Proceedings Track

Visualizing Loss Functions as Topological Landscape Profiles

Editors: List of editors' names

Abstract

In machine learning, a loss function measures the difference between model predictions and ground-truth (or target) values. Visualizing how this loss changes as a neural network's parameters are varied can provide insights into the local structure of the so-called loss landscape (e.g., smoothness) and global properties of the underlying model (e.g., generalization performance). While various methods for visualizing the loss landscape have been proposed, many approaches limit sampling to just one or two directions, ignoring potentially relevant information in this extremely high-dimensional space. This paper introduces a new representation based on topological data analysis that enables the visualization of higher dimensional loss landscapes. In addition to this new topological landscape profile representation, we present an interactive tool for users to explore these landscapes across different models and hyperparameters, enabling more systematic comparisons and informed model exploration. We highlight several use cases, including image segmentation (e.g., UNet) and scientific machine learning (e.g., physics-informed neural networks), showing how visualizing higher-dimensional loss landscapes can provide new insights into model performance and learning dynamics. Through these examples, we provide new insights into how loss landscapes vary across distinct hyperparameter spaces, finding that the topology of the loss landscape is simpler for better-performing models. Interestingly, we observe more variation in the shape of loss landscapes near transitions from low to high model performance.

Keywords: Topological data analysis, loss landscapes, model diagnosis

1. Introduction

A central aim of machine learning (Devlin et al., 2018; Liu et al., 2019; Vaswani et al., 2017; Krizhevsky et al., 2017; Simonyan and Zisserman, 2014; He et al., 2016) is to learn the underlying structure of data. This learning process is governed by a *loss function*, denoted as $\mathcal{L}(\theta)$, where θ is the set of parameters (or weights) defining a neural network. The loss function measures the difference between the outputs of a neural network and ground-truth values. In this way, the loss reflects how good (or bad) the current weights are at making correct predictions and how to adjust these weights during training. Given the important role that the loss function plays during learning, examining it with respect to a neural network's weights—by visualizing the so-called *loss landscape*—can provide valuable insights into both network architecture and learning dynamics (Goodfellow et al., 2014; Im et al., 2016; Li et al., 2018; Yao et al., 2020; Martin and Mahoney, 2021; Martin et al., 2021; Yang et al., 2022b, 2021; Zhou et al., 2023). Indeed, the loss landscape has been essential for understanding certain aspects of deep learning, including, but not limited to, test accuracy, robustness in transfer learning (Djolonga et al., 2021), robustness to out-of-distribution detection (Yang et al., 2022a), robustness to adversarial attack (Kurakin et al., 2016), and generalizability (Cha et al., 2021). In addition, the loss landscape has been characterized in the context of scientific machine learning, e.g., to understand why different

physics-informed architectures and loss functions are often brittle, exhibiting failure modes, and are hard to optimize (Krishnapriyan et al., 2021).

Despite its promise and appeal, *loss landscape visualization* is a complex and often bespoke process. Indeed, exploring and extracting insights from a loss landscape—which is inherently high-dimensional, with as many dimensions as the number of parameters in the model—is challenging to do, especially when trying to visualize directly on a two-dimensional screen. Goodfellow et al. (2014) proposed a random-direction-based approach, where model parameters are interpolated along a one-dimensional path to see how the loss changes. In a later work by Im et al. (2016), an extension of this method was introduced, which involves projecting the loss landscape onto a two-dimensional space using barycentric interpolation between triplets of points and bilinear interpolation between quartets of points. Li et al. (2018) continued improving the resolution of loss landscapes by introducing filter-wise normalization to remove the scaling effects incurred by previous approaches. A more sophisticated approach to visualizing the loss landscape leverages the Hessian to define more relevant directions along which the model can be interpolated. More recently, Yao et al. (2020) used the top two Hessian eigenvectors as directions, thereby capturing more important changes in the underlying loss landscape. While various methods have been proposed, most applications have limited sampling to just one or two directions. Importantly, by restricting the sampling of loss landscapes to two dimensions, whether it be using random or Hessian-based directions, we ignore potentially informative information captured by additional dimensions (e.g., the eigenvectors associated with the third or fourth top eigenvalues of the Hessian matrix).

Towards characterizing higher dimensional loss landscapes, here we take inspiration from topological data analysis. Specifically, we use a merge tree to encode the critical points of a k-dimensional neural network loss landscape, and we represent the merge tree as a topological landscape profile. The merge tree allows us to capture important features in an arbitrary dimensional loss landscape, and using the topological landscape profile, we are able to re-represent this information in two dimensions. We demonstrate the utility of our new topological landscape profile representation by exploring higher dimensional loss landscapes, i.e., sampling along more directions and representing these higher dimensional subspaces as topological landscape profiles. This approach allows us to extract more information from the additional dimensions we consider. While our approach technically can work with arbitrary dimensional loss landscapes, in practice we are limited by sampling. As such, here we limit ourselves to three and four-dimensional loss landscapes.

