Identifying symmetries in the parameter space and data

Nima Dehmamy
IBM Research, MIT-IBM lab
Cambridge MA

With: Bo Zhao, Iordan Ganev, Jianke Yang,
Robin Walters, Rose Yu

Email: nima.dehmamy@ibm.com

Two types of symmetries

Symmetries in the data, processes or phenomena

- Usually, invariances of the distribution of data or features
- Used in group equivariant neural networks

Symmetries in the parameter space

- Invariances of the loss or objective function
- May arise from model and layer architecture

Zhao, Bo, Iordan Ganev, Robin Walters, Rose Yu, and Nima Dehmamy. "Symmetries, flat minima, and the conserved quantities of gradient flow." ICLR 2023, *arXiv preprint arXiv:2210.17216* (2022).

Yang, Jianke, Robin Walters, Nima Dehmamy, and Rose Yu. "Generative Adversarial Symmetry Discovery." *arXiv preprint arXiv:2302.00236* (2023).

Outline

- Parameter space symmetries (loss invariance)
  - Exact and independent of data
  - Data dependent, nonlinear symmetries
  - Use case: Teleportation
- Data symmetries
  - Discovering symmetries using LieGAN

Caspar David Friedrich *Wanderer above the Sea of Fog*, ca. 1817

What is the structure of the Loss landscape

- How much of the landscape can be predicted from the model architecture?
- Are there flat valleys?

https://peakvisor.com/park/grand-canyon-national-park.html

Mode connectivity

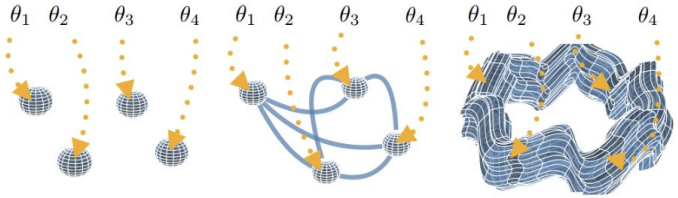


Figure 1. A progressive understanding of the loss surfaces of neural networks. **Left:** The traditional view of loss in parameter space, in which regions of low loss are disconnected (Goodfellow et al., 2015; Choromanska et al., 2015). **Center:** The revised view of loss surfaces provided by work on mode connectivity; multiple SGD training solutions are connected by narrow tunnels of low loss (Garipov et al., 2018; Draxler et al., 2018; Fort & Jastrzebski, 2019). **Right:** The viewpoint introduced in this work; SGD training converges to different points on a connected *volume* of low loss. Paths between different training solutions exist within a large multi-dimensional manifold of low loss. We provide a two dimensional representation of these loss surfaces in Figure A.1.

Benton, Gregory, et al. "Loss surface simplexes for mode connecting volumes and fast ensembling." *ICML*. PMLR, 2021.

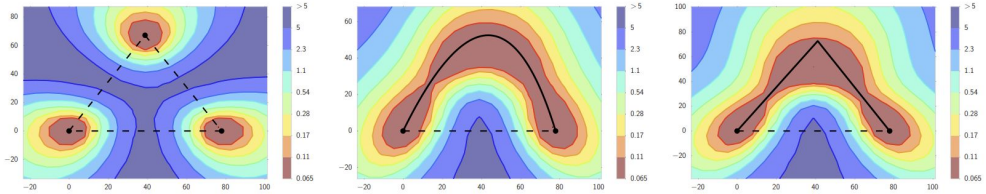
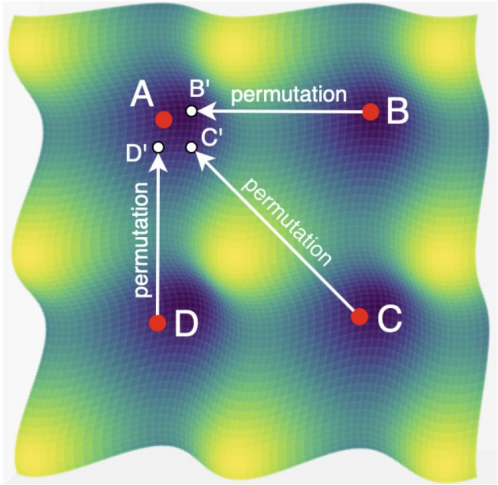


Figure 1: The ℓ_2-regularized cross-entropy train loss surface of a ResNet-164 on CIFAR-100, as a function of network weights in a two-dimensional subspace. In each panel, the horizontal axis is fixed and is attached to the optima of two independently trained networks. The vertical axis changes between panels as we change planes (defined in the main text). **Left:** Three optima for independently trained networks. **Middle** and **Right**: A quadratic Bezier curve, and a polygonal chain with one bend, connecting the lower two optima on the left panel along a path of near-constant loss. Notice that in each panel a direct linear path between each mode would incur high loss.

Garipov, Timur, et al. "Loss surfaces, mode connectivity, and fast ensembling of dnns." (NIPS 2018).

Mode connectivity



Entezari, Rahim, et al. "The role of permutation invariance in linear mode connectivity of neural networks." *arXiv preprint arXiv:2110.06296* (2021).

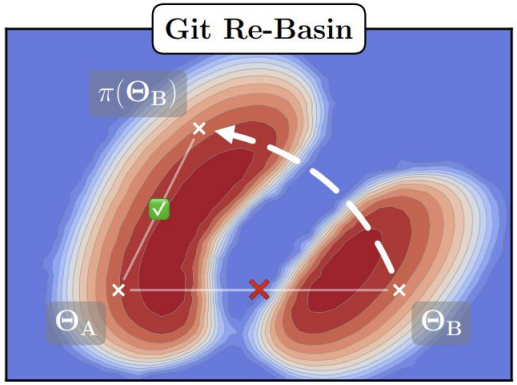
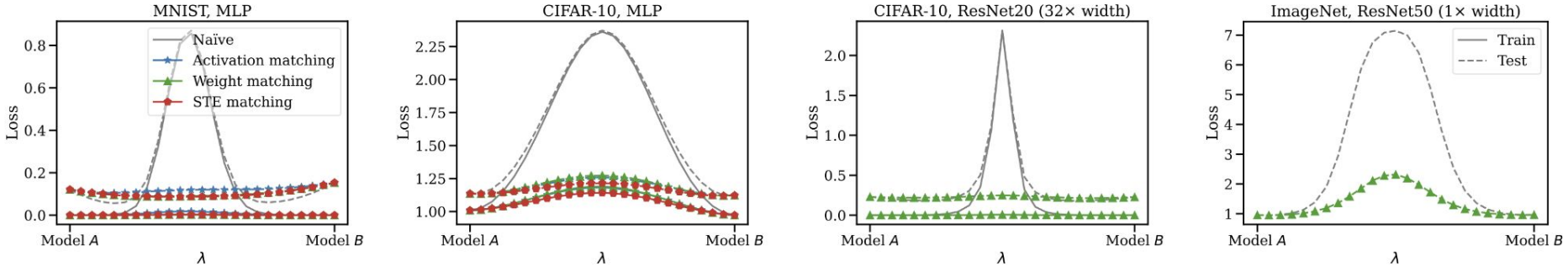


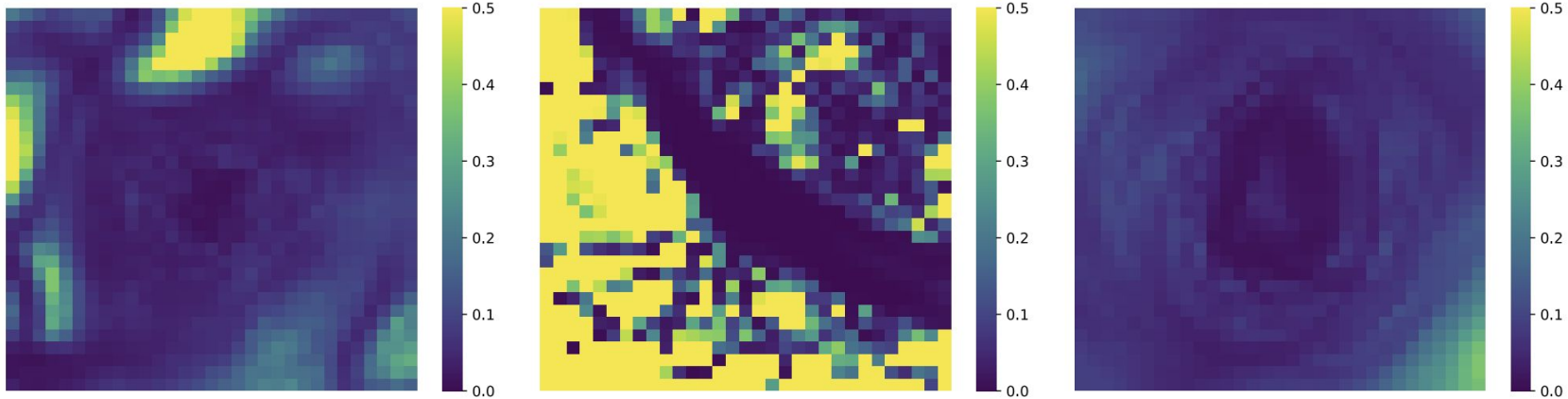
Figure 1: **Git Re-Basin merges models by teleporting solutions into a single basin.** Θ_B is permuted into functionally-equivalent $\pi(\Theta_B)$ so that it lies in the same basin as Θ_A.



Ainsworth, S. K., Hayase, J., & Srinivasa, S. (2022). Git re-basin: Merging models modulo permutation symmetries.

Architecture ⇔ structure of loss landscape

"This partitioning of chaotic and convex regions may explain the **importance of good initialization** strategies, and also the **easy training behavior of "good" architectures**."



