Mechanistic Lens on Mode Connectivity

Ekdeep Singh Lubana[124], **Eric J. Bigelow**[2], **Robert P. Dick**[1], **David Krueger**[3*], **Hidenori Tanaka**[24*]
[1]EECS Department, University of Michigan, Ann Arbor, MI, USA
[2]Center for Brain Science, Harvard University, Cambridge, MA, USA
[3]University of Cambridge, UK
[4]Physics & Informatics Laboratories, NTT Research, Inc., Sunnyvale, CA, USA
[*]Equal advising

Abstract

With the rise of pretrained models, fine-tuning has become increasingly important. However, naive fine-tuning often does not eliminate a model's sensitivity to spurious cues. To understand and address this limitation, we study the geometry of neural network loss landscapes through the lens of mode-connectivity. We tackle two questions: 1) Are models trained on different distributions mode-connected? 2) Can we fine tune a pre-trained model to switch modes? We define a notion of *mechanistic similarity* based on shared invariances and show linearly-connected modes are mechanistically similar. We find naive fine-tuning yields linearly connected solutions and hence is unable to induce relevant invariances. We also propose and validate a method of "mechanistic fine-tuning" based on our gained insights.

1 Introduction

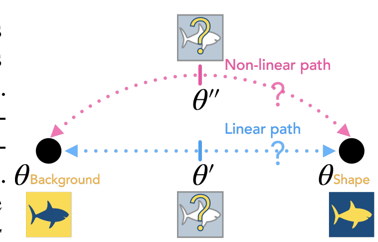


Figure 1: **Mechanistic Lens on Mode Connectivity.** Consider modes that rely on different features (highlighted yellow) to make their predictions. Are such *mechanistically dissimilar* modes connected via paths of high accuracy? Does difference in mechanisms affect the simplicity of their paths? And, can we exploit this connectivity to switch between modes?

Deep neural networks (DNNs) suffer from various shortcomings of robustness [1, 2, 3], often relying on spurious or shortcut cues that do not generalize robustly but are "simpler" to learn [4, 5]. For ex., in vision tasks, models can exploit features such as background or texture to identify object categories; however, shape-related features are likely to be more robust in practice [6, 7, 8]. Invariant prediction and related approaches [9] aim to produce such robust models by accounting for the *causal mechanisms* underlying the data generating process, hence inducing *invariance* to such "spurious" features and learning ones that generalize strongly [10]. Meanwhile, when trained on different data distributions, a model may end up learning different invariances. In this work, we introduce the notion of **mechanistic similarity** to describe models that share invariances, but may otherwise differ in their predictions. Our motivating question is whether fine-tuning can alter a model's learned invariances. Specifically, if a model has learned to rely on spurious features in its training data, can we get it to break that "bad habit" by fine-tuning it on some "clean" data that does not contain such spurious features? We consider this question through the lens of **mode-connectivity** [11, 12], which argues relatively simple paths connect DNN minimizers via paths of high accuracy or low loss.

1.1 Preliminaries / Notations

Model: Consider a neural network $f : \mathcal{X} \times \mathbb{R}^d \to \mathbb{R}^n$. The model's output decision is denoted $\hat{y}(f(x, \theta)) \in \mathcal{Y} = \{1, ..., K\}$ for input $x \in \mathcal{X} \subset \mathbb{R}^n$ and parameters $\theta \in \mathbb{R}^d$. The ground truth target is denoted $y \in \mathcal{Y}$. The model's loss on a dataset $\mathcal{D} \in \mathcal{X} \times \mathcal{Y}$ for parameter setting θ is denoted

ML Safety Workshop, 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

as $\mathcal{L}(f(\mathcal{D};\theta))$. We denote a continuous path connecting two set of parameters θ_1, θ_2 as $\gamma_{\theta_1 \to \theta_2}(t)$, where $\gamma_{\theta_1 \to \theta_2}(0) = \theta_1$, and $\gamma_{\theta_1 \to \theta_2}(1) = \theta_2$. Mode-connectivity is formalized as follows.

Definition 1. ***Mode-Connectivity (along a Path.)*** *Assume modes θ_1, θ_2 achieve $\mathcal{L}(\mathcal{D};\theta) < \epsilon$, for some small value ϵ. We call θ_1, θ_2 mode-connected along $\gamma_{\theta_1 \to \theta_2}(t)$ if moving along the path never yields increase in loss, i.e., $\forall\, t \in [0,1]$, $\mathcal{L}(f(\mathcal{D}, \gamma_{\theta_1 \to \theta_2}(t))) < \epsilon$.*

Prior works have found modes in modern neural networks' landscapes to be connected via relatively simple paths [11, 12, 13, 14]. We thus restrict our experiments to the following: (i) Linear: $\gamma_{\theta_1 \to \theta_2}(t) = t\theta_1 + (1-t)\theta_2$; (ii) Quadratic: $\gamma_{\theta_1 \to \theta_2}(t) = t^2\theta_1 + 2t(1-t)\theta_m + (1-t)^2\theta_2$. Here, θ_m denotes a set of parameters that is explicitly optimized to identify a quadratic path connecting the two modes θ_1, θ_2 (see App. B.4). We note that modes which do not appear to be linearly connected can often be linearly connected via an appropriate permutation of neurons that preserves the model's functional nature (i.e., produces the same outputs) (see App. B.5) [15, 16, 17].

Data: We assume there is a latent space $\mathcal{Z} \subset \mathbb{R}^m$ that instantiates a data-generating process (DGP) $\mathcal{G} : \mathcal{Z} \to \mathcal{X} \times \mathcal{Y}$ and induces the datapoint (x, y) as follows: $(x, y) := \mathcal{G}(z)$. X and Y are assumed conditionally independent given Z, and define $\mathcal{G}_X, \mathcal{G}_Y$ as the components of $\mathcal{G}$ producing X and Y. Similar to prior work on disentanglement [18, 19, 20, 21, 22] and nonlinear ICA [23, 24, 25, 26, 27], we assume $\mathcal{G}_X(.)$ has a valid inverse $\mathcal{G}_X^{-1} : \mathcal{X} \to \mathcal{Z}$ and the latent dimensions are statistically independent, i.e., $z_i \in \mathcal{Z}_i \perp z_j \in \mathcal{Z}_j, \forall\,(i,j), i \neq j$. To empirically validate our claims, we need the ability to intervene on spurious cues in the data and generate counterfactuals (see Def. 2). We thus propose to use synthetic datasets with known spurious cues–e.g., CIFAR-10 with a located box cue; see App. B.3 for details and visualizations.

2 Towards a Definition of Mechanistic Similarity

While a surprising result, it is unclear if connectivity emerges in modes that rely on different mechanisms (e.g., shape vs. background; see Fig. 1). To answer this, we must first design a notion of mechanistic (dis)similarity. The intuition behind our definition will be the following question: *do models that succeed in a similar manner, fail in a similar manner?* We propose to assess failures by measuring a model's response to *relevant* data transformations: if two models use similar mechanisms, they should respond similarly to transformed inputs; by choosing transforms that encode task-relevant vulnerabilities, we can make this definition operationally well-motivated. For ex., randomizing the synthetic cue in Fig. 5, we can assess whether a model relies on the cue. This is analogous to the use of visual illusions (invalid percepts) for designing models of early visual processing in neuro-/cognitive-science [28, 29]. We first define the following.

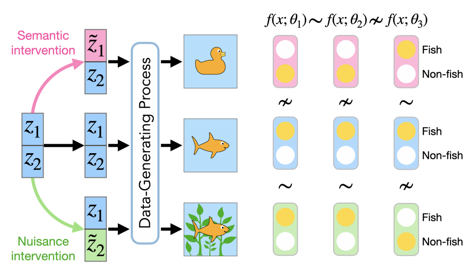


Figure 2: **Summarizing Mechanistic Similarity:** We define mechanistic similarity of two modes by assessing their response to unit interventions on the data-generating process, i.e., interventions on specific dimensions of the latent vector z (e.g., background and shape in the figure). Modes invariant to the same set of interventions (denoted $\sim$) are termed **mechanistically similar**.

Definition 2. ***(Unit Interventions and Counterfactuals.)*** *An isomorphism $\mathcal{A}_i^{\alpha_i} : \mathcal{Z}_i \times \mathcal{Z}_i \to \mathcal{Z}_i$ defines a **unit intervention** on the i^{th} dimension of the state z if it alters its value by adding a predefined scalar α_i. The isomorphism $\mathcal{E} : \mathcal{X} \times \mathbb{R}^m \to \mathcal{X}$ defines a **counterfactual** if it alters a datapoint x by changing its corresponding latent $z = \mathcal{G}_X^{-1}(x)$ via a set of unit interventions $\{\mathcal{A}\} := \{\mathcal{A}_i^{\alpha_i}\}_{i=1}^m$. Specifically, we have, $\mathcal{E}(x;\widehat{\mathcal{A}}) = \mathcal{G}_X \circ \mathcal{A}_m^{\alpha_m} \circ \cdots \circ \mathcal{A}_1^{\alpha_1} \circ \mathcal{G}_X^{-1}(x)$.*

Unit interventions on the data-generating process allow precise manipulation of a state z, while a counterfactual maps the changed state into the observable data space. Due to independence of latent dimensions, our definition of unit interventions easily composes and can model broader notions of interventions [30]; combined with counterfactuals, unit interventions are thus sufficient to assess a model's response to general data transformations, as we show below.

Definition 3. ***(Invariance.)*** *The model $f(.;\theta)$ is termed **invariant** to unit intervention $\mathcal{A}_i$ if counterfactuals generated by $\mathcal{A}_i$ do not yield increase in loss, i.e., $\mathcal{L}(f(\mathcal{D};\theta)) = \mathbb{E}_{\alpha \in \mathcal{Z}_i}\mathcal{L}(f(\mathcal{E}(\mathcal{D};\mathcal{A}_i^{\alpha});\theta))$.*

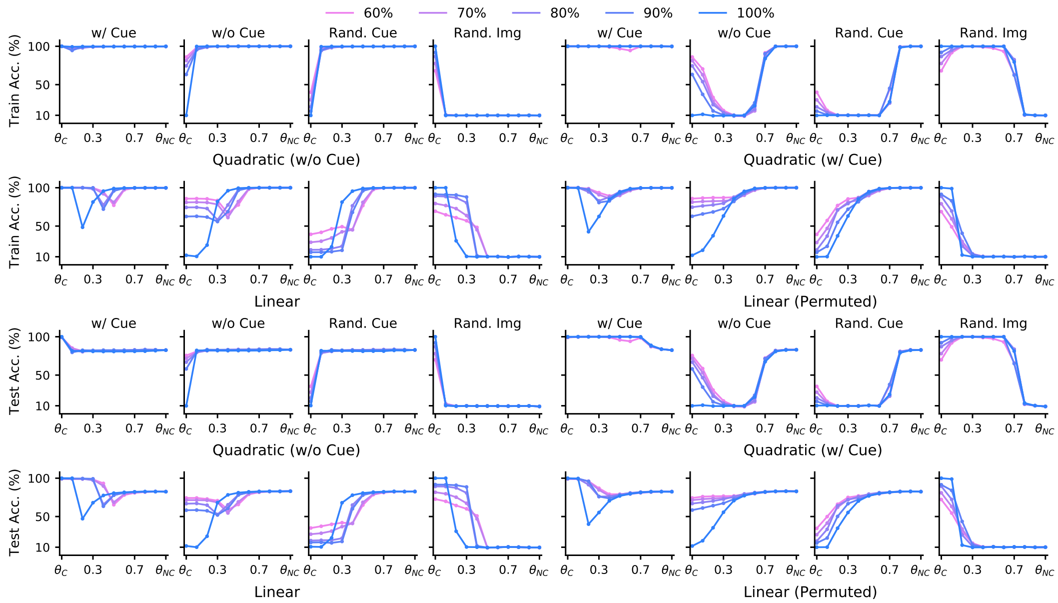


Figure 3: **Non-Linear Connectivity of Mechanistically Dissimilar Modes.** We train ResNet-18 models on our synthetic CIFAR-10 with box-cues and the original dataset (denoted $\theta_{\text{C}}, \theta_{\text{NC}}$, respectively). Line colors denote proportion of dataset that has synthetic cues. Plot titles denote train/test counterfactuals datasets, where either the cue is present (w/ Cue), absent (w/o Cue), randomized (Rand. Cue), or the underlying image is randomized (Rand. Img). We find the two modes can be connected via quadratic, but not linear, paths. Moreover, minimizers near θ_{NC} yield the same performance upon randomization of the cue, while ones near θ_{C} lose performance substantially, showing lack of shared invariances and mechanistic dissimilarity. See App. D for further results.

