What Survives of Path Norms? Path-Lifting as an Intermediate Representation for ReLU Networks

Antoine Gonon [1] **Rémi Gribonval** [2]

Abstract

More than a decade ago, Neyshabur et al. introduced path norms to define complexity measures over classes of functions implemented by ReLU networks, tightening existing bounds by factoring out intrinsic rescaling-invariances of the weight-space parameterization of such networks. While conceptually exciting, this however did not fully match expectations, as path norms bounds often remain several orders of magnitude too large to provide stand-alone quantitative bounds, *e.g.*, on generalization error or Lipschitz constants.

Path-norms are however only the most visible facet of a toolset built on top of *path-lifting* and *path-activations*, two complementary rescaling-invariant representations of ReLU networks. This short perspective paper brings these layers back to the foreground: the lasting contribution of the introduction of path norms is not a single privileged—but overly pessimistic—scalar complexity measure, but a representational toolset for formulating weight-space questions after rescaling has been factored out. As we highlight, this toolset keeps reappearing in invariant embeddings for identifiability, symmetry-aware optimization, conservation laws for gradient flow, pruning, and recent PAC-Bayes analyses, to name a few.

1. Introduction

ReLU networks admit a well-known continuous positive rescaling symmetry, embodied for a single ReLU neuron as:

$$\begin{aligned} f(x) &= v\,\mathrm{ReLU}(\langle u, x\rangle + b), \\ (u, b, v) &\mapsto (\lambda u, \lambda b, \lambda^{-1} v), \qquad \lambda > 0. \end{aligned} \tag{1}$$

Rescaling weights as above leaves the realized function unchanged, and in multilayer networks, the same positive rescaling can be applied independently to each hidden neuron. When applicable, permutations of hidden units add a second, discrete (bounded) symmetry. Because the rescaling orbit is continuous and unbounded, many classical quantities in weight space can vary arbitrarily while the function stays fixed. In particular, plain parameter norms and distances can become arbitrarily large along an equivalence class that corresponds to the same network function.

Path norms were introduced in response to this issue as rescaling-invariant complexity measures (Neyshabur et al., 2015b).[1] They inspired symmetry-aware optimization rules such as Path-SGD (Neyshabur et al., 2015a), and the same intuition has since been extended to modern DAG (directed acyclic graph) ReLU architectures with biases, skip connections, and pooling operators (Gonon et al., 2024). Their quantitative and conceptual limits were first highlighted in the generalization literature, where stand-alone, data-insensitive scalar measures progressively lost credibility (Jiang et al., 2020; Dziugaite et al., 2020). But the practical lesson is broader than generalization bounds: any weight-space method that compares or manipulates parameters through regularization, pruning, merging, editing, posterior design, or perturbation analysis is exposed to gauge arbitrariness unless it is formulated in coordinates adapted to the symmetry.

This literature mixed two objects: coordinates describing networks modulo rescaling, and scalar observables built from those coordinates. The part that remained useful across settings is path-lifting together with path activations: they provide coordinates in which rescaling is factored out while activation dependence remains visible. Some later developments use this representation explicitly; others rely on nearby invariant embeddings or symmetry-aware coordinates motivated by the same geometry (Stock & Gribonval, 2022; Bona-Pellissier et al., 2022; Marcotte et al., 2023;

[1]Institute of Mathematics, EPFL, Lausanne, Switzerland [2]Inria, CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP, UMR 5668, 69342 Lyon Cedex 07, France. Correspondence to: Antoine Gonon <antoine.gonon@epfl.ch>.

Workshop on Weight-Space Symmetries, held in conjunction with the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. 2026.

[1]In layered networks, classical path norms can be read as infima of products of layer norms over all rescaling equivalent parameters, linking them to the norm-based Lipschitz and generalization bounds that were prominent at the time while removing the symmetry breaking introduced by layerwise factorization.

2025; Gonon et al., 2025; Rouchouse et al., 2025). The aim of this paper is a short perspective on this line of work.

2. Path-Lifting and Path Activations

The interest of path-lifting is tied to two basic facts about ReLU networks. First, it captures the continuous rescaling symmetries. Second, together with path activations, it keeps track of which local affine piece (of the piecewise affine function $f_\theta(x)$ realized by the network) is selected by a specific input variable x.