We demonstrate the versatility of our new topological profile representations of loss landscapes and our complementary visualization tool through several use case scenarios. Through these examples, we show the many different ways our tool can be used to extract insights about neural network models based on our topological landscape profiles and by comparing loss landscapes across different hyperparameters. In doing so, we also provide new insights into how loss landscapes vary across distinct hyperparameter spaces, finding that (1) the topology of loss landscapes is simpler for better-performing models, and (2) this topology is often less consistent near transitions from low to high model performance.

2. Background

2.1. Topological Data Analysis

Topological data analysis (TDA) aims to reveal the global underlying structure of data. TDA is particularly useful for studying high-dimensional data or functions, where direct visualization (in two or three dimensions) is inherently not possible. We leverage ideas and algorithms from TDA to study the global structure of the loss function—that is, the shape of the so-called loss landscape. Much of TDA is based on the more general idea of "connectedness." In the context of a loss function, we are interested in the number of minima (i.e., unique sets of parameters for which the loss is locally minimized) and how "prominent" they are (i.e., measuring how many other sets of neighboring parameters have a higher loss than the parameter set that minimizes the loss function). Such information can be obtained from a persistence diagram (i.e., captured by the zero-dimensional persistent homology) and the so-called merge tree.

A *merge tree* (Carr et al., 2003; Heine et al., 2016) tracks connected components of sub-level sets $L^{-}(v) = \{x \in D; x \leq v\}$ as a threshold, v, is increased. The merge tree encodes changes in the loss landscape as nodes in a tree-like structure. The local minima are represented by degree-one nodes, which are connected to other local minima through a single saddle point. The saddle points connecting different minima are represented by degree-three nodes (each connecting two local minima and one other saddle point). Loss functions often display many shallow local minima with low barriers (i.e., the value difference between the minima and connecting saddle point is small) corresponding to "short-lived" connected components that merge quickly with other connected components.

In our work, we use the merge tree to extract the underlying structure of a loss landscape. We then use this extracted information to construct our topological landscape profile representations. Since the merge tree can be computed for an arbitrary dimensional loss landscape, we can use it to construct our representation for higher dimensional loss landscapes, which would otherwise be difficult to visualize.

2.2. Topological Landscape Profiles

To enable the visualization of higher-dimensional loss landscapes, we introduce a new topological landscape profile representation that captures the minima and saddle points encoded by merge trees. This work builds upon Oesterling et al. (2013), who first introduced the idea of representing high-dimensional data clusters (and their nesting) as hills (and their spatial proximity) in a landscape, where the height, width, and shape of these hills correspond to the coherence, size, and stability of the cluster, respectively. To construct the landscape profile, a merge tree is first computed based on the data to encode the distribution (or density) of points. This merge tree is then used to construct the landscape profile, by representing maxima in the merge tree as hills in the landscape, where the size and shape of each hill are determined by characteristics like persistence and the number of points along the corresponding branch. In the context of loss functions, we are more interested in minima than maxima, so here we introduce a new version of this topological landscape profile, using the metaphor of valleys (or basins) rather than hills.

3. Methods

To construct our new topological landscape profile representations, we build on traditional loss landscape sampling approaches and leverage tools from TDA to capture the underlying shape (or topology) of the sampled loss landscapes. First, we select d vectors ($d \leq n$) to define a d-dimensional subspace (Figure 1.1), where n is the dimension of model weights. This subspace can be represented by a d-dimensional loss cube, where each point corresponds to a specific set of parameters. We can therefore calculate the loss for a set of points sampled from this d-dimensional subspace. The sampled points can be represented as an unstructured grid (Figure 1.2). We compute a merge tree to capture the topology of the k-dimensional loss landscape (Figure 1.3). We construct our topological landscape profile based on this merge tree (Figure 1.4). We then visualize these new representations using a complementary interactive visualization tool that makes it easy to explore and compare different topological landscape profiles. In this section, we go into more detail about each of these steps.

