Robust Convergence of Loss Landscapes through Distributional Averaging

Nikita Kiselev
MIPT
Moscow, Russia
kiselev.ns@phystech.edu

Vladislav Meshkov
MIPT
Moscow, Russia
meshkov.vs@phystech.edu

Andrey Grabovoy
MIPT
Moscow, Russia
grabovoy.av@phystech.edu

Abstract

Understanding how a neural network's loss landscape evolves with dataset size is essential for identifying sufficient training data. Prior analyses of this problem have typically been local, focusing on second-order expansions around a single optimum and bounding convergence through Hessian properties. While such studies clarify convergence rates, they provide only a pointwise view of stability. In this paper, we extend the framework to a distributional paradigm. Instead of analyzing convergence at one optimum, we evaluate it in expectation over a parameter distribution. This approach captures how entire neighborhoods of the loss landscape stabilize as additional samples are added. We focus on Gaussian distributions centered at local minima and employ Monte Carlo sampling to estimate convergence in practice. Theoretically, we show that distributional convergence exhibits the same asymptotic rate as the local case, while offering a more robust picture of stability. Empirical studies on image classification tasks confirm these predictions and highlight how architectural choices such as normalization, dropout, and network depth influence convergence. Our results broaden local convergence analyses into a distributional setting, providing stronger guarantees and practical tools for characterizing dataset sufficiency.

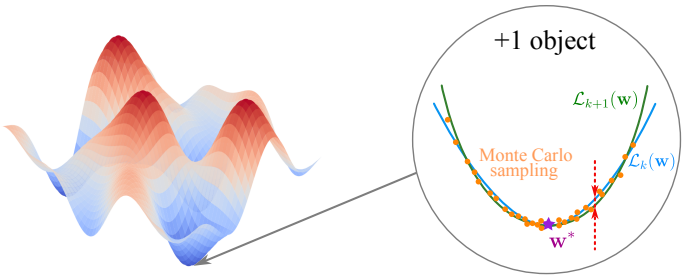
Keywords: Neural networks, Loss landscape, Convergence, Gaussian sampling, Monte Carlo estimation, Dataset size.

1 Introduction

Neural networks achieve strong performance across domains such as image classification, language modeling, and generative modeling. As datasets and models scale, accuracy improves but at growing computational cost. This raises a fundamental question: *how large must the dataset be before additional samples cease to meaningfully alter the optimization landscape?* Answering this is central both for theoretical understanding of generalization and for practical training decisions.

One way to study this problem is through the geometry of the loss landscape. Curvature near minima reflects generalization, stability, and optimization dynamics [1, 2]. Flatter minima are linked to robustness, while sharper ones often overfit [3]. The Hessian, encoding second-order curvature, is central in such analyses [4, 5]. Yet little is known about how these properties evolve with increasing dataset size.

Preprint. Under review.



(a) Loss function landscape (b) Losses difference

Figure 1: **Overview of our observations.** Part (a) shows the loss function landscape, which is a surface in the parameters space. Part (b) shows the losses difference. We propose the Δ_k metric, which in practice is a Monte Carlo estimation of a squared losses difference over a Gaussian distribution near the local minima.

In earlier work we analyzed *loss landscape convergence with sample size* [6]. Using second-order Taylor expansions at local optima, we showed that the difference between losses trained on k and $k+1$ samples vanishes as $k \to \infty$, with rates governed by the Hessian spectral norm. This clarified how local curvature stabilizes with more data, but remained a *pointwise* view tied to a single optimum $\mathbf{w}^*$.

Here we extend that analysis to a *distributional* framework. Instead of one point, convergence is measured in expectation under a parameter distribution $p(\mathbf{w})$. This captures the stability of neighborhoods around optima and allows Monte Carlo estimation in practice. Gaussian neighborhoods offer a natural choice: they probe surrounding curvature while remaining tractable.

We prove that distributional convergence matches the quadratic rate of the pointwise case, while providing a stronger notion of stability by averaging across directions. Experiments on image classification validate the theory and reveal how batch normalization, dropout, and depth influence convergence.

Contributions. Our contributions can be summarized as follows.

- Generalization of pointwise convergence analyses to a distributional framework over arbitrary parameter distributions $p(\mathbf{w})$.
- Introduction of Gaussian neighborhoods around local optima and a Monte Carlo estimator for convergence.
- Theoretical guarantees showing distributional convergence decays at the same asymptotic rate as the local case, but with greater robustness.
- Empirical evaluation across architectures, highlighting how normalization, dropout, and depth reshape landscape stability.

Outline. Section 2 reviews related work. Section 3 introduces notation. Section 4 presents the framework and analysis. Section 5 reports experiments. Section 6 discusses implications and Section 7 concludes.

2 Related Work

Geometry of neural network loss landscapes. The geometry of neural network loss functions has been a central theme in deep learning theory. Early studies analyzed the abundance of local minima and saddle points in high-dimensional spaces [7], and later works revealed connectivity of minima through nearly flat valleys [8–10]. The flatness or sharpness of minima is often linked to generalization, with flatter regions associated with more robust solutions [3, 11]. Visualization approaches [12] and analyses of double descent and training instabilities [13, 14] further highlight how architectural choices, initialization, or optimization interact with the underlying geometry. More recent work has also examined landscape properties in transformers and vision models [15, 16]. These studies provide qualitative and structural insights into geometry, but do not directly address how landscapes converge with increasing dataset size.

Hessian spectra and curvature analyses. The Hessian matrix is a powerful tool for quantifying curvature and stability in trained networks. Empirical studies show that its spectrum typically consists of a large bulk near zero and a few outliers that capture informative directions [4, 5, 17, 18]. Outlier structure has been tied to class-level geometry [19], while random matrix and maximum-entropy models explain power-law tails [20–22]. Decomposition methods, such as Kronecker factorizations, provide scalable approximations and structural interpretations [23–25]. Beyond structural analysis, curvature has been directly connected to generalization and robustness [2, 26–28]. Our earlier studies also fall into this direction: in [6] we introduced a Hessian-based framework for analyzing convergence in fully connected networks, and in [29] extended it to convolutional architectures. However, most Hessian-based analyses remain local, focusing on pointwise curvature near minima rather than distributional stability across neighborhoods.