From one neuron to one hidden layer. For a single ReLU neuron $f(x) = v\mathrm{ReLU}(ux)$, with $u, v, x \in \mathbb{R}$, the product uv is one of the simplest invariants to consider under the rescaling $(u, v) \mapsto (\lambda u, \lambda^{-1} v)$. A one-hidden-layer network is the vector version of the same idea, with one such invariant per hidden unit. The realized function writes:

$$\begin{aligned} f_\theta(x) &= \sum_{j=1}^{m} v_j \mathrm{ReLU}(\langle u_j, x\rangle + b_j) + c \\ &= \sum_{j=1}^{m} a_j(x)\big(\alpha_j^\top x + \beta_j\big) + c, \end{aligned}$$

where $\alpha_j = v_j u_j$ and $\beta_j = v_j b_j$, and $a_j(x) = \mathbf{1}_{\langle u_j, x\rangle + b_j \geq 0}$ is the activation of neuron j. Under the hidden-unit rescaling $(u_j, b_j, v_j) \mapsto (\lambda u_j, \lambda b_j, \lambda^{-1} v_j)$, the coefficients (α_j, β_j) are unchanged, as well as the input-dependent activations $a_j(x)$. The lift stores $\Phi(\theta) = \Phi(u, b, v) := ((\alpha_j, \beta_j))_{j=1}^m$, where $u = (u_j)$, $v = (v_j)$ and $b = (b_j)$, and path activations reduce to $(a_j(x))_{j=1}^m$.

Generalization to the layered setting By collecting the products of weights along each path from input to output, which are invariant under the rescaling symmetries of each hidden unit along any path p, we obtain the *path-lifting*:

$$\Phi(\theta) = \Big(\prod_{e \in p} w_e\Big)_{p \in \mathcal{P}}, \quad \text{where } \theta = (w_e)_e, \tag{2}$$

possibly augmented in modern architectures with the bookkeeping needed for biases, skip connections, pooling or frozen batch-norm (Gonon et al., 2024). The exact formula depends on the architecture, but in each case the lift is constructed so that equivalent parameterizations collapse to the same coordinates, or at least admit a simpler local description modulo symmetry. It is complemented by the *path activations* $a(\theta, x)$, which record which paths are active on the current input (i.e., all neurons along the path are active).

For many modern ReLU DAG parameterizations, these objects lead to a representation of the realized piecewise affine function in which rescaling is factored out, while the active affine piece selected by the input is exposed via path activations. For instance, for a scalar-output network, the function can be written as a sum of path contributions:

$$f_\theta(x) = \left\langle \Phi(\theta) \odot a(\theta, x), A \begin{pmatrix} x \\ 1 \end{pmatrix} \right\rangle, \tag{3}$$

and a similar representation holds for vector outputs. Here, the matrix A is fixed by the architecture and maps which input or bias coordinate should be fed into each path. This equation indeed exposes local affine dependence on the input x: the path activations $a(\theta, x)$ select which paths are active on the current input, and the lift $\Phi(\theta)$ collects the corresponding invariant coefficients that determine the slope and intercept of the selected affine piece.

Local linearity. Since locally around any (θ, x), the activation pattern is fixed, except on measure-zero boundaries, the representation $\theta \mapsto f_\theta(x)$ is locally linear in the lifted coordinates. This local linearity is a key reason why the lift is a natural representation for many weight-space questions, even those that do not explicitly involve path products. For instance, if two networks share the same activation pattern on an input x, then their output difference on that input is directly controlled by the active part of the lift difference,

$$|f_\theta(x) - f_{\theta'}(x)| \leq \|(\Phi(\theta) - \Phi(\theta')) \odot a(\theta, x)\|_1 \|(x, 1)\|_\infty.$$

This type of local estimate is naturally relevant for pruning, quantization, or other weight perturbations, for instance in PAC-Bayes analyses. The main challenge is therefore to characterize path activations $a(\theta, x)$ well enough for observables built from $\Phi(\theta) \odot a(\theta, x)$ to remain both tractable, invariant, and informative.

Coarseness of current observables. So far, tractable observables built from $\Phi(\theta) \odot a(\theta, x)$ have been global norms of $\Phi(\theta)$ that discard the activation information. For instance, path norms collapse $\Phi(\theta)$ to $\|\Phi(\theta)\|_q$, when finer activation-filtered quantities instead involve $\Phi(\theta) \odot a(\theta, x)$; path metrics also discard the activation information and compare two networks through $\|\Phi(\theta) - \Phi(\theta')\|_1$ (Gonon et al., 2025). *These observables are invariant and tractable to compute, but they are also coarse, activation-insensitive, and therefore data-insensitive, yielding worst-case statements*, such as those in Gonon et al. (2025) which is obtained by combining local bounds of the previous form across activation regions. At that point, three ingredients are already present and should be distinguished: the representation, the observable built on top of it, and the downstream task.

The rest of this note illustrates that distinction on optimization questions, where the separation between functional motion and gauge motion is especially transparent.

