Connecting Loss-Landscape Geometry to Generalization Across MLP Depths*

Owen Skriloff
Departments of Mathematics, Department of Chemical and Biological Engineering
Tufts University
Medford, MA
owen.skriloff@tufts.edu

***Abstract*—The relationship between neural network depth and generalization can be illuminated by examining the geometry of the training-loss landscape. In this study, fully-connected multilayer perceptrons of depths 1–10 were trained on a subset of MNIST with 25 random restarts each. We measured both mean-squared training loss and Frobenius-norm curvature at each converged minimum, visualized their joint distributions, and distilled them into two scalar metrics: the RMS radius of parameter clusters and an orthogonal-regression trade-off weight λ_d. Both metrics identified an optimal window around depth four, where minima are flattest, lowest in loss, and most tightly clustered. Finally, a globally optimized curvature penalty $\lambda_G \approx 0.42$ applied to a combined proxy objective reduced the mean absolute error to validation loss by over 15%, demonstrating that Hessian-based regularization can systematically bridge training objectives and generalization across architectures.**

Code: https://github.com/oskril01/loss-curvature-mlp.

***Index Terms*—multilayer perceptron, loss landscape, Hessian curvature, Pareto front, generalization, curvature regularization**

I. Introduction

Deep neural networks often generalize well despite having orders of magnitude more parameters than training examples. Classical capacity measures such as VC-dimension would predict severe overfitting without strong regularization, yet empirical evidence contradicts this expectation [1]. This discrepancy has driven the development of alternative predictors—among them geometric properties of the loss landscape—that more accurately forecast generalization in overparameterized models.

Whereas VC-dimension bounds estimate generalization from model capacity and data, geometric analysis of the loss surface offers a complementary perspective. In particular, minima with low curvature (flat minima) have been empirically linked to improved test performance [2]. For example, small-batch SGD tends to converge to broader, flatter basins that generalize better than the sharper minima found by large-batch training [3]. Large-scale evaluations place curvature-based measures among the most reliable predictors of test error [4], while reparameterization concerns warn against naive sharpness definitions [5]. Recent PAC-Bayesian bounds further formalize these insights, deriving provable generalization guarantees in terms of curvature [6].

Despite extensive research on generalization metrics, the role of network architecture remains underexplored. Depth in multilayer perceptrons exhibits a non-monotonic effect on test error, yet recent work attributes this solely to representation compression rather than examining loss-landscape geometry [7]. Here, we directly investigate how MLP depth influences the curvature of the loss surface and, in turn, generalization performance. By integrating an architecture-invariant curvature metric with training loss in a multi-objective framework, we map how the geometry of local minima evolves as depth varies.

To systematically capture the trade-off between training loss and curvature, we frame them as competing objectives in a Pareto-analysis. For each network depth, we perform multiple random restarts to sample a distribution of local minima, then construct the Pareto front that defines the efficient balance between low loss and low curvature. This multi-objective approach—combined with multistart sampling—allows us to statistically characterize how flatness and fit co-vary across depths.

The remainder of the paper is structured as follows. Section II reviews methods for estimating loss-surface curvature in high dimensions, quantifying the proximity of local minima clusters, and applying multi-objective analysis and details our experimental protocol for varying MLP depth and measuring loss and curvature. Section III presents the empirical results and discussion. Section IV concludes with a summary of finding, implications and directions for future work.

II. Methods

A. Neural Network Architecture and Training

Consider fully-connected feed-forward networks (multilayer perceptrons, MLPs) of varying depth d and a fixed hidden-layer width w. Each defines a map $f_{MLP}^{(d)} : \mathbb{R}^{784} \longrightarrow \mathbb{R}^{10}$ taking a flattened 28x28 MNIST image to a 10-dimensional output. Depths $d \in \{1, 2, \ldots, 10\}$ are explored with $w = 48$ nodes per hidden layer, and use the ReLU activation function after every linear transformation.