Dataset size, stability, and scaling perspectives. A complementary line of work investigates how dataset size affects both generalization and optimization. Empirically, larger datasets tend to produce flatter landscapes and more stable solutions [30], while scaling law studies explore systematic trade-offs among model size, dataset size, and compute [31]. Algorithmic stability theory [32–34] provides generalization guarantees by bounding sensitivity to changes in the training set, offering a perspective closely related to our notion of landscape convergence. Connections to kernel regimes [35, 36] also motivate quadratic and distributional approximations. Beyond these general perspectives, recent work has begun to study sample size determination directly. In particular, authors introduced several methods for linear models based on likelihood bootstrapping [37] and parameters posterior distributions proximity [38]. Yet despite this broad literature, the question of how loss landscapes themselves converge as dataset size increases has remained largely unexplored. Our contribution addresses this gap by formalizing convergence in expectation under parameter distributions, thereby connecting geometric insights from Hessian spectra with stability-style notions from learning theory.

3 Preliminaries

3.1 Notation

We denote a neural network with parameters $\mathbf{w} \in \mathbb{R}^N$ by $f_{\mathbf{w}}$. Let the training dataset be $\mathfrak{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{D}$ of size D, where $\mathbf{x}_i \in \mathcal{X}$ and $\mathbf{y}_i \in \mathcal{Y}$. For a subset $\mathfrak{D}_k = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{k}$ with $k < D$, the empirical loss is

$$\mathcal{L}_k(\mathbf{w}) = \frac{1}{k}\sum_{i=1}^{k} \ell(f_{\mathbf{w}}(\mathbf{x}_i), \mathbf{y}_i) = \frac{1}{k}\sum_{i=1}^{k} \ell_i(\mathbf{w}),$$

where $\ell_i(\mathbf{w})$ is the per-sample loss. The difference in empirical loss when adding the $(k+1)$-th example is

$$\mathcal{L}_{k+1}(\mathbf{w}) - \mathcal{L}_k(\mathbf{w}) = \frac{1}{k+1}\Big(\ell_{k+1}(\mathbf{w}) - \mathcal{L}_k(\mathbf{w})\Big).$$

To quantify landscape change, we define the squared-difference criterion:

$$\Delta_{k+1} = \mathbb{E}_{p(\mathbf{w})}\Big[\big(\mathcal{L}_{k+1}(\mathbf{w}) - \mathcal{L}_k(\mathbf{w})\big)^2\Big],$$

where $p(\mathbf{w})$ is a weighting distribution over parameter space (e.g., Gaussian around a local minimum). Thus, in practice, Δ_{k+1} can be estimated via Monte Carlo:

$$\Delta_{k+1} \approx \frac{1}{B}\sum_{t=1}^{B} \big(\mathcal{L}_{k+1}(\mathbf{w}^{(t)}) - \mathcal{L}_k(\mathbf{w}^{(t)})\big)^2, \quad \mathbf{w}^{(t)} \sim p(\mathbf{w}).$$

3.2 Assumptions

Our theoretical analysis relies on the following assumptions, standard in loss landscape studies:

Assumption 1 (Smoothness)**.** *Each per-sample loss $\ell_i(\mathbf{w})$ is twice continuously differentiable.*

Assumption 2 (Existence of local minima)**.** *For each k, there exists at least one local optimum $\mathbf{w}_k^*$ such that $\nabla\mathcal{L}_k(\mathbf{w}_k^*) = \mathbf{0}$.*

Assumption 3 (Bounded Hessians)**.** *The Hessians at local minima are bounded:*

$$\|\mathbf{H}_i(\mathbf{w}_k^*)\|_2 \leq M_{\mathbf{H}}, \quad \forall i, k,$$

where $\mathbf{H}_i$ *is the Hessian of* ℓ_i*.*

Assumption 4 (Sampling distribution)**.** *Parameter vectors* $\mathbf{w}$ *for evaluating* Δ_k *are drawn from a distribution* $p(\mathbf{w})$ *with mean* $\mathbf{m}$ *and covariance* $\mathbf{\Sigma}$*. A typical choice is Gaussian:* $p(\mathbf{w}) = \mathcal{N}(\mathbf{m}, \sigma^2\mathbf{I})$.

These assumptions allow us to characterize how empirical loss surfaces converge as the dataset size increases, and to derive rates for Δ_k in terms of spectral and Frobenius norm bounds on Hessian differences.

4 Method

We now develop a formal analysis of loss landscape convergence. Our goal is to characterize how the difference between empirical losses $\mathcal{L}_k$ and $\mathcal{L}_{k+1}$ evolves as the dataset size increases, and to provide explicit convergence rates for the squared difference criterion Δ_{k+1}. Unlike previous work that focused on a single optimum, we extend the framework by integrating over a distribution of parameters. This enables a more robust notion of convergence, since it captures how neighborhoods of optima evolve rather than just isolated points.

4.1 Loss landscape change near local optima

Let $\mathbf{w}_k^*$ denote a local minimum of $\mathcal{L}_k$, i.e. $\nabla\mathcal{L}_k(\mathbf{w}_k^*) = 0$. A second-order Taylor expansion around $\mathbf{w}_k^*$ gives

$$\mathcal{L}_k(\mathbf{w}) \approx \mathcal{L}_k(\mathbf{w}_k^*) + \tfrac{1}{2}(\mathbf{w} - \mathbf{w}_k^*)^\top \mathbf{H}^{(k)}(\mathbf{w}_k^*)(\mathbf{w} - \mathbf{w}_k^*),$$

with $\mathbf{H}^{(k)}(\mathbf{w}_k^*)$ denoting the Hessian of $\mathcal{L}_k$ at $\mathbf{w}_k^*$. Applying the same expansion for $\mathcal{L}_{k+1}$ and subtracting yields

$$\mathcal{L}_{k+1}(\mathbf{w}) - \mathcal{L}_k(\mathbf{w}) \approx \mathcal{L}_{k+1}(\mathbf{w}_k^*) - \mathcal{L}_k(\mathbf{w}_k^*) + \tfrac{1}{2}(\mathbf{w} - \mathbf{w}_k^*)^\top \mathbf{A}_k(\mathbf{w} - \mathbf{w}_k^*),$$

where

$$\mathbf{A}_k = \mathbf{H}^{(k+1)}(\mathbf{w}_k^*) - \mathbf{H}^{(k)}(\mathbf{w}_k^*).$$

This local expansion forms the foundation of earlier pointwise analyses. In our setting, however, it also serves as the starting point for distributional averaging: by substituting these expansions into Δ_{k+1}, we capture how curvature fluctuations propagate under $p(\mathbf{w})$.