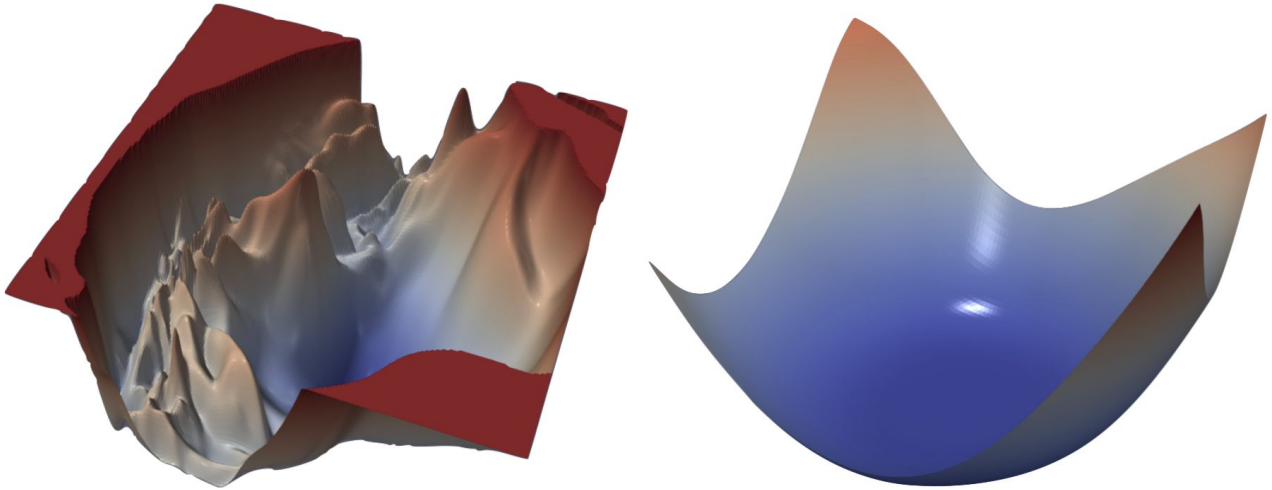
(a) Resnet-56 (b) Resnet-56-noshort (c) DenseNet-121

Figure 7: For each point in the filter-normalized surface plots, we calculate the maximum and minimum eigenvalue of the Hessian, and map the ratio of these two.

Li, Hao, et al. "Visualizing the loss landscape of neural nets." *Advances in neural information processing systems* 31 (2018).

Extended valleys?

1. What **aspects of the architecture** produce extended valleys in the loss?
2. What **parametrizes different points** along a loss valley?



(a) ResNet-110, no skip connections (b) DenseNet, 121 layers

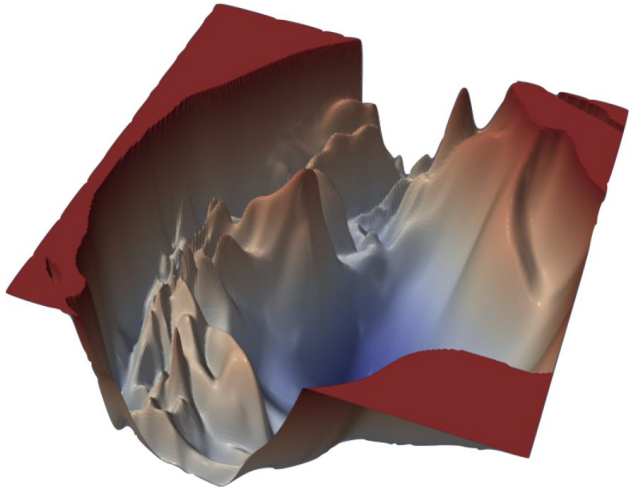
Figure 4: The loss surfaces of ResNet-110-noshort and DenseNet for CIFAR-10.

Li, Hao, et al. "Visualizing the loss landscape of neural nets." *Advances in neural information processing systems* 31 (2018).

Short answer: Origin of low loss valleys

1. What **aspects of the architecture** produce extended valleys in the loss?
- **Answer:** Some valleys arise from **continuous symmetries**

2. What **parametrizes different points** along a loss valley?
- **Answer:** in symmetry induced valleys
  Conserved quantities can parametrize the valley



SYMMETRIES, FLAT MINIMA AND THE CONSERVED QUANTITIES OF GRADIENT FLOW

Bo Zhao[*†]
University of California, San Diego
bozhao@ucsd.edu

Iordan Ganev[*]
Radboud University
iganev@cs.ru.nl

Robin Walters
Northeastern University
rwalters@northeastern.edu

Rose Yu
University of California, San Diego
roseyu@ucsd.edu

Nima Dehmamy
IBM Research
nima.dehmamy@ibm.com

ABSTRACT

Empirical studies of the loss landscape of deep networks have revealed that many local minima are connected through low-loss valleys. Ensemble models sampling different parts of a low-loss valley have reached SOTA performance. Yet, little is known about the theoretical origin of such valleys. We present a general framework for finding continuous symmetries in the parameter space, which carve out low-loss valleys. Importantly, we introduce a novel set of nonlinear, data-dependent symmetries for neural networks. These symmetries can transform a trained model such that it performs similarly on new samples. We then show that conserved quantities associated with linear symmetries can be used to define coordinates along low-loss valleys. The conserved quantities help reveal that using common initialization methods, gradient flow only explores a small part of the global minimum. By relating conserved quantities to convergence rate and sharpness of the minimum, we provide insights on how initialization impacts convergence and generalizability. We also find the nonlinear action to be viable for ensemble building to improve robustness under certain adversarial attacks.

ICLR 2023

Definitions

- **“Loss”:** combination of model architecture and loss function (e.g. MSE or cross entropy loss)

  Ex: $Loss(W; X, Y) = MSE(F(W, X), Y)$ (F representing a neural net)

$$\mathcal{L} : \mathbf{Param} \times \mathbf{Data} \to \mathbb{R}, \qquad \mathcal{L}(\boldsymbol{\theta}, (x, y)) = \mathrm{Cost}(y, F_{\boldsymbol{\theta}}(x)).$$

- **“Symmetry”:** Any transformation on model weights or data which **keep loss invariant**.

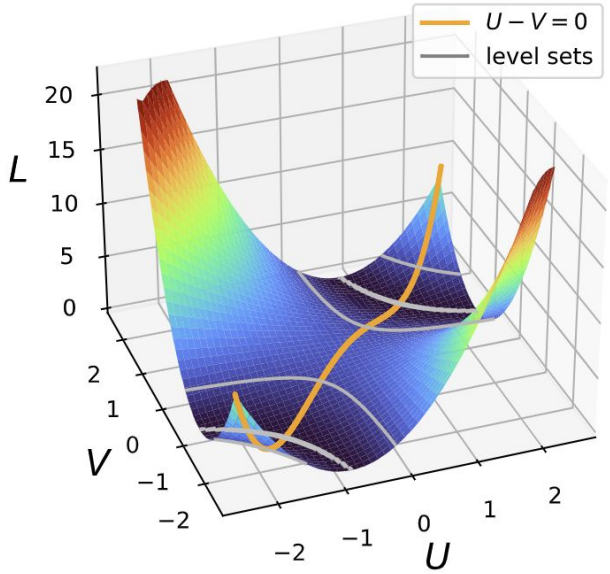
$$\mathcal{L}(g \cdot \boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}), \qquad \forall \boldsymbol{\theta} \in \mathbf{Param}, \quad g \in G$$

Continuous symmetries should produce valleys

- If there exist continuous symmetries of model parameters,
- Example: Linear regression, two layer linear network

$$L(W; X, Y) = \frac{1}{n}\|Y - WX\|^2, \quad W = UV$$

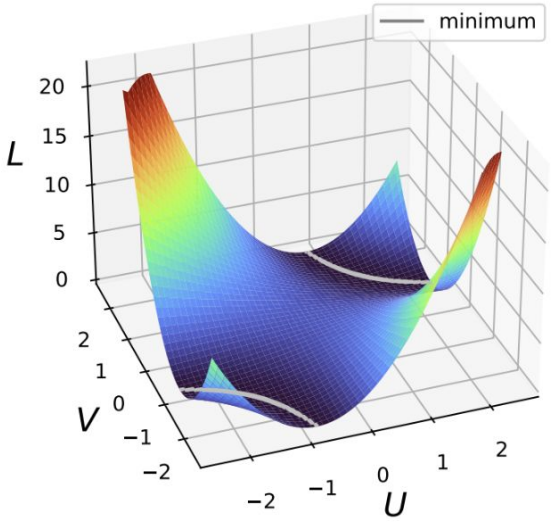
- For any linear reparamtrization $W = UV$:



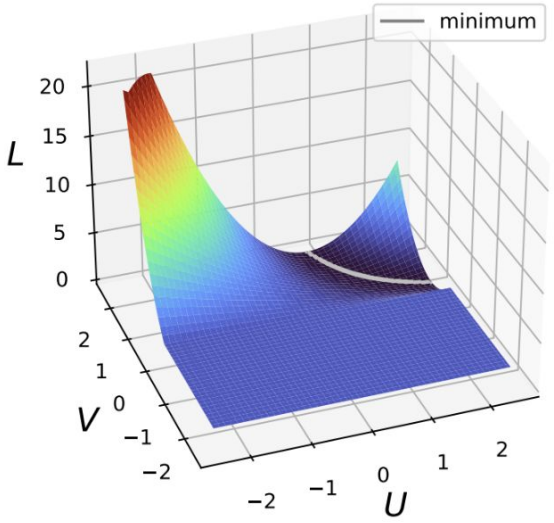
Nonlinear case

- More tricky: continuous symmetries not guaranteed to exist for generic nonlinear activation… But

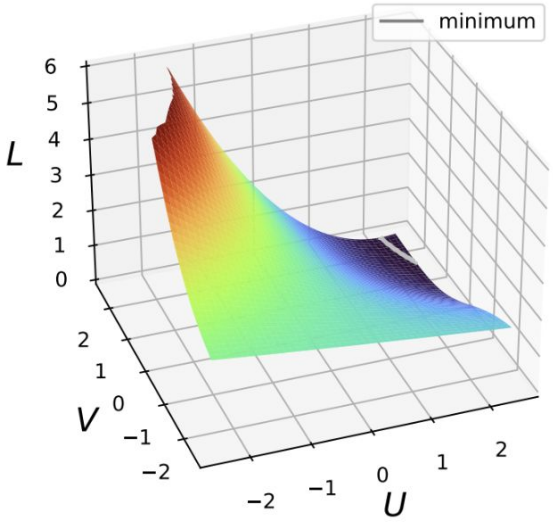
$$L(U,V) = \tfrac{1}{2}|2 - U\sigma(V)|^2$$



(a) linear



(b) ReLU



(c) sigmoid

General case using “equivariant” activation

Consider a subnetwork with $U = W_i$ and $V = W_{i-1}$

$$F(x) = U\sigma(Vx) \text{ for } (U, V) \in \mathbf{Param} = \mathbb{R}^{m\times h} \times \mathbb{R}^{h\times n} \text{ and } x \in \mathbb{R}^n$$

Let $G \subseteq \mathrm{GL}_h(\mathbb{R})$ be a subgroup and $\pi : G \to \mathrm{GL}_h(\mathbb{R})$ be a representation.

