LOSS-LEVEL CONNECTIVITY ACROSS OPTIMIZATION TRAJECTORIES

Anonymous authors
Paper under double-blind review

ABSTRACT

The *training data* and *optimization process* jointly shape how deep neural networks navigate their loss landscapes. This interaction motivates studying not only the geometry of optimization, but also how that geometry relates to the model's evolving behavior on the training data. Yet, existing studies have largely focused on loss-landscape geometry around converged solutions, where learning from the training data is largely complete. Mode connectivity exemplifies this line of work by asking whether independently trained solutions can be connected through paths of low loss. This perspective naturally raises a question during training: *how is this geometry organized while the model is still learning the structure of the training data?* However, extending this perspective to intermediate training stages is less straightforward, because the objective itself changes continuously as the network continues to learn from the data. To separate this changing objective value from the geometry being examined, we study connectivity between independently trained states at matched objective levels, which we refer to as *loss-level connectivity*. In the settings we examine, matched-loss states can be connected by fitted nonlinear paths that approximately preserve the endpoint-defined objective level across multiple stages of training. We then analyze example-wise residuals along these paths. These residuals distinguish controlled subsets of training examples even after matching their endpoint losses, and vary systematically under label-corruption and training interventions. These findings show that path-based measurements can contain information that is not captured by scalar endpoint loss alone.

1 INTRODUCTION

The training data and optimization process jointly shape how deep neural networks (DNNs) navigate their high-dimensional loss landscape. Despite the practical success of modern optimization algorithms (Kingma & Ba, 2014; Loshchilov & Hutter, 2017), our understanding of the geometry during the training process remains limited. Existing studies for loss landscapes of DNNs have largely focused on the geometry around converged solutions (Garipov et al., 2018; Foret et al., 2020; Izmailov et al., 2018), where learning from the training data is almost complete. Among these, Garipov et al. (2018) studied mode connectivity, which has provided an insightful geometric perspective by showing that apparently distinct trained solutions can often belong to a common low-loss path connected by simple nonlinear paths.

However, focusing on the geometry around converged solutions limits our understanding of how such geometric structure is organized while the model is still learning the training data. This naturally raises a question as follows:

RQ1 Does such connectivity emerge only once optimization reaches a solution, or is related geometric structure present throughout training?

However, extending the usual connectivity perspective to intermediate training states remains less straightforward since the objective value continues to change throughout training, unlike near converged modes, even though some prior works (Freeman & Bruna, 2016; Gotmare et al., 2018) have already shown that connected structure can appear before convergence. Their perspectives primarily characterize whether two intermediate states can be connected through a path that remains sufficiently

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

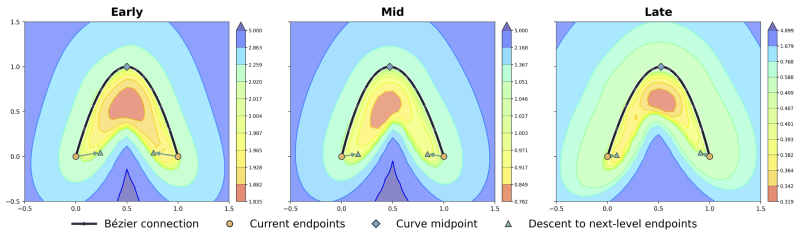

Figure 1: Visualization of loss-level connected geometry for ResNet-18 on CIFAR-100. Each panel connects checkpoints from two independently trained optimization trajectories that reach comparable ℓ_2-regularized loss values at different stages of training using a quadratic Bézier curve. **Left**: early-stage connectivity. **Middle**: mid-stage connectivity. **Right**: late-stage connectivity.

low in objective value. Such a criterion does not distinguish a path that connects the states while *preserving* a specific loss level from one that connects them simply through lower loss regions. This distinction becomes particularly important for interpreting the relationship between connectivity and the learning states of each example during training as we mentioned above. We therefore investigate loss-level-preserving connectivity at multiple stages of training to compare fitted-path geometry while controlling for differences in endpoint objective value. This allows us to examine how connected geometry itself is organized while the model is still learning. We refer to this loss-level preserving connectivity throughout the paper as *loss-level connectivity (LLC)*.

Using this perspective, we select matched-loss states from two independently trained trajectories using two random seeds and fit simple nonlinear paths between them. In the examined settings, we recover approximately loss-level-preserving nonlinear paths at multiple stages before convergence. We also observe such recoverability in the examined cross-optimizer setting. These results indicate that simple loss-level-preserving paths are not restricted to converged solutions. These observations suggest that loss-level connectivity reflects a recurring structure of the loss landscape rather than a phenomenon restricted to converged modes. Having established this connected geometry, we next ask what it reflects about the training data. Since the loss landscape is jointly determined by the model, objective function, and data, we keep the model and objective fixed and examine how the data is reflected in the observed loss-level connectivity. This leads to our second question:

RQ2 What example-wise information appears along loss-level connections?

In controlled label-corruption experiments, different subsets of training examples exhibit distinct residual along the same fitted connection. These differences persist after matching the subsets by endpoint loss and vary with corruption ratio and training interventions. Thus, example-wise LLC provides an along-path measurement that can distinguish training subsets even when their scalar endpoint losses are similar.

In summary, we observe approximate loss-level connectivity across the examined training stages, architectures, and optimizer settings. We then use example-wise path residuals to show that matched endpoint objective values can conceal systematic differences in along-path behavior.

2 Connectivity Before Modes

We begin by studying loss-level connectivity between independently trained optimization trajectories. While mode connectivity has primarily been examined between converged solutions, our goal is to ask whether a related level-preserving geometric structure is already present during optimization in this section. To this end, we first define approximate LLC and examine whether such connectivity exists before convergence and whether it emerges through training or is already present at initialization. Unless stated otherwise, we use ResNet18 (He et al., 2016) and CIFAR-100 (Krizhevsky et al., 2009) with SGD with momentum (SGDM) as a main experimental setting. For a different experimental setting, please refer to Appendix A.