4.2 Squared difference criterion

The convergence measure defined in Section 3 can thus be expressed as

$$\Delta_{k+1} = \mathbb{E}_{p(\mathbf{w})}\Big[\big(\mathcal{L}_{k+1}(\mathbf{w}) - \mathcal{L}_k(\mathbf{w})\big)^2\Big].$$

Expanding the square and using standard variance decomposition gives

$$\Delta_{k+1} = \mathbb{D}_{p(\mathbf{w})}\big[\mathcal{L}_{k+1}(\mathbf{w}) - \mathcal{L}_k(\mathbf{w})\big] + \Big(\mathbb{E}_{p(\mathbf{w})}[\mathcal{L}_{k+1}(\mathbf{w}) - \mathcal{L}_k(\mathbf{w})]\Big)^2.$$

For a Gaussian sampling distribution $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}_k^*, \sigma^2\mathbf{I})$ centered at $\mathbf{w}_k^*$, and under the natural assumption that $\mathcal{L}_{k+1}(\mathbf{w}_k^*) \approx \mathcal{L}_k(\mathbf{w}_k^*)$, the mean term vanishes. We then obtain the simplified form

$$\Delta_{k+1} = \frac{\sigma^4}{4}\Big(\mathrm{Tr}^2(\mathbf{A}_k) + 2\mathrm{Tr}(\mathbf{A}_k^2)\Big).$$

The Monte Carlo estimator provides a scalable way to approximate this expectation in practice, making it possible to track Δ_k across architectures and datasets. Importantly, this aligns the theory with empirical visualization tools: we can evaluate Δ_k in the same neighborhoods where landscape plots are drawn.

4.3 Bounding the trace terms

The traces can be controlled by spectral properties of $\mathbf{A}_k$. By Cauchy–Schwarz,

$$\mathrm{Tr}(\mathbf{A}_k) \le \sqrt{N}\,\|\mathbf{A}_k\|_F, \qquad \mathrm{Tr}(\mathbf{A}_k) \le N\|\mathbf{A}_k\|_2,$$

while

$$\mathrm{Tr}(\mathbf{A}_k^2) = \|\mathbf{A}_k\|_F^2 \le N\|\mathbf{A}_k\|_2^2.$$

Thus,

$$\mathrm{Tr}^2(\mathbf{A}_k) + 2\mathrm{Tr}(\mathbf{A}_k^2) \;\le\; N^2\|\mathbf{A}_k\|_2^2 + 2N\|\mathbf{A}_k\|_2^2.$$

Lemma 1 (Trace bound)**.** *For any symmetric matrix $\mathbf{A}_k \in \mathbb{R}^{N\times N}$,*

$$\mathrm{Tr}^2(\mathbf{A}_k) + 2\mathrm{Tr}(\mathbf{A}_k^2) \;\le\; N(N+2)\|\mathbf{A}_k\|_2^2.$$

This bound ensures that the distributional convergence measure remains controlled by well-understood spectral quantities, connecting our generalization directly back to Hessian-based analyses of prior work.

4.4 Bounding the Hessian difference

By construction,

$$\mathbf{A}_k = \frac{1}{k+1}\Big(\mathbf{H}_{k+1}(\mathbf{w}_k^*) - \frac{1}{k}\sum_{i=1}^{k}\mathbf{H}_i(\mathbf{w}_k^*)\Big).$$

If $\|\mathbf{H}_i(\mathbf{w}_k^*)\|_2 \le M_{\mathbf{H}}$ for all i, then

$$\|\mathbf{A}_k\|_2^2 \le \frac{2M_{\mathbf{H}}^2}{(k+1)^2}.$$

Intuitively, this expresses that as the dataset grows, the variability in curvature introduced by adding one new example becomes negligible, which formalizes the geometric notion of "sample sufficiency."

4.5 Convergence rate

Substituting the above bounds into the expression for Δ_{k+1} yields

$$\Delta_{k+1} \le \frac{\sigma^4}{4}\,N(N+2)\,\|\mathbf{A}_k\|_2^2 \le \frac{\sigma^4 N(N+2)M_{\mathbf{H}}^2}{2(k+1)^2}.$$

Theorem 1 (Convergence rate of loss landscapes)**.** *Under Assumptions 1–4, the squared difference between empirical loss landscapes at sample sizes k and $k+1$ satisfies*

$$\Delta_{k+1} = \mathcal{O}\Big(\frac{1}{(k+1)^2}\Big).$$

Thus, as the dataset grows, loss landscapes converge quadratically fast in sample size.

This result shows that extending from pointwise to distributional averaging does not change the asymptotic convergence rate: both decay quadratically with dataset size. However, the distributional setting provides a much stronger guarantee, since it demonstrates that entire neighborhoods around minima stabilize, not just single points.

5 Experiments

We now validate our theoretical framework through empirical studies. Our experiments address three main questions: (1) How does the loss landscape geometry evolve with architectural choices such as dropout, batch normalization, and network depth? (2) How does the proposed convergence measure Δ_k behave in practice? (3) Can Δ_k provide a geometric interpretation of sample size sufficiency analogous to scaling law studies?

5.1 Experimental Setup

We use the CIFAR-10 dataset (50k training and 10k test images, 10 balanced classes). Networks are trained for classification with cross-entropy loss. Unless otherwise stated, optimization is performed with Adam (learning rate 3×10^{-4}, batch size 64).