3. Optimization as an Illustration

The previous section introduced path-lifting as a way to describe ReLU networks modulo rescaling, and pinpointed

the coarseness issues of the path norm approach historically introduced in the Lipschitz and generalization settings.

To make the payoff of path-lifting and path-activations more concrete, it is useful to shift away from these historical settings and look instead at optimization. Optimization is a natural illustration because moving along a rescaling orbit in parameter space leaves the realized function (and the path lifting / path activations) unchanged, yet can strongly modify Euclidean gradients and the local landscape seen by the raw coordinates. In that sense, the potential of path-lifting on the optimization side comes from its ability to disentangle functional motion from gauge motion. Adopting this viewpoint, three broad questions arise:

- How to *better understand* the original dynamics in raw coordinates
- How to *improve* the raw coordinates dynamics by moving at each step to a different representative along the rescaling orbit (via teleportation or penalization).
- How to *change the descent geometry itself* by running dynamics directly in another space, so that equivalent parameterizations receive equivalent updates.

The point here is not to rank these directions, but to show that the same invariant formulation helps study all three.

To keep the discussion concrete, consider again the one-hidden-layer model and a fixed training set $(x_i, y_i)_{i=1}^n$. On any open set Ω of parameter space where the activation pattern of each training sample is fixed, one has

$$f_\theta(x_i) = \sum_{j=1}^{m} a_{ij}\big(\alpha_j^\top x_i + \beta_j\big) + c, \qquad a_{ij} \in \{0,1\},$$

with a_{ij} constant on Ω. The empirical loss can therefore be written locally as

$$\mathcal{L}(u,b,v) = \widetilde{\mathcal{L}}_\Omega(\Phi(u,b,v))\,,$$

This local factorization is the common starting point below: on each activation region, the loss depends on the parameters through invariant coefficients, while the activation pattern is frozen in the background and incorporated into $\widetilde{\mathcal{L}}_\Omega$.

Understanding the raw dynamics – via conservation laws. The first question is to keep gradient flow in the original coordinates and ask what the dynamics can preserve. For a local factorization $\mathcal{L} = \widetilde{\mathcal{L}}_\Omega \circ \Phi$, any quantity H that is conserved for every choice of $\widetilde{\mathcal{L}}_\Omega$ must satisfy

$$D\Phi(\theta)\nabla H(\theta) = 0,$$

or equivalently $\nabla H(\theta) \in \ker D\Phi(\theta)$. The search for conserved quantities is therefore reduced to linear algebra on the Jacobian of the local invariant map. Already for a single hidden unit, with $\Phi(u,b,v) = (vu, vb)$,

$$D\Phi(u,b,v)[\delta_u,\delta_b,\delta_v] = \big(v\delta_u + u\delta_v,\ v\delta_b + b\delta_v\big),$$

so $\ker D\Phi(u,b,v) = \mathrm{span}\{(u,b,-v)\}$, and one recovers the familiar conserved quantity $\|u\|_2^2 + b^2 - v^2$ as being provably the *only* conserved quantity. The point here is that an explicit symmetry-adapted factorization turns the search for all independent conservation laws into a finite-dimensional problem, and a natural factorization is available for ReLU networks thanks to the path-lifting locally linearizing the representation. This is the viewpoint developed by Marcotte et al. (2023): symmetry-adapted factorizations are used to compute or certify the available conservation laws, first for ReLU networks and later for ResNets and Transformers (Marcotte et al., 2025). This viewpoint has also led to the analysis of the dynamics of $z(t) := \Phi(\theta(t))$ when $\theta(t)$ follows the gradient flow dynamics, showing via Lie algebra calculus that $z(t)$ satisfies an ordinary differential equation (ODE) determined by initialization (Marcotte et al., 2026).

Improving raw training by moving along the orbit. A second line of works aims to improve dynamics in raw coordinates by moving at each step to a different representative along the rescaling orbit. The motivation is that the raw dynamics is not invariant, so future performance can be improved by choosing a better representative inside the same equivalence class. Various criteria can be considered for selecting a representative. For instance, one can try to select a representative that maximizes the norm of the raw gradient, or maximizes balance between the layers. They all amount to selecting a representative θ^+ from the same equivalence class as θ that minimizes some criterion Q:

$$\theta^+ \in \arg\min_{\theta' \sim \theta} Q(\theta'),$$

where $\theta' \sim \theta$ means that θ' is obtained from θ by rescaling hidden units and therefore represents the same function. In the one-hidden-layer case, θ' takes the form $(\lambda_j u_j, \lambda_j b_j, \lambda_j^{-1} v_j)_{j=1}^m$. Different works then choose different criteria Q. Equi-normalization selects a norm-balanced representative by iterative rescaling (Stock et al., 2019). Neural teleportation proposes orbit moves guided by a gradient-based criterion (Armenta et al., 2020). Path-conditioned training chooses initialization-time rescalings using a conditioning criterion expressed in path space (Lebeurrier et al., 2026). These methods do not answer the same question, but they all start from the same observation: raw optimization depends on the chosen representative even when the function does not. The lift is useful here because it characterizes the equivalence class and can serve to define the criterion Q in a way that is invariant to rescaling, for instance by using the path representation to define a conditioning criterion that does not depend on the choice of representative (Lebeurrier et al., 2026).

