DLN experiments

February 19, 2026

1 Golden Path hypothesis

We understand the learning dynamics of gradient flow under assumptions:

- Balanced weights (small initialization)
- Decoupled modes
- Covariance matrices are simultaneously diagonalizable (properties of the data)

Under such assumptions, gradient flow is analytically solvable and is described in detail in the supplementary materials of A mathematical theory of semantic development in deep neural networks . With Avi, Strang, etc we extended the learning dynamics to stochastic gradient flow under the same assumption (pdf attached in the overleaf project: Diffusion of SGD on Deep Linear Networks Preprint.pdf). The relevant propositions are proposition 2.5 (modewise SDEs) and proposition 4.2 (induced marginal distribution from Fokker Planck). I would prioritize speed for these experiments at first: qualitative results, descriptive stats before doing proper hypothesis testing. Some quick and dirty experiment to try first:

- Compute difference between rank pattern of a few SGD trajectories and GD
- Compute distance as frobenius norm between GD and SGD weight matrices of each layer
- Check distance in function space for GD and SGD is small

1.1 Experimental Protocol: Tracking Rank Patterns under GD and SGD.

There are a lot of internal symmetries, what we should expect instead is that the parameters would have same complexity: weight matrices have same rank patterns. Consider a deep linear network of depth L with weights $\{W_\ell(t)\}_{\ell=1}^{L}$ trained on the same dataset and initialization for both full-batch gradient descent (GD) and stochastic gradient descent (SGD). Fix a global singular value threshold parameter $\varepsilon > 0$ (e.g. 10^{-5}) that will be used consistently across all runs and checkpoints.

1. Training and Checkpointing.

1. Run one GD trajectory and S independent SGD trajectories (same initialization, different batch orderings we care about the effect of gradient noise).

2. Choose a checkpoint alignment rule (recommended: loss-aligned checkpoints at fixed loss values and escape checkpoints defined by the first significant drop from a saddle plateau).

3. At each selected checkpoint time t_k, save all layer weights $\{W_\ell(t_k)\}_{\ell=1}^{L}$.

2. Rank Pattern Computation. For each saved checkpoint:

1. For all $1 \le i \le j \le L$, compute the contiguous partial products

$$P_{ij}(t_k) = W_j(t_k) \cdots W_i(t_k).$$

2. Compute the singular values $\{\sigma_m(P_{ij})\}$ and define the numerical rank (or use a ready made python function to compute rank)

$$r_{ij}(t_k) = \#\{m : \sigma_m(P_{ij}) > \varepsilon\, \sigma_1(P_{ij})\}.$$

3. Record the singular value gap at the threshold (ratio σ_r/σ_{r+1}) to monitor rank ambiguity.

3. Multiplicity Pattern (Optional not a priority). Compute the multiplicity pattern via the discrete inclusion–exclusion transform

$$m_{ij}(t_k) = r_{ij}(t_k) - r_{i,j+1}(t_k) - r_{i-1,j}(t_k) + r_{i-1,j+1}(t_k),$$

with zero boundary conventions. If small negative values occur due to numerical noise, clip to zero.

4. Aggregation for SGD. At each checkpoint t_k, compute:

- The mean rank pattern $\bar{r}_{ij}^{\mathrm{SGD}}(t_k)$ across the S runs.
- The mean multiplicity pattern $\bar{m}_{ij}^{\mathrm{SGD}}(t_k)$.
- Entrywise standard deviations and bootstrap confidence intervals.

5. GD vs. SGD Comparison Metrics. For each checkpoint compute:

$$d_r(t_k) = \sum_{i \le j} \left| r_{ij}^{\mathrm{GD}}(t_k) - \bar{r}_{ij}^{\mathrm{SGD}}(t_k) \right|,$$

$$d_m(t_k) = \sum_{i \le j} \left| m_{ij}^{\mathrm{GD}}(t_k) - \bar{m}_{ij}^{\mathrm{SGD}}(t_k) \right|. \quad \text{OPTIONAL}$$

Plot $d_r(t_k)$ and $d_m(t_k)$ as functions of aligned training time, with confidence bands for SGD. Emphasize behavior in temporal windows surrounding saddle escape events.