To limit runtime, MLPs were trained on a subset of MNIST comprising $n_{\text{train}} = 5000$ examples and validated on $n_{\text{val}} = 1000$ held-out images. All pixel intensities are rescaled to [0,1], and labels are encoded as one-hot vectors. Both training

*Submitted for EE 143 Final Project, May 6th, 2025 and validation losses via mean-squared error were measured, for example

$$\ell_{\text{train}}(\theta) = \frac{1}{n_{\text{train}}} \sum_{i=1}^{n_{\text{train}}} \left\| f_\theta(x_i) - y_i \right\|_2^2,$$

with an analogous definition for $\ell_{\text{val}}(\theta)$.

For each depth d, $R = 25$ independent restarts $\theta^{(0)} \sim \mathcal{N}(0, I)$ were performed to sample multiple local minima of the training loss. To estimate parameters θ^* for each restart, optimization with Adam (learning rate $\alpha = 10^{-3}$, default momentum parameters) minimizes the loss function and halts when $\left|\ell_{\text{train}}^{(t)} - \ell_{\text{train}}^{(t-1)}\right| < 10^{-6}$ or after 10^5 iterations. From each depth d and restart i, the following are recorded:

- The converged parameters $\theta^*_{d,i}$
- The final training loss $\ell_{train}(\theta^*_{d,i})$
- The validation loss $\ell_{val}(\theta^*_{d,i})$
- The local curvature $\kappa_{d,i}$ (defined in Geometric Metric section)

These collections $\{\theta^*_{d,i}, \ell_{train}(\theta^*_{d,i}), \ell_{val}(\theta^*_{d,i}), \kappa_{d,i}\}_{i=1}^{R}$ for each depth d serve as the dataset for all subsequent analyses.

B. Geometric Metrics

Local curvature of the loss landscape at each minimizer θ^* was quantified by $\kappa = \|H(\theta^*)\|_F$, $H(\theta^*) = \nabla^2 \ell(\theta^*)$, where $\|\cdot\|_F$ denotes the Frobenius norm. Recall that for symmetric matrix $H \in \mathbb{R}^{n \times n}$,

$$\|H\|_F = \sqrt{\operatorname{tr}(H^T H)} = \sqrt{\sum_{k=1}^{n} \lambda_k^2}$$

where $\{\lambda_k\}$ are the eigenvalues of H. This choice yields a single scalar measuring curvature magnitude in all parameter directions. This approach follows prior work which shows the effectiveness of the Frobenius norm of the Hessian as a flatness metric [8].

The Frobenius norm of the Hessian was chosen for two reasons. First, summing the squares of all eigenvalues captures the total "sharpness" of the loss surface around θ^*, avoids cancellation between positive and negative curvature, and does not rely on any single principal direction. Second, although the full Hessian spectrum contains richer information, $\|H\|_F$ provides a single scalar summary of that spectrum and can be efficiently approximated. In particular, Hutchinson's stochastic trace estimator replaced an $\mathcal{O}(n^3)$ eigen-decomposition with an $\mathcal{O}(mn^2)$ procedure [9],

$$\operatorname{tr}(H^2) \approx \frac{1}{m} \sum_{j=1}^{m} \|H\, v_j\|_2^2,$$

which uses m Rademacher probes. A probe count of $m = 45$ was used to yield an approximate relative error of $\mathcal{O}(1/\sqrt{m}) \approx 15\%$ [10].

To quantify the spatial clustering of local minima at each depth d, the root-mean-squared radius $\rho_{\text{RMS}}(d)$ was defined. Let the converged parameter vectors $\{\theta_{d,i}\}_{i=1}^{25}$ be stacked as the rows of a matrix $\Theta_d \in \mathbb{R}^{25 \times N}$. The Gram matrix $\Sigma_d = \Theta_d \, \Theta_d^T$ was formed, and the RMS radius was defined:

$$\rho_{\text{RMS}}(d) = \sqrt{\frac{\operatorname{tr}(\Sigma_d)}{25}}.$$

This scalar captures the average Euclidean norm of the minimizers and thus provides a direct measure of how tightly the 25 restarts clustered in parameter space at each network depth, and is analogous to spread measures used in other studies [11].

C. Pareto Analysis and λ Selection

The joint behavior of training loss ℓ_{train} and Hessian-based curvature κ was treated as a two-objective problem. For each depth d, the collection $\{\ell^i_{train}, \kappa^i\}_{i=1}^{25}$ was studied. Pareto-front analysis identified the efficient boundary where no objective could be improved without worsening the other.

