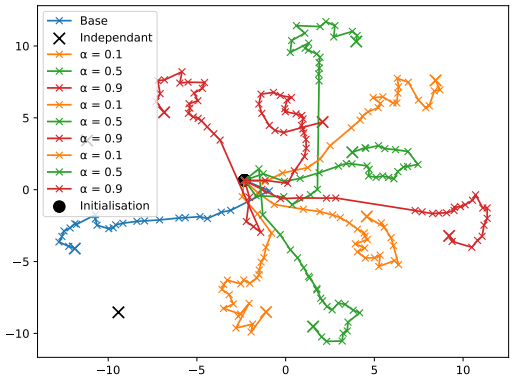


Figure 8: CIFAR100 Functional Landscape for Self distillation on MobileNet Architecture Across Random Seeds 2,24 and 42