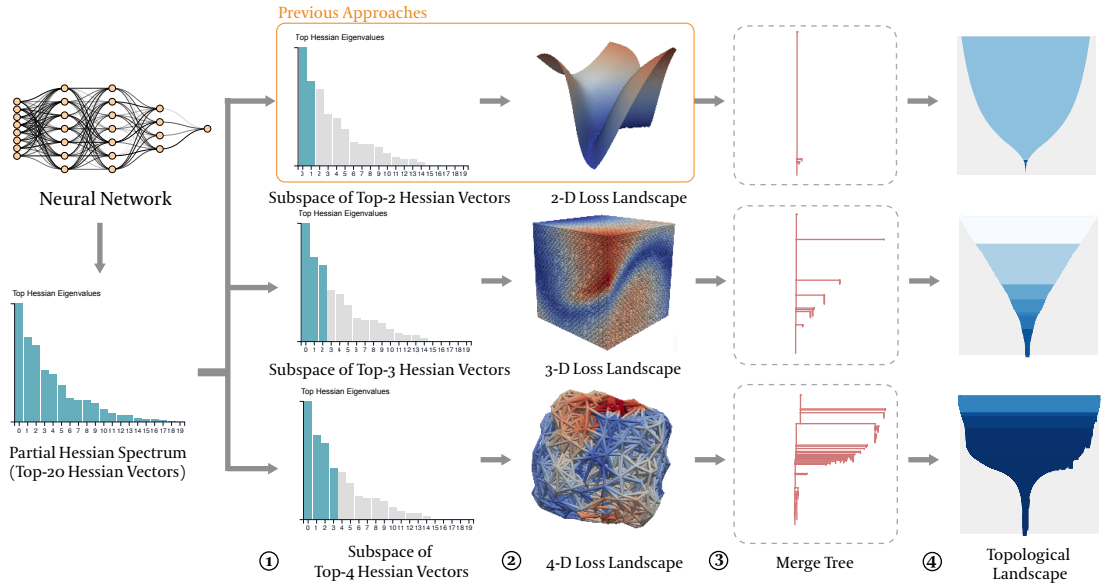

Figure 1: Our topological landscape profiles enable the visualization of higher dimensional loss landscapes by capturing their underlying shape (or topology). Here we show loss landscapes based on the top k Hessian eigenvectors. See Section 3 for details.

3.1. Loss Landscape Construction and Representation

In this work, we limit our analysis to Hessian-based loss landscapes. We calculate the top d Hessian eigenvectors using PyHessian (Yao et al., 2020) (Figure 1.1) and then sample along the subspace spanned by these directions (Figure 1.2). The idea is that by using the eigenvectors associated with the top d largest eigenvalues, we can visualize the most significant local loss fluctuations for a given model. Given the d orthogonal directions, we generalize the approach taken by Li et al. (2018) by expanding the subspace beyond two

dimensions. Formally, we perturb trained model parameters along the d directions and evaluate the loss $\mathcal{L}$ as follows:

$$f(\alpha_1 ... \alpha_d) = \mathcal{L}(\theta^* + \Sigma_{i=1}^{d}(\alpha_i \delta_i)), \tag{1}$$

where $\alpha_1 ... \alpha_d$ are the coordinates in the d-dimensional subspace, δ_i is the i-th direction in that subspace, and θ^* is the original model. As such, each coordinate will correspond to a point that contains a computed loss value, and the entire point set forms an d-dimensional loss landscape.

Given a d-dimensional loss landscape, we can represent the sampled points as an unstructured grid, where each vertex in the grid is associated with d coordinates and a scalar loss value. Before we can characterize how the loss changes throughout the landscape (i.e., as parameters are perturbed from one vertex to the next), we need to define the spatial proximity (or connectivity) of vertices in the grid based on the similarity of their coordinates. Here we use a scalable, approximate nearest neighbor algorithm to construct a neighborhood graph representation of the loss landscape (Dong et al., 2011). The *neighborhood graph*, proposed by Jaromczyk and Toussaint (1992), of a dataset D is a graph $G = (D, E)$ where two points u and v are connected by an edge $(u, v) \in E$ if they are *similar*. Here we focus on the k-nearest neighbor graph, where each point is connected to the k most similar points. We also use a *symmetric* version of this graph, where points are only considered neighbors if each point is a neighbor of the other. In this case, an edge (u, v) is pruned from the graph if u is not one of the k nearest points to v, or vice versa. We note that this approach involves selecting an appropriate value for the k parameter. Here we use $k = 4 \times d$, such that the connectivity is similar to the spatial proximity of pixels in an image (i.e., each pixel having $k = 8$ neighbors, corresponding to the left, right, top, bottom, and all four corners).

3.2. Topological Structures and Landscape Profiles

After defining the subspace and computing the loss landscape, we perform topological data analysis to extract and summarize the most important features. In this work, we use a merge tree to extract key information from the loss landscape, which we then use to define our topological landscape profile. We compute the merge tree for each loss landscape using Topology ToolKit (TTK), developed by Bin Masood et al. (2021).

Given a merge tree, we then construct the topological landscape profile using the method proposed by Oesterling et al. (2013). In this representation, each branch (in the merge tree) ending in a local minimum is represented by a basin (in the landscape profile), and each sub-branch ending in a saddle point is represented as a sub-basin, below which other basins are placed. In either case, each basin (or sub-basin) is represented by a set of rectangles encoding the cumulative size of the branch (or sub-branch), from bottom to top, such that the top of the basin is as wide as the number of points found along the corresponding branch in the merge tree.