To construct a stable Pareto front, the two-dimensional convex hull of the 25 points $(\ell^i_{\text{train}}, \kappa^i)$ was computed. All interior (dominated) points were discarded, and the "lower-left" chain of hull vertices (monotonic in κ and non-increasing in ℓ_{train}) was retained as the front F_d at depth d. An orthogonal (total-least-squares) regression line $\ell_{\text{train}} = a + b\,\kappa$ was then fit to the points in F_d. The trade-off weight $\lambda^d = -\,b$ was chosen so that minimizing the combined objective $\ell_{\text{train}} + \lambda_d\,\kappa$ traces out the Pareto front, following the classic linear (Lagrangian) scalarization method [12]. This single scalar λ_d therefore encapsulates the flatness–fit trade-off among the best minima at depth d.

To determine a single trade-off weight that best predicts validation performance, a proxy performance curve $P_d(\lambda) = \ell^d_{\text{train}} + \lambda\,\kappa^d$ was constructed alongside the validation loss curve $V_d = \ell^d_{\text{val}}$. Both $P(\lambda) = \{P_d(\lambda)\}_{d=1}^{10}$ and $V = \{V_d\}_{d=1}^{10}$ were min-max normalized across depths. The global trade-off weight λ_G was selected by

$$\lambda_G = \arg\min_{\lambda} \sum_{d=1}^{10} |P_d(\lambda) - V_d|.$$

This λ_G is the single scalar that makes the proxy performance curve align most closely with the held-out validation loss across all depths.

III. Results and Discussion

The Results are presented in three sections, each addressing a different aspect of how depth influences the geometry and predictive power of local minima in the loss landscape. Section A investigates the raw distribution of training loss and curvature samples at each depth, characterizing how both central tendency and spread evolve as the architecture deepens. Section B introduces two summary statistics—the RMS radius ρ_{RMS} to capture how tightly the converged parameters cluster, and the per-depth trade-off weight λ_d to measure the local balance between fit and flatness. Finally, Section C integrates these insights into a single global performance proxy $P_d(\lambda_G) = \ell_{\text{train}} + \lambda_G\,\kappa$, demonstrating how an appropriately chosen curvature penalty can systematically regularize training loss to better align with generalization.

A. Loss–Curvature Dispersion Across Depths

Depth exerts a pronounced, non-monotonic effect on the joint distribution of training loss and Hessian curvature (Fig. 1). In very shallow networks ($d = 1$–3), the 50% confidence ellipses occupy the middle-right region of the plot, corresponding to high loss and moderate curvature; their large area reflects considerable variability across the 25 restarts. As depth increases to $d = 4$–6, these ellipses shift sharply down and to the left: mean loss decreases by roughly an order of magnitude, curvature falls by more than half, and the ellipses contract noticeably, indicating that all restarts converge to similarly flat, low-loss minima. Beyond this mid-range, further increases in depth drive the ellipses slightly back up in loss and markedly upward in curvature, with the shapes expanding again—evidence of renewed sharpness and greater heterogeneity among deep minima. This re-emergent sharpness also parallels the generalization gap seen in large-batch training regimes [3].