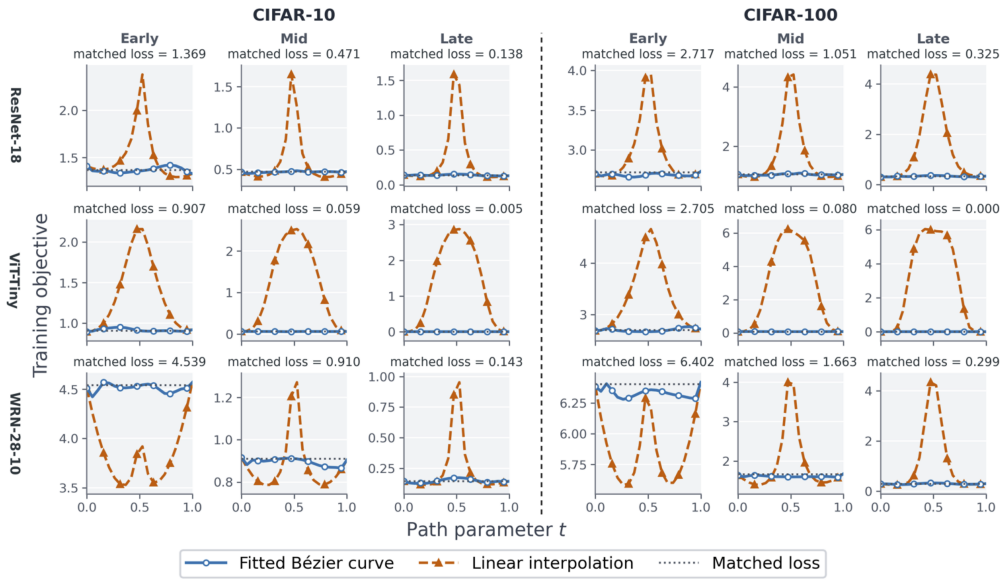

Figure 2: Loss-level connectivity across architectures, datasets, and training stages. We compare linear interpolation and fitted quadratic Bézier curve between two checkpoints that exhibit similar objective values to each other from two independently trained optimization trajectories for ResNet-18, ViT-Tiny, and WRN-28-10 on CIFAR-10 and CIFAR-100. Columns correspond to early, mid, and late stages of training. Across architectures and datasets, linear interpolation can deviate substantially from the endpoint-defined loss level, whereas the quadratic Bézier curve provides a substantially better level-preserving connection. All Bézier curves are fitted using the ℓ_2-regularized cross-entropy objective except for ViT-Tiny, which is using AdamW optimizer (Loshchilov & Hutter, 2017). For the results of other networks, please refer to the Appendix B.

2.1 Loss-Level Connectivity Across Optimization Trajectories

Let θ_A and θ_B denote two independently obtained parameter states with objective values $L_A = L(\theta_A)$ and $L_B = L(\theta_B)$ such that $L_A \approx L_B$. Then, we first define the reference loss as follows:

$$\ell(t; \theta_A, \theta_B) = (1-t)L_A + tL_B, \qquad t \in [0,1]. \tag{1}$$

For a path $\gamma : [0,1] \to \Theta$ connecting the two states, we measure its deviation $e(t; \theta_A, \theta_B) = L(\gamma(t)) - \ell(t)$ and normalize it as follows:

$$D_{max}(\gamma; \theta_A, \theta_B) = \max_{t\in[0,1]} \frac{|e(t; \theta_A, \theta_B)|}{(L_A + L_B)/2}, \quad D_{avg}(\gamma; \theta_A, \theta_B) = \mathbb{E}_{t\sim[0,1]} \left[\frac{|e(t; \theta_A, \theta_B)|}{(L_A + L_B)/2} \right]. \tag{2}$$

A small value of D_{max} indicates that the two states can be connected while approximately preserving their objective level. Similarly, A small value of D_{avg} represents that the path is generally well fitted to their objective level. We uniformly sampled 21 points from a uniform grid over $[0,1]$, and calculated the D_{max} and D_{avg}. Throughout our experiments, we use quadratic Bézier curves $\gamma_{bez}(t) = (1-t)^2\theta_A + 2t(1-t)C + t^2\theta_B$, where C is a control point, as the nonlinear path. Linear interpolation $\gamma_{lin}(t) = (1-t)\theta_A + t\theta_B$ is used as a reference baseline. Note that we omit θ_A and θ_B from $e(t; \theta_A, \theta_B)$, $D(\gamma; \theta_A, \theta_B)$, and $l(t; \theta_A, \theta_B)$ for convenience from now on.

Bézier curve fitting procedure. For each pair of independent optimization trajectories, we select states at matched objective levels and re-evaluate their objective value using a deterministic evaluation protocol.[1] We retain endpoint pairs whose relative objective mismatch is sufficiently small, and then similar to Garipov et al. (2018), we optimize the control point C of $\gamma_{bez}(t)$ to minimize the deviations

[1]To ensure comparable objective values across trajectories, we disable stochastic data augmentation and evaluate all checkpoints on the same fixed training examples.

from the reference loss level $l(t)$ in Eq. 1. Unless stated otherwise, all path evaluations use the same batch normalization (Ioffe & Szegedy, 2015) recalibration and objective evaluation protocol.[2] Details of fitting procedure of $\gamma_{bez}(t)$ is elaborated in Appendix A.

2.2 Does Mode Connectivity Exist Before There Are Modes?

We first investigate whether independently trained trajectories are already connected before they reach their final solutions. Accordingly, we fit Bézier curves at the early, middle, and late stages of training across multiple network architectures. As shown in Fig. 1 and Fig. 2, for states with matched objective values, linear interpolation can exhibit a substantial high-loss barrier, indicating that the endpoints are not trivially connected by a straight path. In contrast, a simple quadratic Bézier curve can maintain a substantially more stable objective value between the endpoints of each curve throughout the training process. Thus, the nonlinear connectivity commonly associated with converged modes is already visible well before optimization has produced a final minimum. These observations suggest that independently trained trajectories should not be viewed as moving through completely isolated trajectories until convergence. Thus, in the examined settings, independently trained states at comparable objective levels can already be joined before convergence by a simple fitted nonlinear path.

2.3 Beyond Parameter Redundancy: Functional Variation Along the Path

After observing there is LLC on the loss landscape, a natural concern is that two endpoints may appear different in parameter space while implementing essentially the same function (Dinh et al., 2017; Simsek et al., 2021; Entezari et al., 2021). In that case, their connectivity would primarily reflect parameter redundancy rather than a meaningful relation between distinct learned models. We therefore examine the functional variation along the fitted curve using prediction disagreement and centered-logit distance at multiple values of $t \in [0, 1]$. Prediction disagreement measures the fraction of examples whose predicted class differs between two corresponding models, and centered-logit distance measures the distance between their logits after removing the per-example mean across classes. As shown in Fig. 3, each point in $\gamma_{bez}(t)$ differ in their predictions. Thus, the observed connectivity is not simply a consequence of redundant symmetries in parameter space. Instead, the LLC geometry can contain functionally distinct learned models. This implies that LLC contains nontrivial functional degrees of freedom along which model behavior can change.

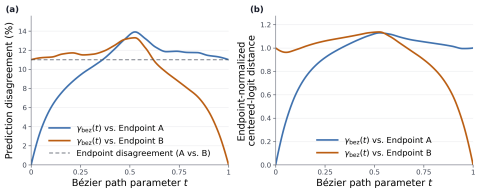

Figure 3: Functional variations along the fitted curve. (a) and (b) are prediction disagreement and centered-logit distance between points on path and endpoints, respectively.

2.4 Do SGD and Adam Travel Through the Same Geometry?

We next investigate the LLC is specific to the optimizer type. For example, SGDM and Adam (Kingma & Ba, 2014) induce different optimization dynamics, and may therefore explore different regions of parameter space even when they reach comparable objective values. To test whether the connected geometry is shared across different optimizer families, we construct cross-optimizer endpoint pairs by selecting an SGDM state and an Adam state at matched objective levels. We then fit the Bézier paths used in our within-optimizer analysis at multiple stages during training process. Despite the different optimization dynamics, SGDM and Adam states can be connected through Bézier curve as shown in Fig. 4. Thus, the LLC is not confined to trajectories generated by the same optimizer family. Thus, the recoverability of such simple paths is not restricted to endpoint pairs generated by the same optimizer family in these experiments.

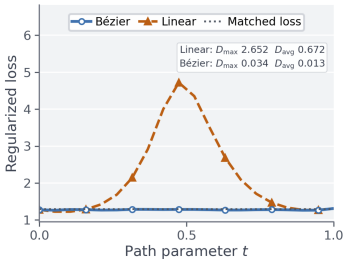

Figure 4: Cross-optimizer LLC at an intermediate training stage.