Deliverables.

- For different regimes (saddle to saddle, intermediate, NTK) and different data rank: consider teacher matrix that is not full rank, full rank and degenerate.
- Heatmaps of r_{ij} for GD and mean SGD at selected checkpoints.
- Trajectories of $d_r(t)$ with uncertainty bands.
- Diagnostics of singular value gaps to identify unstable rank transitions.

1.2 Rank computation

We estimate the numerical rank of a matrix $A \in \mathbb{R}^{m\times n}$ using two complementary notions. First, the *relative-threshold rank* is defined as

$$r_\varepsilon(A) \;=\; \#\{i : \sigma_i(A) > \varepsilon\,\sigma_1(A)\},$$

where $\sigma_1(A) \geq \cdots \geq \sigma_{\min(m,n)}(A)$ are the singular values and $\varepsilon > 0$ is a fixed threshold. This definition is invariant under global rescaling $A \mapsto cA$, since both σ_i and σ_1 scale linearly in c, so the inequality $\sigma_i > \varepsilon\sigma_1$ is unchanged. Such scale invariance is essential in deep linear networks, where norms of partial products may vary substantially during training while their spectral *shape* (relative magnitudes) carries the structural information of interest. The threshold ε should be chosen above the effective noise floor of the pipeline: singular values below this scale are dominated by non-structural perturbations arising from (i) stochastic gradient noise in SGD, which induces diffusion-like fluctuations in the weights, (ii) finite-sample effects that lift small modes through sampling variability, and (iii) floating-point roundoff, which limits reliable resolution to $O(u\|A\|)$, where u is machine precision. In practice, ε is selected so that modes counted by r_ε are stable across seeds and checkpoints, while smaller singular values exhibit variance comparable to their magnitude.

Second, we track an *effective rank*, defined for example via the entropy of the normalized singular values

$$r_{\mathrm{eff}}(A) \;=\; \exp\Big(-\sum_i p_i \log p_i\Big), \qquad p_i = \frac{\sigma_i(A)}{\sum_j \sigma_j(A)},$$

or alternatively through a participation ratio such as

$$r_{\mathrm{PR}}(A) \;=\; \frac{\big(\sum_i \sigma_i^2(A)\big)^2}{\sum_i \sigma_i^4(A)}.$$

These quantities are also invariant under global scaling and provide a continuous measure of spectral complexity. Although they do not reproduce the discrete rank pattern required for inclusion–exclusion multiplicity analyses, they vary smoothly with the spectrum and are therefore more robust to small perturbations or near-degenerate singular value gaps. Tracking effective rank alongside r_ε allows us to distinguish genuine structural rank transitions from gradual spectral reshaping or noise-dominated fluctuations.

Protocol: Calibration of Numerical Rank Estimation

The goal of this protocol is to calibrate the numerical rank estimator

$$r_\varepsilon(A) = \#\{i : \sigma_i(A) > \varepsilon\,\sigma_1(A)\},$$

by validating it in controlled settings where the ground-truth rank behavior is known. We perform two complementary calibration experiments and a systematic sensitivity analysis over ε.

I. Synthetic Perturbation Model: $A = M + N$. Objective. Determine how well r_ε recovers a known rank r in the presence of controlled noise, and estimate a practical noise floor.

Setup.

1. Fix dimensions $d_{\text{out}}, d_{\text{in}}$.
2. Construct a rank-r signal matrix

$$M = U \operatorname{diag}(s_1, \ldots, s_r, 0, \ldots, 0)\, V^\top,$$

where U, V are orthogonal and $s_1 \geq \cdots \geq s_r > 0$.