If $$\boxed{\sigma(gz) = \pi(g)\sigma(z)}$$

Then, this subnetwork has the symmetry

$$\boxed{g \cdot U = U\pi(g^{-1}), \qquad g \cdot V = gV}$$

Examples of $\sigma(gz) = \pi(g)\sigma(z)$

- **Linear network:** Has GL symmetry (any invertible matrix g)

$$\sigma(x) = x. \text{ One can take } \pi(g) = g \text{ and } G = \mathrm{GL}_h(\mathbb{R})$$

- **Homogeneous:** is symmetric under positive rescaling (diagonal matrices with positive diagonal entries) $g = \mathrm{diag}(\mathbf{c})$ with $\mathbf{c} = (c_1, \ldots, c_h) \in \mathbb{R}^h_{>0}$

$$\sigma(cz) = c^\alpha \sigma(z)$$

$$[\sigma(gz)]_i = \sigma(c_i z_i) = c_i^\alpha \sigma(z_i) = [g^\alpha \sigma(z)]_i$$

  - **ReLU and LeakyReLU:** Homogeneous of degree 1. Have positive rescaling symmetry

- **Radial rescaling activation:** $\sigma(z) = f(\|z\|)z$
  Symmetric under the orthogonal group $g \in O(h)$ (that is, $g^T g = I$)

$$\sigma(gz) = f(\|gz\|)(gz) = g(f(\|z\|)z) = g\sigma(z)$$

Symmetry moves us along flat valley

- Is there a way to define coordinates along flat directions?
- Are some places better than others, or should we make an ensemble from them (like “fast ensembling” paper)?

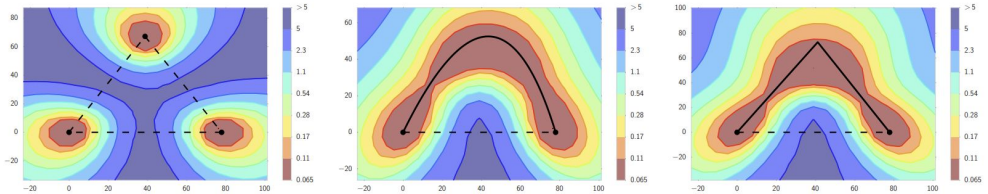
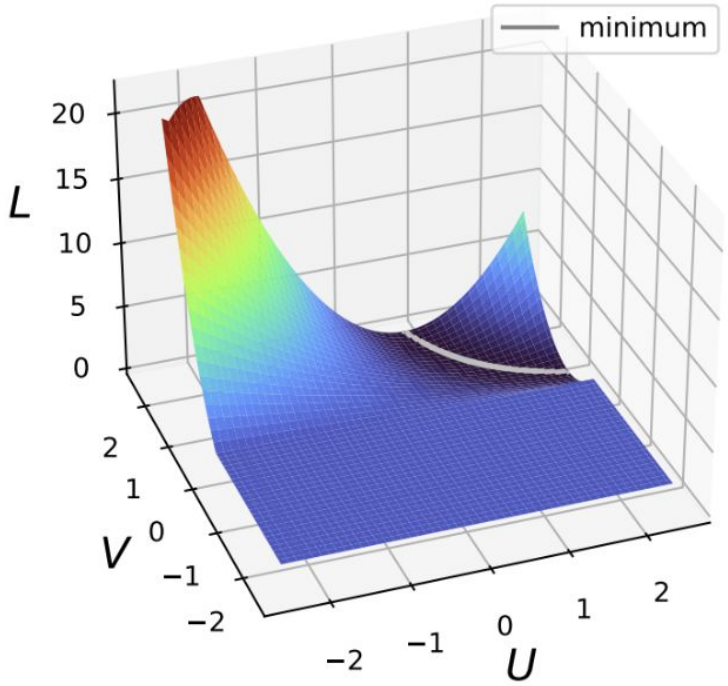


Figure 1: The ℓ_2-regularized cross-entropy train loss surface of a ResNet-164 on CIFAR-100, as a function of network weights in a two-dimensional subspace. In each panel, the horizontal axis is fixed and is attached to the optima of two independently trained networks. The vertical axis changes between panels as we change planes (defined in the main text). **Left:** Three optima for independently trained networks. **Middle** and **Right**: A quadratic Bezier curve, and a polygonal chain with one bend, connecting the lower two optima on the left panel along a path of near-constant loss. Notice that in each panel a direct linear path between each mode would incur high loss.



Continuous Symmetry ⇒ Extended local minima

Proposition 1.4. *Suppose G is a symmetry of $\mathcal{L}$ acting linearly on* **Param**. *Then the gradients of $\mathcal{L}$ at any $\boldsymbol{\theta}$ and $g \cdot \boldsymbol{\theta}$ are related via:*

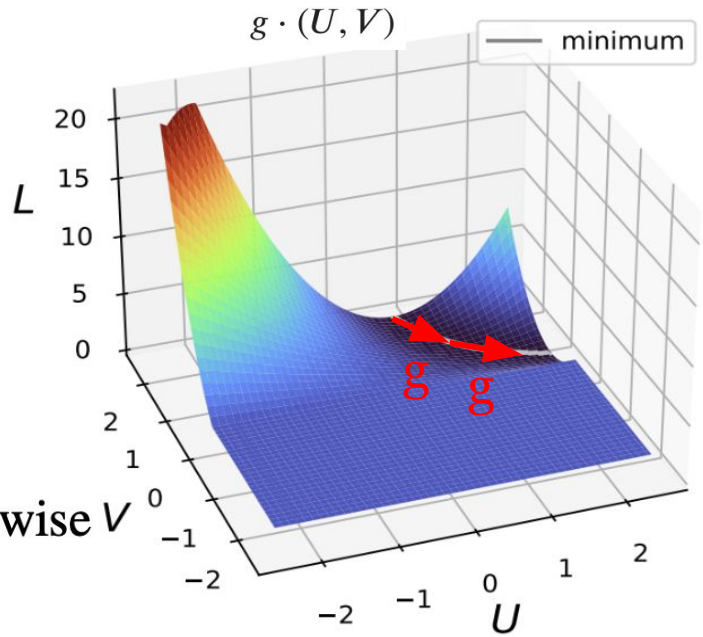
$$\nabla_{g\cdot\boldsymbol{\theta}}\mathcal{L}\ \rho(g) = \nabla_{\boldsymbol{\theta}}\mathcal{L}, \qquad \forall g \in G,\ \forall \boldsymbol{\theta} \in \mathbf{Param} \tag{6}$$

Moreover, if $\boldsymbol{\theta}^*$ is a critical point (resp. local minimum) of $\mathcal{L}$, then so is $g \cdot \boldsymbol{\theta}^*$.

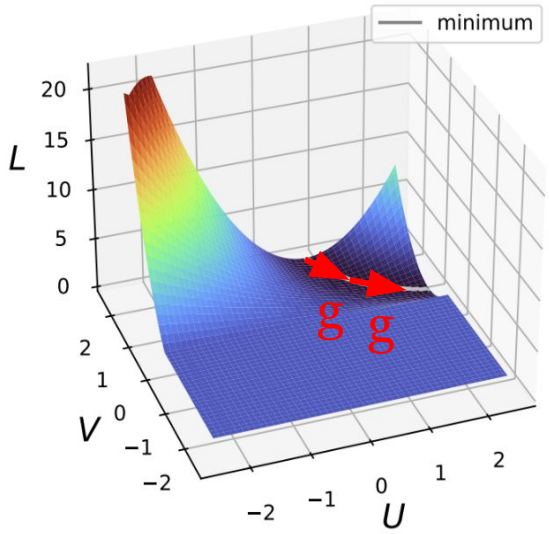
Dimension of generic orbit of group:

- **Linear:** $h^2 - \max(0, h-n)\max(0, h-m)$
- **Homogeneous:** $\min(h, \max(n, m))$
- **Radial:**

$\binom{h}{2}$ if $h \leq \max(n, m)$ and $\binom{h}{2} - \binom{h-\max(m,n)}{2}$ otherwise



How can we parameterize **where** along a minimum we are?



Q: Parameterizing **where** along a minimum we are?

A: Conserved quantities

During gradien flow (GF) some quantities remain constant

GF: $\dot{\boldsymbol{\theta}}(t) = d\boldsymbol{\theta}(t)/dt = -\varepsilon \nabla_{\boldsymbol{\theta}(t)} \mathcal{L}.$

Conservation of Q:

$$dQ(\boldsymbol{\theta}(t))/dt = 0.$$

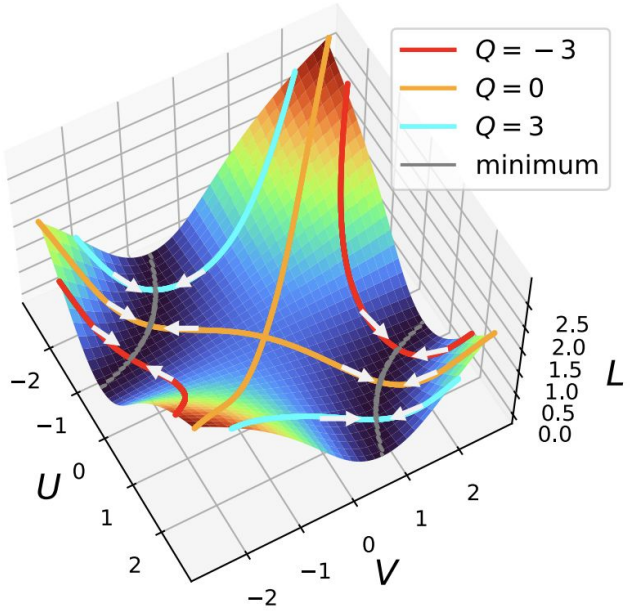


Figure 2: Gradient flow for $L(U, V) = \frac{1}{2}\|Y - UVX\|^2$, where $U, V \in \mathbb{R}$, $Y = 2$, and $X = 1$. Trajectories corresponding to different values of Q intersect the minima at different points.