Lemma 1. ***(Exhaustiveness of Unit Interventions.)*** *If $f(.;\theta)$ is invariant to unit interventions $\mathcal{A}_i$ and $\mathcal{A}_j$, it must be invariant to their composition; if it is not invariant to $\mathcal{A}_i$ or $\mathcal{A}_j$, it cannot be invariant to their composition. That is, unit interventions* ***exhaustively*** *characterize a model.*

Essentially, the above lemma states that studying a model's response to unit interventions is sufficient to characterize it: if a model is invariant to a set of unit interventions, it must be invariant to their composition; similarly, lack of invariance to a single unit intervention is sufficient to preclude invariance to the composition of those interventions. The lemma thus allows us to define mechanistic similarity of two modes as sharing of invariances to the same unit interventions.

Definition 4. ***(Mechanistic Similarity.)*** *Consider a set of unit interventions $\{\mathcal{A}\} := \{\mathcal{A}_i^{\alpha_i}\}$, where $i \in [m]$. For parameters θ, denote the subset of interventions the model is invariant to as $\mathcal{I}(\theta) \subset \{\mathcal{A}\}$. Then, $f(.;\theta_1)$ and $f(.;\theta_2)$ are said to be* ***mechanistically similar*** *if $I(\theta_1) = I(\theta_2)$.*

3 Relating Mechanistic Similarity and Mode Connectivity

We first present the following proposition which follows directly from the results by [31, 32, 33] and answers the question: *is it even possible for mechanistically dissimilar modes to be connected?*

Proposition 1. ***(Mechanistically Dissimilar Modes are Connected.)*** *Assume θ_1, θ_2 are two mechanistically dissimilar modes of loss $\mathcal{L}(f(\mathcal{D};\theta))$ on a given dataset $\mathcal{D}$. Given sufficient overparamterization, there exists a continuous path that connects the two modes (in the sense of Def. 1).*

That is, even if two modes use completely different mechanisms to fit a dataset $\mathcal{D}$, as long as they achieve zero loss on it, there will exist a continuous path that connects them. However, as we mentioned before, beyond the surprising fact that modes in the landscape are connected at all, the further intriguing result is that they are connected via relatively simple paths. We empirically demonstrate that this property continues to hold for mechanistically dissimilar modes, *provided one uses non-linear connectivity paths*. Specifically, we train VGG-13 and ResNet-18 models on synthetic datasets and plot accuracy on counterfactual datasets (e.g., see Fig. 5 & App. D). We analyze quadratic paths identified using without cue data, with cue data, linear path, and linear path after permuting neurons to match in activations (see App. B.5). We see that we can identify quadratic, *but not linear*, paths that connect mechanistically dissimilar modes in the sense of Def. 1. Thus,

Table 1: We train ResNet-18 models on our synthetic CIFAR-10 with box cues and fine-tune the trained models using 2500 "clean" samples without cues. Test accuracy (%) on test counterfactuals with no Cue (NC), with Cue (C), Randomized Cue (RC), and Randomized Image (RI) is reported. We compare our method, Connectivity-Based Fine-Tuning (CBFT), against Fine-tuning with a medium/small learning rate ($FT_{M/S}$), LLR [34], and LPFT [35]. $\sim$ denotes invariance is desirable, i.e., accuracy should be similar to that on NC; $\downarrow$ indicates lower accuracy is desirable; best results are in bold. Unlike CBFT, we see all baselines yield large degradations in absence of cues and achieve very high accuracy even when the underlying image is randomized. See App. B.3 for further results.

	60% Cue data				70% Cue data				80% Cue data				90% Cue data			
C-10	$NC^{\uparrow}$	$C^{\sim}$	$RC^{\sim}$	$RI^{\downarrow}$	$NC^{\uparrow}$	$C^{\sim}$	$RC^{\sim}$	$RI^{\downarrow}$	$NC^{\uparrow}$	$C^{\sim}$	$RC^{\sim}$	$RI^{\downarrow}$	$NC^{\uparrow}$	$C^{\sim}$	$RC^{\sim}$	$RI^{\downarrow}$
FT_M	**75.7**	98.4	23.6	83.4	**75.8**	98.6	27.7	78.6	71.3	97.7	37.6	63.6	67.2	95.4	49.6	46.6
FT_S	75.8	98.7	17.5	90.1	74.9	98.8	16.3	91.1	69.9	98.4	15.7	90.9	64.7	97.9	15.3	90.7
LLR	71.6	95.1	36.3	57.1	70.9	95.8	29.9	65.8	65.1	81.8	27.0	53.2	59.3	70.7	24.6	40.7
LPFT	70.6	88.1	21.0	70.7	69.6	87.3	18.7	72.5	64.4	63.8	18.8	48.0	59.7	56.6	19.8	37.8
CBFT	74.1	**71.5**	**73.4**	**8.75**	73.2	**69.2**	**72.3**	**8.60**	70.0	**70.0**	**69.5**	**9.68**	**67.9**	**72.5**	**68.1**	**13.1**

mechanistic similarity affects the functional form of connectivity paths between modes. Moreover, we see different points along the connectivity paths respond differently to counterfactuals, indicating *lack of mechanistic similarity along the path*. Building on these results, we next argue that mechanistically similar modes must be connected via linear paths and vice versa.

Conjecture 1. ***(Mechanistic Similarity Enforces Linear Connectivity.)*** *If, up to permutations of neurons, θ_1, θ_2 show linear connectivity on a dataset $\mathcal{D}$, then they must be mechanistically similar. If they cannot be connected linearly, the modes must be mechanistically dissimilar.*

In App. F, we show the above conjecture holds locally by analyzing the landscape up to a second-order approximation. We provide extensive experiments to demonstrate the conjecture holds true in real settings (see Fig. 4 & App. E). Specifically, we train VGG-13 and ResNet-18 models on our synthetic datasets and fine-tune these trained models on the original data without cues for 100 epochs, different initial learning rates, and a step-decay schedule. We see that whenever linear connectivity is exhibited, the modes respond similarly to counterfactual datasets (i.e., they are mechanistically similar). Meanwhile, when linear connectivity does not emerge, the fine-tuned mode responds differently on counterfactuals: e.g., fine-tuning using a large learning rate exhibits clear invariance to the spurious cue, while the original mode does not. Our results thus provide nuance to prior claims that all modes are linearly connected up to permutations [15, 16]: we find the landscape is a collection of basins of linearly connected modes that follows similar mechanisms to produce their outputs. The training pipeline's inherent biases (e.g., simplicity bias in SGD [4, 5, 36]) shows preferential behavior for certain basins, due to which linear connectivity may emerge (up to permutations) for the same pipeline but slightly different settings that do not shift bias towards another basin (e.g., changed initializations). Meanwhile, two mechanistically dissimilar modes cannot be connected linearly, exhibiting an increase in loss along the linear path between them.

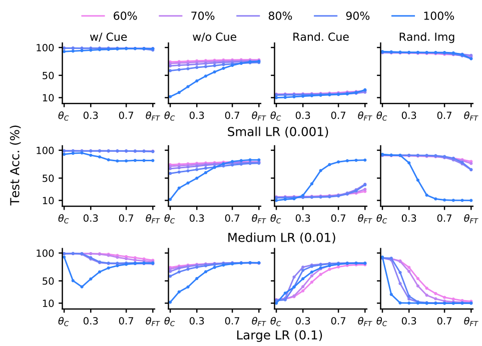


Figure 4: **Linear Connectivity and Mechanistic Similarity.** We train ResNet-18 on box cue CIFAR-10 (denoted θ_C) and fine-tune on the without cue data (θ_{FT}) using different learning rates (LR). Plots show accuracy along linear paths (after permutation matching); line colors indicate proportion of dataset with cues; titles denote evaluation data. We see that for small/medium learning rates, θ_C, θ_{FT} exhibit linear connectivity on data with cue; correspondingly, counterfactual behavior is shared, indicating mechanistic similarity. Increasing the learning rate breaks this linear connectivity; correspondingly, models respond differently to counterfactuals and are mechanistically dissimilar. See App. E for further results.

Mechanistic Fine-Tuning: Consider a mode θ_C that we want to fine-tune on some minimal "clean" data to create a mode θ_{FT} that does not use some undesirable mechanism. This premise follows from recent work on removing reliance on spurious cues in DNNs [34, 35]. We now show our developed insights can be used to address this problem (see App. C for details). Specifically, we propose to regularize the fine-tuning process to induce a high loss barrier between linear interpolation of the current state of θ_{FT} and θ_C. As per our analysis, the existence of this barrier will imply lack of shared invariances and hence mechanistic dissimilarity. To ensure the unshared invariance corresponds to ignoring vs. using the spurious cue, we further add an invariance loss that asks class-centroids produced by θ_{FT} on data with and without cue to be the same. We find this approach, termed Connectivity-Based Fine-Tuning (CBFT), outperforms recent baselines [34, 35] and naive fine-tuning on clean and counterfactual data, indicating emergence of desired invariances (see Tab. 1 & App. 3).

References

[1] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[2] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018.

[3] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *arXiv preprint arXiv:2006.09994*, 2020.

[4] Dimitris Kalimeris, Gal Kaplun, Preetum Nakkiran, Benjamin Edelman, Tristan Yang, Boaz Barak, and Haofeng Zhang. Sgd on neural networks learns functions of increasing complexity. *Advances in neural information processing systems*, 32, 2019.

[5] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33:9573–9585, 2020.

[6] Katherine Hermann, Ting Chen, and Simon Kornblith. The origins and prevalence of texture bias in convolutional neural networks. *Advances in Neural Information Processing Systems*, 33:19000–19015, 2020.

[7] Katherine Hermann and Andrew Lampinen. What shapes feature representations? exploring datasets, architectures, and training. *Advances in Neural Information Processing Systems*, 33:9995–10006, 2020.

[8] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

[9] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[10] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

[11] 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.

[12] 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.

[13] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pages 3259–3269. PMLR, 2020.

[14] Gregory Benton, Wesley Maddox, Sanae Lotfi, and Andrew Gordon Gordon Wilson. Loss surface simplexes for mode connecting volumes and fast ensembling. In *International Conference on Machine Learning*, pages 769–779. PMLR, 2021.

[15] Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. *arXiv preprint arXiv:2110.06296*, 2021.

[16] Samuel K. Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git Re-Basin: Merging Models modulo Permutation Symmetries. *arXiv preprint arXiv:2209.04836*, 2022.

[17] Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. *Advances in Neural Information Processing Systems*, 33:22045–22055, 2020.

[18] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR, 2019.

[19] Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In *International Conference on Machine Learning*, pages 6348–6359. PMLR, 2020.

[20] Luigi Gresele, Paul K Rubenstein, Arash Mehrjou, Francesco Locatello, and Bernhard Schölkopf. The incomplete rosetta stone problem: Identifiability results for multi-view nonlinear ica. In *Uncertainty in Artificial Intelligence*, pages 217–227. PMLR, 2020.

[21] Luigi Gresele, Julius Von Kügelgen, Vincent Stimper, Bernhard Schölkopf, and Michel Besserve. Independent mechanism analysis, a new concept? *Advances in neural information processing systems*, 34:28233–28248, 2021.

[22] Julius Von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-supervised learning with data augmentations provably isolates content from style. *Advances in neural information processing systems*, 34:16451–16467, 2021.

[23] Aapo Hyvarinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ica. *Advances in neural information processing systems*, 29, 2016.

[24] Aapo Hyvarinen and Hiroshi Morioka. Nonlinear ICA of temporally dependent stationary sources. In *Artificial Intelligence and Statistics*, pages 460–469. PMLR, 2017.