To visualize high-dimensional loss landscapes, we project onto 2D random subspaces. Specifically, given an optimum $\mathbf{w}^*$, we sample two random normalized directions $\mathbf{u}, \mathbf{v}$ with the same parameter norm ratios as $\mathbf{w}^*$, and evaluate

$$f(\alpha, \beta) = \mathcal{L}(\mathbf{w}^* + \alpha\mathbf{u} + \beta\mathbf{v}).$$

This follows the common visualization protocol of Li et al. [12].

5.2 Effects of Dropout and Batch Normalization on the Loss Landscape

In the following experiments, 4-layer convolutional networks were trained with 64 channels in the hidden layers, a kernel size of 3, and ReLU activations. ReLU and batch normalization were added depending on the experiment. Figure 2 demonstrates how the loss function landscape of convolutional networks changes when dropout and/or batch normalization are added to the neural network. Although the example is quite toy-like, it clearly illustrates the transformation of the landscape. Specifically, one can see that the introduction of batch normalization leads to a substantial "narrowing" of the landscape, while adding dropout afterward causes it to widen again.

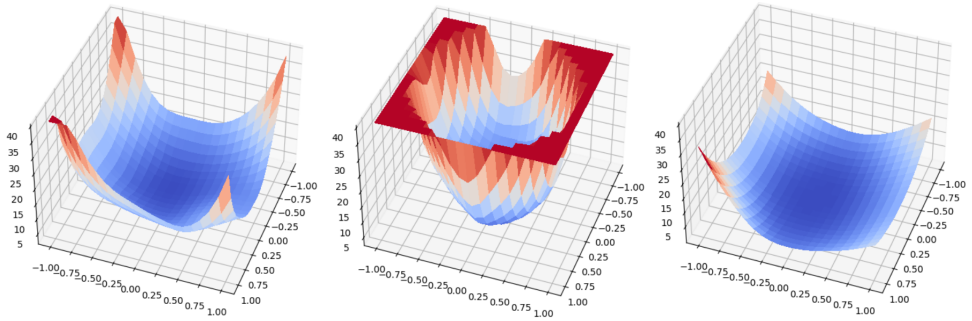


Figure 2: Convolutional networks (4 layers). **Left:** with dropout + batch norm, **center:** without dropout, **right:** without both. Batch normalization narrows the landscape, dropout widens it, while removing both produces unstable minima.

The next experiments involved training a 4-layer fully connected neural network with 128-unit hidden layers using ReLU activations. Batch normalization and dropout were incorporated depending on the specific task configuration. Figure 3 presents a similar experiment but with a fully connected network instead of a convolutional one, where the resulting loss landscape exhibits different properties. In particular, the landscape transforms less predictably and varies differently across different parameter space directions. When both batch normalization and dropout are applied simultaneously, the landscape becomes notably wider — an effect more pronounced than in convolutional networks.

5.3 Loss Landscape Dynamics Near Optima with Varying Convolutional Blocks

The next three experiments involve training a convolutional neural network with ReLU activations and 64 channels in the hidden layers, while varying the number of layers. Figure 4 demonstrates how the loss landscape evolves in the vicinity of the optimum as the number of convolutional blocks changes. The visualization reveals that increasing network depth leads to a significantly wider landscape and more complex geometry. While shallow networks (2 layers) exhibit nearly parabolic shapes, deeper architectures produce more complex structure.

5.4 Delta calculation method hyperparameters influence

This experiment employed a fully connected network with 4 hidden layers of 128 units each, ReLU activation, along with batch normalization and dropout.

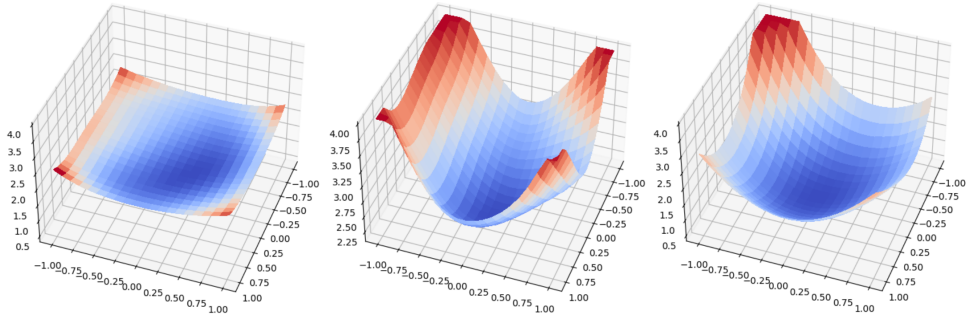


Figure 3: Fully connected networks (4 hidden layers, 128 units). **Left:** with dropout + batch norm, **center:** without dropout, **right:** without both. Unlike convolutional models, the combination of dropout and batch norm produces the widest landscapes.

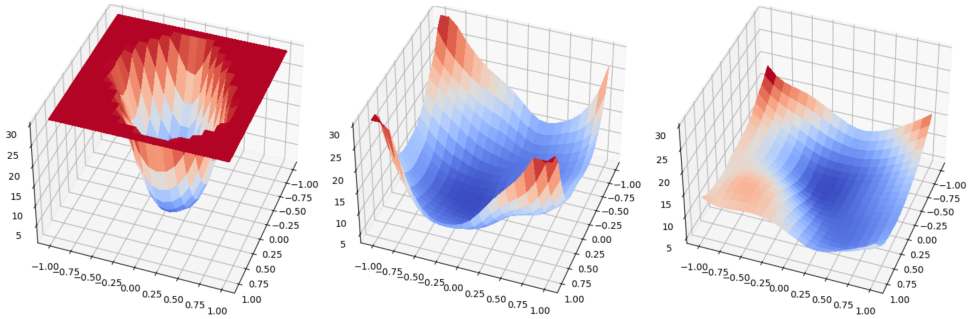


Figure 4: This image shows the loss landscape of convolutional models with varying numbers of layers ($2 \rightarrow 4 \rightarrow 6$), illustrating how the loss landscape changes with increasing network depth.