3. Generate noise matrices N under different models:
   - *Additive Gaussian noise:* $N_{ij} \sim \mathcal{N}(0, \sigma^2)$, with σ chosen to match empirically observed SGD-induced perturbations.
   - *Finite-sample proxy:* construct M via empirical covariance from n samples and vary n i.e. $\hat{M} = \frac{1}{n}\sum_{k=1}^{n} y_k x_k^\top$ with x_k, y_k such that $y_k = Mx_k$ and x_k i.i.d. from a standard distribution and whitened.
   - *Precision stress test:* repeat computations in float32 and float64.
4. Form $A = M + N$.

Procedure.

1. Compute singular values $\{\sigma_i(A)\}$.
2. For a grid of thresholds $\varepsilon \in \mathcal{E}$ (log-spaced), compute $r_\varepsilon(A)$ and compute effective rank $r_{\text{eff}}(A)$.
3. Repeat over multiple noise realizations.

Diagnostics.

- Recovery rate: $\mathbb{P}(r_\varepsilon(A) = r)$ over noise draws.
- Stability: variance of $r_\varepsilon(A)$ across draws.
- Gap analysis: record σ_r/σ_{r+1}.

Outcome. Identify a range of ε such that: (i) $r_\varepsilon(A)$ recovers r with high probability, and (ii) the estimated rank is stable across noise realizations. This determines a lower bound on ε consistent with the effective noise floor.

II. DLN Calibration in Saddle-to-Saddle Regime. Objective. Validate that r_ε reproduces the expected stagewise rank growth in deep linear networks.

Setup.

1. Consider a depth-$L = 3$ deep linear network.
2. Choose a teacher matrix M of rank r with ordered singular values $s_1 > \cdots > s_r$.
3. Train using:
   - Full-batch gradient descent (GD).
   - S independent SGD runs (same initialization, different data orderings).

4. Work in the saddle-to-saddle regime (small initialization, balanced weights).

Procedure.

1. At loss plateaus (e.g. fixed loss values or escape events), compute the end-to-end matrix $W(t)$.
2. Compute singular values and $r_\varepsilon(W(t))$ for each $\varepsilon \in \mathcal{E}$.

Expected behavior. Under stagewise learning dynamics:

- The numerical rank should start at 0 (up to noise).
- Rank should increase approximately one unit at a time.

Diagnostics.

- Cross-seed stability: variance of r_ε across SGD runs.
- Gap monitoring: σ_r/σ_{r+1} at each transition.

Outcome. Select ε values for which rank growth is: (i) monotone up to rare transition noise, (ii) predominantly unit-step (iii) stable across SGD seeds.

III. Sensitivity Sweep over ε. Procedure.

1. Define a log-spaced grid $\mathcal{E}$ (e.g. 10^{-7} to 10^{-3} in float64).
2. Repeat previous analyses for each ε.

Report.

- Plot rank trajectories for multiple ε.
- Report regions where qualitative conclusions are invariant to ε.
- Identify a plateau of ε values yielding stable and interpretable rank patterns.

Interpretation. A well-calibrated numerical rank estimator should:

1. Correctly recover known rank under controlled perturbations.
2. Exhibit stable stagewise growth in DLN training.
3. Remain qualitatively unchanged across a nontrivial interval of ε.

If no such ε plateau exists, then discrete rank is intrinsically ambiguous in the regime under study, and conclusions should rely more heavily on continuous spectral measures (e.g. effective rank).

1.3 Experimental Protocol: Singular-Value Patterns (Top–K) under GD vs. SGD.

We can also check the singular values of the individual layers for GD and SGD and check that they are approximately, or eventually become approximately the same throughout training. We compare GD and SGD by tracking the singular values of all contiguous partial products in a deep linear network of depth L. Fix a target truncation level K (e.g. $K = 5$ or 10) and a numerical floor $\delta > 0$ (e.g. $\delta = 10^{-30}$) used to avoid taking $\log(0)$. Use the *same* initialization $W_\ell(0)$ for GD and for every SGD run; vary only the mini-batch shuffling/permutation seed across SGD runs.