[25] Aapo Hyvarinen, Hiroaki Sasaki, and Richard Turner. Nonlinear ICA using auxiliary variables and generalized contrastive learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 859–868. PMLR, 2019.

[26] Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR, 2020.

[27] Ilyes Khemakhem, Ricardo Monti, Robert Leech, and Aapo Hyvarinen. Causal autoregressive flows. In *International conference on artificial intelligence and statistics*, pages 3520–3528. PMLR, 2021.

[28] Hidenori Tanaka, Aran Nayebi, Niru Maheswaranathan, Lane McIntosh, Stephen Baccus, and Surya Ganguli. From deep learning to mechanistic understanding in neuroscience: the structure of retinal prediction. *Advances in neural information processing systems*, 32, 2019.

[29] David Marr. Early processing of visual information. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, 275(942):483–519, 1976.

[30] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Towards causal representation learning. *arXiv preprint arXiv:2102.11107*, 2021.

[31] Quynh Nguyen. On connected sublevel sets in deep learning. In *International conference on machine learning*, pages 4790–4799. PMLR, 2019.

[32] Berfin Simsek, François Ged, Arthur Jacot, Francesco Spadaro, Clément Hongler, Wulfram Gerstner, and Johanni Brea. Geometry of the loss landscape in overparameterized neural networks: Symmetries and invariances. In *International Conference on Machine Learning*, pages 9722–9732. PMLR, 2021.

[33] C Daniel Freeman and Joan Bruna. Topology and geometry of half-rectified network optimization. *arXiv preprint arXiv:1611.01540*, 2016.

[34] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint arXiv:2204.02937*, 2022.

[35] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022.

[36] Mohammad Pezeshki, Oumar Kaba, Yoshua Bengio, Aaron C Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. *Advances in Neural Information Processing Systems*, 34:1256–1272, 2021.

[37] Horia Mania, John Miller, Ludwig Schmidt, Moritz Hardt, and Benjamin Recht. Model similarity mitigates test set overuse. *Advances in Neural Information Processing Systems*, 32, 2019.

[38] Pratyush Maini, Saurabh Garg, Zachary Chase Lipton, and J Zico Kolter. Characterizing Datapoints via Second-Split Forgetting. In *ICML 2022: Workshop on Spurious Correlations, Invariance and Stability*, 2022.

[39] Sara Hooker, Aaron Courville, Gregory Clark, Yann Dauphin, and Andrea Frome. What do compressed deep neural networks forget? *arXiv preprint arXiv:1911.05248*, 2019.

[40] Aron Beekman, Louk Rademaker, and Jasper van Wezel. An introduction to spontaneous symmetry breaking. *SciPost Physics Lecture Notes*, page 011, 2019.

[41] Haruki Watanabe. Counting rules of Nambu–Goldstone modes. *Annual Review of Condensed Matter Physics*, 11:169–187, 2020.

[42] Vedant Nanda, Till Speicher, Camila Kolling, John P Dickerson, Krishna Gummadi, and Adrian Weller. Measuring representational robustness of neural networks through shared invariances. In *International Conference on Machine Learning*, pages 16368–16382. PMLR, 2022.

[43] Michel Besserve, Naji Shajarisales, Bernhard Schölkopf, and Dominik Janzing. Group invariance principles for causal generative models. In *International Conference on Artificial Intelligence and Statistics*, pages 557–565. PMLR, 2018.

[44] Michel Besserve, Arash Mehrjou, Rémy Sun, and Bernhard Schölkopf. Counterfactuals uncover the modular structure of deep generative models. *arXiv preprint arXiv:1812.03253*, 2018.

[45] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *International conference on machine learning*, pages 9786–9796. PMLR, 2020.

[46] Binxu Wang and Carlos R Ponce. A Geometric Analysis of Deep Generative Image Models and Its Applications. In *International Conference on Learning Representations*, 2020.

[47] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

[48] Mitchell Wortsman, Maxwell C Horton, Carlos Guestrin, Ali Farhadi, and Mohammad Rastegari. Learning neural network subspaces. In *International Conference on Machine Learning*, pages 11217–11227. PMLR, 2021.

[49] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pages 23965–23998. PMLR, 2022.

[50] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? *Advances in neural information processing systems*, 33:512–523, 2020.

[51] Pu Zhao, Pin-Yu Chen, Payel Das, Karthikeyan Natesan Ramamurthy, and Xue Lin. Bridging mode connectivity in loss landscapes and adversarial robustness. *arXiv preprint arXiv:2005.00060*, 2020.

[52] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Dilan Gorur, Razvan Pascanu, and Hassan Ghasemzadeh. Linear mode connectivity in multitask and continual learning. *arXiv preprint arXiv:2010.04495*, 2020.

[53] Jeevesh Juneja, Rachit Bansal, Kyunghyun Cho, João Sedoc, and Naomi Saphra. Linear Connectivity Reveals Generalization Strategies. *arXiv preprint arXiv:2205.12411*, 2022.

[54] Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitriy Pyrkin, Sergei Popov, and Artem Babenko. Editable neural networks. *arXiv preprint arXiv:2004.00345*, 2020.

[55] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*, 2021.

[56] Shibani Santurkar, Dimitris Tsipras, Mahalaxmi Elango, David Bau, Antonio Torralba, and Aleksander Madry. Editing a classifier by rewriting its prediction rules. *Advances in Neural Information Processing Systems*, 34:23359–23373, 2021.

[57] Andrea Dittadi, Frederik Träuble, Francesco Locatello, Manuel Wüthrich, Vaibhav Agrawal, Ole Winther, Stefan Bauer, and Bernhard Schölkopf. On the transfer of disentangled representations in realistic settings. *arXiv preprint arXiv:2010.14407*, 2020.

[58] Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre Alvise-Rebuffi, Ira Ktena, Taylan Cemgil, et al. A fine-grained analysis on distribution shift. *arXiv preprint arXiv:2110.11328*, 2021.

[59] Sjoerd Van Steenkiste, Francesco Locatello, Jürgen Schmidhuber, and Olivier Bachem. Are disentangled representations helpful for abstract visual reasoning? *Advances in Neural Information Processing Systems*, 32, 2019.

[60] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *International Conference on Learning Representations*, 2017.

[61] David Klindt, Lukas Schott, Yash Sharma, Ivan Ustyuzhaninov, Wieland Brendel, Matthias Bethge, and Dylan Paiton. Towards nonlinear disentanglement in natural data with temporal sparse coding. *arXiv preprint arXiv:2007.10930*, 2020.

[62] Puja Trivedi, Ekdeep Singh Lubana, Mark Heimann, Danai Koutra, and Jayaraman J Thiagarajan. Analyzing Data-Centric Properties for Contrastive Learning on Graphs. *arXiv preprint arXiv:2208.02810*, 2022.

[63] Samuel Ritter, David GT Barrett, Adam Santoro, and Matt M Botvinick. Cognitive psychology for deep neural networks: A shape bias case study. In *International conference on machine learning*, pages 2940–2949. PMLR, 2017.

[64] SciPy. SciPy Documentation. `https://docs.scipy.org/doc/scipy-0.18.1/reference/generated/scipy.optimize.linear_sum_assignment.html`, 2016.

[65] Robert Hecht-Nielsen. On the algebraic structure of feedforward network weight spaces. In *Advanced Neural Computers*, pages 129–135. Elsevier, 1990.

[66] Norman Tatro, Pin-Yu Chen, Payel Das, Igor Melnyk, Prasanna Sattigeri, and Rongjie Lai. Optimizing mode connectivity via neuron alignment. *Advances in Neural Information Processing Systems*, 33:15300–15311, 2020.

A Related Work

Model Similarity: Prior works have tried to assess similarity of two models trained under different settings via the notion of prediction mismatch [37, 38, 39], which involves finding which samples two models produce different outputs on. In contrast to such works, we propose to assess model simlarity via their response to *relevant* data transformations, where relevance depends on the task and user choices. If two models use similar mechanisms, they should respond similarly to transformed inputs; by choosing transforms that encode task-relevant vulnerabilities, we can make this definition operationally well-motivated. For example, by randomizing the synthetic cue in Figure 5, we can assess whether the model relies on the cue or not. This is analogous to the use of visual illusions (inaccurate perceptual inferences) for determining valid computational models of early visual processing in cognitive science [29] and neuroscience [28], and relates to the idea of Goldstone's theorem in physics [40, 41], which claims moving in the direction of an operator's symmetry does not yield increase in system's energy. Intuitively, then, we can expect the number of shared symmetries to be a valid proxy for how benign a path must be for two mechanistically dissimilar modes to be connected. We also note our work is among the first works to examine the mechanistic similarity of different networks. While many previous works have asked whether a neural network learns the true causal mechanisms generating the data, ours is the second work to examine the mechanistic similarity of different networks, after Nanda et al. [42], who introduce a method ("STIR") for measuring how similar the mechanisms learned by two networks are. We also note the idea of identifying easily manipulable latents to infer interpretable modules is also popular in generative models [43, 44, 45, 46].

Mode connectivity. Existence of a single, continuous manifold connecting global minimizers was first identified theoretically by [33] and empirically discovered in concurrent works under the title of "mode connectivity" by [11] and [12]. A geometrical characterization of the manifold connecting minimizers was provided by [32], who showed the manifold is *primarily* composed of affine subspaces. Mode connectivity results have been used for designing and analyzing algorithms for several practically relevant applications, such as ensembling [14, 47, 48, 49], network pruning [13, 15], fine-tuning [50], adversarial robustness [51], and multi-task/continual learning [52]. This last work is closely related to ours: they find that they can train a model which is linearly mode connected to models trained on individual tasks. During the course of this work, we became aware of the contemporary paper by [53], who empirically investigate if minimizers connected via linear paths follow similar strategies to produce their decisions. Their analysis focuses on NLP tasks and follows an alternative analysis; hence, their results can be regarded as further verification of our claims about linear mode connectivity on a different modality.

Fine-tuning. Fine-tuning is a well-established practice in deep learning. The most basic fine-tuning method is to treat the pre-trained model as an initialization, and continue training with new data. A variant is to train only a subset of parameters, such as the final classification layer [34].While [34] argue that "last layer re-training is sufficient for robustness to spurious correlations", our more in-depth evaluation shows that this method actually does not eliminate sensitivity to spurious features. Our findings are more congruent with those of [50], who find that fine-tuned models tend to remain linearly connected to the pretraining mode, suggesting that fine-tuning may fail to make fundamental changes to a model's behavior.

Model editing. Model editing refers to fine-tuning approaches that aim to make a targeted change to a particular aspect of model's behavior without incidentally affecting other aspects. For instance, [54] give the example of correcting a prediction errors on a particular example without changing a model's prediction on other examples. Most work to date on model editing aims to make such changes that are "local" in input space, i.e. only affecting the model's "understanding" of who the current prime minister of the UK is [55]. Mechanistic fine-tuning shares this high-level goal of "targeted" fine-tuning, however, we aim to make edits that are local in a different sense: we want to make a targeted change to the causal mechanism the model implements to make predictions. Specifically, we aim to make a model invariant to (e.g. spurious/nuisance) features that is was not already invariant to (or vice versa), without changing its learned representations of the features themselves, or its invariance to other features. Such a change would tend to influence many of a model's predictions, making approaches such as MEND [55] inappropriate for our setting. We believe [56] is the work most similar to ours in this respect; three significant differences are: (i) their method only works

	Linearly Mode Connected	Nonlinearly Mode Connected	Linearly Mech. Connected	Nonlinearly Mech. Connected
Mechanistically Similar	✓	✓	✓	✓
Mechanistically Dissimilar	✗	✓	✗	✗

Table 2: Summary of our (empirical) findings regarding mode connectivity of mechanistically (dis)simmilar modes. The ✓mark indicates that such (pairs of) models exhibit that type of connectivity.

given a large model pre-trained on lots of data, (ii) they rely on style transfer, (iii) they only tune some of the layers.