[2]For path evaluation, we do one additional forward to recalibrate the batch-normalization running statistics for each point along the path.

2.5 Summary of This Section

The results of this section can be summarized as follows. First, in the examined settings, simple nonlinear paths can approximately preserve the endpoint-defined objective level between independently trained states before convergence. Second, such path recoverability is observed from early training stages rather than only near converged solutions. Third, the examined cross-optimizer endpoint pairs also admit similar fitted paths, indicating that this phenomenon is not restricted to same-optimizer pairs. Together, These observations raises a more fundamental question:

> What determines the structure of this connected geometry?

Since the loss landscape is jointly shaped by the model, the data, and the objective function, we next hold the model and objective function fixed and investigate how the data is reflected in the observed loss-level connections. To do this, we leverage the LLC as an analysis tool in following section.

3 What Example-wise Information Appears Along Loss-Level Connections?

Having established approximate loss-level connections, we next examine their example-wise behavior under controlled label corruption. Label corruption provides known subsets of corrupted and uncorrupted examples, allowing us to test whether the two subsets exhibit systematically different residual profiles along the same fitted path.

3.1 A Global Connection Hides Heterogeneous Example-Wise Behavior

A LLC preserves the training objective only at the aggregate level. This does not necessarily imply that different subsets of the training data behave similarly along the fitted curve. To examine what is hidden beneath the aggregated objective, we keep the fitted curve fixed and separately measure the example-wise signed residuals $s_i = \mathbb{E}_t\left[e_i(t)\right]$, where i is the index of example and $e_i(t)$ is the example-wise $e(t)$ in Sec. 2.1, for each corrupted and uncorrupted examples along the path. We first calculate example-wise s_i along the fitted curve on the label-corrupted dataset. We then average the s_i separately over the corrupted and uncorrupted subsets. At a low corruption ratio $p = 0.2$, the two subsets exhibit substantially different residual behavior as shown in Fig. 5. However, the magnitude of this heterogeneity changes systematically with the corruption ratio. The residual profiles of corrupted and uncorrupted examples become more similar as p increases. Because the fitted path is optimized with respect to the aggregate training objective, one possible explanation is that changing the prevalence of corrupted examples changes how strongly each subset contributes to the fitted path. This observation alone does not identify a specific learning mechanism. We therefore next test whether the separation persists after matching endpoint losses, and in Sec. 3.3 vary the training contribution of corrupted examples while holding the corruption pattern fixed.

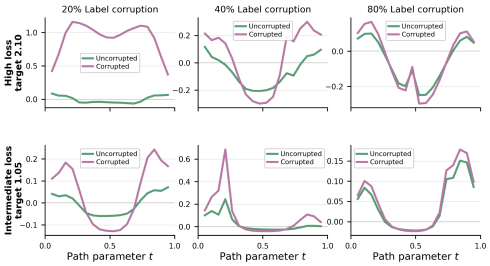

Figure 5: Mean signed residuals along the curve under different corruption ratios ($p = 0.2, 0.4$, and 0.8). The upper and lower rows refer to high and intermediate loss stages, respectively.

3.2 Example-wise LLC Differences Persis After Endpoint-Loss Matching

The preceding analysis shows that corrupted and uncorrupted examples can behave differently along the same aggregated LLC, and that the pattern of this difference changes systematically with the corruption ratio. We next ask whether this behavior reflects genuinely different along-path structure or simply arises from differences in their per-example losses at the two endpoints. To separate these possibilities, we match corrupted and uncorrupted examples according to their losses at both endpoints so that they have similar loss values. After matching, the endpoint-loss feature used in this analysis has AUROC close to chance.

Table 1: The results of LLC behavior along the curve for corrupted and uncorrupted subsets after matching corrupted and uncorrupted examples on their losses at both endpoints. Results are shown for CIFAR-100/ResNet-18 at corruption ratios $p = 0.2$ and $p = 0.6$, after matching corrupted and uncorrupted examples to have similar losses at both endpoints. Note that the N. LLC refers to normalized shape-only LLC. AUROCs are computed from out-of-fold predictions using stratified five-fold cross-validation. For other ratios p, please refer to the Appendix C.

Matched loss	$p = 0.2$				$p = 0.6$			
	Matched examples	Endpoint loss AUROC	Raw LLC AUROC	N. LLC AUROC	Matched examples	Endpoint loss AUROC	Raw LLC AUROC	N. LLC AUROC
2.104	9,392	0.486	0.833	0.760	17,402	0.493	0.627	0.612
1.895	9,466	0.493	0.773	0.706	17,514	0.497	0.624	0.595
1.046	9,429	0.493	0.664	0.688	17,081	0.498	0.657	0.658
0.843	9,501	0.501	0.720	0.689	16,505	0.497	0.691	0.616
0.735	9,354	0.491	0.735	0.694	16,048	0.492	0.697	0.643

Table 2: Endpoint-loss-matched example-wise LLC separation for the CIFAR-100/ResNet-18 at $p = 0.5$ memorization intervention under various λ settings. The values under Endpoint, Raw LLC, and N. LLC are five-fold stratified out-of-fold AUROCs.

Mem. target	$\lambda = 0.25$				$\lambda = 0.5$				$\lambda = 1$				$\lambda = 2$			
	Loss	Endpoint	Raw LLC	N. LLC	Loss	Endpoint	Raw LLC	N. LLC	Loss	Endpoint	Raw LLC	N. LLC	Loss	Endpoint	Raw LLC	N. LLC
1%	4.385	0.478	0.693	0.554	4.073	0.481	0.669	0.554	4.020	0.482	0.664	0.608	4.885	0.492	0.570	0.540
10%	3.657	0.477	0.843	0.740	3.932	0.487	0.822	0.703	3.879	0.483	0.802	0.735	3.908	0.490	0.841	0.816
50%	2.301	0.489	0.726	0.675	3.008	0.491	0.762	0.689	3.656	0.488	0.799	0.736	4.104	0.490	0.832	0.786
90%	1.153	0.496	0.778	0.760	1.537	0.497	0.669	0.665	1.872	0.492	0.657	0.635	2.236	0.495	0.627	0.609
99%	0.710	0.495	0.808	0.731	0.901	0.497	0.746	0.650	1.037	0.497	0.698	0.637	1.124	0.500	0.626	0.606

Its AUROC remains close to chance level across all examined loss levels for both $p = 0.2$ and $p = 0.6$ as shown in Table 1. The raw LLC reaches AUROCs of 0.664–0.833 at $p = 0.2$ and 0.624–0.697 at $p = 0.6$. Thus, two examples can have nearly indistinguishable losses at the same pair of endpoint states while behaving systematically differently between those states under sparse label corruption. This indicates that the distinction observed in Sec. 3.1 is not simply from a consequence of one group being harder at the endpoints. At the same time, the strength of this distinction again depends on the corruption ratio. The LLC separation is substantially stronger at $p = 0.2$ than at $p = 0.6$, following the same trend observed in the signed residuals of Sec. 3.1. The persistence of this distinction after endpoint-loss matching shows that the observed separation is not eliminated by matching the two endpoint losses. Taken together, these results establish a narrower point. Along-path LLC residuals contain information about corruption status that is not captured by the matched endpoint-loss values used in this analysis. The strength of this separation varies with the corruption ratio. The residual variance in the Table 3 provides a complementary view of this difference. At $p = 0.2$, corrupted examples exhibit substantially larger along-path residual variation $v_i = \mathbb{E}_t\left[(e_i(t) - s_i)^2\right]$ than uncorrupted examples even after their endpoint losses are matched. The variance gap also becomes smaller as the corruption ratio increases, mirroring the reduction in AUROC and signed-residual separation. The same composition-dependent trend therefore appears in several different characterizations of the along-path behavior. Finally, we test whether the remaining distinction can be explained solely by the s_i or the scale $\sqrt{v_i}$ of the example-wise residual $e_i(t)$. We therefore center and normalize $e_i(t)$ for each example, removing its mean and scale while retaining only its shape along the curve. The results show that corrupted and uncorrupted examples still remain distinguishable after this normalization as shown in Table 1, although the separation again weakens at higher corruption ratios. Thus, the distinction is not captured solely by the average level or variation of $e_i(t)$. Thus, the evolution of the residual along the curve contains information for distinguishing corrupted and uncorrupted examples beyond its per-example mean

Table 3: The mean example-wise variance along the curve. The U and C refer to the uncorrupted and corrupted example, respectively.