1. Training and Checkpointing.

1. Run one full-batch GD trajectory and S independent SGD trajectories (same initialization, different data-order seeds).
2. Choose a checkpoint alignment rule (recommended: loss-aligned checkpoints at fixed loss values, or escape-aligned checkpoints defined around plateau-to-drop events).
3. At each checkpoint time t_k, save the full set of layer weights $\{W_\ell(t_k)\}_{\ell=1}^{L}$ for GD and for each SGD run.

2. Partial Products and Top–K Spectra. For each saved checkpoint t_k (and each run):

1. For all $1 \le i \le j \le L$, compute the contiguous partial products
$$P_{ij}(t_k) = W_j(t_k) \cdots W_i(t_k).$$
2. Compute the top–K singular values of each $P_{ij}(t_k)$:
$$\sigma_1^{(ij)}(t_k) \ge \sigma_2^{(ij)}(t_k) \ge \cdots \ge \sigma_K^{(ij)}(t_k) \ge 0,$$
using a deterministic SVD routine (or randomized SVD with a fixed tolerance and oversampling parameter).
3. Form the *log-singular-value pattern* tensor entries
$$s_{ij,k}(t_k) = \log\big(\max\{\sigma_k^{(ij)}(t_k), \delta\}\big), \qquad k = 1, \dots, K.$$
4. (Optional but recommended) Also record the *shape-normalized* spectrum
$$\tilde{s}_{ij,k}(t_k) = \log\Big(\frac{\max\{\sigma_k^{(ij)}(t_k), \delta\}}{\max\{\sigma_1^{(ij)}(t_k), \delta\}}\Big),$$
to separate scale (σ_1) from spectral shape.

3. Aggregation across SGD Runs. At each checkpoint t_k, compute the SGD mean patterns entrywise in log-space:
$$\bar{s}_{ij,k}^{\mathrm{SGD}}(t_k) = \frac{1}{S}\sum_{s=1}^{S} s_{ij,k}^{(s)}(t_k), \qquad \bar{\tilde{s}}_{ij,k}^{\mathrm{SGD}}(t_k) = \frac{1}{S}\sum_{s=1}^{S} \tilde{s}_{ij,k}^{(s)}(t_k),$$

and estimate uncertainty using bootstrap confidence intervals over the S runs (resample runs with replacement).

4. GD vs. SGD Discrepancy Metrics. For each checkpoint t_k, quantify the discrepancy between GD and SGD using ℓ_1 distances:

$$d_\sigma(t_k) = \sum_{1\le i\le j\le L} \sum_{k=1}^{K} \left| s_{ij,k}^{\mathrm{GD}}(t_k) - \bar{s}_{ij,k}^{\mathrm{SGD}}(t_k) \right|,$$

and, if using normalized spectra,

$$d_{\mathrm{shape}}(t_k) = \sum_{1\le i\le j\le L} \sum_{k=1}^{K} \left| \tilde{s}_{ij,k}^{\mathrm{GD}}(t_k) - \bar{\tilde{s}}_{ij,k}^{\mathrm{SGD}}(t_k) \right|.$$

Optionally use interval-dependent weights w_{ij} (e.g. $w_{ij} = j - i + 1$) to emphasize long products:

$$d_{\sigma,w}(t_k) = \sum_{i\le j} w_{ij} \sum_{k=1}^{K} \left| s_{ij,k}^{\mathrm{GD}}(t_k) - \bar{s}_{ij,k}^{\mathrm{SGD}}(t_k) \right|.$$

5. Reporting (Focused on Saddle Escape).

- Plot $d_\sigma(t_k)$ and (if applicable) $d_{\mathrm{shape}}(t_k)$ versus aligned time, with bootstrap confidence bands from SGD.
- Provide heatmaps of $s_{ij,1}(t_k) = \log \sigma_1^{(ij)}(t_k)$ and selected $k > 1$ slices (and/or $\tilde{s}$ slices) at key checkpoints before, during, and after escape.
- Report the same analyses in an escape-centered window $t_k \in [t_{\mathrm{esc}} - \Delta, t_{\mathrm{esc}} + \Delta]$ where t_{esc} is the escape time defined by the chosen alignment rule.