We introduce this topological landscape profile representation of loss functions to effectively capture more information from higher-dimensional loss landscapes, in such a way that can still be visualized. While this topological representation and the merge tree used to create it both capture important features of the high-dimensional space, it also discards

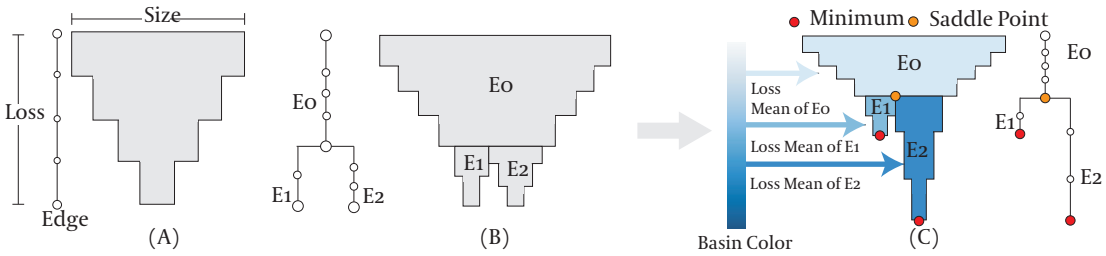

Figure 2: Representing the merge tree as a topological landscape profile. In (A) we show a single basin corresponding to a merge tree with a single branch, and in (B) we show multiple basins corresponding to multiple branches. In (C) we color the basins based on their average loss.

some important information by design. Here, we reincorporate some of this discarded information back into our representation, for example, by using the loss values to color the different basins. As shown in Figure 2.C, we compute the average loss across the points in each basin, and we use darker blues to represent lower average loss values. Thus, deeper basins are represented by a darker blue color, evoking the idea of deeper ocean depths. In addition to coloring the topological landscape profile, we also annotate the basins with the critical points, including saddle points (orange dots) and minima (red dots). Interestingly, the distribution (or density) of saddle points and minima reflects local characteristics of the loss landscape, such as locally sharp or locally flat.

4. Experiments

4.1. Visualizing Different Physical Constraints

In our first experiment, we look at a set of physics-informed neural network (PINN) models trained to solve simple convection problems (Krishnapriyan et al., 2021). Here we aim to investigate the PINN's soft regularization and how it helps (or fails to help) the optimizer find an optimal solution to a seemingly simple convection problem. We show how the shape and complexity of our topological landscape profiles change as a physical "wave speed" parameter is increased and the PINN fails to solve this seemingly simple physical problem. Specifically, we consider the one-dimensional convection problem, a hyperbolic partial differential equation that is commonly used to model transport phenomena:

$$\frac{\partial u}{\partial t} + \beta \frac{\partial u}{\partial x} = 0, \;\; x \in \Omega, \;\; t \in [0, T] \tag{2}$$

$$u(x, 0) = h(x), \;\; x \in \Omega \tag{3}$$

where β is the convection coefficient and $h(x)$ is the initial condition. The general loss function for this problem is

$$L(\theta) = \frac{1}{N_u} \sum_{i=1}^{N_u} (\hat{u} - u_0^i)^2 + \frac{1}{N_f} \sum_{i=1}^{N_f} \lambda_i (\frac{\partial \hat{u}}{\partial t} + \beta \frac{\partial \hat{u}}{\partial x})^2 + L_B \tag{4}$$

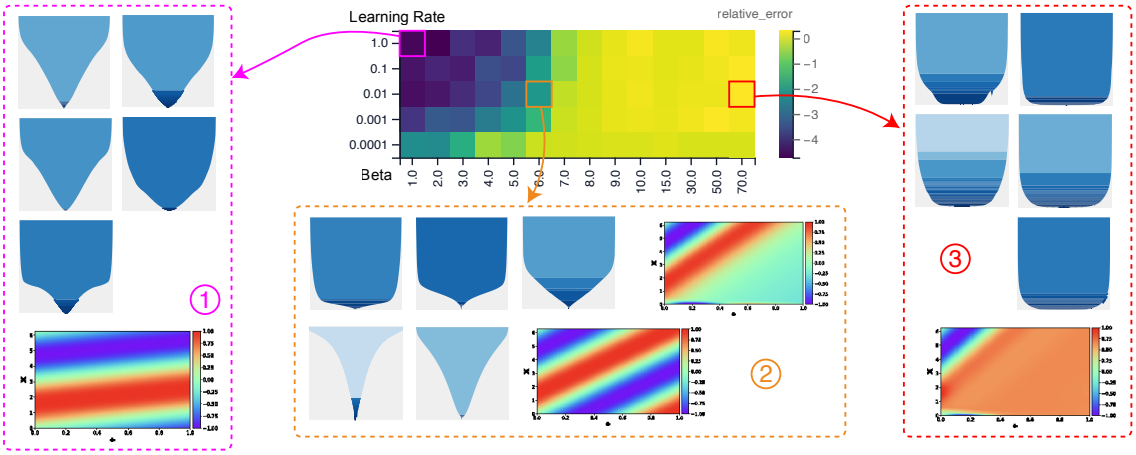

Figure 3: Analyzing the loss function of a physics-informed neural network (PINN) trained to solve simple physical convection problems. See Section 4.1 for details.

where $\hat{u} = NN(\theta, x, t)$ is the output of the NN, and L_B is the boundary loss. While increasing the physical wave speed parameter, β, should not necessarily make this a harder problem to solve, it can make PINN models harder to train. Interestingly, Krishnapriyan et al. (2021) related these failure modes to changes in the corresponding loss landscape, showing that it becomes increasingly complicated, such that optimizing the model becomes increasingly difficult. Here we explore these failure modes in more detail using three-dimensional and four-dimensional Hessian-based loss landscapes, finding more variability in the shape of loss landscapes near the transition between high and low-performing models.