Loss	p	U	C	C/U
High	0.2	1.406	3.404	**2.42**
High	0.4	0.875	1.400	**1.60**
High	0.8	0.752	0.786	**1.05**
Inter.	0.2	0.260	0.557	**2.14**
Inter.	0.4	0.123	0.262	**2.13**
Inter.	0.8	0.130	0.136	**1.05**
Low	0.2	0.100	0.356	**3.56**
Low	0.4	0.129	0.211	**1.64**
Low	0.8	0.070	0.072	**1.03**

and scale. Accordingly, example-wise LLC provides an along-path descriptor whose discriminative signal persists after endpoint-loss matching and per-example mean/scale normalization.

3.3 Example-wise LLC Varies Under A Memorization-Pressure Intervention

The preceding analysis may be confounded by changes in data composition, since varying the corruption ratio also changes the relative prevalence of corrupted examples. To address this issue, we design an additional controlled experiment in which the corruption pattern is fixed at $p = 0.5$, while only the relative contribution of corrupted examples during network training is varied. Specifically, we control the pressure to fit corrupted labels using a weighting parameter λ as follows:

$$\mathcal{L}_\lambda = \frac{(1-p)\mathcal{L}_{\text{uncorr}} + \lambda p\, \mathcal{L}_{\text{corr}}}{(1-p) + \lambda p}, \tag{3}$$

where $\mathcal{L}_{\text{uncorr}}$ and $\mathcal{L}_{\text{corr}}$ denote the average losses over the uncorrupted and corrupted subsets, respectively. We vary $\lambda \in \{0.25, 0.5, 1, 2\}$, while keeping the dataset, corruption pattern, and training setup fixed. Importantly, λ is used only during network training, and all Bézier curves are subsequently fitted using the same uniformly weighted objective over all training examples. Fig. 6 shows the signed example-wise residual distributions at different memorization states. Across λ and memorization targets, the residual distributions and LLC-based AUROCs vary substantially. Separation is often strong during intermediate or late stages, although the trend is not uniformly monotonic across all λ settings as shown in Table 2. Because the corruption pattern and the curve-fitting objective are held fixed, these results show that example-wise LLC statistics change under interventions that alter the training pressure on corrupted labels. They do not by themselves isolate memorization as the only causal factor. Together, these results show that the example-wise structure of LLC changes with memorization state even when data composition and the Bézier-fitting objective are held fixed. For the residual variance-related experiments under various λ settings, please refer to the Appendix C.

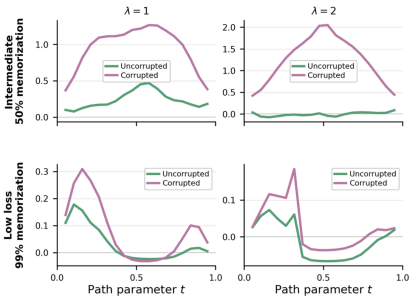

Figure 6: Mean signed residuals along the curve under different λ settings ($\lambda = 1.0$ and 2.0). The upper and lower rows refer to intermediate and late training stages, respectively.

3.4 Example-Wise Loss-Level Connectivity Stabilizes During Training

The preceding analysis shows that the example-wise differences revealed by LLC persist after matching corrupted and uncorrupted examples on their endpoint losses. We next ask whether this structure is specific to a particular fitted connection or instead emerges consistently across independently trained trajectories. We focus on the magnitude of the example-wise residuals $q^i_{avg} = \mathbb{E}_t\left[|e_i(t)|\right]$ and examine both their reproducibility across trajectory pairs and their persistence over training. We first compare example-wise rankings induced by q^i_{avg} across different trajectory pairs (*e.g.* (seed a, seed b) and (seed c, seed d)) at comparable learning stage. Near initialization, high-deviation examples differ substantially across pairs. However, as training progresses, the rankings become increasingly consistent, and largely the same high-deviation examples are recovered across independent fitted connections as shown in Fig. 7(a). This suggests that the emerging example-wise structure is not specific to a particular stochastic trajectory. We next examine whether this structure persists across training stages within a single trajectory pair. The results show that the rankings near initialization differ substantially from those observed after learning begins, indicating that the pattern is not fixed from initialization. However, during training, the correlation between successive stages increases as shown in Fig. 7(b). Thus, examples with large LLC deviations increasingly remain high-deviation examples at later stages. Finally, we compare these rankings across different model architectures (ResNet-18, ResNet-50, WRN-28-10, and ViT-Tiny) in Fig. 7(c). The rankings are positively correlated across the examined architectures, with stronger agreement among similar architectures than between CNNs and ViT-Tiny. Thus, some of the example ranking is shared across these models, while its architecture-dependent variation indicates that the ranking is not universal. Across the examined independent runs, high-deviation examples also become increasingly reproducible as training progresses. Taken together, LLC increasingly identifies reproducible and

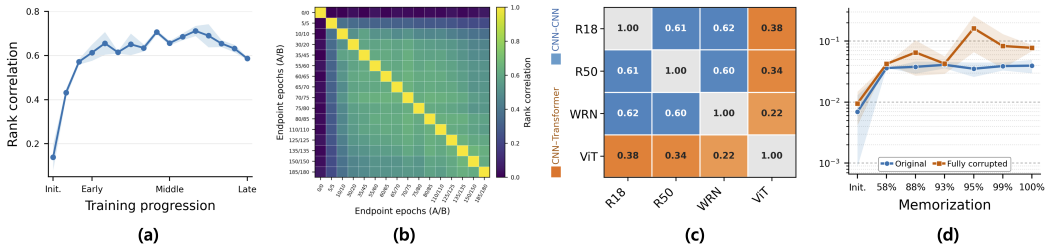

Figure 7: The results of example-wise loss-level connectivity analyses. (a) Rank correlation of example-wise LLC deviation magnitude across different independently trained trajectory pairs. (b) Temporal rank correlation of example-wise LLC deviation across matched-loss stages within a trajectory pair. (c) Cross-architecture rank correlation of example-wise LLC deviation across ResNet-18, ResNet-50, WRN-28-10, and ViT-Tiny. (d) D_{max} for clean-label and fully randomized-label training across matched objective levels as random-label memorization progresses. The x axis refers to the accuracy of fully corrupted dataset at matched objective level. Note that results in (a) and (d) report the mean and standard deviation across three independent seed pairs.

persistent high-deviation examples as training progresses. Although stochastic optimization produces different trajectories and fitted curves, these curves progressively highlight similar examples. Rather than serving only as a geometric description of a specific trajectory pair, LLC may therefore provide a reproducible characterization of which training examples interact most strongly with the connected loss-level geometry. The partial agreement across architectures further suggests that this structure contains both data-dependent and architecture-dependent components. At the same time, broader generalization across different optimization settings remains an open question for future study.