B Training Setup and Dataset Visualizations

B.1 Training details

When training from scratch (e.g., in Fig. 3), we train models for 100 epochs with a batch-size of 256. Learning rate starts at 0.1 and is dropped by a factor of 10 at the 40^{th} and 80^{th} epochs. No data augmentations are used. When fine-tuning to assess linear connectivity (e.g., in Fig. 4), we train models for a further 100 epochs on data without cues using different initial learning rates, but the same step-decay schedule (decay factor of 0.1 at decay epochs 40 and 80). When using synthetic datasets, if a proportion c is to be assigned the cue feature, we use the first $c\%$ samples of all classes to assign them the respective cues. We do not store the samples beforehand and use manually designed dataloaders that allow for easy manipulation of samples in an online manner, enabling straightforward counterfactual evaluations. *We note the code is currently under review, but will be released soon.*

B.2 Fine-Tuning details

We train models on the synthetic data with cue features, reserving 2500 training samples as "clean" data for further fine-tuning to remove reliance on the spurious cue. Depending on the method, the fine-tuning setup involves different hyperparameters. For consistency, we follow the use of a cosine schedule for fine-tuning on clean data, as done by Kirichenko et al. [34] and Kumar et al. [35].

Naive Fine-Tuning. We use different initial learning rates, including medium (0.01) and small (0.001). For a large learning rate, we note that while fine-tuning on a minimal set does induce good invariance properties, the performance on the original, without cue data (called NC in tables) is often rather poor. Hence, we disinclude those results.

LLRT (Kirichenko et al. [34]). We freeze the model parameters at their current state, remove the final linear layer, and replace it with a randomly initialized one. The layer is fine-tuned on clean data for 100 epochs with a cosine decay schedule that starts at a LR of 30.

LPFT (Kumar et al. [35]). First, we follow the protocol above for LLRT and get a new linear layer. Thereafter, the entire model is fine-tuned on clean data for 20 epochs with initial learning rates of 0.01, 0.001, and 0.0001. The best retrieved results on validation data are reported.

CBFT. We initialize the model at the spurious mode θ_{C} and first run a warmup epoch without the invariance loss (i.e., only the barrier loss and clean data loss are used). This is similar to the LLRT step in LPFT and helps move the model away from θ_{C}. Thereafter, the model is fine-tuned for 20 epochs with an initial learning rate of 0.01.

B.3 Data Generating Process and Visualizations

Since our goal is to assess the role of mechanistic similarity on connectivity of modes, we need models that we know for a fact rely on different mechanisms for making their decisions. To this end, we propose to use easily manipulable synthetic datasets. Such datasets have been used by prior works for better understanding several important topics, such as transfer learning [57], domain generalization [58, 59], disentanglement [60, 61], self- and semi-supervised learning [22, 62, 19], and inductive biases of neural networks [6, 7, 63]. Our data generation process (DGP) is illustrated in Figure 5 and involves augmenting the natural DGP with *synthetic cue features* that are conditioned on the sample category. By design, the dataset is easy to intervene on for creating counterfactual samples (see Def. 2). We specifically focus on interventions which break the cue's correlation with respect to target category by randomizing it (e.g., uniformly changing location of the box cue in the "located CIFAR-10" dataset). Vice-versa, we also analyze the case where the cue remains intact, but the underlying image is randomized (e.g., putting a cat instead of a dog). This helps analyze how much a given mode relies on possibly semantically relevant features that originate from the source image, especially under partial correlation. We highlight that such low-complexity cues can be viewed as stand-ins for spurious or shortcut features that are commonplace in realistic settings [2, 1], allowing us to analyze if modes that use spurious versus non-spurious features are connected.

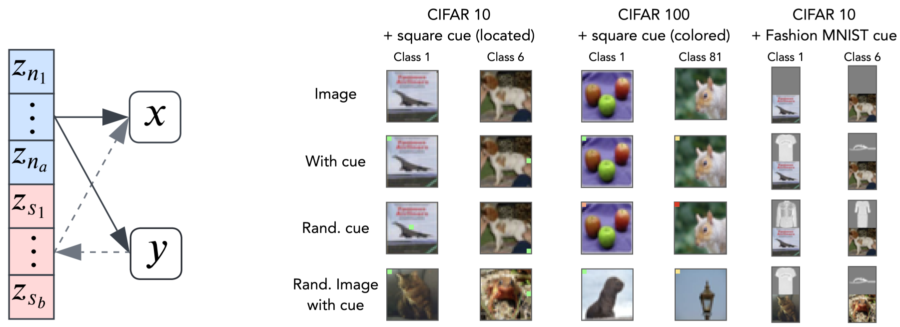


Figure 5: **Data-Generating Process (left).** We design evaluation datasets by augmenting the latents of the natural DGP (z_n) with synthetic cues (z_s). By conditioning (denoted grey, dotted line) the values of these cues on sample category (y), we can induce correlation between the desired output of a model and input features that are irrelevant to the task. If the cues are designed to be linearly separable, the simplicity bias of neural networks [5] will ensure the model preferentially uses them for making its decisions. **Datasets (right).** We use three datasets: (1) CIFAR-10 with a 3×3 box whose location depends on the sample category; (2) CIFAR-100 with 3×3 boxes colored according to the first digit of the object label, and located according to the second digit; (3) and Dominoes [5], wherein CIFAR-10 images are concatenated same class FashionMNIST images.

B.4 Identifying Quadratic Paths

The qudaratic path is defined as follows.

$$\gamma_{\theta_1 \to \theta_2}(t) = t^2\theta_1 + 2t(1-t)\theta_m + (1-t)^2\theta_2. \tag{1}$$

The set of parameters θ_m can be thought of as vertex of a parabola that helps anchor the curve. To identify this set of parameters, we follow [11] and train points randomly sampled from the quadratic path to achieve zero loss on a given dataset $\mathcal{D}$. That is,

$$\theta_m = \underset{\theta}{\operatorname{argmin}}\, \mathbb{E}_{x \in \mathcal{D}, t \in [0,1]}(\mathcal{L}(f(x; \gamma_{\theta_1 \to \theta_2)(t)})). \tag{2}$$

Note that consequently, a quadratic connectivity path necessarily depends on the dataset used for its identification and it is not mandatory that it will generalize across datasets/distributions. This is precisely what we see in our results in Figure 3, where we are able to identify quadratic mode-connectivty between two sets of parameters on a given dataset, but those paths do not generalize to counterfactual datasets.

B.5 Finding Permutations for Linear Connectivity

Given two sets of parameters θ_1, θ_1, identifying the linear path between them involves merely interpolating the parameters. [15] argue that parameters discovered using SGD can always be linearly mode connected up to permutations, i.e., there exists a permutation π that connects the $\pi(\theta_1), \theta_2$ in the sense of Def. 1. To empirically analyze this claim in our work, we identify π by maximizing the similarity of activations produced by model with parameters θ_1 and θ_2. That is,

$$\pi^* = \arg\min_{\pi} ||f(x; \pi(\theta_1)) - f(x; \theta_2)||. \tag{3}$$

Given the above problem involves discrete optimization, we propose to solve it greedily by computing representations at each layer of the two models, finding a permutation that matches the representations maximally, and then repeating the process for the next layer. For finding the permutation, we use SciPy's linear assignment solver [64]. We use representations over a batch-size of 512 and run the matching process over the entire dataset.

Let $\pi_{<l}$ denotes permutations before layer l.

Initialize: θ_1, θ_2, dataset $\mathcal{D}$, model f, ${\pi_l}_1^L = I$, where I is identity permutation.
$l \leftarrow 1$
$L \leftarrow \#$ of Layers
for $x \in \mathcal{D}$ **do**
  $l \leftarrow 1$
  for $l \leq L$ **do**
    $\pi_l = \arg\min_\pi ||f_l(x; \pi_{<l}(\theta_1)) - f_l(x; \theta_2)||$ ▷ f_l denotes representation at layer l
    $l \leftarrow l + 1$
  end for
end for

Our algorithm is able to recover the ground-truth permutation if a given model's neurons are intentionally randomized (while maintaining functional connectivity). This serves as a sanity check that confirms the validity of our technique.

C Mechanistic Fine Tuning: Overriding Decision-Making Rules by Driving a Model Beyond Mechanistic Barrier

Fine tuning is commonly used as a strategy to improve sample efficiency by taking a pre-trained model as an initialization and training it further on a new small data set. This section explores the idea of **mechanistic fine tuning**, where we specifically aim to override existing mechanisms of a pretrained model by fine tuning it on a small out-of-distribution dataset. Note that here we are not trying to teach the model how to perform new extra tasks (e.g., classifying new image categories, etc.), but rather attempting to override its existing mechanisms, e.g., for how it classifies images with a fixed set of classes.

Mechanistic fine-tuning with connectivity-based fine-tuning (CBFT) Consider a set of parameters θ_{C} found by training a model $f(.)$ on dataset $\mathcal{D}_{\text{C}}$ that has some spurious or shortcut cue that a model can use to achieve zero loss. Practically, this situation is commonplace because spurious or shortcut cues are often highly correlated with the actual target category [2]. Recent work has aimed to use minimal data with "clean" samples, i.e., samples which lack the spurious cue to remove the model's reliance on such cues [34, 35], while trying to maintain any other useful features it has learned. We now show that our newfound understanding of neural network loss landscapes from the perspective of mechanistic similarity can be used to address this problem. To this end, we recall from Figure 4 that when fine-tuning models with different learning rates on original datasets, the only situation θ_{FT} demonstrates invariance to cues embedded in the dataset is when it is not linearly connected to the original mode θ_{C}. As per Conjecture 1, this lack of linear connectivity indicates θ_{FT} and θ_{C} are mechanistically dissimilar due to lack of a shared invariance. We thus argue that **a valid strategy to remove a model's reliance on spurious cues is by forcing it to move to a different region in the landscape that does not exhibit linear connectivity to the** θ_C.

Operationalizing this idea has a challenge however: there can be multiple regions that do not linearly connect with θ_{C} and we speficially want ones that boast our desired invariances–i.e., regions that are invariant to the spurious cue. We propose to circumvent this issue by assuming the existence of a minimal "clean" dataset $\mathcal{D}_{\text{NC}}$, similar to prior work, that can be used to enforce desired invariances on the model's representations. Specifically, assume $\mathcal{L}(.,.)$ denotes a classification loss, such as cross-entropy and $\mathcal{D}^i$ denotes the subset of a dataset $\mathcal{D}$ corresponding to samples that belong to the i^{th} class in an $K-$class classification problem. Then, we can use the following two-step procedure for connectivity-based fine tuning (CBFT):

$$\begin{aligned} &\text{(i)} \quad \mathcal{L}_1(\theta) = \arg\min_\theta |\lambda_1 - \mathcal{L}_{\text{CE}}(\hat{y}(\mathcal{D}_{\text{C}}; \gamma_{\theta\to\theta_{\text{C}}}(t)), y)|; \\ &\text{(ii)} \quad \mathcal{L}_2(\theta) = \arg\min_\theta \mathcal{L}_{\text{CE}}(\hat{y}(\mathcal{D}_{\text{NC}}; \theta), y) + \frac{\lambda_2}{K} \left\| \mathbb{E}_{x\in\mathcal{D}_{\text{C}}^k}(f(x;\theta)) - \mathbb{E}_{x\in\mathcal{D}_{\text{NC}}^k}(f(x;\theta)) \right\|. \end{aligned} \tag{4}$$

In the above, $\gamma_{\theta\to\theta_{\text{C}}}(t)$ denotes the linear path between θ and θ_{C}. The first step aims to maximize the model loss $\mathcal{L}_{\text{CE}}$ along the linear path up to an upper bound λ_1; meanwhile, the second loss aims to ensure the model predicts correct labels on the clean dataset $\mathcal{D}_{\text{NC}}$, while simultaneously promoting invariance to spurious cues by producing the same average representation across all classes on both $\mathcal{D}_{\text{NC}}$ and $\mathcal{D}_{\text{C}}$. We note the set of parameters θ is initialized to θ_{C}; the method turns out to be fairly robust to the values of λ_1 and λ_2, so we set both to 1 and never tune them.