Relating conserved Q to symmetries

$$\frac{dQ_i}{dt} = \left\langle \frac{d\theta}{dt}, \frac{\partial Q}{\partial \theta} \right\rangle = -\left\langle \varepsilon \nabla_\theta \mathcal{L}, \nabla_\theta Q \right\rangle$$

Symmetry condition: $\mathcal{L}(g \cdot \boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}),$

Infinitesimal version $\langle \nabla_\theta \mathcal{L}, M \cdot \boldsymbol{\theta} \rangle = 0 \quad \Rightarrow \quad \left\langle \varepsilon^{-1} \dot{\boldsymbol{\theta}} \, , \, M \cdot \boldsymbol{\theta} \right\rangle = 0$

Where M is a Lie algebra element $\overline{M}_{\boldsymbol{\theta}} := \left. \frac{d}{dt} \right|_{t \to 0} (\exp_M(t) \cdot \boldsymbol{\theta})$

Proposition 2.1. *Suppose the action of G on* **Param** *is linear*[6] *and leaves $\mathcal{L}$ invariant. For any $M \in \mathfrak{g}$, the function $Q_M : \mathbf{Param} \to \mathbb{R}$ is a conserved quantity:*

$$Q_M(\boldsymbol{\theta}) = \langle \boldsymbol{\theta} \, , \, M \cdot \boldsymbol{\theta} \rangle \tag{17}$$

The space of distinct conserved quantities of the form Q_M for $M \in \mathfrak{g}$ is in one-to-one correspondence with the space of symmetric matrices in $\mathfrak{g}$.

Conserved Q for linear symmetries

$M \in \mathfrak{g}$	M symmetric	M anti-symmetric
differential equation	conserved quantity	differential equation
$\dot{\theta}^T M \theta = 0$	$Q_M(\theta) = \theta^T M \theta$	$\sum_{i<j} m_{ij} r_{ij}^2 \dot{\phi}_{ij} \equiv 0$

- All conserved Qs in literature (known as **“imbalance”**) are for **symmetric** M
- **Linear:** $Q_M(U, V) = \mathrm{Tr}\left[(VV^T - U^T U)M\right]$
- **Homogeneous (e.g. ReLU):** $\mathbf{Q} = \mathrm{diag}\left[VV^T - \alpha U^T U\right]$
- For antisymmetric M (e.g. O(h) rotation symmetry) **angular momenta** of successive layers cancel. We could not identify an explicit for for Q for O(h).

Need for data-dependent symmetries.

- The symmetries discussed so far were **linear** and data-independent

Need for data-dependent symmetries. A symmetry of $\mathcal{L}$ which is independent of the input data transforms a set of parameters $\boldsymbol{\theta}$ to $\boldsymbol{\theta}' = g \cdot \boldsymbol{\theta}$ such that the loss doesn't change $\mathcal{L}(\boldsymbol{\theta}', X) = \mathcal{L}(\boldsymbol{\theta}, X)$. However, this means that we have $\mathcal{L}(\boldsymbol{\theta}', X') = \mathcal{L}(\boldsymbol{\theta}, X)$ even on new data X', suggesting it is impossible to use these symmetries to improve performance on OOD data. Hence, we conclude that for a symmetry to yield any benefit on unseen data, it should be at least *data-dependent*. Therefore, after reviewing linear symmetries, we introduce a set of data-dependent, nonlinear symmetries.

Non-linear group action

General Equivariance: $\sigma(gz) = c(g,z)\sigma(z) \qquad \forall g \in \mathrm{GL}_h \; \forall z \in \mathbb{R}^h$

$$c(g,z) = R_{\sigma(gz)} R^{-1}_{\sigma(z)}$$

Theorem 4.1. *Suppose $\sigma(z)$ is nonzero for any $z \in \mathbb{R}^h$. Then there is an action $\mathrm{GL}_h \times (\mathbf{Param} \times \mathbb{R}^n) \to \mathbf{Param} \times \mathbb{R}^n$ given by*

$$g \cdot (U, V, x) = (U R_{\sigma(Vx)} R^{-1}_{\sigma(gVx)} \; , \; gV \; , \; x). \tag{7}$$

The evaluation of the feedforward function at x unchanged: $F_{(U,V)}(x) = F_{(UR_{\sigma(Vx)}R^{-1}_{\sigma(gVx)}, gV)}(x)$.

Application: Teleportation

Zhao, B., Dehmamy, N., Walters, R., & Yu, R. (2022). Symmetry Teleportation for Accelerated Optimization. *Neurips 2022*.

Symmetries can be used to move to points on the level-set with steeper gradients

$$\frac{d\mathcal{L}(\boldsymbol{w})}{dt} = \left\langle \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}}, \frac{d\boldsymbol{w}}{dt} \right\rangle = -\left[\nabla \mathcal{L}\right]^T \eta \nabla \mathcal{L} = -\|\nabla \mathcal{L}\|_\eta^2,$$

Proposition 5.1. *Let $\boldsymbol{w}' = g \cdot \boldsymbol{w}$ be a point we teleport to. Let $J = \partial \boldsymbol{w}'/\partial \boldsymbol{w}$ be the Jacobian. Symmetry teleportation using g accelerates the rate of decay in $\mathcal{L}$ if it satisfies*

$$\left\| \left[J^{-1}\right]^T \nabla \mathcal{L}(\boldsymbol{w}) \right\|_\eta^2 > \|\nabla \mathcal{L}(\boldsymbol{w})\|_\eta^2 . \tag{10}$$

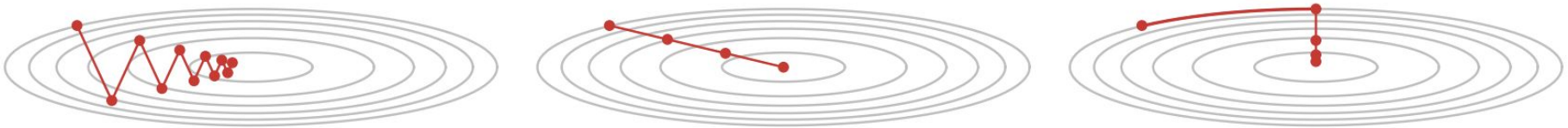


Figure 1: Left to right: gradient descent, second-order methods, proposed method.

Ex: Rosenbrock

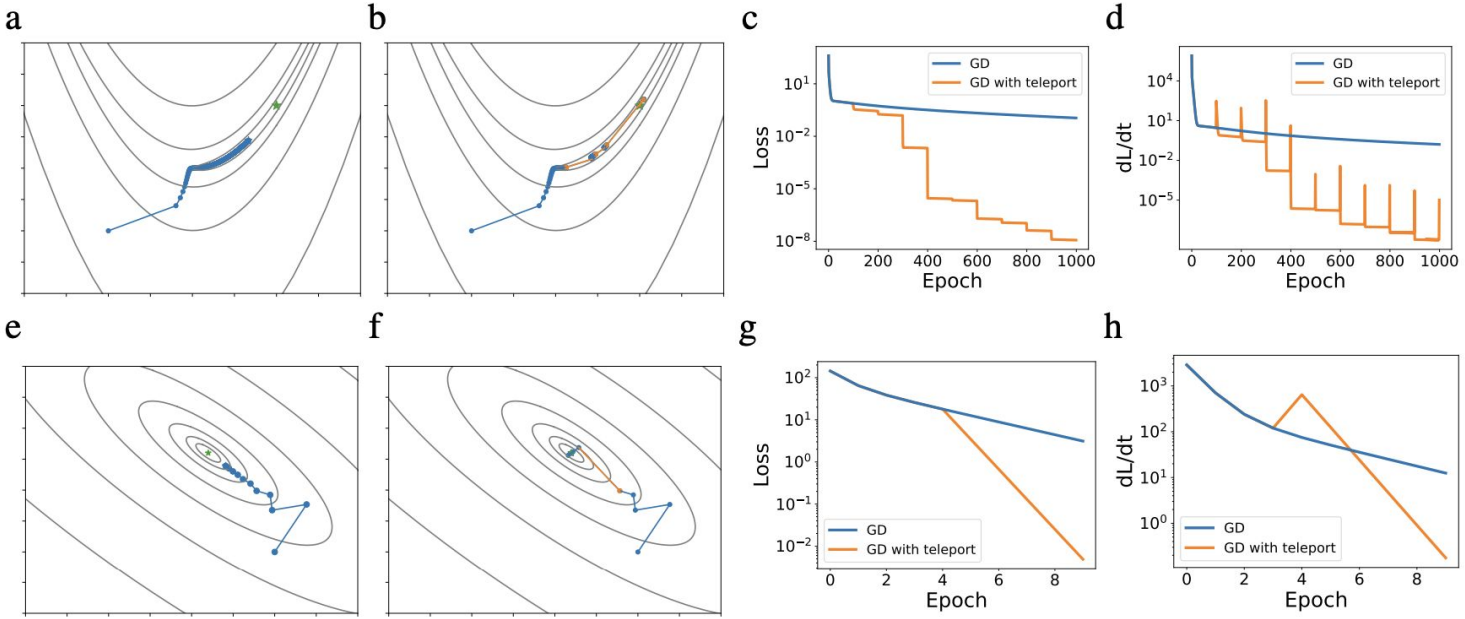


Figure 2: Optimization of the Rosenbrock function (top row) and Booth function (bottom row) using (a) gradient descent and (b) the proposed algorithm. Contours represent the level sets of the loss function. Loss $\mathcal{L}$ and convergence rate $d\mathcal{L}/dt$ are shown in (c) and (d). Teleportation helps move the parameters towards the target.