3.5 How Does the Connected Geometry Change With Memorization?

We next ask whether the connected geometry changes depending on what the model is learning fundamentally different structure from the data. To examine this, we compare training on the original dataset ($p = 0$) with training under completely randomized labels ($p = 1$), where optimization increasingly requires memorization of arbitrary input–label associations (Zhang et al., 2016; Arpit et al., 2017). For each setting, we select independently trained endpoint pairs at the same target objective levels. We then fit the Bézier curve and compare $D_{\max}$ in Eq. 2. At earlier matched levels, the connections obtained with original and randomized labels exhibit relatively similar deviations. However, as training progresses, $D_{\max}$ becomes larger and more variable for fully randomized-label training than for clean-label training in the examined seed pairs as shown in Fig. 7(d). This observation is consistent with the hypothesis that simple level-preserving paths become harder to recover as arbitrary-label fitting progresses. However, random-label training changes multiple aspects of the learned functions and optimization dynamics, so this experiment does not uniquely attribute the difference to the absence of shared label structure.

4 Related Works

4.1 Loss-Landscape Geometry and Mode Connectivity.

A substantial body of work has sought to characterize the geometry of neural-network loss landscapes, including interpolation between solutions, the structure of wide or flat regions, and the topology of low-loss sets (Goodfellow et al., 2014; Nguyen & Hein, 2017; Izmailov et al., 2018; Foret et al., 2020). A particularly influential line of work is mode connectivity. Garipov et al. (2018); Draxler et al. (2018) showed that independently trained solutions that appear separated under linear interpolation can often be connected by simple nonlinear paths with nearly constant low loss. Subsequent work provided theoretical accounts of such connectivity through overparameterization, stability, and connected sublevel sets (Kuditipudi et al., 2019; Nguyen, 2019; Nguyen et al., 2018), and proposed broader geometric views in which solutions lie within interconnected low-loss structures (Fort & Jastrzebski, 2019). These observations have further been extended from one-dimensional paths to learned subspaces, simplicial complexes, and nonlinear surfaces connecting multiple solutions (Benton et al., 2021; Ren et al., 2025). In contrast to connectivity defined only by the existence of a sufficiently

low-loss path, our focus is on *loss-level geometry*: we condition independently reached states on comparable objective values and ask whether a simple connection can approximately preserve their current loss level.

4.2 Connectivity Across Optimization.

Several works indicate that connected structure is not restricted to final solutions. Freeman & Bruna (2016) studied the geometry of level sets and reported connected level-set structure throughout learning. Gotmare et al. (2018) demonstrated that mode connectivity can also serve as a tool for analyzing training trajectories and models obtained under different initialization and training conditions. Relatedly, Frankle et al. (2020) used linear mode connectivity between runs exposed to different SGD noise to characterize when training becomes stable, finding that such stability can emerge early in optimization. These results motivate studying connectivity beyond converged minima, but address different notions of geometric structure and stability. Our analysis specifically compares states from *independently evolving trajectories at matched loss levels* and measures the extent to which their connecting path preserves the endpoint-defined loss profile. This loss-conditioned view allows geometric changes along optimization to be separated from the trivial effect of simply attaining a lower objective value, and enables us to trace the same notion of geometry from initialization through intermediate training.

5 Discussion

Our experiments show that, in the settings examined, fitted quadratic Bézier paths can approximately preserve matched endpoint objective levels at multiple stages of training. Matching endpoint objective values controls for one source of variation when comparing these fitted paths across training. In particular, simple level-preserving connections become less consistently recoverable under fully randomized-label training, the fitted paths exhibit larger and more variable $D_{\max}$ at later matched objective levels in the examined seed pairs. Our measurements are necessarily conditional on the chosen path family and fitting procedure. Failure to recover a quadratic Bézier connection under a fixed fitting budget does not establish topological disconnection; rather, it indicates reduced recoverability of a simple level-preserving path under the same protocol. Likewise, mean and maximum deviations capture complementary notions of path fidelity. We therefore view the present results not as a complete characterization of the underlying level set, but as evidence that its observable structure changes systematically with the learning process. This perspective opens several directions for future work. Richer path families may reveal geometric structure that quadratic Bézier curves cannot capture, while larger-scale models and datasets will test whether the same phenomena persist beyond the controlled settings studied here. More broadly, it will be important to understand which aspects of optimization, data structure, and memorization determine the emergence and degradation of coherent loss-level connectivity. If future work establishes that these path-based measurements provide information beyond richer endpoint statistics and generalize beyond the controlled settings studied here, they may become useful diagnostics of training dynamics.

Generative AI Use

We used generative AI tools to assist with refining and sanity-checking research hypotheses and experimental plans, and supporting research code development. We additionally used these tools for literature search and polishing text for clarity and readability. All AI-assisted outputs were reviewed by the authors. Experimental results and quantitative analyses were obtained from experiments conducted by the authors and were checked against the corresponding code and saved outputs. Literature-related suggestions were verified against the original sources.

References

Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning*. PMLR, 2017.

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*. PMLR, 2021.

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

Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. Essentially no barriers in neural network energy landscape. In *International Conference on Machine Learning*. PMLR, 2018.

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.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Stanislav Fort and Stanislaw Jastrzebski. Large scale structure of neural network loss landscapes. *Advances in Neural Information Processing Systems*, 32, 2019.

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

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

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov, and Andrew G Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in Neural Information Processing Systems*, 31, 2018.

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

Akhilesh Gotmare, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. Using mode connectivity for loss landscape analysis. *arXiv preprint arXiv:1806.06977*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. pmlr, 2015.

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.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Rohith Kuditipudi, Xiang Wang, Holden Lee, Yi Zhang, Zhiyuan Li, Wei Hu, Rong Ge, and Sanjeev Arora. Explaining landscape connectivity of low-cost solutions for multilayer nets. *Advances in Neural Information Processing Systems*, 32, 2019.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *Advances in Neural Information Processing Systems Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Quynh Nguyen. On connected sublevel sets in deep learning. In *International Conference on Machine Learning*. PMLR, 2019.

Quynh Nguyen and Matthias Hein. The loss surface of deep and wide neural networks. In *International Conference on Machine Learning*. PMLR, 2017.

Quynh Nguyen, Mahesh Chandra Mukkamala, and Matthias Hein. On the loss landscape of a class of deep neural networks with no bad local valleys. *arXiv preprint arXiv:1809.10749*, 2018.

Jie Ren, Pin-Yu Chen, and Ren Wang. Revisiting mode connectivity in neural networks with bezier surface. In *International Conference on Learning Representations*, 2025.