In Figure 5, we observe how Δ_k depends on the parameter σ in the distribution $p(\mathbf{w})$. Notably, the dependence is significant, indicating that this parameter strongly affects the locality of the estimated landscape change.

In the same figure (**right**), we observe the dependence on the dim used for landscape evaluation. Notably, the results exhibit high noise levels, suggesting this parameter likely does not require fine-tuning. Since visualization tests confirmed reproducibility across different random seeds, we subsequently use dim=2 for convenience. Figure 6 demonstrates the variation of $(\mathcal{L}_{k+1} - \mathcal{L}_k)^2 p(\mathbf{w})$ (the integrand quantity), for different values of k. Specifically, we observe that as k increases while keeping $p(\mathbf{w})$ parameters constant, local behavior becomes increasingly dominant.

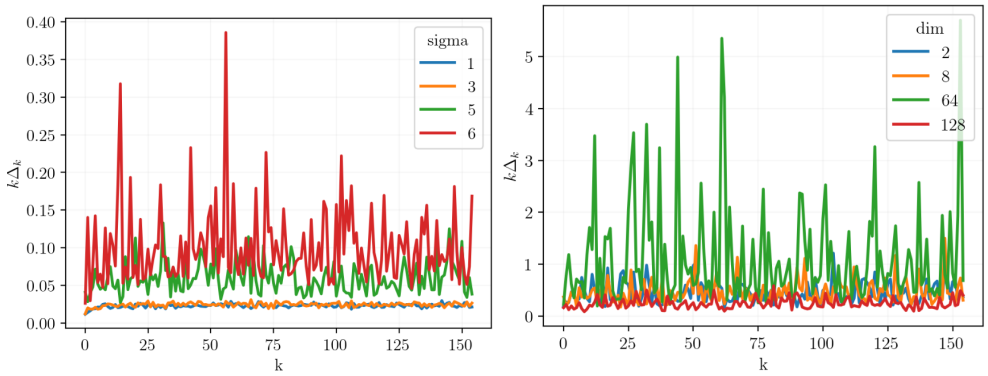
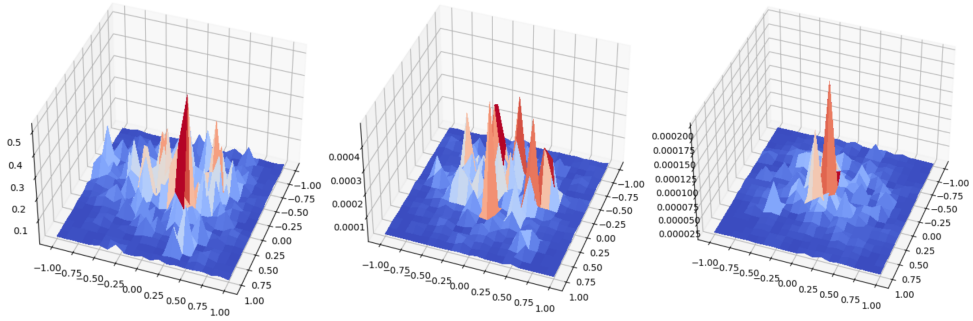


Figure 5: The **left** plot demonstrates how the estimated quantity varies with changes in the distribution parameter (specifically, the variance σ^2) of $p(w)$. The **right** plot shows its dependence on the number of random directions (dim).



(a) $(\mathcal{L}_1 - \mathcal{L}_2)^2 p(\mathbf{w})$ (b) $(\mathcal{L}_{50} - \mathcal{L}_{49})^2 p(\mathbf{w})$ (c) $(\mathcal{L}_{100} - \mathcal{L}_{101})^2 p(\mathbf{w})$

Figure 6: This figure shows how the $\big(\mathcal{L}_{k+1}(\mathbf{w}) - \mathcal{L}_k(\mathbf{w})\big)p(\mathbf{w})$ changes as the sample size used to estimate it varies.

6 Discussion

Our results show that empirical loss landscapes converge quadratically fast with dataset size, as captured by the metric Δ_k. The experiments confirm this trend: as k increases, $(\mathcal{L}_{k+1} - \mathcal{L}_k)^2 p(\mathbf{w})$ smooths out and local fluctuations diminish. We now discuss the implications and boundaries of these findings.

Geometric notion of sample sufficiency. Performance metrics such as accuracy or cross-entropy are standard proxies for dataset sufficiency. Our framework suggests a complementary, geometric criterion: once Δ_k falls below a threshold, the loss surface becomes stable enough that additional samples no longer significantly alter optimization geometry. This view connects data sizing to measurable landscape properties, offering a tool for reasoning about when a dataset is "large enough."

Impact of architectural choices. We observed that architectural elements reshape convergence behavior. Batch normalization narrows valleys in convolutional networks, dropout widens them, and deeper models introduce more complex curvature patterns. In fully connected architectures, the interplay of dropout and normalization differs qualitatively. These findings highlight that convergence speed and stability are not solely functions of dataset size but also of design decisions, underlining the need to analyze architecture and data jointly.

Assumptions and limitations. Our theoretical guarantees rely on quadratic approximations around local minima, Gaussian parameter distributions, and bounded Hessians. Real networks may violate these assumptions: multimodal loss landscapes and heavy-tailed curvature spectra are common, especially in large-scale transformers and language models. Furthermore, our experiments were confined to medium-scale image classification. Assessing whether distributional convergence persists in large architectures, alternative data modalities, and non-Gaussian parameter distributions remains an open challenge.

7 Conclusion

We developed a distributional framework for studying loss landscape convergence, showing that neighborhoods around optima stabilize at the same quadratic rate as pointwise analyses. This provides a stronger and more practical notion of stability, with empirical confirmation across several architectures. Looking ahead, extending this framework to large-scale models, exploring non-Gaussian parameter distributions, and connecting convergence measures to generalization bounds represent promising directions for future work.

References

[1] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30, 2017.