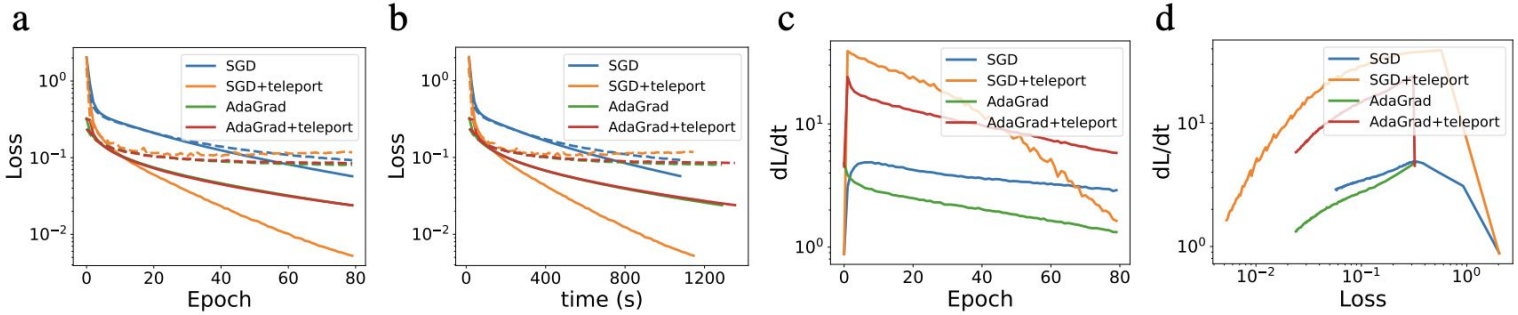


Figure 4: MNIST classification using gradient descent with and without teleportation. Solid lines are training loss and dashed lines are validation loss.

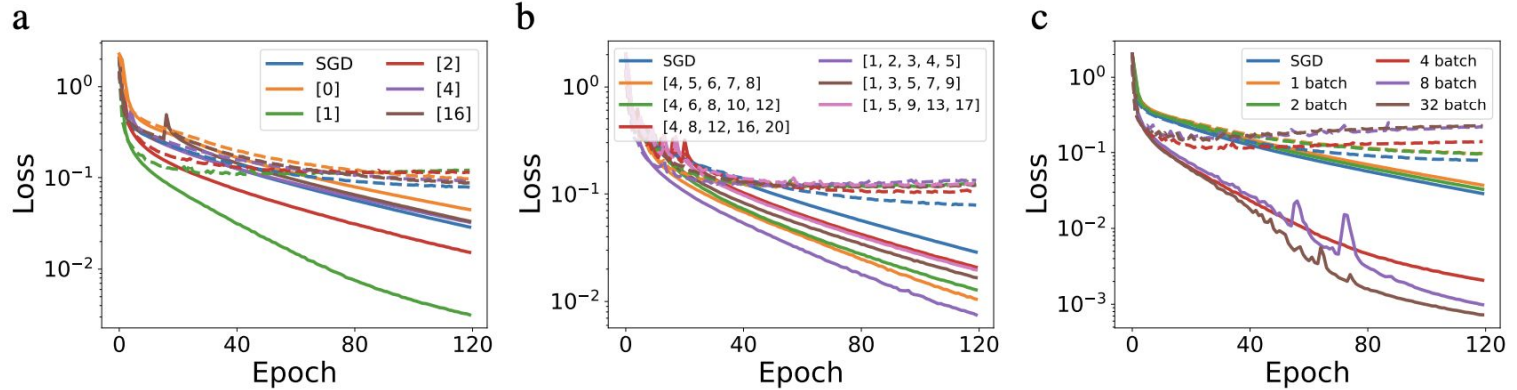


Figure 5: Teleportation (a) once at different epoch, (b) 5 times with different teleportation schedules, and (c) using different number of mini-batches. The lists in the legend of (a) and (c) denote the epoch numbers in teleportation schedule where teleportation happens.

Conclusion

- Model architecture can result in a lot of continuous symmetries in the loss landscape
- The symmetries can be used to move to more favorable parts along minima
- Nonlinear, data-dependent symmetries may be useful for generalization
- Further works is needed

Generative Adversarial Symmetry Discovery

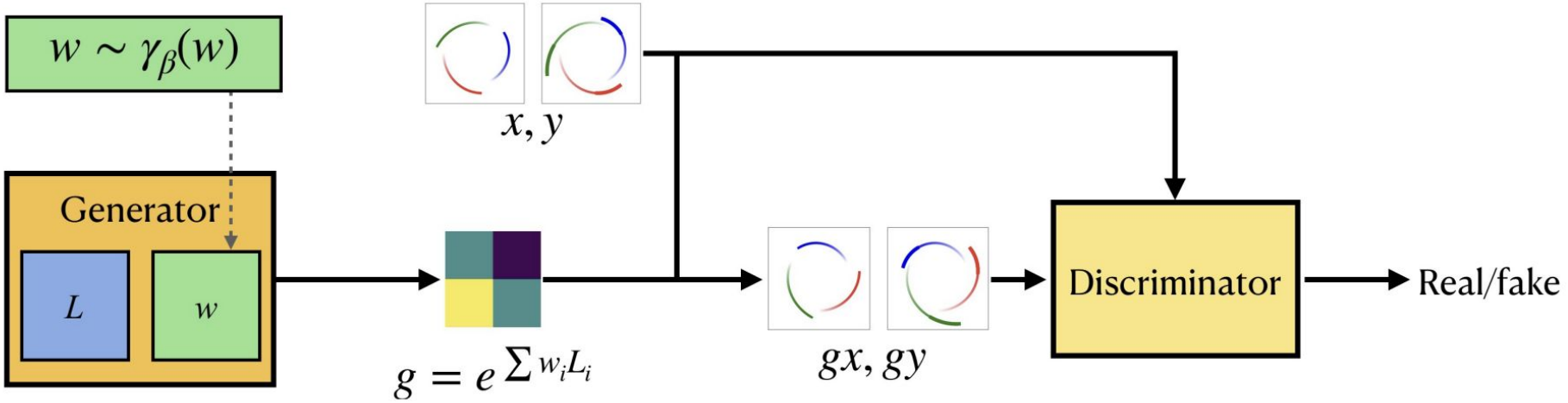
Jianke Yang [1] **Robin Walters** [*2] **Nima Dehmamy** [*3] **Rose Yu** [1]

Table 1. Comparison of different models' capability of discovering different kinds of symmetries

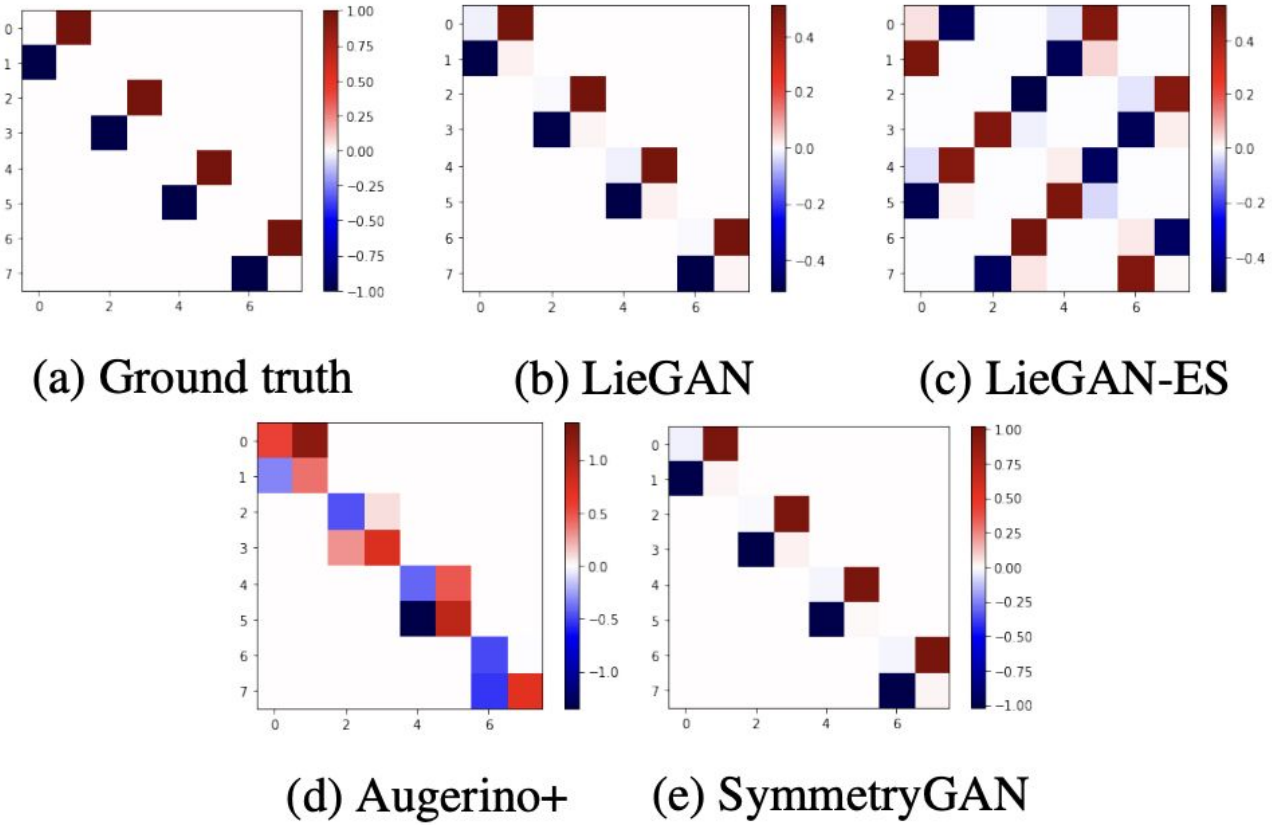
SYMMETRY	MSR	AUGERINO	LIEGAN
DISCRETE	✓	✗	✓
CONTINUOUS	✗	✗	✓
GIVEN GROUP SUBSET	✗	✓	✓
UNKNOWN GROUP SUBSET	✗	✗	✓