Distance in parameter space Simply check distance in parameter space, that is, test the difference in parameter space between:

$$\mathbf{E}_B ||\theta_{GD} - \theta_{SGD}||_2$$

Average over batch partitions. I would also plot $||\theta_{GD}||$ with small initialization to have a better sense of scale (it should correspond to parameters at various saddle during training) as well as the ratio:

$$\frac{\mathbf{E}_B ||\theta_{GD} - \theta_{SGD}||_2}{||\theta_{GD}||_2}$$

Check cosine similarity (averaged over batching):

$$\cos(\theta_{GD}, \theta_{SGD}) = \frac{\theta_{GD} \cdot \theta_{SGD}}{\|\theta_{GD}\|_2 \, \|\theta_{SGD}\|_2}$$

Check layerwise distance (might be useful but we might drop it):

$$d_l = \frac{\mathbf{E}_B ||W_{l,GD} - W_{l,SGD}||_F}{||W_{l,GD}||_F}$$

and aggregate:

$$d = \frac{1}{L}\sum_l d_l$$

First we want some qualitative results (descriptive: no hypothesis testing). Prediction: I expect this to be false at many training times because just because of the internal symmetries: sgd and gd could select different symmetries. However because of mode alignment it might becomes partially true as we cross saddle points until it eventually become true at the end of training. But the most interesting version is that training select model of similar complexity (internal rank patterns and singular values of internal layers).

Distance in function space In function space, GPH is true at convergence because the loss is quadratic and convex in function space and we know from Implicit Regularization or Implicit Conditioning? Exact Risk Trajectories of SGD in High Dimensions that GPH holds in function space. We might still want to check

$$\frac{\mathbf{E}_B||W_{GD} - W_{SGD}||_F}{||W_{GD}||_F}$$

Just to be sure.

Balanced assumptions To directly assess the balance condition between adjacent layers, we measure the normalized residual of the balance matrix

$$G_l = W_l W_l^\top - W_{l+1}^\top W_{l+1}.$$

Specifically, we define

$$r_l \;=\; \frac{\left\|W_l W_l^\top - W_{l+1}^\top W_{l+1}\right\|_F}{\left\|W_l W_l^\top\right\|_F + \left\|W_{l+1}^\top W_{l+1}\right\|_F},$$

which is invariant to overall weight scaling. Exact balance corresponds to $r_l = 0$, while small values of r_l indicate approximate balance up to numerical tolerance. We track r_l throughout training to evaluate whether adjacent layers satisfy the balance condition.

Mode decoupling To empirically test mode coupling (and its eventual suppression), fix a teacher matrix $\mathbf{M} = \mathbf{USV}^\top$ with singular vectors $\{\mathbf{u}_\alpha\}_{\alpha=1}^r$ and $\{\mathbf{v}_\beta\}_{\beta=1}^r$, and let $\mathbf{W}(t)$ denote the student weight matrix at training time t. We define the *mode coefficients* by

$$w_{\alpha\beta}(t) \;\triangleq\; \mathbf{u}_\alpha^\top \mathbf{W}(t)\, \mathbf{v}_\beta,$$

and call the *cross modes* those with $\alpha \neq \beta$. A convenient scalar summary of cross-coupling is the Frobenius energy off the diagonal,

$$E_{\text{cross}}(t) \;\triangleq\; \sum_{\alpha\neq\beta} \left|w_{\alpha\beta}(t)\right|^2 \quad \text{(optionally normalized by } E_{\text{tot}}(t) = \sum_{\alpha,\beta} |w_{\alpha\beta}(t)|^2\text{)}.$$