[2] Stanislav Fort and Surya Ganguli. Emergent properties of the local geometry of neural loss landscapes. *arXiv preprint arXiv:1910.05929*, 2019.

[3] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9:1–42, 01 1997.

[4] Levent Sagun, Utku Evci, V Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

[5] Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In *International Conference on Machine Learning*, pages 2232–2241. PMLR, 2019.

[6] N. S. Kiselev and A. V. Grabovoy. Unraveling the hessian: A key to smooth convergence in loss function landscapes. *Doklady Mathematics*, 110(1):S49–S61, 2024.

[7] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Artificial intelligence and statistics*, pages 192–204. PMLR, 2015.

[8] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in neural information processing systems*, 31, 2018.

[9] Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In *International conference on machine learning*, pages 1309–1318. PMLR, 2018.

[10] Quynh Nguyen and Matthias Hein. The loss surface of deep and wide neural networks. In *International conference on machine learning*, pages 2603–2612. PMLR, 2017.

[11] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pages 1019–1028. PMLR, 2017.

[12] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.

[13] Sidak Pal Singh, Aurelien Lucchi, Thomas Hofmann, and Bernhard Schölkopf. Phenomenology of double descent in finite-width neural networks. *arXiv preprint arXiv:2203.07337*, 2022.

[14] Lawrence Wang and Stephen Roberts. The instabilities of large learning rate training: a loss landscape view. *arXiv preprint arXiv:2307.11948*, 2023.

[15] Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pre-training or strong data augmentations. *arXiv preprint arXiv:2106.01548*, 2021.

[16] Youngwan Lee, Jeffrey Ryan Willette, Jonghee Kim, and Sung Ju Hwang. Visualizing the loss landscape of self-supervised vision transformer. *arXiv preprint arXiv:2405.18042*, 2024.

[17] Vardan Papyan. The full spectrum of deepnet hessians at scale: Dynamics with sgd training and sample size. *arXiv preprint arXiv:1811.07062*, 2018.

[18] Vardan Papyan. Traces of class/cross-class structure pervade deep learning spectra. *Journal of Machine Learning Research*, 21(252):1–64, 2020.

[19] Vardan Papyan. Measurements of three-level hierarchical structure in the outliers in the spectrum of deepnet hessians. *arXiv preprint arXiv:1901.08244*, 2019.

[20] Jeffrey Pennington, Samuel Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. *Advances in neural information processing systems*, 30, 2017.

[21] Zeke Xie, Qian-Yuan Tang, Yunfeng Cai, Mingming Sun, and Ping Li. On the power-law hessian spectrums in deep learning. *arXiv preprint arXiv:2201.13011*, 2022.

[22] Zhenyu Liao and Michael W Mahoney. Hessian eigenspectra of more realistic nonlinear models. *Advances in Neural Information Processing Systems*, 34:20104–20117, 2021.

[23] Yikai Wu, Xingyu Zhu, Chenwei Wu, Annie Wang, and Rong Ge. Dissecting hessian: Understanding common structure of hessian in neural networks. *arXiv preprint arXiv:2010.04261*, 2020.

[24] Sidak Pal Singh, Gregor Bachmann, and Thomas Hofmann. Analytic insights into structure and rank of neural network hessian maps. *Advances in Neural Information Processing Systems*, 34: 23914–23927, 2021.

[25] Sidak Pal Singh, Thomas Hofmann, and Bernhard Schölkopf. The hessian perspective into the nature of convolutional neural networks. *arXiv preprint arXiv:2305.09088*, 2023.

[26] Haotian Ju, Dongyue Li, and Hongyang R Zhang. Robust fine-tuning of deep neural networks with hessian-based generalization guarantees. In *International conference on machine learning*, pages 10431–10461. PMLR, 2022.

[27] Van-Anh Nguyen, Quyen Tran, Tuan Truong, Thanh-Toan Do, Dinh Phung, and Trung Le. Agnostic sharpness-aware minimization. *arXiv preprint arXiv:2406.07107*, 2024.

[28] Lachlan Ewen MacDonald, Jack Valmadre, and Simon Lucey. On progressive sharpening, flat minima and generalisation. *arXiv preprint arXiv:2305.14683*, 2023.

[29] Vladislav Meshkov, Nikita Kiselev, and Andrey Grabovoy. Convnets landscape convergence: Hessian-based analysis of matricized networks. In *2024 Ivannikov Ispras Open Conference (ISPRAS)*, pages 1–10, 2024. doi: 10.1109/ISPRAS64596.2024.10899113.

[30] Lei Wu, Zhanxing Zhu, and Weinan E. Towards understanding generalization of deep learning: Perspective of loss landscapes, 2017. URL `https://arxiv.org/abs/1706.10239`.

[31] 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.

[32] Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2002.

[33] Andre Elisseeff, Theodoros Evgeniou, and Massimiliano Pontil. Stability of randomized learning algorithms. *Journal of Machine Learning Research*, 6(3):55–79, 2005.

[34] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*, pages 1225–1234. PMLR, 2016.

[35] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[36] Jaehoon Lee, Samuel Schoenholz, Jeffrey Pennington, Ben Adlam, Lechao Xiao, Roman Novak, and Jascha Sohl-Dickstein. Finite versus infinite neural networks: an empirical study. *Advances in Neural Information Processing Systems*, 33:15156–15172, 2020.

[37] N. S. Kiselev and A. V. Grabovoy. Sample size determination: Likelihood bootstrapping. *Computational Mathematics and Mathematical Physics*, 65(3):416–423, 2025. doi: 10.1134/S0965542524702002.

[38] N. S. Kiselev and A. V. Grabovoy. Sample size determination: Posterior distributions proximity. *Computational Management Science*, 22(1):1, 2025. doi: 10.1007/s10287-024-00528-9.