LieGAN

- Learn Lie algebra to generate transformations to fool discriminator



2-body gravitation system



Discover Lorentz Symmetry from particle physics data

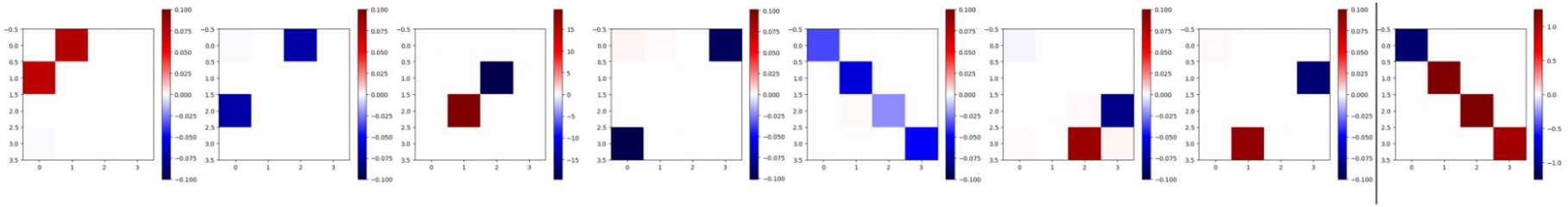


Figure 4. Left: LieGAN discovers an approximate $\mathrm{SO}(1,3)^+$ symmetry in top tagging dataset, where channels 0, 1, 3 indicate boost along x-, y- and z-axis and channels 2, 5, 6 correspond to $\mathrm{SO}(3)$ rotation. Right: Computed invariant metric of the discovered symmetry by solving Equation (10).

Thank you

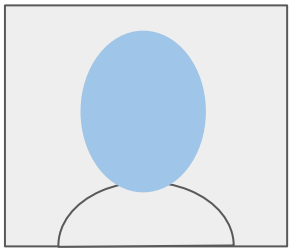
Bo Zhao
UCSD

Iordan Ganev
Radboud Uni.

Robin Walters
Northeastern Uni.

Rose Yu
UCSD

Jianke Yang
UCSD



Email: nima.dehmamy@ibm.com

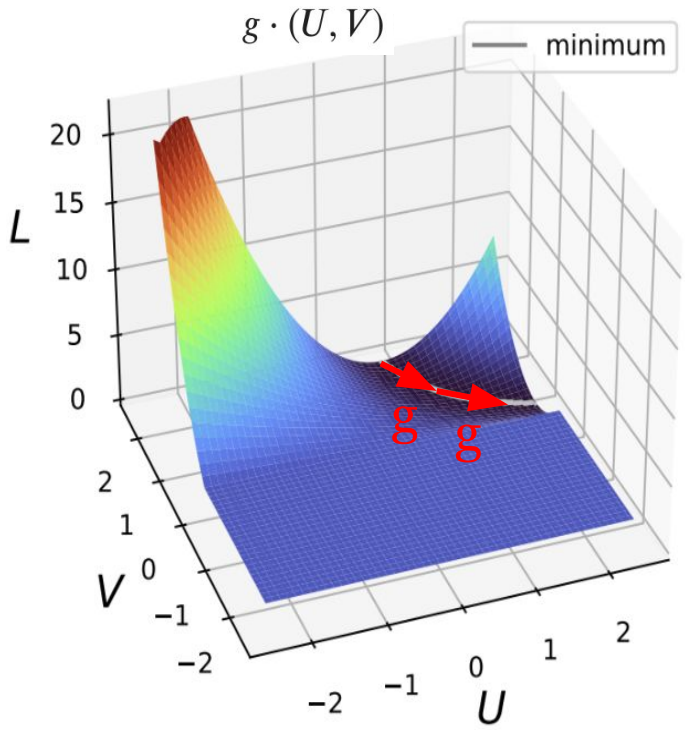
Thank You

Noether's Theorem

Every continuous symmetry leads to conserved quantities during dynamics

- Can conserved quantities define coordinates along loss valleys?
- Is Noether's theorem applicable to

  gradient descent?



Imbalance: A conserved quantity in gradient descent (GD)

Consider the following formulation for factorizing a low-rank matrix:

$$\min_{\boldsymbol{U}\in\mathbb{R}^{d_1\times r},\boldsymbol{V}\in\mathbb{R}^{d_2\times r}} f(\boldsymbol{U},\boldsymbol{V}) = \frac{1}{2}\left\|\boldsymbol{U}\boldsymbol{V}^\top - \boldsymbol{M}^*\right\|_F^2,$$

Layer imbalance $\frac{1}{8}\left\|\boldsymbol{U}^\top\boldsymbol{U} - \boldsymbol{V}^\top\boldsymbol{V}\right\|_F^2.$

Is conserved during GD

Algorithmic Regularization in Learning Deep Homogeneous Models: Layers are Automatically Balanced*

Simon S. Du† **Wei Hu**‡ **Jason D. Lee**§

Abstract

We study the implicit regularization imposed by gradient descent for learning multi-layer homogeneous functions including feed-forward fully connected and convolutional deep neural networks with linear, ReLU or Leaky ReLU activation. We rigorously prove that gradient flow (i.e. gradient descent with infinitesimal step size) effectively enforces the differences between squared norms across different layers to remain *invariant* without any explicit regularization. This result implies that if the weights are initially small, gradient flow automatically balances the magnitudes of all layers. Using a discretization argument, we analyze gradient descent with positive step size for the non-convex low-rank asymmetric matrix factorization problem without any regularization. Inspired by our findings for gradient flow, we prove that gradient descent with step sizes $\eta_t = O\left(t^{-\left(\frac{1}{2}+\delta\right)}\right)$ $(0 < \delta \leqslant \frac{1}{2})$ automatically balances two low-rank factors and converges to a bounded global optimum. Furthermore, for rank-1 asymmetric matrix factorization we give a finer analysis showing gradient descent with constant step size converges to the global minimum at a globally linear rate. We believe that the idea of examining the invariance imposed by first order algorithms in learning homogeneous models could serve as a fundamental building block for studying optimization for learning deep models.

NeurIPS 2018

Imbalance: A conserved quantity in gradient descent (GD)

For homogeneous nonlinearity, imbalance is conserved during GD

Theorem 2.1 (Balanced incoming and outgoing weights at every neuron). *For any* $h \in [N-1]$ *and* $i \in [n_h]$, *we have*

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\|\boldsymbol{W}^{(h)}[i,:]\|^2 - \|\boldsymbol{W}^{(h+1)}[:,i]\|^2\right) = 0. \tag{6}$$

- Could imbalance be related to symmetries?

Rescaling ⇒ Imbalance

Noether's theorem shows that when a model is **invariant under rescaling** of weights of two layers, the **imbalance** between the two layers is conserved

Translation: $\langle \theta_{\mathcal{A}}(t), \mathbb{1} \rangle = \langle \theta_{\mathcal{A}}(0), \mathbb{1} \rangle$

Scale: $|\theta_{\mathcal{A}}(t)|^2 = |\theta_{\mathcal{A}}(0)|^2$

Rescale: $|\theta_{\mathcal{A}_1}(t)|^2 - |\theta_{\mathcal{A}_2}(t)|^2 = |\theta_{\mathcal{A}_1}(0)|^2 - |\theta_{\mathcal{A}_2}(0)|^2$

Published as a conference paper at ICLR 2021

NEURAL MECHANICS: SYMMETRY AND BROKEN CONSERVATION LAWS IN DEEP LEARNING DYNAMICS

Daniel Kunin*, Javier Sagastuy-Brena, Surya Ganguli, Daniel L.K. Yamins, Hidenori Tanaka*†

Stanford University
† Physics & Informatics Laboratories, NTT Research, Inc.

ABSTRACT

Understanding the dynamics of neural network parameters during training is one of the key challenges in building a theoretical foundation for deep learning. A central obstacle is that the motion of a network in high-dimensional parameter space undergoes discrete finite steps along complex stochastic gradients derived from real-world datasets. We circumvent this obstacle through a unifying theoretical framework based on intrinsic symmetries embedded in a network's architecture that are present for *any* dataset. We show that any such symmetry imposes stringent geometric constraints on gradients and Hessians, leading to an associated conservation law in the continuous-time limit of stochastic gradient descent (SGD), akin to Noether's theorem in physics. We further show that finite learning rates used in practice can actually break these symmetry induced conservation laws. We apply tools from finite difference methods to derive *modified gradient flow*, a differential equation that better approximates the numerical trajectory taken by SGD at finite learning rates. We combine modified gradient flow with our framework of symmetries to derive exact integral expressions for the dynamics of certain parameter combinations. We empirically validate our analytic expressions for learning dynamics on VGG-16 trained on Tiny ImageNet. Overall, by exploiting symmetry, our work demonstrates that we can analytically describe the learning dynamics of various parameter combinations at finite learning rates and batch sizes for state of the art architectures trained on *any* dataset.

How?

- Noether’s theorem works specifically with “Lagrangians”.
- The dynamics is defined via a variational principle
- How can GD be written in this language, as a variational equation?

How?