Changing the optimization geometry. A third line of works does not try to understand or improve the raw dynamics, but rather to define a new descent geometry in which equivalent parameterizations receive equivalent updates, akin to running optimization directly in the quotient space. Path-SGD in (Neyshabur et al., 2015a) is a good example. It replaces the Euclidean step by a rescaling-invariant update of the form

$$w_e^+ = w_e - \eta\, \kappa_e(w)^{-1} \frac{\partial \mathcal{L}}{\partial w_e},$$

where $\kappa_e(w)$ is derived from a path-wise regularizer (Neyshabur et al., 2015a). Here the path structure is not used to select a representative, but to define a metric in which equivalent parameterizations induce equivalent steps. This is a different question from understanding raw gradient flow or moving along the orbit, but it starts from the same observation: Euclidean descent in raw coordinates mixes functional motion with gauge motion.

These three questions are genuinely different, but a common technical tool to study all of them is an invariant representation of the parameters that factors out symmetries. The path-lifting is a natural such representation, useful for distinguishing what belongs to the raw dynamics, what belongs to the choice of representative inside an equivalence class, and how the optimization geometry mixes the two.

4. Conclusion

The perspective described in this paper suggests that path-lifting has been more useful as a symmetry-adapted representation than as a source of stand-alone complexity scalars—path norms—built on top of it. Optimization was used as an illustration precisely because it makes that point visible beyond the Lipschitz and generalization settings historically associated with such scalar constructions. The point is not that the full lift should be computed in practice, but that it provides a clean starting point for deciding which tractable observables or procedures are relevant for a given question.

References

Armenta, M. A., Judge, T., Painchaud, N., Skandarani, Y., Lemaire, C., Sanchez, G. G., Spino, P., and Jodoin, P.-M. Neural teleportation, 2020.

Bona-Pellissier, J., Malgouyres, F., and Bachoc, F. Local identifiability of deep relu neural networks: the theory. In *Advances in Neural Information Processing Systems*, 2022.

Dziugaite, G. K., Drouin, A., Neal, B., Rajkumar, N., Caballero, E., Wang, L., Mitliagkas, I., and Roy, D. M. In search of robust measures of generalization. In *Advances in Neural Information Processing Systems*, 2020.

Gonon, A., Brisebarre, N., Riccietti, E., and Gribonval, R. A path-norm toolkit for modern networks: Consequences, promises and challenges. In *International Conference on Learning Representations*, 2024.

Gonon, A., Brisebarre, N., Riccietti, E., and Gribonval, R. A rescaling-invariant lipschitz bound based on path-metrics for modern relu network parameterizations. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 20047–20074. PMLR, 2025.

Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., and Bengio, S. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020.

Lebeurrier, A., Vayer, T., and Gribonval, R. Path-conditioned training: a principled way to rescale relu neural networks, 2026.

Marcotte, S., Gribonval, R., and Peyré, G. Abide by the law and follow the flow: Conservation laws for gradient flows. In *Advances in Neural Information Processing Systems*, 2023.

Marcotte, S., Gribonval, R., and Peyré, G. Transformative or conservative? conservation laws for resnets and transformers. In *International Conference on Machine Learning*, 2025.

Marcotte, S., Peyré, G., and Gribonval, R. Intrinsic training dynamics of deep neural networks. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=IlyesljaNb.

Neyshabur, B., Salakhutdinov, R., and Srebro, N. Path-sgd: Path-normalized optimization in deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 2422–2430, 2015a.

Neyshabur, B., Tomioka, R., and Srebro, N. Norm-based capacity control in neural networks. In *Proceedings of the 28th Conference on Learning Theory*, pp. 1376–1401. PMLR, 2015b.

Rouchouse, D., Gonon, A., Gribonval, R., and Guedj, B. Non-vacuous generalization bounds: Can rescaling invariances help?, 2025. OpenReview submission.

Stock, P. and Gribonval, R. An embedding of relu networks and an analysis of their identifiability. *Constructive Approximation*, 2022.

Stock, P., Graham, B., Gribonval, R., and Jégou, H. Equi-normalization of neural networks. In *International Conference on Learning Representations*, 2019.