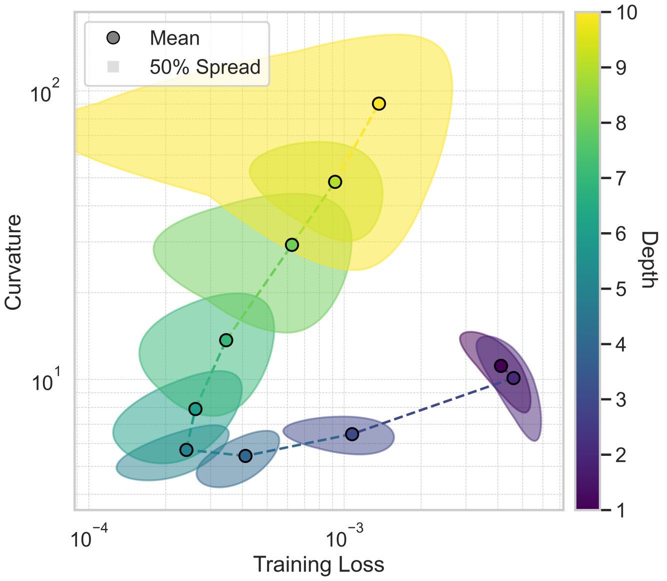

Fig. 1. Fifty-percent confidence ellipses and mean points of training-loss vs. Hessian-curvature at converged parameters for MLPs of depths 1–10. Each semi-transparent ellipse encloses the central 50% of all 25 restarts (ℓ, κ) values at that depth. Black-outlined circles mark the per-depth means, and dashed lines connect these means in order of increasing depth. The colormap indicates network depth from 1 (purple) to 10 (yellow).

Two distinct regimes emerge in this trajectory. From $d = 1 - 5$, loss plummets rapidly while curvature declines more gradually; from $d = 5 - 10$, curvature rises steeply even as loss increases only modestly. Note that the vertical axis (curvature) spans over an order of magnitude more variability than the horizontal (loss), so these trends are best understood in relation to their own scales and the normalized shifts between depths. The crossover around $d = 4$–6 identifies an optimal depth window where minima are both flattest and lowest in loss, and are the most homogeneous. This non-monotonic depth-generalization curve matches theoretical predictions for MLPs [13] and mirrors the performance gains observed in deep residual networks [14].

These findings suggest that very shallow, under-parameterized networks are confined to steep, high-loss basins, whereas excessively deep, over-parameterized models induce sharper minima despite higher representational capacity. The pronounced contraction of the ellipses at mid-depth implies that in this regime, gradient descent schemes like Adam reliably locate broad, low-loss basins, a geometry type that empirical evidence links to strong generalization.

B. Scalar Characterization of Local Minima

Two metrics condense each depth's cloud of 25 optima into interpretable scalars (Fig. 2). The top panel shows the RMS radius $\rho_{RMS}(d)$ of the converged parameter vectors. The resulting U-shaped curve shows that for shallow $d = 1, 2$ and deep ($d = 7 - 10$) networks, the minima are spread out across the parameter space, consistent with mode-connectivity spread measures in other network families [11]. However, for mid-depth models ($d = 4 - 6$, the minima form a more concentrated cluster.

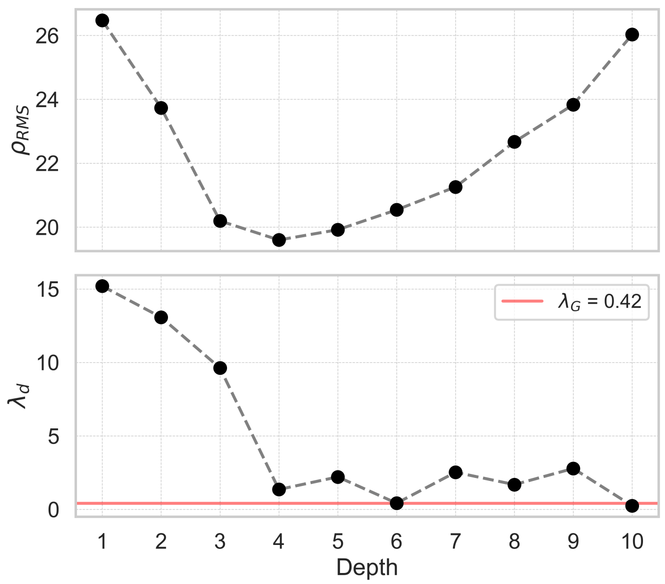

Fig. 2. Summary statistics of MLP minima across depths d. (Top) RMS radius $\rho_{\text{RMS}}(d)$ of the 25 converged parameter vectors at each depth, showing a U-shaped dependence: minima cluster most tightly around d=4. (Bottom) Local flatness–fit trade-off weights λ_d extracted from the Pareto fronts of loss vs. curvature, plotted against depth. The horizontal red line marks the globally-optimal weight $\lambda_G = 0.36$ that best aligns the combined proxy $P = \ell_{train} + \lambda \, \kappa$ with held-out validation loss.