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*. PMLR, 2021.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

A Experimental Details

A.1 Training Trajectories

We construct independent optimization trajectories on CIFAR-10, CIFAR-100, and SVHN. For CIFAR-10 and CIFAR-100, we use ResNet-18, ResNet-50, WRN-28-10, and ViT-Tiny/4. For SVHN, we use ResNet-18. CIFAR models are trained using the standard random-crop and horizontal-flip augmentation, while SVHN uses random cropping without horizontal flipping. Evaluation and all curve-related analyses use deterministic, augmentation-free versions of the corresponding training sets. Table 4 summarizes the optimization settings used to generate the trajectories. Unless otherwise specified, SGD uses momentum 0.9 and cosine learning-rate decay, while Adam uses $(\beta_1, \beta_2) = (0.9, 0.999)$ and $\epsilon = 10^{-8}$. ViT-Tiny is trained with AdamW using a five-epoch linear warmup followed by cosine decay. Checkpoints are saved every three epochs and at the end of training.

Table 4: Training configurations used to generate optimization trajectories.

Dataset	Model	Optimizer	LR	WD	Epochs	Warmup
CIFAR-10/100	ResNet-18	SGD	5×10^{-2}	5×10^{-4}	200	–
CIFAR-10/100	ResNet-18	Adam	10^{-3}	5×10^{-4}	200	–
CIFAR-10/100	ResNet-50	SGD	5×10^{-2}	5×10^{-4}	200	–
CIFAR-10/100	WRN-28-10	SGD	5×10^{-2}	5×10^{-4}	200	–
CIFAR-10/100	ViT-Tiny/4	AdamW	3×10^{-4}	5×10^{-2}	300	5 epochs
SVHN	ResNet-18	SGD	10^{-2}	5×10^{-4}	200	–

All experiments use batch size 128. For CIFAR-10/100, we normalize each channel using mean $(125.3, 123.0, 113.9)/255$ and standard deviation $(63.0, 62.1, 66.7)/255$. For SVHN, we use mean $(0.4377, 0.4438, 0.4728)$ and standard deviation $(0.1980, 0.2010, 0.1970)$.

A.2 Matched-Loss Endpoint Selection

For each pair of independently trained trajectories, we select checkpoints that reach comparable objective values. For SGD and Adam, the matching objective includes both cross-entropy and the ℓ_2 penalty corresponding to weight decay,

$$L(\theta) = L_{\mathrm{CE}}(\theta) + \frac{\lambda_{\mathrm{wd}}}{2}\|\theta\|_2^2. \tag{4}$$

For AdamW, which trains the model with decoupled weight decay, we use cross-entropy alone. We first determine the range of objective values shared by the two trajectories and choose target loss levels within this common range. For each target, we search over checkpoint pairs whose relative endpoint-loss difference satisfies

$$\frac{|L_A - L_B|}{\frac{1}{2}(L_A + L_B)} \le 0.10. \tag{5}$$

Candidate endpoints are then re-evaluated on the full deterministic training set with recalibrated batch-normalization running statistics, and the pair with the smallest resulting endpoint gap is selected. The target loss used for subsequent curve fitting is

$$L_\star = \frac{L_A + L_B}{2}. \tag{6}$$

A.3 Quadratic Bézier Curve Fitting

Given matched endpoints θ_A and θ_B, we fit a quadratic Bézier curve

$$\gamma(t) = (1-t)^2\theta_A + 2t(1-t)C + t^2\theta_B, \qquad t \in [0,1], \tag{7}$$

where only the control point C is optimized. We initialize the control point at the midpoint, $C = (\theta_A + \theta_B)/2$, and optimize it to keep the objective close to the matched target level as follows:

$$\mathcal{J}(t) = \left[\log L(\gamma(t)) - \log L_\star\right]^2. \tag{8}$$

Table 5: Optimization settings for quadratic Bézier curve fitting.

Setting	Value
Optimizer	Adam
Initial learning rate	10^{-3}
Betas	$(0.9, 0.999)$
Weight decay	0
LR scheduler	Reduce-on-plateau ($\times 0.3$)
Minimum learning rate	10^{-5}
Fitting budget	200 epochs
Batch size	512
Validation interval	5 epochs
Validation grid	21 uniformly spaced t values
Final evaluation grid	20 uniformly spaced t values

Table 5 summarizes the fitting configuration. We optimize the control point C with Adam for at most 200 epochs and evaluate the fitted curve every five epochs using 20 uniformly spaced values of t. During optimization, one stratified random value of t is sampled for each minibatch.

For each validation, we measure its mean and maximum relative deviation from the target loss,

$$E_{\text{mean}} = \frac{1}{M} \sum_{j=1}^{M} \frac{|L(\gamma(t_j)) - L_\star|}{L_\star}, \qquad E_{\max} = \max_j \frac{|L(\gamma(t_j)) - L_\star|}{L_\star}. \tag{9}$$

We regard a fit as successful when $E_{\text{mean}} \leq 0.02$ and $E_{\max} \leq 0.05$. Among successful checkpoints, we select the one with the best validation score $S = E_{\text{mean}} + 0.5E_{\max}$. If no checkpoint satisfies the success criterion within the fitting budget, we instead use the checkpoint with the best validation score encountered during fitting. For models with batch normalization, running statistics are recomputed on the full augmentation-free training set before endpoint and curve evaluation. The final quadratic Bézier curve and the corresponding linear interpolation path $\gamma_{\text{lin}}(t) = (1-t)\theta_A + t\theta_B$ are evaluated using the same objective, evaluation grid, and BN-recalibration procedure.

A.4 Details about AUROC

We evaluate whether example-wise LLC distinguishes corrupted from uncorrupted examples after controlling for endpoint losses. Before classification, examples are deterministically matched 1:1 using their two log-transformed endpoint cross-entropies. We use a 50×50 empirical-quantile grid, optimal within-cell Hungarian matching, and retain pairs within a standardized-distance caliper of 0.25.

For example i, we define the signed LLC residual along the fitted Bézier path as

$$r_i(t) = \ell_i(t) - [(1-t)\ell_i(0) + t\ell_i(1)].$$

Using the 18 interior path points, we compare three feature sets: (i) the two endpoint losses, (ii) the full 18-dimensional signed residual vector (Raw LLC), and (iii) the same vector after per-example mean centering and L_2 normalization (N. LLC), which removes overall offset and amplitude while retaining pathwise shape.

Each feature set is evaluated independently using L2-regularized logistic regression ($C = 1$) with shuffled stratified five-fold cross-validation. Feature standardization is fitted only on each training fold, and AUROC is computed from the pooled out-of-fold probabilities over all matched examples.

We report 95% percentile confidence intervals from 2,000 class-stratified bootstrap resamples of the fixed out-of-fold prediction–label pairs. Models, matching, and cross-validation are not refitted during bootstrapping, so the intervals are conditional on the matched cohort and its OOF predictions.

B Additional Loss-Level Connectivity Results

This section provides additional results supporting the loss-level connectivity analysis in Sec. 2. We report the complete set of matched-loss levels for additional dataset–architecture combinations and

Table 6: The results of LLC behavior along the curve for corrupted and uncorrupted subsets after controlling for endpoint difficulty. Results are shown for CIFAR-100/ResNet-18 at corruption ratios $p = 0.4$ and $p = 0.8$, after matching corrupted and uncorrupted examples to have similar losses at both endpoints. Note that the N. LLC refers to normalized shape-only LLC. AUROCs are computed from out-of-fold predictions using stratified five-fold cross-validation.