In Figure 3, we show a heat map corresponding to the average relative error across different values of the physical wave speed parameter and across different learning rates. Interestingly, we observe that the error increases with this physical parameter, but more slowly for higher learning rates. The smallest learning rate displays higher error rates even for smaller values of the physical parameter. When looking at the loss landscapes, we observe consistently more funnel-like loss landscapes for the smaller values of β, corresponding to lower error (Figure 3.1). In contrast, we observe a consistently more bowl-like loss landscape for the larger values of β, corresponding to higher error (Figure 3.3). The funnel-like landscapes likely correspond to when the PINN models find a physically reasonable solution, albeit constrained to a smaller space of solutions by the physical wave speed parameter. In other words, since the solution is constrained by the physical parameter, perturbing the model results in a faster increase in the loss, given that the physical problem is no longer satisfied. In contrast, the more bowl-like landscapes correspond to the failure to find a reasonable solution, such that perturbing the model does not immediately change the already high loss. Note, the landscapes corresponding to these failure modes also include more saddle points and are otherwise more complex.

To verify that these representations are stable across different model initializations, we show five different landscapes for each hyperparameter configuration, corresponding to the same model trained using different random seeds. We see the landscapes look similar across different random seeds for the low and high values of the physical wave speed parameter.

Moreover, we observe more variation in the loss landscapes near the transition from low to high error (Figure 3.2). This suggests that while the error starts to increase near the transition, only some of the models are failing whereas other may be finding physically reasonable solutions, as indicated by the funnel-like loss landscapes.

In Figure A.5 we compare the topological landscape profiles based on three- and four-dimensional loss landscapes. An important insight here is that, in higher dimensions, we observe many more critical points and that the basins in the much spikier landscape can be mapped back to the wider basins in the topological landscape profiles based on the three-dimensional loss landscapes. Overall the global shape of the topological landscape profile looks similar when comparing the same random seed. Moreover, this highlights an important feature of our new representations—the ability to visualize higher dimensional loss landscapes, i.e., sampling along more than just one or two dimensions.

4.2. Visualizing Loss Landscapes Over Training

In our second experiment, we explore how loss landscapes change throughout training and across different learning rates. To do this, we study UNet models with a learnable CRF-RNN layer (Avaylon et al., 2022) trained on the Oxford-IIIT Pet dataset (Parkhi et al., 2012). We trained the models using five different random seeds across seven different learning rates for 30 epochs. For each checkpoint, we computed two-dimensional loss landscapes based on the top two Hessian eigenvectors. The model was perturbed using a distance of 0.01 and layerwise normalization was adopted (Li et al., 2018).