⇒ If we approximate gradient descent using a second-order continuous **gradient flow**, the dynamics can be derived from a variational Lagrangian

x = model parameters (weights)

V = loss

$$x(t+\delta t) = x + \delta t\dot{x} + \frac{\delta t^2}{2}\ddot{x}$$

$$\frac{\delta x}{\delta t} = -\varepsilon \nabla V$$

2nd order gradient flow (GF)

$$\dot{x} = -\varepsilon \nabla V - \frac{\delta t}{2}\ddot{x}$$

Lagrangian formulation of 2nd order gradient flow

Bregman Lagrangian (also used in *Tanaka & Kunin NeurIPS 2021*)
(x=weights, V=loss, ε=learning rate, $\gamma = 2/\delta x$)

$$L = e^{\gamma t}\left(\frac{1}{2}\dot{x}^i \varepsilon_{ij}^{-1} \dot{x}^j - \gamma V(x)\right)$$

Variational (Euler-Lagrange) equations:

$$0 = \frac{\partial L}{\partial x} - \frac{d}{dt}\frac{\partial L}{\partial \dot{x}} \quad \Rightarrow \quad \boxed{\dot{x} = -\varepsilon \nabla V - \frac{1}{\gamma}\ddot{x}} \qquad \boxed{\dot{x} = -\varepsilon \nabla V - \frac{\delta t}{2}\ddot{x}}$$

When $\gamma \to \infty$, we get the first-order gradient flow:

$$\dot{x} = -\varepsilon \nabla V$$

Noether's theorem for 2nd order GF

Let δx denote an infinitesimal continuous symmetry transformation. Then the following Q is conserved ($dQ/dt=0$) during the GF dynamics

$$Q = \delta x \frac{\partial L}{\partial \dot{x}}$$

Let $\delta x = Tx$ where T is an infinitesimal symmetry generator (e.g. Lie algebra)

$$Q = \delta x \frac{\partial L}{\partial \dot{x}} = (Tx) \cdot (e^{\gamma t} \varepsilon^{-1} \dot{x}) = e^{\gamma t} (Tx)^T \varepsilon^{-1} \dot{x} = e^{\gamma t} x^T T^T \varepsilon^{-1} \dot{x}$$

Conserved quantity decays to zero

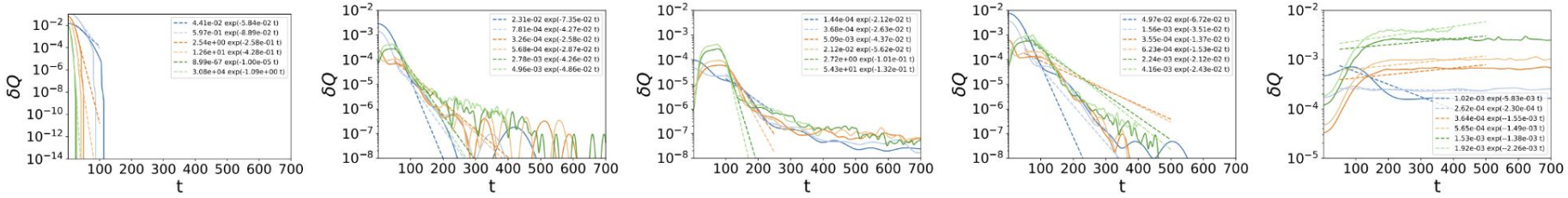
For symmetries with linear action (matrix product)

Let $Q_0 = x^T T^T \varepsilon^{-1} \dot{x}$. Then

$$\frac{dQ}{dt} = e^{\gamma t}\left(\gamma Q_0 + \frac{dQ_0}{dt}\right) = 0$$

$$\gamma Q_0 = -\frac{dQ_0}{dt} \quad \Rightarrow \quad Q_0(t) = e^{-\gamma t} Q_0(0)$$

⇒ in the limit of 1st order GF ($\gamma \to \infty$)
Noether's theorem predicts $Q_0(t)$ decays to zero exponentially



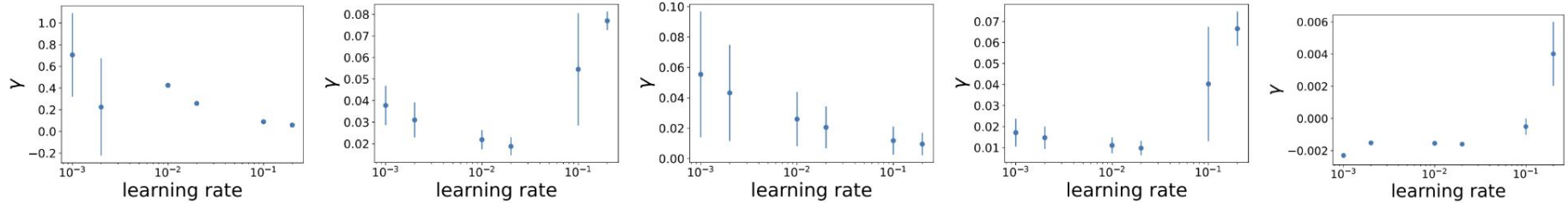
(a) sigma=1.0, m=5, n=10, h=50

(b) sigma=0.1, m=5, n=10, h=50

(c) sigma=0.01, m=5, n=10, h=50

(d) sigma=0.1, m=5, n=10, h=10

(e) sigma=0.1, m=20, n=30, h=10



(f) sigma=1.0, m=5, n=10, h=50

(g) sigma=0.1, m=5, n=10, h=50

(h) sigma=0.01, m=5, n=10, h=50

(i) sigma=0.1, m=5, n=10, h=10

(j) sigma=0.1, m=20, n=30, h=10

Figure 8: Dynamics of Q with U, V of different sizes and standard deviation (sigma). First row: δQ vs. t, with learning rate in [0.2, 0.1, 0.02, 0.01, 0.002, 0.001]. Second row: γ vs. lr.

Examples

For symmetries with linear group action (matrix product)

Symmetry group is a subgroup of GL_n(general linear group: invertible matrices)

T can be split into:

1. S: symmetric (scaling and hyperbolic)
2. A: antisymmetric (rotations)

S: generalizes imbalance $$\frac{dQ_{GF}}{dt} = 2x^T S\dot{x}$$

For UV:

$$Q_{GF} = \frac{1}{2}\left(\mathrm{Tr}[USU^T] - \mathrm{Tr}[V^T SV]\right)$$

Antisymmetric generators (rotations)

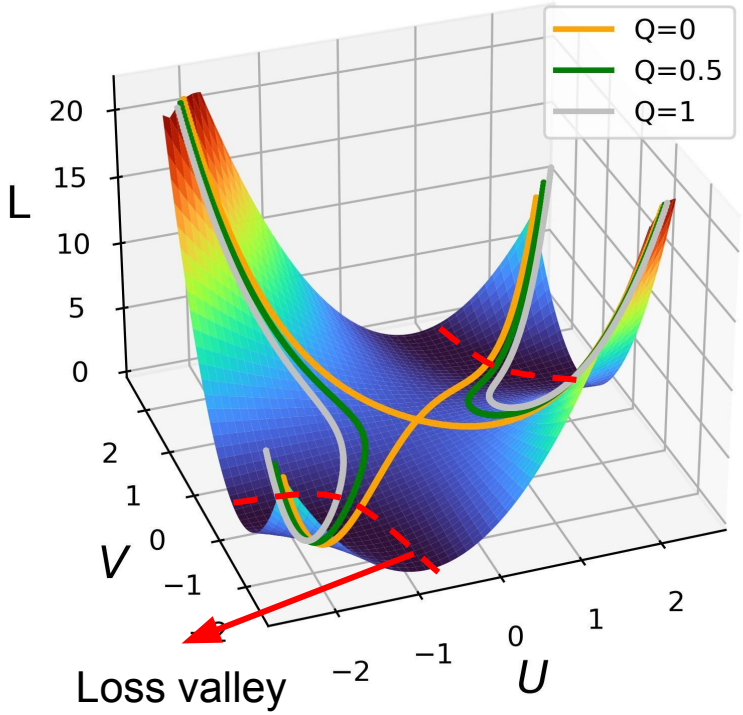
Conserved Q: a particular angle is conserved

Ex: 2D rotation

$$
\begin{aligned}
J = x^T A\dot{x} &= \begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} \\
&= x_1\dot{x_2} - x_2\dot{x_1} \\
&= (r\cos\theta)(\dot{r}\sin\theta + r\cos\theta\dot{\theta}) - (r\sin\theta)(\dot{r}\cos\theta - r\sin\theta\dot{\theta}) \\
&= r^2\dot{\theta}
\end{aligned}
$$

Conserved quantities parametrize loss valleys

- Q can be used to define coordinate along symmetry-induced loss valleys



More general nonlinear symmetries

- Won’t exist in general
- In some cases, a data-dependent symmetry can be defined

$$\mathcal{L} = e^{\gamma t}\left(\frac{|\dot{U}|^2 + |\dot{V}|^2}{2} - \gamma L(U\sigma(VX))\right)$$

$$g \cdot (U, V, X) = (g \cdot U, g \cdot V, g \cdot X) = (U\sigma(VX)\sigma(gVX)^{-1}, gV, X).$$

Noether's Learning Dynamics: Role of Symmetry Breaking in Neural Networks

Hidenori Tanaka[1*] , **Daniel Kunin**[2]
[1]Physics & Informatics Laboratories, NTT Research, Inc., Sunnyvale, CA, USA
[2]Stanford University, Stanford, CA, USA

Abstract

In nature, symmetry governs regularities, while symmetry breaking brings texture. In artificial neural networks, symmetry has been a central design principle to efficiently capture regularities in the world, but the role of symmetry breaking is not well understood. Here, we develop a theoretical framework to study the *geometry of learning dynamics* in neural networks, and reveal a key mechanism of explicit symmetry breaking behind the efficiency and stability of modern neural networks. To build this understanding, we model the discrete learning dynamics of gradient descent using a continuous-time Lagrangian formulation, in which the learning rule corresponds to the kinetic energy and the loss function corresponds to the potential energy. Then, we identify *kinetic symmetry breaking* (KSB), the condition when the kinetic energy explicitly breaks the symmetry of the potential function. We generalize Noether's theorem known in physics to take into account KSB and derive the resulting motion of the Noether charge: *Noether's Learning Dynamics* (NLD). Finally, we apply NLD to neural networks with normalization layers and reveal how KSB introduces a mechanism of *implicit adaptive optimization*, establishing an analogy between learning dynamics induced by normalization layers and RMSProp. Overall, through the lens of Lagrangian mechanics, we have established a theoretical foundation to discover geometric design principles for the learning dynamics of neural networks.