Matched loss	$p = 0.4$				$p = 0.8$			
	Matched examples	Endpoint loss AUROC	Raw LLC AUROC	N. LLC AUROC	Matched examples	Endpoint loss AUROC	Raw LLC AUROC	N. LLC AUROC
2.104	16,458	0.491	0.661	0.640	9,955	0.492	0.569	0.572
1.895	17,041	0.492	0.683	0.653	9,936	0.494	0.560	0.550
1.046	16,754	0.494	0.710	0.647	9,839	0.500	0.573	0.555
0.843	16,569	0.495	0.724	0.669	9,884	0.494	0.601	0.544
0.735	15,787	0.495	0.752	0.663	9,884	0.487	0.598	0.543

Table 7: Endpoint-loss-matched mean pointwise residual variance for the CIFAR-100/ResNet-18 $p = 0.5$ memorization intervention across $\lambda \in \{0.25, 0.5, 1, 2\}$.

Mem. target	$\lambda = 0.25$			$\lambda = 0.5$			$\lambda = 1$			$\lambda = 2$		
	U	C	C/U	U	C	C/U	U	C	C/U	U	C	C/U
1%	1.194	1.071	0.90	0.814	0.704	0.86	0.400	0.345	0.86	0.084	0.086	1.04
10%	2.218	2.532	1.14	1.922	1.999	1.04	1.448	1.414	0.98	0.865	0.832	0.96
50%	1.448	2.573	1.78	2.117	4.089	1.93	2.268	3.441	1.52	2.082	2.613	1.26
90%	0.094	0.295	3.13	0.492	0.671	1.37	0.770	0.991	1.29	1.038	1.218	1.17
99%	0.030	0.069	2.28	0.135	0.309	2.28	0.135	0.246	1.83	0.053	0.089	1.69

independently trained trajectory pairs. In all cases, we compare the fitted quadratic Bézier path with the corresponding linear interpolation using the same endpoint-selection, fitting, and evaluation procedures described in Appendix A.

B.1 Across Datasets and Architectures

We first examine whether the LLC behavior extends across different datasets and model architectures. Fig. 8 report the complete results for CIFAR-10 and CIFAR-100 with ResNet-18, ResNet-50, WRN-28-10, and ViT-Tiny. For each dataset–architecture combination, we use a representative pair of independently trained trajectories and show both the quadratic Bézier connection and linear interpolation at every matched loss level.

Across these settings, the quadratic Bézier path consistently provides a substantially closer connection to the matched objective level than linear interpolation. The same qualitative behavior is observed across early, intermediate, and late stages of training, supporting the main-text observation that loss-level connectivity is not specific to a particular dataset or architecture.

B.2 Across Independent Trajectory Pairs

We next examine whether the observed connectivity depends on the particular pair of optimization trajectories being connected. Fig. 9 reports the complete results for multiple independently trained trajectory pairs on CIFAR-100 and SVHN (Netzer et al., 2011) with ResNet-18.

Although the precise shape of the fitted path varies across trajectory pairs, the overall pattern remains consistent. The quadratic Bézier connections stay substantially closer to the matched objective level than the corresponding linear paths across the examined stages.

C Corruption-Ratio Dependence of Example-Wise LLC Behavior

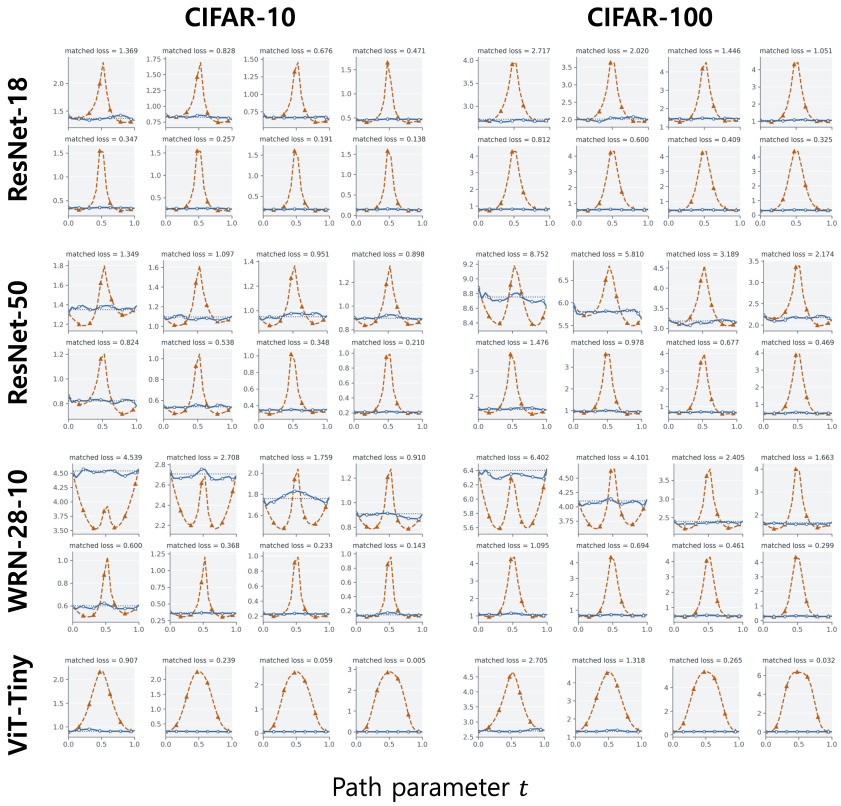

Figure 8: Loss-level connectivity across architectures, datasets, and training stages.

This section extends the example-wise analysis in Sec. 3 by varying the label-corruption ratio. For each corruption ratio p, Fig. 10, 11, and 12 report the loss-level connection for each p. Also, we measure the mean signed residuals of corrupted and uncorrupted examples along the fitted path under various λ settings in Fig. 13.

Next, Table 6 report AUROCs under various p as a extension of the Table 1. We also report the variance along the curve under corrupted dataset in Table 8, which is a extension of Table 3. Additionally, we report the residual variance in Table 7.

Table 8: The mean example-wise variance along the curve. The U and C refer to the uncorrupted and corrupted example, respectively.

Loss	p	U	C	C/U
High	0.6	0.780	0.924	**1.19**
Inter.	0.6	0.061	0.120	**1.96**
Low	0.6	0.014	0.021	**1.54**

Across multiple training stages, most of these results agrees that the difference between the two groups becomes progressively smaller as the corruption ratio increases. At low corruption ratios, corrupted and uncorrupted examples exhibit clearly distinct signed-residual profiles along the same global connection. However, as corrupted examples constitute a larger fraction of the training set, their residual behavior becomes increasingly similar to that of the uncorrupted subset.

This systematic dependence on p is consistent with the interpretation in Sec. 3.1 that the global loss-level connection reflects the aggregate behavior of the training data. When corrupted examples occupy only a small portion of the dataset, their behavior differs more strongly from that represented by the global connection. As their prevalence increases, the fitted connection increasingly incorporates their behavior, reducing the contrast between the two subsets.