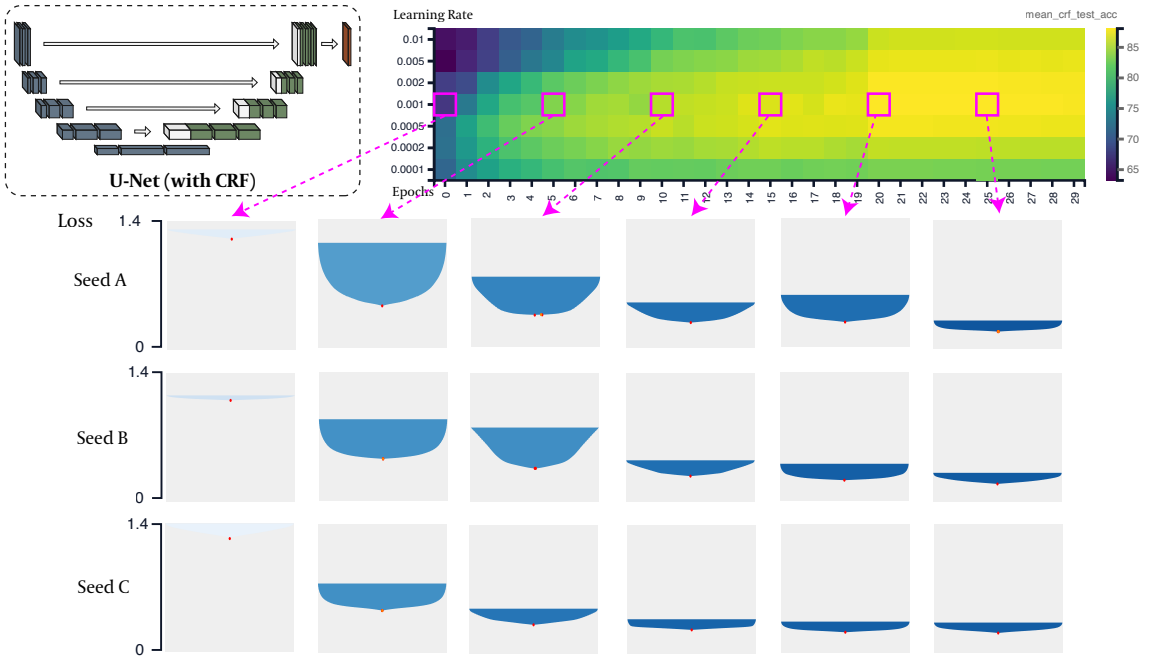

Figure 4: Loss landscapes over training for UNet models with a CRF layer trained on the Oxford-IIIT Pet dataset. See Section 4.2 for details.

In Figure 4 and Figure B.6, we show the same heat map corresponding to average test accuracy over training and across different learning rates. We observe that the test accuracy improves over training, with some variation across the different learning rates. In Figure 4, we consider how the loss landscape changes over training. When looking at the loss landscapes for three different random seeds, after zooming in, we observe an initially shallow loss landscape but with the global minimum at a much higher loss compared to the end of training. As training proceeds, we see that the global minimum becomes lower, but the basin itself becomes deeper with the edges remaining at much higher loss values. As the global minimum continues to drop, we also observe additional flattening of the basin, such that all points have a much lower loss compared to the beginning of training. Interestingly, the flat basin at a much higher loss corresponds to a phase of learning where perturbing the model in any one direction doesn't really increase the already high loss. After five epochs, the much deeper basin reflects a less stable model, where perturbing the model results in relatively higher loss. As training proceeds, we observe a flattening of the basin, which means the model becomes more stable, as perturbations result in smaller changes in loss. In Figure B.6, we consider how the loss landscape changes across different learning rates. When looking at the loss landscapes for three different random seeds, after zooming in, we observe consistent variation in the depth and shape of the loss landscape as the learning rate is varied. This variation is also reflected in the test accuracy scores shown in the heat map. Interestingly, we observe deeper basins when the learning rate is too small or too big, indicating that the trained models are less stable compared to those with shallower basins.

5. Conclusion and Future Work

In this paper, we introduced a new topological landscape profile representation of neural network loss landscapes. To demonstrate the many different ways this new representation of loss landscapes can be used, we explored several different machine learning examples, including image segmentation (e.g., UNet-CRF) and scientific machine learning (e.g., PINNs). Along the way, we provided new insights into how loss landscapes vary across distinct hyperparameter spaces, finding that the topology of the loss landscape is simpler for better-performing models and that this topology is more variable near transitions from low to high model performance. Moreover, by using a merge tree to extract the most important features from a computed loss landscape, we are able to construct a new representation encoding these features. By separating this new representation from the original space in which the loss landscape was sampled, our approach opens up the door to visualizing higher-dimensional loss landscapes.

While we only explore up to four dimensions here, our approach can be extended to much higher dimensional spaces. The limiting factor is sampling, which requires exponentially many more resources as the number of dimensions increases. However, future advances towards more efficient sampling could be combined with our current approach to reveal the higher dimensional structure of loss functions. Complementary advances in sampling more global loss landscapes (combining multiple independently trained models) could also be combined with our tool. In that case, we would expect to see more distinct basins in our topological landscape profiles.

References

Matthew Avaylon, Robbie Sadre, Zhe Bai, and Talita Perciano. Adaptable deep learning and probabilistic graphical model system for semantic segmentation. *Advances in Artificial Intelligence and Machine Learning*, 02:288–302, 2022. doi: http://dx.doi.org/10.54364/aaiml.2022.1119.

Talha Bin Masood, Joseph Budin, Martin Falk, Guillaume Favelier, Christoph Garth, Charles Gueunet, Pierre Guillou, Lutz Hofmann, Petar Hristov, Adhitya Kamakshidasan, Christopher Kappe, Pavol Klacansky, Patrick Laurin, Joshua Levine, Jonas Lukasczyk, Daisuke Sakurai, Maxime Soler, Peter Steneteg, Julien Tierny, Will Usher, Jules Vidal, and Michal Wozniak. An overview of the Topology ToolKit. In *Topological Methods in Data Analysis and Visualization VI*, pages 327–342. Springer International Publishing, 2021. doi: 10.1007/978-3-030-83500-2_16.

Hamish Carr, Jack Snoeyink, and Ulrike Axen. Computing contour trees in all dimensions. *Computational Geometry*, 24(2):75–94, 2003. ISSN 0925-7721. doi: https://doi.org/10.1016/S0925-7721(02)00093-7. URL https://www.sciencedirect.com/science/article/pii/S0925772102000937. Special Issue on the Fourth CGC Workshop on Computational Geometry.

Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems*, 34:22405–22418, 2021. doi: https://doi.org/10.48550/arXiv.2102.08604.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. doi: https://doi.org/10.48550/arXiv.1810.04805.

Josip Djolonga, Jessica Yung, Michael Tschannen, Rob Romijnders, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Matthias Minderer, Alexander D'Amour, Dan Moldovan, Gelly Sylvain, Neil Houlsby, Xiaohua Zhai, and Mario Lucic. On robustness and transferability of convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16458–16468, 2021. doi: https://doi.org/10.48550/arXiv.2007.08558.

Wei Dong, Charikar Moses, and Kai Li. Efficient k-nearest neighbor graph construction for generic similarity measures. In *Proceedings of the 20th international conference on World wide web*, pages 577–586, 2011.

Ian J Goodfellow, Oriol Vinyals, and Andrew M Saxe. Qualitatively characterizing neural network optimization problems. *arXiv preprint arXiv:1412.6544*, 2014. doi: https://doi.org/10.48550/arXiv.1412.6544.

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*, pages 770–778, 2016. doi: https://doi.org/10.48550/arXiv.1512.03385.

Christian Heine, Heike Leitte, Mario Hlawitschka, Federico Iuricich, Leila De Floriani, Gerik Scheuermann, Hans Hagen, and Christoph Garth. A Survey of Topology-based Methods in Visualization. *Computer Graphics Forum*, 35(3):643–667, 2016. doi: https://doi.org/10.1111/cgf.12933.

Daniel Jiwoong Im, Michael Tao, and Kristin Branson. An empirical analysis of the optimization of deep network loss surfaces. *arXiv preprint arXiv:1612.04010*, 2016. doi: https://doi.org/10.48550/arXiv.1612.04010.

Jerzy W Jaromczyk and Godfried T Toussaint. Relative neighborhood graphs and their relatives. *Proceedings of the IEEE*, 80(9):1502–1517, 1992. doi: 10.1109/5.163414.

Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael W Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Advances in Neural Information Processing Systems*, 34:26548–26560, 2021. doi: https://doi.org/10.48550/arXiv.2109.01050.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. doi: https://dl.acm.org/doi/10.1145/3065386.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016. doi: https://doi.org/10.48550/arXiv.1611.01236.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 2018. doi: https://doi.org/10.48550/arXiv.1712.09913.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*, 2019. doi: https://doi.org/10.48550/arXiv.1907.11692.

Charles H Martin and Michael W Mahoney. Implicit Self-Regularization in Deep Neural Networks: Evidence from Random Matrix Theory and Implications for Learning. *The Journal of Machine Learning Research*, 22(1):7479–7551, 2021. doi: https://doi.org/10.48550/arXiv.1810.01075.

Charles H Martin, Tongsu Peng, and Michael W Mahoney. Predicting trends in the quality of state-of-the-art neural networks without access to training or testing data. *Nature Communications*, 12(1):4122, 2021. doi: https://doi.org/10.1038/s41467-021-24025-8.

Patrick Oesterling, Christian Heine, Gunther H. Weber, and Gerik Scheuermann. Visualizing nd point clouds as topological landscape profiles to guide local data analysis. *IEEE Transactions on Visualization and Computer Graphics*, 19(3):514–526, 2013. doi: 10.1109/TVCG.2012.120.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on Computer Vision and Pattern Recognition*, pages 3498–3505. IEEE, 2012. doi: 10.1109/CVPR.2012.6248092.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. doi: https://doi.org/10.48550/arXiv.1409.1556.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 2017. doi: https://doi.org/10.48550/arXiv.1706.03762.

Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized Out-of-Distribution Detection: A Survey, 2022a.

Yaoqing Yang, Liam Hodgkinson, Ryan Theisen, Joe Zou, Joseph E Gonzalez, Kannan Ramchandran, and Michael W Mahoney. Taxonomizing local versus global structure in neural network loss landscapes. *Advances in Neural Information Processing Systems*, pages 18722–18733, 2021. doi: https://doi.org/10.48550/arXiv.2107.11228.

Yaoqing Yang, Ryan Theisen, Liam Hodgkinson, Joseph E Gonzalez, Kannan Ramchandran, Charles H Martin, and Michael W Mahoney. Evaluating natural language processing models with generalization metrics that do not need access to any training or testing data. *arXiv preprint arXiv:2202.02842*, 2022b. doi: https://doi.org/10.48550/arXiv.2202.02842.

Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. PyHessian: Neural networks through the lens of the hessian. In *2020 IEEE international conference on big data (Big data)*, pages 581–590. IEEE, 2020. doi: https://doi.org/10.48550/arXiv.1912.07145.

Yefan Zhou, Yaoqing Yang, Arin Chang, and Michael W Mahoney. A three-regime model of network pruning. In *International Conference on Machine Learning*, pages 42790–42809. PMLR, 2023.