A Appendix: Proofs

A.1 Proof of Lemma 1 (Trace bound)

Proof. Let $\lambda_1, \ldots, \lambda_N$ be the eigenvalues of the symmetric matrix $\mathbf{A}_k$. Then

$$\mathrm{Tr}(\mathbf{A}_k) = \sum_{i=1}^{N} \lambda_i, \qquad \mathrm{Tr}(\mathbf{A}_k^2) = \sum_{i=1}^{N} \lambda_i^2.$$

By Cauchy–Schwarz,

$$\Big(\sum_{i=1}^{N} \lambda_i\Big)^2 \le N \sum_{i=1}^{N} \lambda_i^2.$$

Thus,

$$\mathrm{Tr}^2(\mathbf{A}_k) + 2\mathrm{Tr}(\mathbf{A}_k^2) \le N \sum_{i=1}^{N} \lambda_i^2 + 2 \sum_{i=1}^{N} \lambda_i^2 = (N+2) \sum_{i=1}^{N} \lambda_i^2.$$

Finally, since $\max_i |\lambda_i| = \|\mathbf{A}_k\|_2$,

$$\sum_{i=1}^{N} \lambda_i^2 \le N \|\mathbf{A}_k\|_2^2.$$

Substituting gives

$$\mathrm{Tr}^2(\mathbf{A}_k) + 2\mathrm{Tr}(\mathbf{A}_k^2) \le N(N+2)\|\mathbf{A}_k\|_2^2,$$

which proves the claim. □

A.2 Proof of Theorem 1 (Convergence rate of loss landscapes)

Proof. From the derivation in Section 4, under Gaussian sampling around a local optimum $\mathbf{w}_k^*$ we have

$$\Delta_{k+1} = \frac{\sigma^4}{4}\Big(\mathrm{Tr}^2(\mathbf{A}_k) + 2\mathrm{Tr}(\mathbf{A}_k^2)\Big).$$

Applying Lemma 1 gives

$$\Delta_{k+1} \le \frac{\sigma^4}{4} N(N+2) \, \|\mathbf{A}_k\|_2^2.$$

Now recall that

$$\mathbf{A}_k = \frac{1}{k+1}\left(\mathbf{H}_{k+1}(\mathbf{w}_k^*) - \frac{1}{k}\sum_{i=1}^{k} \mathbf{H}_i(\mathbf{w}_k^*)\right).$$

By the bounded Hessian assumption $\|\mathbf{H}_i(\mathbf{w}_k^*)\|_2 \le M_{\mathbf{H}}$ for all i, we have

$$\|\mathbf{A}_k\|_2 \le \frac{1}{k+1}\Big(\|\mathbf{H}_{k+1}(\mathbf{w}_k^*)\|_2 + \frac{1}{k}\sum_{i=1}^{k} \|\mathbf{H}_i(\mathbf{w}_k^*)\|_2\Big) \le \frac{2M_{\mathbf{H}}}{k+1}.$$

Therefore,

$$\|\mathbf{A}_k\|_2^2 \le \frac{4M_{\mathbf{H}}^2}{(k+1)^2}.$$

Substituting into the upper bound for Δ_{k+1} yields

$$\Delta_{k+1} \le \frac{\sigma^4}{4} N(N+2) \cdot \frac{4M_{\mathbf{H}}^2}{(k+1)^2} = \frac{\sigma^4 N(N+2) M_{\mathbf{H}}^2}{(k+1)^2}.$$

Thus,

$$\Delta_{k+1} = \mathcal{O}\Big(\frac{1}{(k+1)^2}\Big),$$

which proves the theorem. □

B Appendix: Experiments

In the main section, we analyzed landscapes at local optima obtained after fixed training epochs. Here we repeat the experiments but redefine $\mathbf{w}^*$ as the *validation-optimal parameters*, i.e., those corresponding to the best validation epoch (typically occurring early, between epochs 3–6). This adjustment provides a clearer view of generalization behavior, contrasting with the purely optimization-focused landscapes examined earlier.

B.1 Loss landscapes near validation optima

Setup. We trained fully connected networks with 4 hidden layers of size 128 and batch size 64. Depending on the configuration, batch normalization and dropout blocks were inserted into each layer. For convolutional networks, we fixed 4 layers with 64 channels, kernel size 3, and padding 1.

Results. Figure 7 shows that in fully connected models, batch normalization produces narrower optima, while dropout widens the valleys. In convolutional models (Figure 8), adding both batch normalization and dropout leads to sharper landscapes compared to vanilla networks. Figure 9 compares networks of varying depth (2, 4, and 6 layers), revealing that deeper networks exhibit stronger boundary effects and more complex geometry at validation-optimal points.

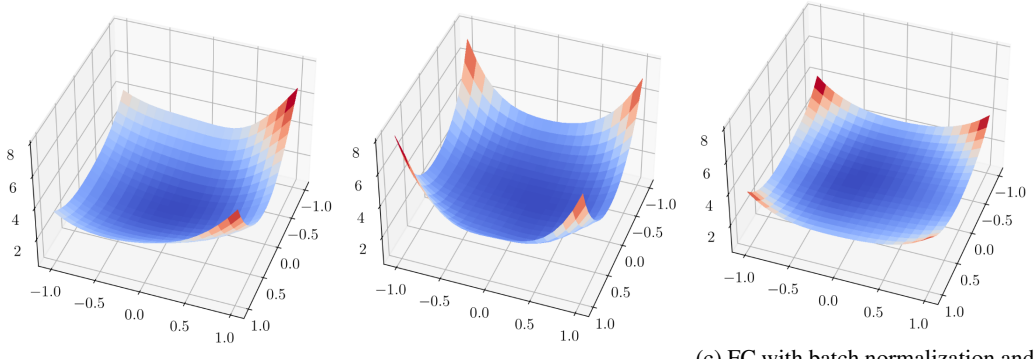
Takeaway. The structure observed in the main experiments persists when evaluating at the best validation epoch, but landscapes are generally narrower and more sensitive at the boundaries, highlighting the connection between validation performance and curvature.

B.2 Effects of hidden channel size

Setup. We trained convolutional networks with varying numbers of hidden channels while keeping other hyperparameters fixed. Landscapes were evaluated within neighborhoods of equal norm to ensure comparability across models.