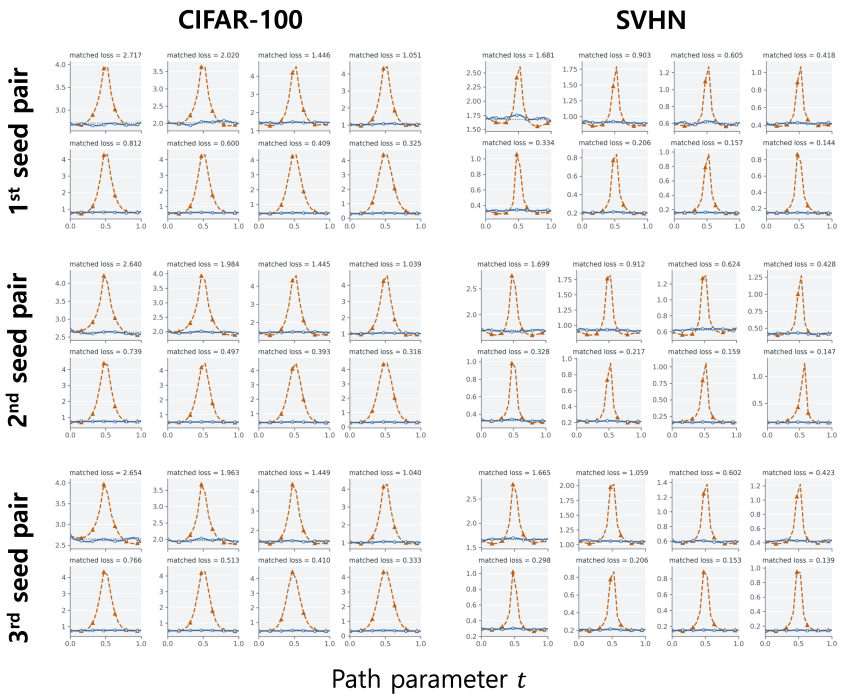

Figure 9: Loss-level connectivity across independently trained trajectory pairs on ResNet-18.

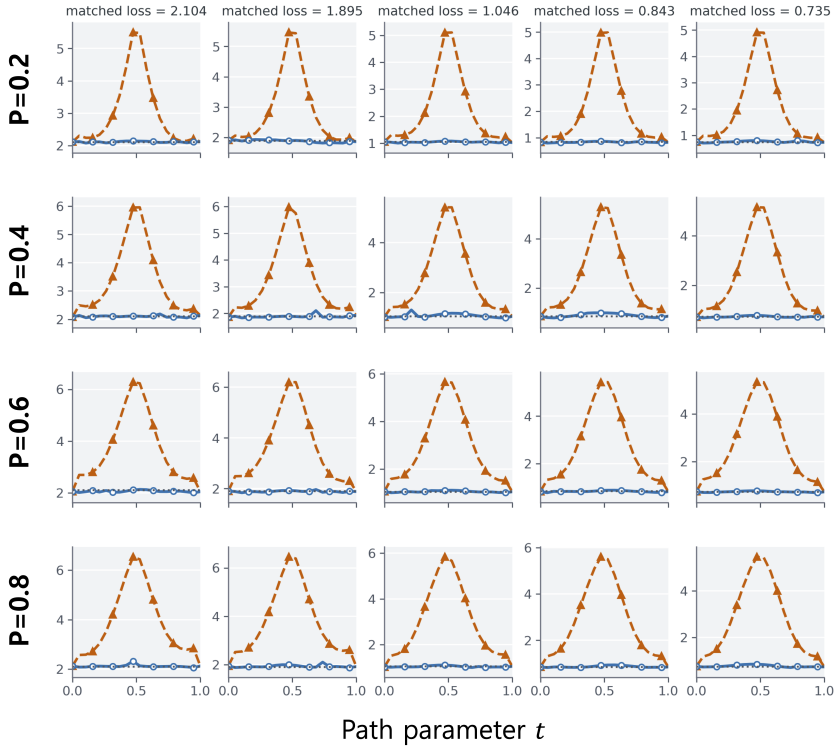

Figure 10: Loss-level connectivity under various corruption ratio p on ResNet-18/CIFAR100.

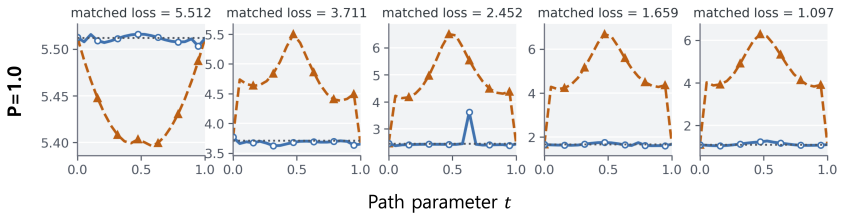

Figure 11: Loss-level connectivity under fully corrupted CIFAR100 on ResNet-18/CIFAR100.

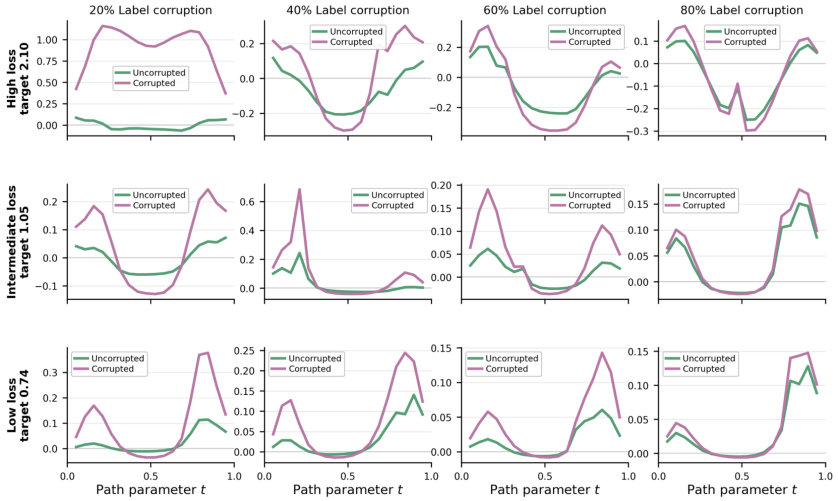

Figure 12: Mean signed residuals along the curve under different corruption ratios ($p = 0.2, 0.4, 0.6,$ and 0.8). The upper, middle, and bottom rows refer to high, intermediate, and low loss stages, respectively.

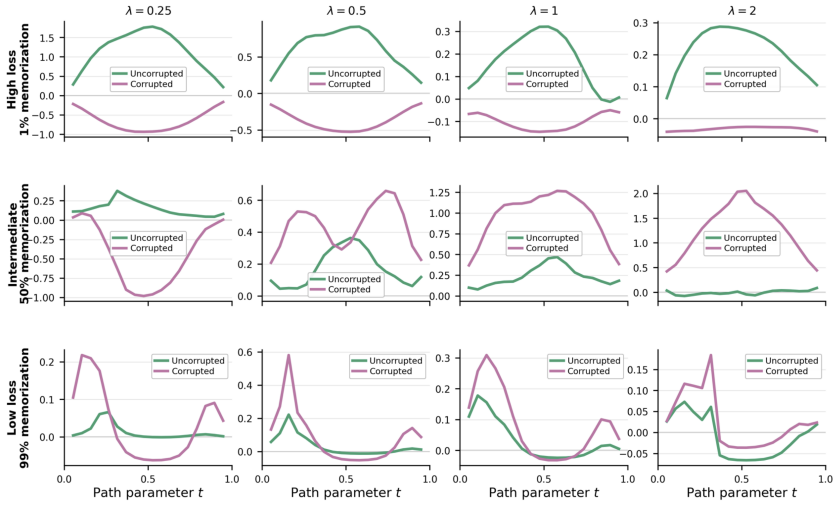

Figure 13: Mean signed residuals along the curve under different λ settings. The upper, middle, and bottom rows refer to high, intermediate, and low training stages, respectively.