The bottom panel shows the per-depth trade-off weights λ_d, each defined by the direction orthogonal to a total-least-squares fit of the Pareto front at that depth. Intuitively, λ_d quantifies the "cost" in curvature required to achieve a one-unit decrease in training loss along the optimal trade-off contour. In contrast to the U-shaped RMS radius, λ_d decreases from $d = 1 - 4$, indicating ever more favorable flatness-vs-fit ratios, and then remains relatively low and stable for $d = 4 - 10$.

This plateau in λ_d beyond $d = 4$ means that, although the absolute loss and curvature values (and their variances) continue to change with depth, the *relative* trade-off between them remains similar. In other words, once the network is deep enough, every additional "unit" of loss improvement always costs the same amount of curvature increase, regardless of whether those minima sit in a tighter or more dispersed cloud. This stability of the cost ratio suggests a depth-invariant geometric structure: deeper models may find minima with higher or more variable curvature or loss, but the slope of the Pareto frontier (the fundamental balance between fit and flatness) no longer shifts. Architecturally, it indicates that beyond a certain depth, adding layers changes where minima lie in absolute terms but not how steeply loss and curvature trade off against one another.

C. Global Curvature-Regularized Performance Proxy

A global trade-off weight λ_G was selected to test whether the combined objective $P(\lambda) = \ell_{train} + \lambda\, \kappa$ could predict held-out generalization across all depths. The procedure outlined in Section II C yielded $\lambda_G \approx 0.42$. The normalized proxy curve $P(\lambda_G)$ tracked normalized validation losses very closely (Fig. 3).

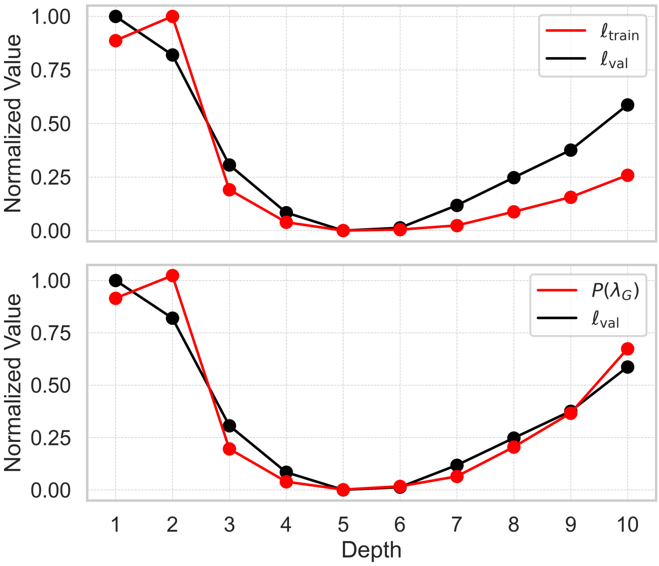

Fig. 3. Normalized training and validation losses (top) and normalized performance proxy $P(\lambda_G)$ versus validation loss (bottom) as a function of network depth. In the top panel, red and black lines trace the min–max–normalized mean training loss ℓ_{train} and validation loss ℓ_{val}, respectively. In the bottom panel, the red line shows the normalized proxy $P_d(\lambda_G) = \ell_{\text{train}} + \lambda_G\, \kappa$ using the globally optimized weight $\lambda_G = 0.42$, plotted alongside the same validation loss curve in black.

In quantitative terms, the proxy $P(\lambda_G)$ reduced the mean absolute error (MAE) to validation loss by roughly 15.7% compared to using training loss alone. This improvement was most pronounced at depths $d > 5$, where the raw training loss begins to under-predict generalization error. This occurs since deep networks often achieve very low training loss, but suffer larger validation gaps. By adding the curvature term with weight λ_G, the proxy inflates the loss for these deeper models, reflecting their sharper minima, and brings the curve back into alignment with held-out performance.

IV. Conclusions

This work examined how varying the depth of multilayer perceptrons shapes the geometry of their loss landscapes and, in turn, influences generalization. Training ten architectures (depths 1–10) with 25 random restarts each on MNIST yielded a rich dataset of converged parameters, training and validation losses, and Hessian-based curvature at each minimum.