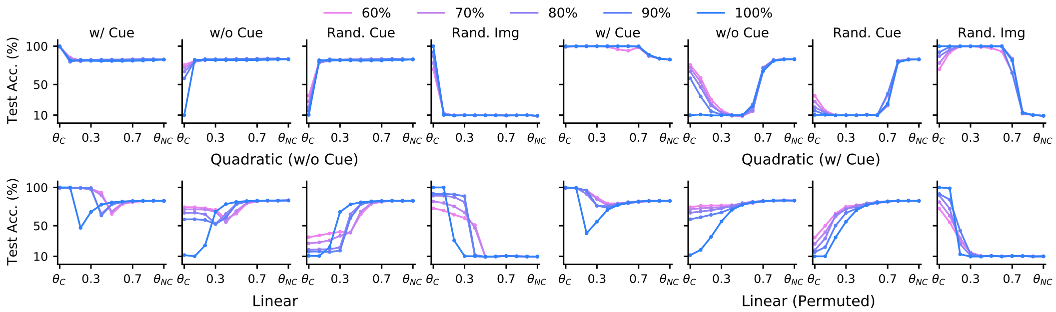
In Table 3, we empirically validate CBFT against recent baselines used for removing a model's reliance on spurious cues. As we show, while these methods are performant, they do not work well on counterfactuals datasets, e.g., they continue to perform well even if we randomize the image! In contrast, we see that not only CBFT performs better on clean data, but it in fact shows the desired behaviors: sensitivity to randomization of the image and invariance to spurious cues.

Table 3: We train ResNet-18 models on our synthetic CIFAR-10 with box cues and fine-tune the trained models using 2500 "clean" samples without cues. Test accuracy (%) on test counterfactuals with no Cue (NC), with Cue (C), Randomized Cue (RC), and Randomized Image (RI) is reported. We compare our method, Connectivity-Based Fine-Tuning (CBFT), against Fine-tuning with a medium/small learning rate ($\text{FT}_{\text{M/S}}$), LLR [34], and LPFT [35]. $\sim$ denotes invariance is desirable, i.e., accuracy should be similar to that on NC; $\downarrow$ indicates lower accuracy is desirable; best results are in bold. We generally see that all baselines yield large degradations in its absence of cues; even achieving very high accuracy when the underlying image is randomized. Meanwhile, CBFT is able to break reliance on cues, inducing representations that are often completely invariant to its presence.

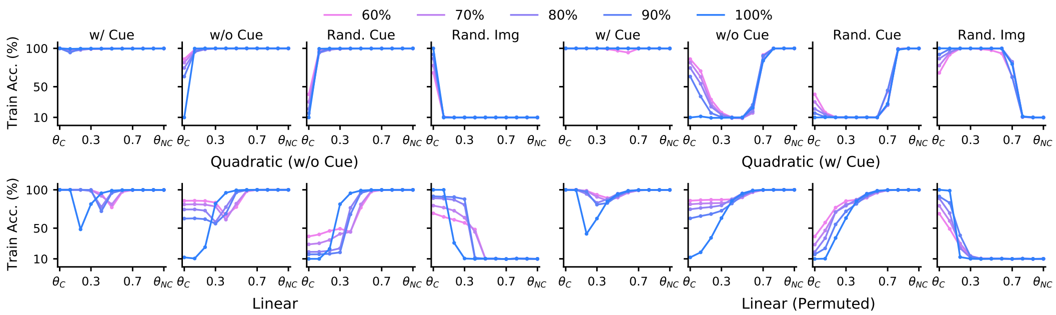
	60% Cue data				70% Cue data				80% Cue data				90% Cue data			
C-10	NC$^{\uparrow}$	C$^{\sim}$	RC$^{\sim}$	RI$^{\downarrow}$	NC$^{\uparrow}$	C$^{\sim}$	RC$^{\sim}$	RI$^{\downarrow}$	NC$^{\uparrow}$	C$^{\sim}$	RC$^{\sim}$	RI$^{\downarrow}$	NC$^{\uparrow}$	C$^{\sim}$	RC$^{\sim}$	RI$^{\downarrow}$
FT_{M}	**75.7**	98.4	23.6	83.4	**75.8**	98.6	27.7	78.6	71.3	97.7	37.6	63.6	67.2	95.4	49.6	46.6
FT_{S}	75.8	98.7	17.5	90.1	74.9	98.8	16.3	91.1	69.9	98.4	15.7	90.9	64.7	97.9	15.3	90.7
LLR	71.6	95.1	36.3	57.1	70.9	95.8	29.9	65.8	65.1	81.8	27.0	53.2	59.3	70.7	24.6	40.7
LPFT	70.6	88.1	21.0	70.7	69.6	87.3	18.7	72.5	64.4	63.8	18.8	48.0	59.7	56.6	19.8	37.8
CBFT	74.1	**71.5**	**73.4**	**8.75**	73.2	**69.2**	**72.3**	**8.60**	70.0	**70.0**	**69.5**	**9.68**	**67.9**	**72.5**	**68.1**	**13.1**
C-100	NC$^{\uparrow}$	C$^{\sim}$	RC$^{\sim}$	RI$^{\downarrow}$	NC$^{\uparrow}$	C$^{\sim}$	RC$^{\sim}$	RI$^{\downarrow}$	NC$^{\uparrow}$	C$^{\sim}$	RC$^{\sim}$	RI$^{\downarrow}$	NC$^{\uparrow}$	C$^{\sim}$	RC$^{\sim}$	RI$^{\downarrow}$
FT_{m}	**44.4**	99.2	12.8	85.3	**40.3**	99.6	12.3	89.8	33.6	99.0	11.4	90.5	25.2	79.2	9.79	57.9
FT_{s}	43.1	99.6	10.3	93.6	38.2	99.7	10.5	95.7	32.5	99.6	10.4	97.0	24.5	39.4	4.87	30.9
LLR	35.5	99.2	12.1	89.0	31.5	98.6	11.3	89.6	25.3	96.7	10.6	89.4	18.9	75.1	9.1	58.7
LPFT	35.1	93.2	10.3	82.3	31.1	90.2	9.89	78.5	25.6	89.6	9.70	80.8	18.7	**28.6**	**4.42**	**19.6**
CBFT	42.7	**65.0**	**36.4**	**14.6**	38.5	**66.7**	**34.7**	**21.2**	**34.6**	**69.3**	**23.0**	**27.9**	**28.5**	72.9	23.2	46.0
Dom.	NC$^{\uparrow}$	C$^{\sim}$	RC$^{\sim}$	RI$^{\downarrow}$	NC$^{\uparrow}$	C$^{\sim}$	RC$^{\sim}$	RI$^{\downarrow}$	NC$^{\uparrow}$	C$^{\sim}$	RC$^{\sim}$	RI$^{\downarrow}$	NC$^{\uparrow}$	C$^{\sim}$	RC$^{\sim}$	RI$^{\downarrow}$
FT_{m}	**77.4**	96.8	43.8	56.1	**76.6**	96.6	42.7	58.7	**74.1**	95.7	41.7	61.3	**68.8**	95.1	40.0	57.5
FT_{s}	76.4	96.9	37.5	62.4	76.8	96.6	32.5	66.5	73.2	96.4	30.8	67.7	67.3	95.2	31.2	65.6
LLR	74.6	94.4	39.8	53.0	73.9	93.2	36.3	54.7	70.8	84.8	33.1	46.6	63.3	77.0	31.2	39.0
LPFT	73.2	92.5	38.0	51.8	72.7	88.0	34.8	50.9	69.4	34.8	33.1	39.1	61.2	60.8	31.2	26.6
CBFT	72.0	**64.9**	**67.5**	**9.9**	71.5	**70.0**	**59.2**	**12.1**	70.8	**69.7**	**65.9**	**11.9**	67.2	**68.7**	**61.5**	**14.9**

D Non-Linear Connectivity of Mechanistically Dissimilar Modes

We train VGG-13 and ResNet-18 models on our synthetic CIFAR-10 / CIFAR-100 / Dominoes datasets with cue features (see subsection B.3) and the original datasets themselves. Parameters of the corresponding models are denoted $\theta_{\text{C}}, \theta_{\text{NC}}$. We identify connectivity paths along pairs of parameters, specifically evaluating quadratic paths identified using the data without cue (denoted Quadratic w/o Cue), quadratic path identified using data with cue (denoted Quadratic w/ Cue), linear path (denoted Linear), and linear path after permuting θ_{C} to maximally match θ_{NC}'s activations (denoted Linear Permuted). In the following, plot titles denote evaluation dataset, including datasets where either the cue is present (denoted w/ Cue), absent (denoted w/o Cue), randomized (denoted Rand. Cue), or the underlying image is randomized but the cue remains the same (denoted Rand. Img). Line colors denote the proportion of dataset that has synthetic cues. Across all our results, we see the set of parameters θ_{NC} yields the same performance upon randomization of the cue, while θ_{C} loses its performance substantially–i.e., the two modes are mechanistically dissimilar due to lack of shared invariances (see Def. 4). Nonetheless, we can identify quadratic (but not linear) paths that connect these mechanistically dissimilar modes in the sense of Def. 1, hence corroborating Prop.1 across several datasets and model architectures, and showing *mechanistically dissimilar modes can be connected via relatively simple paths as well*. However, different points on the connectivity paths respond differently to counterfactuals, indicating *mechanistic dissimilarity* despite connectivity via low-loss paths.

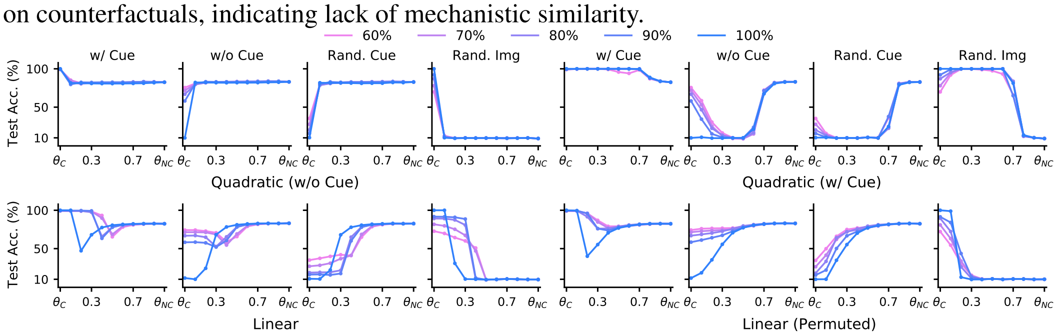


(a) Test Accuracy.

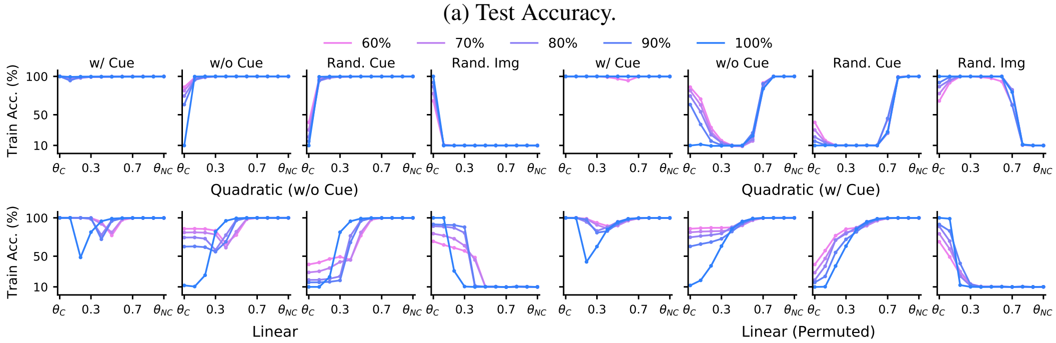


(b) Train Accuracy.

Figure 6: **VGG-13 on CIFAR-10 with Box Cue**. We plot test/train accuracy curves along different connectivity paths and see thorough corroboration of our claims in the main text: Mechanistically dissimilar minimizers can be connected via nonlinear paths on a given dataset, but behave different on counterfactuals, indicating lack of mechanistic similarity.