Thank you

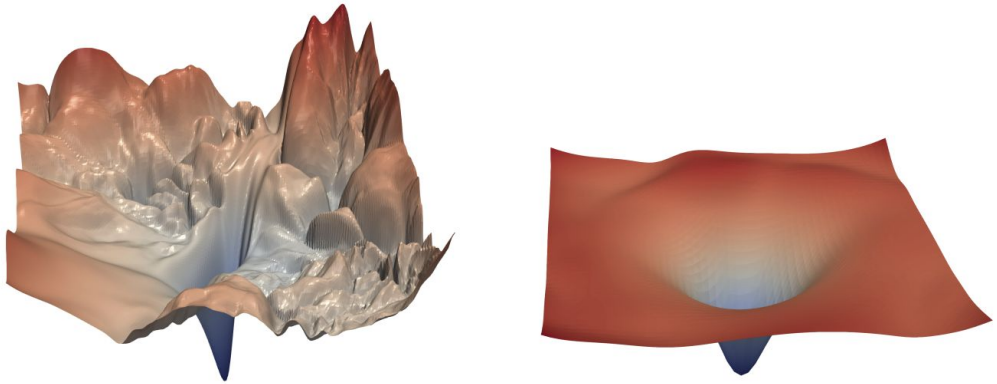
Bo Zhao
UCSD

Iordan Ganev
Radboud Uni.

Robin Walters
Northeastern Uni.

Rose Yu
UCSD

More general symmetries for nonlinear NN

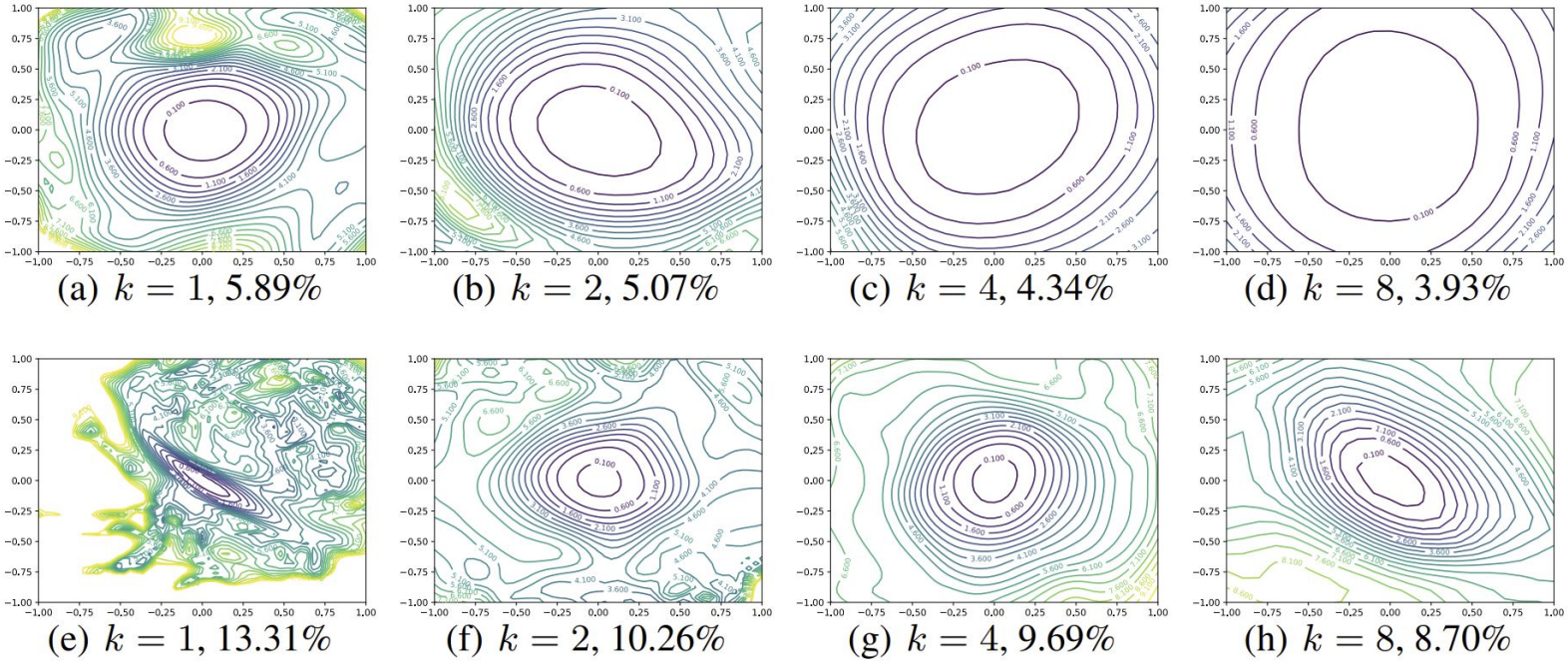


(a) without skip connections (b) with skip connections

Figure 1: The loss surfaces of ResNet-56 with/without skip connections. The proposed filter normalization scheme is used to enable comparisons of sharpness/flatness between the two figures.

Li, Hao, et al. "Visualizing the loss landscape of neural nets." *Advances in neural information processing systems* 31 (2018).

Wider models ⇒ larger flat minima



(a) $k = 1$, 5.89% (b) $k = 2$, 5.07% (c) $k = 4$, 4.34% (d) $k = 8$, 3.93%

(e) $k = 1$, 13.31% (f) $k = 2$, 10.26% (g) $k = 4$, 9.69% (h) $k = 8$, 8.70%

Figure 6: Wide-ResNet-56 on CIFAR-10 both with shortcut connections (top) and without (bottom). The label $k = 2$ means twice as many filters per layer. Test error is reported below each figure.

Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs

Timur Garipov[*1,2] **Pavel Izmailov**[*3] **Dmitrii Podoprikhin**[*4]
Dmitry Vetrov[5] **Andrew Gordon Wilson**[3]

[1]Samsung AI Center in Moscow, [2]Skolkovo Institute of Science and Technology,
[3]Cornell University,
[4]Samsung-HSE Laboratory, National Research University Higher School of Economics,
[5]National Research University Higher School of Economics

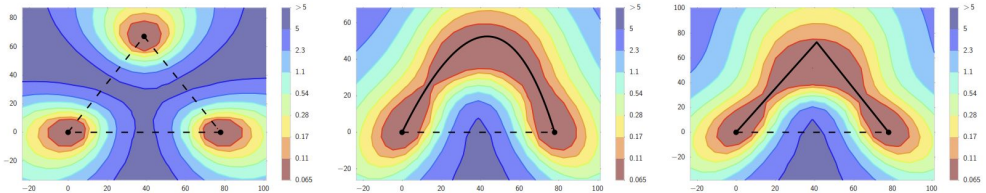


Figure 1: The ℓ_2-regularized cross-entropy train loss surface of a ResNet-164 on CIFAR-100, as a function of network weights in a two-dimensional subspace. In each panel, the horizontal axis is fixed and is attached to the optima of two independently trained networks. The vertical axis changes between panels as we change planes (defined in the main text). **Left:** Three optima for independently trained networks. **Middle** and **Right**: A quadratic Bezier curve, and a polygonal chain with one bend, connecting the lower two optima on the left panel along a path of near-constant loss. Notice that in each panel a direct linear path between each mode would incur high loss.

Abstract

The loss functions of deep neural networks are complex and their geometric properties are not well understood. We show that the optima of these complex loss functions are in fact connected by simple curves over which training and test accuracy are nearly constant. We introduce a training procedure to discover these high-accuracy pathways between modes. Inspired by this new geometric insight, we also propose a new ensembling method entitled Fast Geometric Ensembling (FGE). Using FGE we can train high-performing ensembles in the time required to train a single model. We achieve improved performance compared to the recent state-of-the-art Snapshot Ensembles, on CIFAR-10, CIFAR-100, and ImageNet.

Garipov, Timur, et al. "Loss surfaces, mode connectivity, and fast ensembling of dnns." (NIPS 2018).

Loss Surface Simplexes for Mode Connecting Volumes and Fast Ensembling

Gregory W. Benton [1] **Wesley J. Maddox** [1] **Sanae Lotfi** [1] **Andrew Gordon Wilson** [1]

Abstract

With a better understanding of the loss surfaces for multilayer networks, we can build more robust and accurate training procedures. Recently it was discovered that independently trained SGD solutions can be connected along one-dimensional paths of near-constant training loss. In this paper, we show that there are in fact mode-connecting simplicial complexes that form multi-dimensional manifolds of low loss, connecting many independently trained models. Inspired by this discovery, we show how to efficiently build simplicial complexes for fast ensembling, outperforming independently trained deep ensembles in accuracy, calibration, and robustness to dataset shift. Notably, our approach only requires a few training epochs to discover a low-loss simplex, starting from a pre-trained solution. Code is available at https://github.com/g-benton/loss-surface-simplexes.

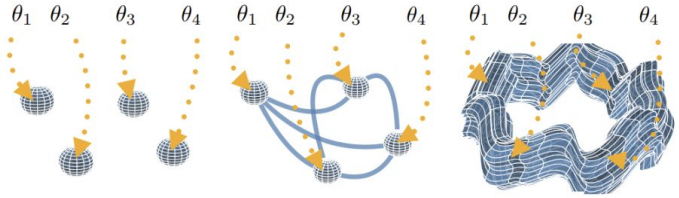


Figure 1. A progressive understanding of the loss surfaces of neural networks. **Left:** The traditional view of loss in parameter space, in which regions of low loss are disconnected (Goodfellow et al., 2015; Choromanska et al., 2015). **Center:** The revised view of loss surfaces provided by work on mode connectivity; multiple SGD training solutions are connected by narrow tunnels of low loss (Garipov et al., 2018; Draxler et al., 2018; Fort & Jastrzebski, 2019). **Right:** The viewpoint introduced in this work; SGD training converges to different points on a connected *volume* of low loss. Paths between different training solutions exist within a large multi-dimensional manifold of low loss. We provide a two dimensional representation of these loss surfaces in Figure A.1.