Analysis of the full loss–curvature distributions revealed a non-monotonic depth effect. Shallow networks $(d \leq 3)$ consistently fell into sharp, high-loss basins with large inter-restart variability, while mid-depth models $(d = 4–6)$ achieved both the lowest loss and the flattest, most homogeneous minima. Beyond that range, deeper networks again exhibited sharper curvature and increased dispersion, despite maintaining low training loss.

Two concise metrics (the RMS radius of the 25 optima and the orthogonal-regression-derived trade-off weight λ_d) both reached their optimal values around depth four, confirming that this regime offers the most reliable, flat solutions. Finally, a single global weight $\lambda_G \approx 0.42$ applied to the combined proxy $P(\lambda_G) = \ell_{train} + \lambda_G\, \kappa$ reduced the mean absolute error to held-out validation loss by over 15%. This demonstrates that augmenting training loss with a curvature term provides an effective, depth-agnostic regularizer that systematically bridges the gap between optimization and generalization.

These findings highlight an optimal window of network depth where capacity, optimization dynamics, and loss-landscape flatness align to produce robust generalization. However, our reliance on the Frobenius norm of the Hessian as a flatness metric has clear limitations: by aggregating squared eigenvalues it obscures the distinction between a few sharp directions and many small ones, offers no insight into the most sensitive curvature directions, and can be dominated by noise in high-dimensional settings. Future work could extend this geometric framework to convolutional and residual architectures, investigate dynamic curvature penalties during training, explore alternative norms that better capture directional sharpness, and develop more efficient curvature-estimation techniques for large-scale models.

Acknowledgment

The author gratefully acknowledges the support of Professor Usman Khan for his instruction and guidance on this project.

References

[1] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

[2] S. Hochreiter and J. Schmidhuber, "Flat minima," *Neural Computation*, vol. 9, no. 1, pp. 1–42, 1997.

[3] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, “On large-batch training for deep learning: Generalization gap and sharp minima,” in *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

[4] Y. Jiang, B. Neyshabur, H. Mobahi, D. Krishnan, and S. Bengio, “Fantastic generalization measures and where to find them,” in *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020.

[5] L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio, “Sharp minima can generalize for deep nets,” *arXiv preprint arXiv:1703.04933*, 2017.

[6] M. Haddouche, P. Viallard, U. Simsekli, and B. Guedj, “A pac-bayesian link between generalisation and flat minima,” in *Proceedings of The 36th International Conference on Algorithmic Learning Theory*, ser. Proceedings of Machine Learning Research, G. Kamath and P.-L. Loh, Eds., vol. 272. PMLR, 24–27 Feb 2025, pp. 481–511. [Online]. Available: https://proceedings.mlr.press/v272/haddouche25a.html

[7] R. Patel, L. Nguyen, and Y. Chen, “The tunnel effect: Building data representations in deep neural networks,” in *NeurIPS*, 2023.

[8] D. Granziol and G. W. Taylor, “Flatness is a false friend,” *arXiv preprint arXiv:2006.09091*, 2020, see Section 3.1 where the Hessian’s Frobenius norm (along with trace and spectral norms) is evaluated as a flatness metric.

[9] M. F. Hutchinson, “A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines,” *Communications in Statistics – Simulation and Computation*, vol. 18, no. 3, pp. 1059–1076, 1989.

[10] H. Avron and S. Toledo, “Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix,” *Journal of the ACM*, vol. 58, no. 2, pp. 8:1–8:34, 2011.

[11] K. Garipov, P. Izmailov, D. Podoprikhin, D. P. Vetrov, and A. G. Wilson, “Loss surfaces, mode connectivity, and fast ensembling of dnns,” in *Advances in Neural Information Processing Systems*, vol. 31, 2018, pp. 8789–8798.

[12] R. T. Marler and J. S. Arora, “Survey of multi-objective optimization methods for engineering,” *Structural and Multidisciplinary Optimization*, vol. 26, no. 6, pp. 369–395, 2010.

[13] S. Sun, W. Chen, L. Wang, X. Liu, and T.-Y. Liu, “On the depth of deep neural networks: A theoretical view,” in *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

[14] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.