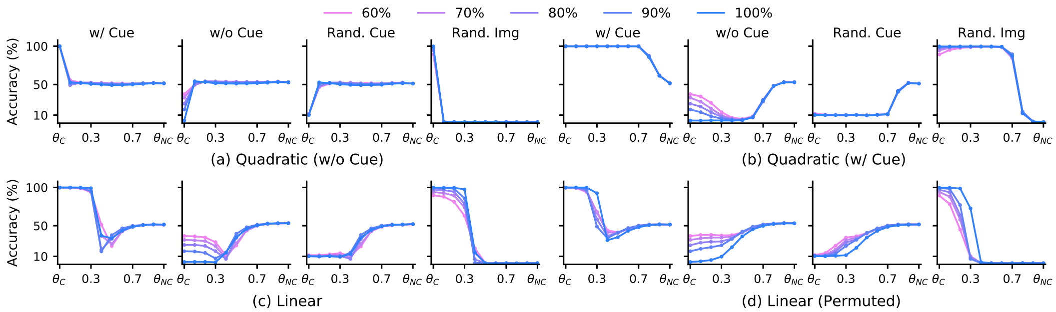


(a) Test Accuracy.

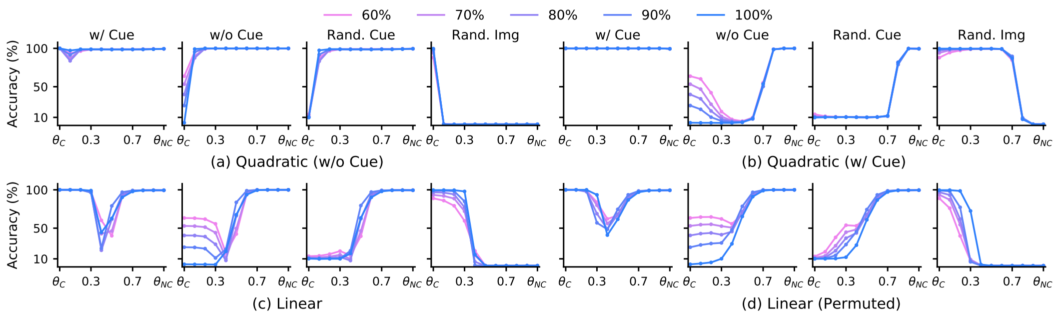


(b) Train Accuracy.

Figure 7: **ResNet-18 on CIFAR-10 with Box Cue**. We plot test/train accuracy curves along different connectivity paths and see thorough corroboration of our claims in the main text: Mechanistically dissimilar minimizers can be connected via nonlinear paths on a given dataset, but behave different on counterfactuals, indicating lack of mechanistic similarity.

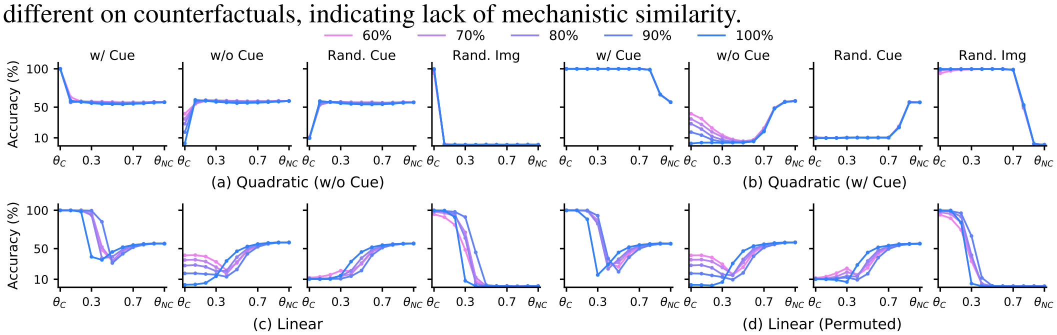


(a) Test Accuracy.

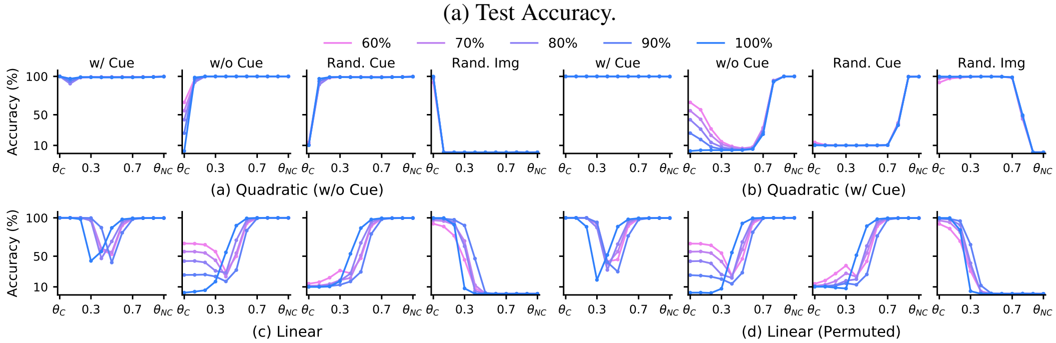


(b) Train Accuracy.

Figure 8: **VGG-13 on CIFAR-100 with Box/Color Cue**. We plot test/train accuracy curves along different connectivity paths and see thorough corroboration of our claims in the main text: Mechanistically dissimilar minimizers can be connected via nonlinear paths on a given dataset, but behave different on counterfactuals, indicating lack of mechanistic similarity.

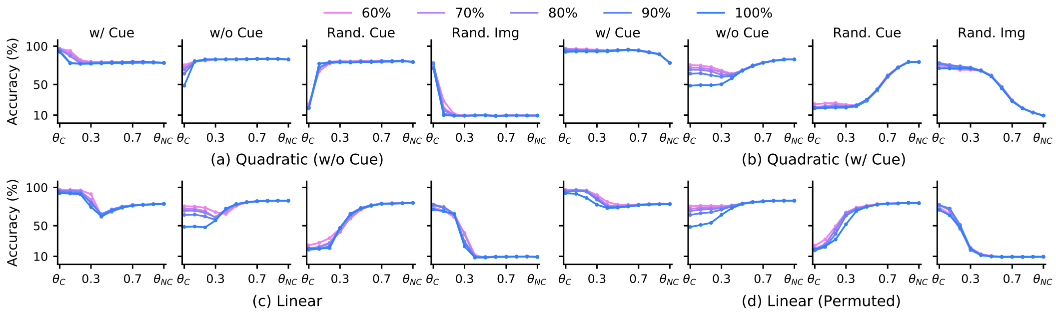


(a) Test Accuracy.

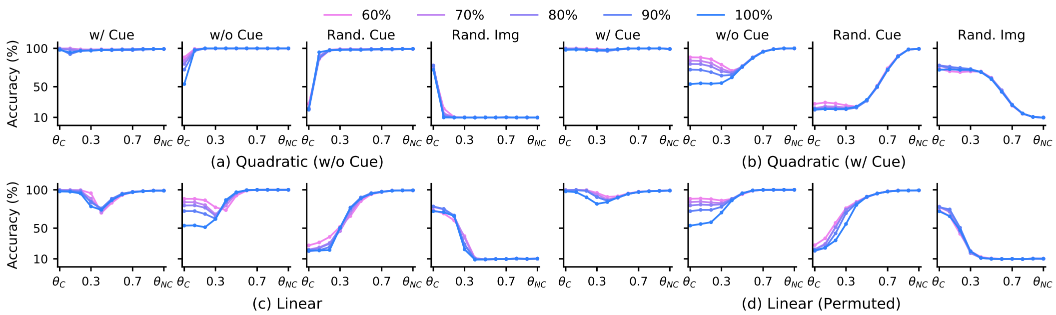


(b) Train Accuracy.

Figure 9: **ResNet-18 on CIFAR-100 with Box/Color Cue**. We plot test/train accuracy curves along different connectivity paths and see thorough corroboration of our claims in the main text: Mechanistically dissimilar minimizers can be connected via nonlinear paths on a given dataset, but behave different on counterfactuals, indicating lack of mechanistic similarity.

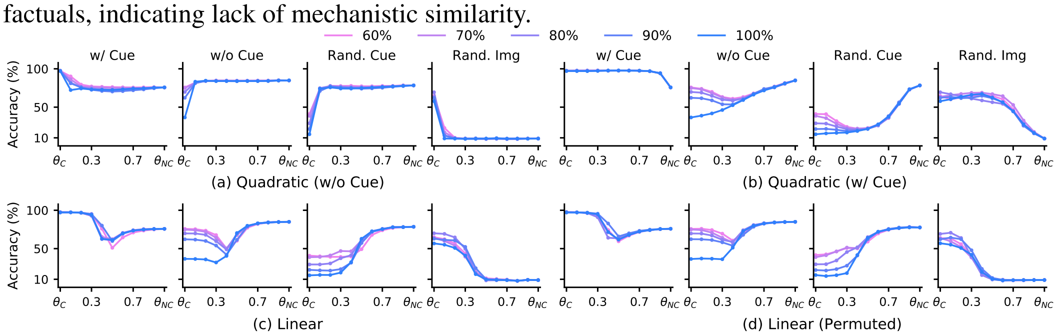


(a) Test Accuracy.

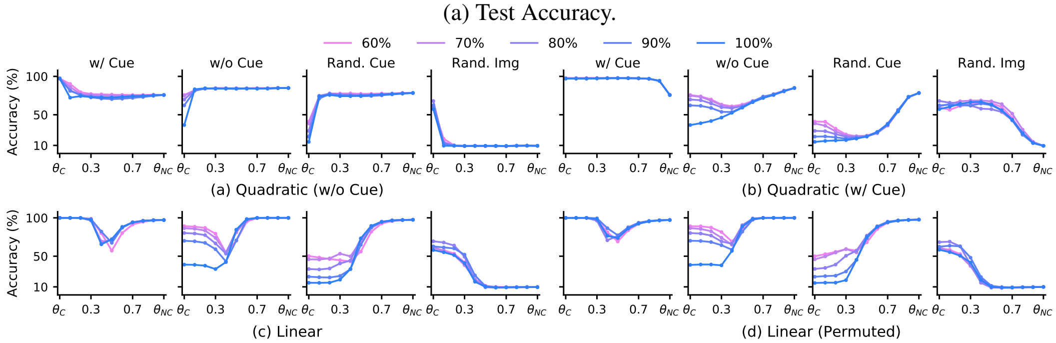


(b) Train Accuracy.

Figure 10: **VGG-13 on Dominoes**. We plot test/train accuracy curves along different connectivity paths and see thorough corroboration of our claims in the main text: Mechanistically dissimilar minimizers can be connected via nonlinear paths on a given dataset, but behave different on counterfactuals, indicating lack of mechanistic similarity.



(a) Test Accuracy.



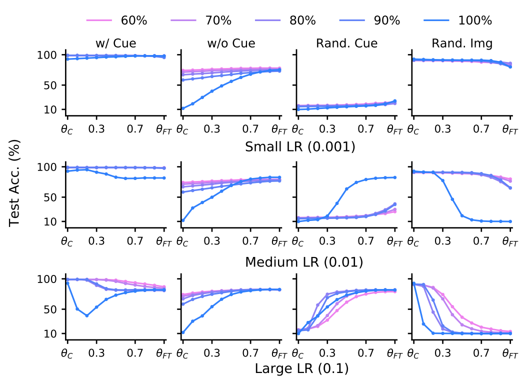
(b) Train Accuracy.

Figure 11: **ResNet-18 on Dominoes**. We plot test/train accuracy curves along different connectivity paths and see thorough corroboration of our claims in the main text: Mechanistically dissimilar minimizers can be connected via nonlinear paths on a given dataset, but behave different on counterfactuals, indicating lack of mechanistic similarity.

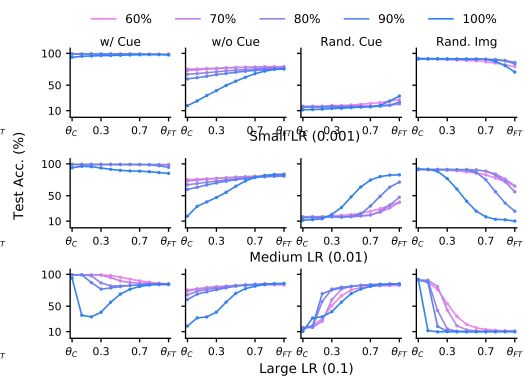
E Linear Connectivity and Mechanistic Similarity

We train VGG-13 and ResNet-18 models on our synthetic CIFAR-10 / CIFAR-100 / Dominoes datasets with cue features (see App. B.3). Corresponding models are denoted θ_{C}. These models are then fine-tuned on the original CIFAR-10/CIFAR-100 datasets that do not have any cue features. We use different learning rates (LR) and train for 100 epochs with a step-decay schedule (decay at epoch 40, 80 by a factor of 0.1). Corresponding models are denoted θ_{FT}. In the following, plot titles denote evaluation dataset, including datasets where either the cue is present (denoted w/ Cue), absent (denoted w/o Cue), randomized (denoted Rand. Cue), or the underlying image is randomized but the cue remains the same (denoted Rand. Img). Line colors denote the proportion of dataset that has synthetic cues.