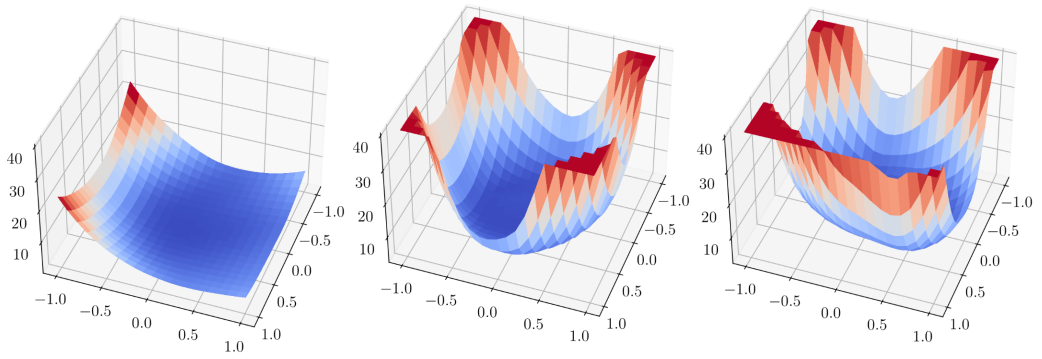
Results. As shown in Figure 10, increasing the number of channels leads to progressively wider landscapes, with geometric deformation accelerating in deeper networks. This suggests that larger representational capacity amplifies curvature effects, though fixed-norm comparisons introduce dimensionality biases.

Takeaway. Channel count strongly influences curvature geometry at validation optima, reinforcing the role of network capacity in shaping landscape stability.



(a) vanilla FC (b) FC with batch normalization (c) FC with batch normalization and dropout

Figure 7: Fully connected networks near validation-optimal parameters. Batch normalization narrows the optimum, dropout widens it.

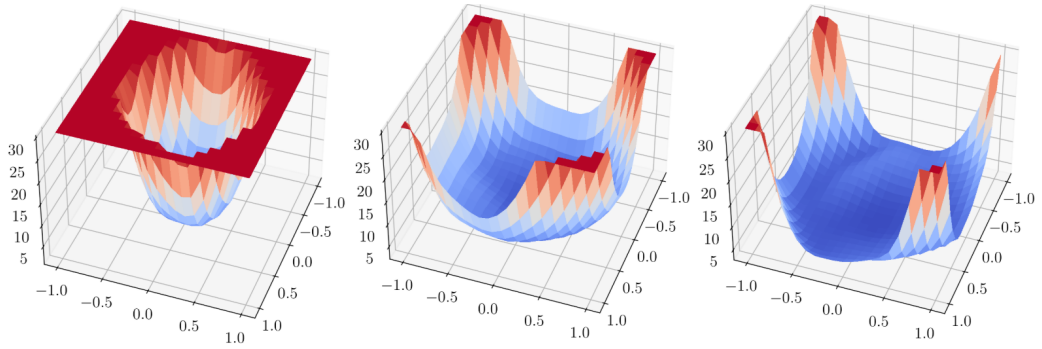


(a) vanilla ConvNet

(b) ConvNet with batch normalization

(c) ConvNet with batch normalization and dropout

Figure 8: Convolutional networks near validation-optimal parameters. Batch normalization and dropout sharpen the landscape relative to the vanilla case.

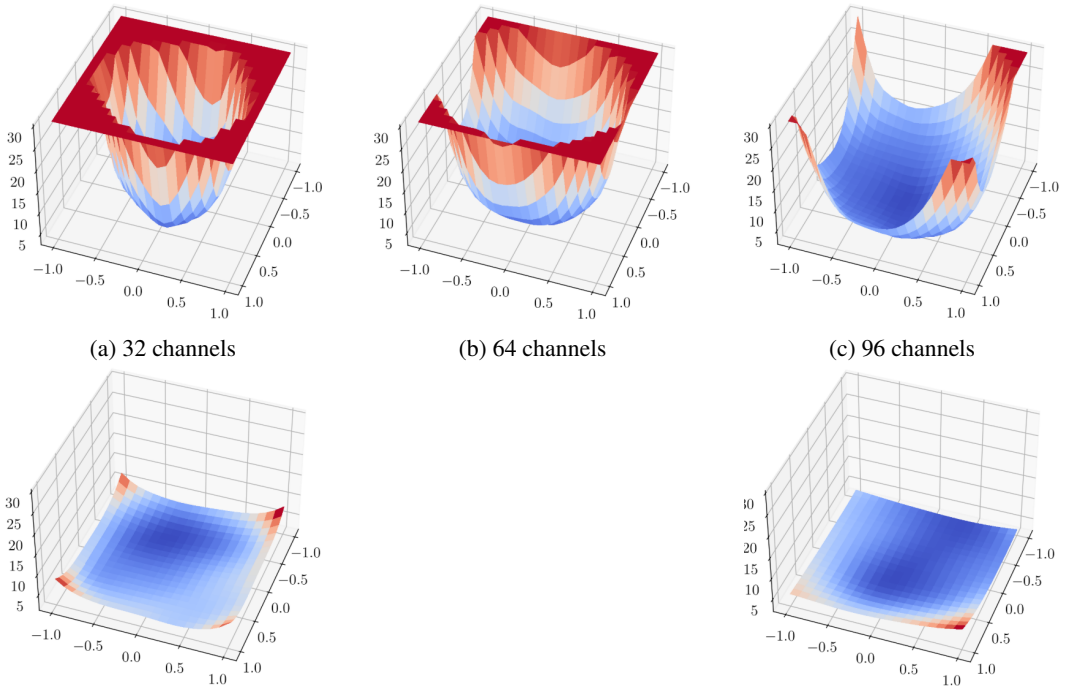


(a) 2 layers

(b) 4 layers

(c) 6 layers

Figure 9: Loss landscapes of convolutional models with different depths near validation optima. Deeper networks exhibit stronger boundary effects and more complex curvature.



(a) 32 channels (b) 64 channels (c) 96 channels

(d) 128 channels (e) 256 channels

Figure 10: Effect of hidden channel size on loss landscapes at validation optima. Wider networks show progressively broader and more deformed landscapes.