Appendix A. Visualizing Different Physical Constraints

In Figure A.5, we show topological landscape profiles for two different random initializations (from left to right) of a physics-informed neural network (PINN). Note, that the landscapes look different for the different random initializations because we are looking at a model corresponding to the transition from low to high error. Overall the global shape of the topological landscape profile looks similar when comparing the same random seed across three- and four-dimensional loss landscapes. In four dimensions, we observe many more critical points and that the basins in the much spikier landscape can be mapped back to the wider basins in the topological landscape profiles based on the three-dimensional loss landscapes. These visualizations also highlight an important feature of our topological landscape profile representations—the ability to visualize higher dimensional loss landscapes.

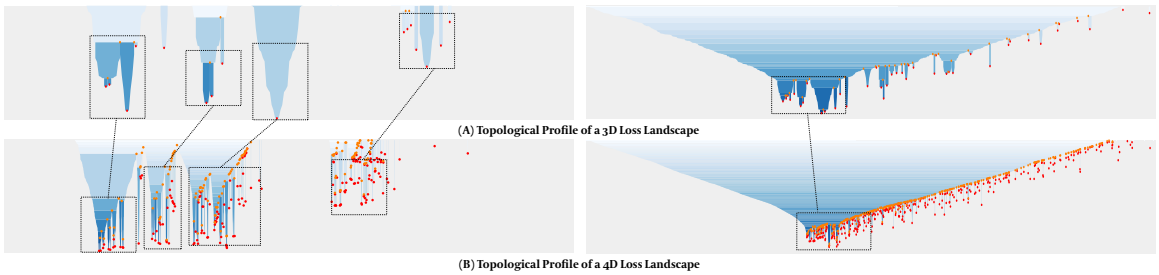

Figure 5: Comparing topological landscape profiles based on (A) three-dimensional and (B) four-dimensional loss landscapes. See Section 4.1 for details.

Appendix B. Visualizing Loss Landscapes Over Training

In Figure B.6, we show the loss landscape changes across different learning rates. When looking at the loss landscapes for three different random seeds, after zooming in, we observe consistent variation in the depth and shape of the loss landscape as the learning rate is varied. See Section 4.2 for details.

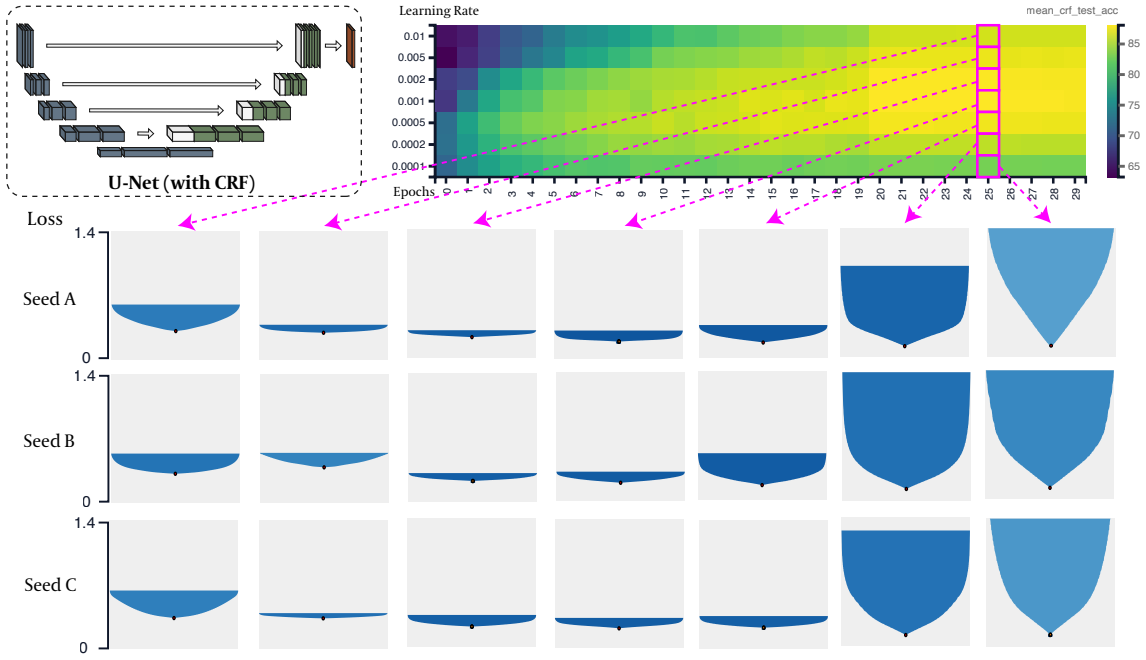

Figure 6: Loss landscapes across learning rates for UNet models with a CRF layer trained on the Oxford-IIIT Pet dataset. See Section 4.2 for details.