Across all our results, we see that for a small enough learning rate, θ_{FT} exhibits linear connectivity with θ_{C} on the synthetic dataset (in the sense of Def.1); correspondingly, counterfactual evaluations illustrate linear connectivity as well. This indicates the models respond similarly to interventions on the dataset and are hence mechanistically similar and connected (see Def. 4). Meanwhile, increasing the learning rate induces barriers along the linear path. Correspondingly, we find linear connectivity does not hold on the synthetic dataset and models respond differently to counterfactual evaluations. That is, they are mechanistically dissimilar and not connected.

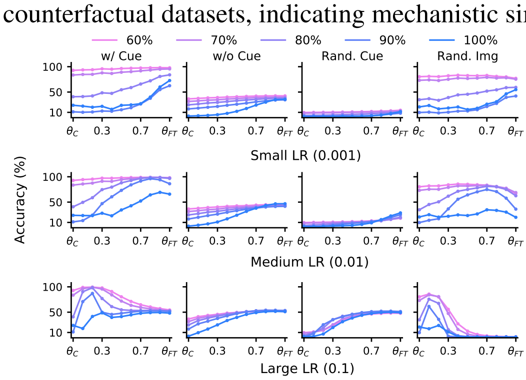


(a) VGG.

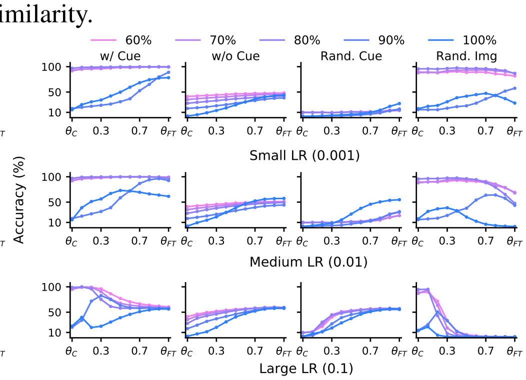


(b) ResNet18.

Figure 12: **Fine-tuning of models trained on CIFAR-10 with Box Cue**. We plot test accuracy curves along the linear path between θ_{C} and θ_{FT} and see thorough corroboration of our claims in the main text: Linearly connected minimizers exhibit mechanistic similarity, behaving identically on counterfactual datasets, indicating mechanistic similarity.

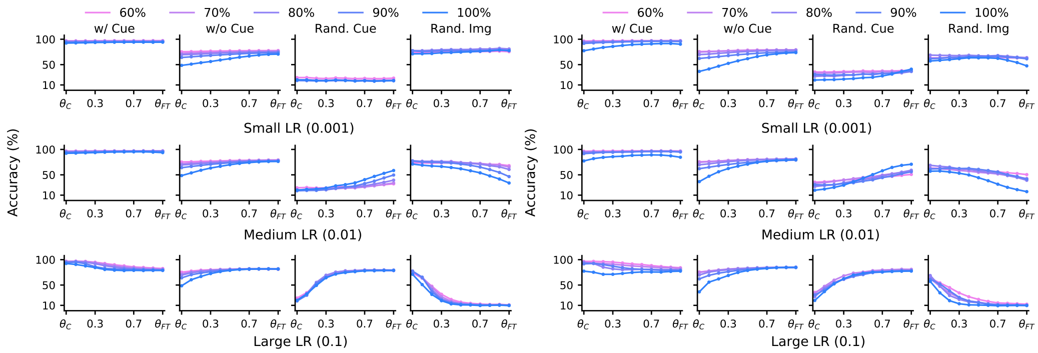


(a) VGG.



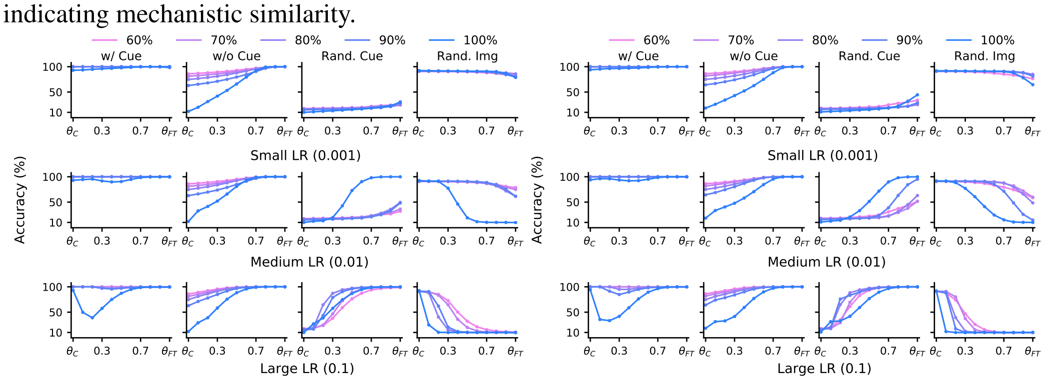
(b) ResNet18.

Figure 13: **Fine-tuning of models trained on CIFAR-100 with Box/Color Cue**. We plot test accuracy along the linear path between θ_{C} and θ_{FT} and see thorough corroboration of our claims in the main text: Linearly connected minimizers exhibit mechanistic similarity, behaving identically on counterfactual datasets, indicating mechanistic similarity.



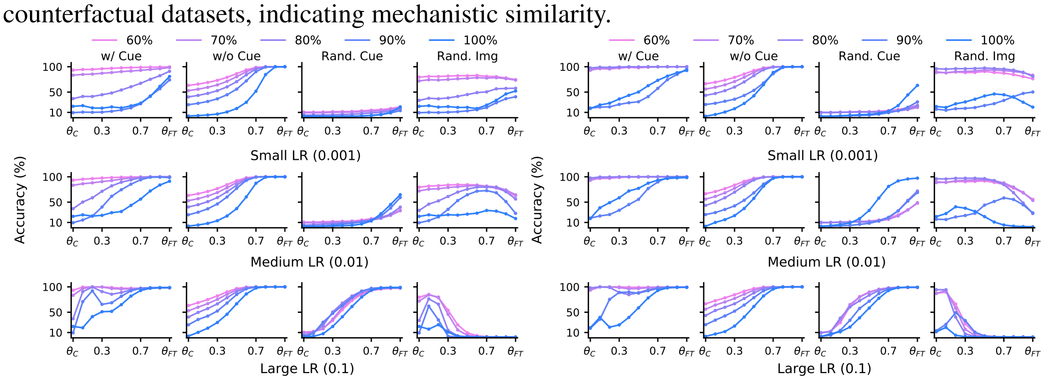
(a) VGG. (b) ResNet18.

Figure 14: **Fine-tuning of models trained on Dominoes**. We plot test accuracy along the linear path between θ_{C} and θ_{FT} and see thorough corroboration of our claims in the main text: Linearly connected minimizers exhibit mechanistic similarity, behaving identically on counterfactual datasets, indicating mechanistic similarity.



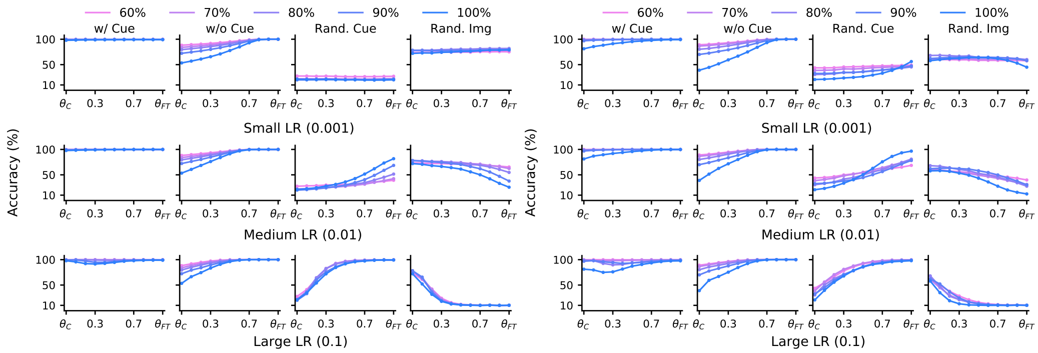
(a) VGG. (b) ResNet18.

Figure 15: **Fine-tuning of models trained on CIFAR-10 with Box Cue**. We plot train accuracy curves along the linear path between θ_{C} and θ_{FT} and see thorough corroboration of our claims in the main text: Linearly connected minimizers exhibit mechanistic similarity, behaving identically on counterfactual datasets, indicating mechanistic similarity.



(a) VGG. (b) ResNet18.

Figure 16: **Fine-tuning of models trained on CIFAR-100 with Box/Color Cue**. We plot train accuracy curves along the linear path between θ_{C} and θ_{FT} and see thorough corroboration of our claims in the main text: Linearly connected minimizers exhibit mechanistic similarity, behaving identically on counterfactual datasets, indicating mechanistic similarity.



(a) VGG. (b) ResNet18.

Figure 17: **Fine-tuning of models trained on Dominoes**. We plot test accuracy curves along the linear path between θ_{C} and θ_{FT} and see thorough corroboration of our claims in the main text: Linearly connected minimizers exhibit mechanistic similarity, behaving identically on counterfactual datasets, indicating mechanistic similarity.

F Lemma / Claims

F.1 Lemma 1.

Lemma 1. *If $f(.;\theta)$ is invariant to unit interventions $\mathcal{A}_i$ and $\mathcal{A}_j$, it must be invariant to their composition; if it is not invariant to $\mathcal{A}_i$ or $\mathcal{A}_j$, it cannot be invariant to their composition.*

Proof. Assume the parameterization θ exhibits invariance to the intervention $\mathcal{A}_i$. Independently, consider another intervention $\mathcal{A}_j$. Then, $f(\mathcal{E}(x;\{\mathcal{A}_i,\mathcal{A}_j\});\theta) = f(\mathcal{G}_X \circ \mathcal{A}_i \circ \mathcal{A}_j \circ \mathcal{G}_X^{-1}(x);\theta) = f(\mathcal{G}_X \circ \mathcal{A}_i \circ \mathcal{G}_X^{-1}(\mathcal{E}(x;\mathcal{A}_j));\theta) = f(\mathcal{E}(\mathcal{E}(x;\mathcal{A}_j);\mathcal{A}_i);\theta) = f(\mathcal{E}(x;\mathcal{A}_j);\theta)$, where the last equality happens due to the assumed invariance of $\mathcal{A}_i$. Now, if θ exhibits invariance to $\mathcal{A}_j$ as well, we have $f(\mathcal{E}(x;\{\mathcal{A}_i,\mathcal{A}_j\});\theta) = f(\mathcal{E}(x;\mathcal{A}_j);\theta) = f(x;\theta)$, i.e., the parameterization θ is invariant to the simultaneous operation (i.e., composition) of $\mathcal{A}_i$ and $\mathcal{A}_j$. Meanwhile, if θ is invariant $\mathcal{A}_i$, but not to $\mathcal{A}_j$, we have $f(\mathcal{E}(x;\{\mathcal{A}_i,\mathcal{A}_j\});\theta) = f(\mathcal{E}(x;\mathcal{A}_j);\theta) \neq f(x;\theta)$, i.e., the parameterization θ is not invariant to the simultaneous operation (i.e., composition) of $\mathcal{A}_i$ and $\mathcal{A}_j$.

Note that the derivation above did not rely on the fact that the interventions are "unit", in the sense that they act on independent dimensions. However, if one considers general interventions that can act on multiple dimensions of the latent space simultaneously, then a given intervention can undo the effects of another one. For example, assume a parameterization is not invariant to interventions that yield rotations, but nothing else. Then, two invariant interventions can make an object rotate by equal and opposite angles, while changing some other dimensions of the latent state that the model is invariant to. In this case, the interventions end up undoing their effect, and the overall state change does not yield any influence on the model output. By assuming unit interventions that enforce transformations on specific dimensions, we can circumvent this failure mode. □

F.2 Proposition 1.

Proposition 1. ***(Mechanistically Dissimilar Modes are Connected.)*** *Assume θ_1, θ_2 are two mechanistically dissimilar modes of loss $\mathcal{L}(f(\mathcal{D};\theta))$ on a given dataset $\mathcal{D}$. Given sufficient overparamterization, there exists a continuous path that connects the two modes (in the sense of Def. 1).*

Proof. The proof follows trivially from the results of [32, 31]. Therein, it is shown all loss minimizers lie on a single continuous manifold given sufficient overparameterization. That is, regardless of the underlying mechanism leading to zero loss, the minimizer will necessarily lie on the manifold of parameterizations achieving zero loss. □

F.3 Conjecture 1.

Conjecture 1. ***(Mechanistic Similarity Enforces Linear Connectivity.)*** *If, up to permutations of neurons, θ_1, θ_2 show linear connectivity on a dataset $\mathcal{D}$, then they must be mechanistically similar. If they cannot be connected linearly, the modes must be mechanistically dissimilar.*

Neural networks boast the well-known permutation symmetry phenomenon in their structure: permuting neurons, while accounting for the fan-in and fan-out weights, yields a model that is functionally the same [65, 15, 32]. That is, after permutation, the model encodes the exact same function as the original model; in the language of this paper, we can say the model uses the exact same mechanisms producing its outputs before and after the permutation. To avoid this degeneracy, we will assume that we are analyzing two minimizers θ_1, θ_2 that necessarily are *not* permutations of each other. In practice, one can run recent methodologies on "neural alignment" to ensure this assumption is valid [16, 17, 66].

Proof. As per Lemma 1, we only need to establish invariance / covariance to unit interventions for characterizing the mechanisms underlying a model's decision rules and, correspondingly, ascertain mechanistic similarity between two model parameterizations. To that end, we consider a unit intervention $\mathcal{A}_i$ that we assume the minimizer θ_1 is invariant to. We will analyze the loss of the model parameterized with linear interpolation of θ_1, θ_2 on a counterfactual sample $\mathcal{E}(x; \mathcal{A}_i)$ generated using intervention $\mathcal{A}_i$. For brevity, we denote the latent state of z as $z = \mathcal{G}_X^{-1}(x)$; correspondingly, we denote the intervened latent state as $\mathcal{A}_i^{\alpha_i}(z) = z + \Delta z$, where Δz is 0 in all but the i^{th} dimension, where it is equal to $\Delta z_i = \alpha_i$. We can thus write: $\mathcal{E}(x; \mathcal{A}_i^{\alpha_i}) = \mathcal{G}_X \circ \mathcal{A}_i^{\alpha_i} \circ \mathcal{G}_X^{-1}(x) = \mathcal{G}_X(\tilde{z}) = \mathcal{G}_X(z + \Delta z)$.

We now consider the parameterization along a general path $\gamma_{\theta_1 \to \theta_2}(t)$ such that $\gamma_{\theta_1 \to \theta_2}(0) = \theta_1$ and $\gamma_{\theta_1 \to \theta_2}(1) = \theta_2$. We assess its loss on the counterfactual data via a second-order expansion along the data-generating process:

$$
\begin{aligned}
& L\left(f\left(\mathcal{E}\left(x; \mathcal{A}_i^{\alpha_i}\right); \gamma_{\theta_1 \to \theta_2}(t)\right)\right) \\
& = L\left(f\left(\mathcal{G}_X\left(z + \Delta z\right); \gamma_{\theta_1 \to \theta_2}(t)\right)\right), \\
& = L\left(f\left(\mathcal{G}_X\left(z\right); \gamma_{\theta_1 \to \theta_2}(t)\right)\right) + \left(\Delta z\right)^T \nabla_z L\left(f\left(\mathcal{G}_X\left(z\right); \gamma_{\theta_1 \to \theta_2}(t)\right)\right) \\
& \qquad + \frac{1}{2}\left(\Delta z\right)^T \nabla_z^2 L\left(f\left(\mathcal{G}_X\left(z\right); \gamma_{\theta_1 \to \theta_2}(t)\right)\right)\left(\Delta z\right) + \mathcal{O}(\alpha_i^3), \\
& \approx L\left(f\left(\mathcal{G}_X\left(z\right); \gamma_{\theta_1 \to \theta_2}(t)\right)\right) + \alpha_i \frac{\partial}{\partial z_i} L\left(f\left(\mathcal{G}_X\left(z\right); \gamma_{\theta_1 \to \theta_2}(t)\right)\right) \\
& \qquad + \frac{1}{2}\left(\alpha_i\right)^2 \frac{\partial^2}{\partial z_i^2} L\left(f\left(\mathcal{G}_X\left(z\right); \gamma_{\theta_1 \to \theta_2}(t)\right)\right).
\end{aligned} \tag{5}
$$

The parameterization along a general path connecting the two minimizers θ_1, θ_2 can be written in the following form: $\gamma_{\theta_1 \to \theta_2}(t) = \theta_1 + \Delta\theta(t, 1)$, where $\Delta\theta(t, 1) = \gamma_{\theta_1 \to \theta_2}(t) - \theta_1$. Then, expanding the loss achieved by the model with this parameterization on the original data up to second-order along the change in parameters, we get the following.

$$
\begin{aligned}
& L\left(f\left(\mathcal{G}_X\left(z\right); \gamma_{\theta_1 \to \theta_2}(t)\right)\right) = L\left(f\left(\mathcal{G}_X\left(z\right); \theta_1 + \Delta\theta(t, 1)\right)\right) \\
& = L\left(f\left(\mathcal{G}_X\left(z\right); \theta_1\right)\right) + \left(\Delta\theta(t, 1)\right)^T \nabla_\theta L\left(f\left(\mathcal{G}_X\left(z\right); \theta_1\right)\right) \\
& \qquad + \frac{1}{2}\left(\Delta\theta(t, 1)\right)^T \nabla_\theta^2 L\left(f\left(\mathcal{G}_X\left(z\right); \theta_1\right)\right)\left(\Delta\theta(t, 1)\right) + \mathcal{O}(||\Delta\theta(t, 1)||^3), \\
& \approx \frac{1}{2}\left(\Delta\theta(t, 1)\right)^T \nabla_\theta^2 L\left(f\left(\mathcal{G}_X\left(z\right); \theta_1\right)\right)\left(\Delta\theta(t, 1)\right),
\end{aligned} \tag{6}
$$

where the loss and the gradient term can be ignored because θ_1 is a minimizer of the loss on dataset $\mathcal{D}$. Now, substituting Equation 6 into Equation 5, we get the following.

$$
\begin{aligned}
&L\left(f\left(\mathcal{E}\left(x;\mathcal{A}_i^{\alpha_i}\right);\gamma_{\theta_1\to\theta_2}(t)\right)\right)\\
&=L\left(f\left(\mathcal{G}_X\left(z\right);\gamma_{\theta_1\to\theta_2}(t)\right)\right)+\alpha_i\frac{\partial}{\partial z_i}L\left(f\left(\mathcal{G}_X\left(z\right);\gamma_{\theta_1\to\theta_2}(t)\right)\right)+\frac{1}{2}\left(\alpha_i\right)^2\frac{\partial^2}{\partial z_i^2}L\left(f\left(\mathcal{G}_X\left(z\right);\gamma_{\theta_1\to\theta_2}(t)\right)\right),\\
&=\frac{1}{2}\Delta\theta(t,1)^T\nabla_\theta^2 L\left(f\left(\mathcal{G}_X\left(z\right);\theta_1\right)\right)\Delta\theta(t,1)\\
&\qquad+\alpha_i\frac{\partial}{\partial z_i}\left(\frac{1}{2}\Delta\theta(t,1)^T\nabla_\theta^2 L\left(f\left(\mathcal{G}_X\left(z\right);\theta_1\right)\right)\Delta\theta(t,1)\right)\\
&\qquad+\frac{1}{2}\left(\alpha_i\right)^2\frac{\partial^2}{\partial z_i^2}\left(\frac{1}{2}\Delta\theta(t,1)^T\nabla_\theta^2 L\left(f\left(\mathcal{G}_X\left(z\right);\theta_1\right)\right)\Delta\theta(t,1)\right),\\
&=\frac{1}{2}\Delta\theta(t,1)^T\nabla_\theta^2\left[L\left(f\left(\mathcal{G}_X\left(z\right);\theta_1\right)\right)+\alpha_i\frac{\partial}{\partial z_i}L\left(f\left(\mathcal{G}_X\left(z\right);\theta_1\right)\right)+\frac{1}{2}\left(\alpha_i\right)^2\frac{\partial^2}{\partial z_i^2}L\left(f\left(\mathcal{G}_X\left(z\right);\theta_1\right)\right)\right]\Delta\theta(t,1),\\
&=\frac{1}{2}\Delta\theta(t,1)^T\nabla_\theta^2\left[L\left(f\left(\mathcal{G}_X\left(z\right);\theta_1\right)\right)+\left(\Delta z\right)^T\nabla_z L\left(f\left(\mathcal{G}_X\left(z\right);\theta_1\right)\right)+\frac{1}{2}\left(\Delta z\right)^T\nabla_z^2 L\left(f\left(\mathcal{G}_X\left(z\right);\theta_1\right)\right)\left(\Delta z\right)\right]\Delta\theta(t,1),\\
&\approx\frac{1}{2}\Delta\theta(t,1)^T\nabla_\theta^2\left[L\left(f\left(\mathcal{G}_X\left(z+\Delta z\right);\theta_1\right)\right)\right]\Delta\theta(t,1),\\
&=\frac{1}{2}\Delta\theta(t,1)^T\nabla_\theta^2\left[L\left(f\left(\mathcal{E}\left(x;\mathcal{A}_i^{\alpha_i}\right);\theta_1\right)\right)\right]\Delta\theta(t,1),\\
&=\frac{1}{2}\Delta\theta(t,1)^T\nabla_\theta^2\left[L\left(f\left(x;\theta_1\right)\right)\right]\Delta\theta(t,1),
\end{aligned}
\tag{7}
$$

where the last equality follows because of the assumed invariance of θ_1 to the intervention $\mathcal{A}_i^{\alpha_i}$.

We break the argument into two cases:

1. **Linear case:** If the connectivity path $\gamma_{\theta_1\to\theta_2}(t)$ is linear, the change in loss moving from θ_1 to θ_2, along the displacement vector $\theta_2-\theta_1$, is zero. Since θ_1 is a minimizer, this implies the displacement vector lies in the null-space of the Hessian, i.e., $\Delta\theta(2,1)^T\nabla_\theta^2\left[L\left(f\left(x;\theta_1\right)\right)\right]\Delta\theta(2,1)=0$. Correspondingly, for any point in this linear path, we have, $\Delta\theta(t,1)^T\nabla_\theta^2\left[L\left(f\left(x;\theta_1\right)\right)\right]\Delta\theta(t,1)=0\,\forall\,t\in[0,1]$. Substituting this relation into Equation 7, we get $L\left(f\left(\mathcal{E}\left(x;\mathcal{A}_i^{\alpha_i}\right);\gamma_{\theta_1\to\theta_2}(t)\right)\right)=0$ and all parameterizations along the linear path share invariances with the parameterization θ_1.
2. **Non-Linear case:** If the connectivity path $\gamma_{\theta_1\to\theta_2}(t)$ is not linear, then there exists an interpolation along the linear path connecting minimizers θ_1,θ_2 that has a loss higher than the two minimizers. That is, the displacement vector $\Delta\theta(t,1)$ does not lie in the null-space of the Hessian and $\Delta\theta(t,1)^T\nabla_\theta^2\left[L\left(f\left(x;\theta_1\right)\right)\right]\Delta\theta(t,1)\neq 0$. Substituting this relation into Equation 7, we get $L\left(f\left(\mathcal{E}\left(x;\mathcal{A}_i^{\alpha_i}\right);\gamma_{\theta_1\to\theta_2}(t)\right)\right)\neq 0$.

□