In experiments, we record $\{w_{\alpha\beta}(t)\}$ throughout training and verify that (i) for each fixed α, the set $\{|w_{\alpha\beta}(t)| : \beta \neq \alpha\}$ decays toward 0 and (ii) $E_{\text{cross}}(t)$ (or $E_{\text{cross}}(t)/E_{\text{tot}}(t)$) decreases toward 0

before the corresponding diagonal mode $w_{\alpha\alpha}(t)$ becomes large. Concretely, define a small threshold $\varepsilon > 0$ and compare the first times

$$t_{\text{cross}}(\alpha) \triangleq \inf\Big\{t : \max_{\beta\neq\alpha} |w_{\alpha\beta}(t)| \le \varepsilon\Big\}, \qquad t_{\text{diag}}(\alpha) \triangleq \inf\Big\{t : |w_{\alpha\alpha}(t)| \ge c\, s_\alpha\Big\},$$

for a constant $c \in (0,1)$ (e.g. $c = 0.5$) and teacher singular value s_α; the empirical claim is that typically $t_{\text{cross}}(\alpha) < t_{\text{diag}}(\alpha)$ across modes α, seeds, and batch partitions. We report trajectories of $\max_{\beta\neq\alpha} |w_{\alpha\beta}(t)|$, $|w_{\alpha\alpha}(t)|$, and $E_{\text{cross}}(t)$ (mean $\pm$ s.e.m. over runs), thereby directly checking that all cross modes are driven to zero prior to (or at least no later than) the growth of the corresponding aligned diagonal modes.

2 Sampling saddle points

We leverage the algebraic parameterization of first-order critical points in deep linear networks (DLNs) given by the rank condition and the cyclic conditions. The objective is to turn these polynomial constraints into a numerical sampler that produces many (typically gauge-inequivalent) saddle points.

Rank strata and target projection. Let the teacher (or population) matrix be

$$M = U\,\mathrm{diag}(s_1, \ldots, s_{d_{\text{out}}})\, V^\top,$$

and fix a rank r and a subset S of singular directions with $|S| = r$. Define U_S as the columns of U indexed by S, and the corresponding projector $P_S := U_S U_S^\top$. A rank-r critical stratum is characterized by the end-to-end product

$$W := W_L \cdots W_1 = P_S M, \qquad \mathrm{rank}(W) = r.$$

In practice, we choose $r < r_{\max}$ to avoid global minimizers and bias toward saddle points.

Gauge-fixed normal form and residual variables. We work in a fixed representative of the hidden-layer gauge orbit. Each hidden dimension d_ℓ is split into a signal part of dimension r and a residual part of dimension $d_\ell - r$. In this gauge, the layers are parameterized as

$$W_\ell \sim \begin{pmatrix} I_r & 0 \\ 0 & Z_\ell \end{pmatrix}, \qquad \ell = 2, \ldots, L-1,$$

with residual blocks $Z_\ell \in \mathbb{R}^{(d_\ell - r)\times(d_{\ell-1}-r)}$. The first and last layers embed the signal path and attach the residuals:

$$W_L = [\,U_S \;\; U_Q Z_L\,], \qquad W_1 = \begin{pmatrix} U_S^\top M \\ Z_1 \end{pmatrix},$$

where U_Q spans the orthogonal complement of U_S. The signal blocks are chosen so that the product along the signal path equals $P_S M$ exactly, ensuring the rank condition by construction.

Cyclic and fiber constraints on the residuals. In this normal form, first-order criticality imposes polynomial constraints on the residual blocks $Z = (Z_1, \dots, Z_L)$:

- **Fiber (product) constraint:**

$$\mu(Z) := Z_L Z_{L-1} \cdots Z_1 = 0.$$

- **Cyclic constraints:**

$$d\mu_\ell(Z) := Z_{\ell-1} \cdots Z_1 \, \sigma \, Z_L \cdots Z_{\ell+1} = 0, \qquad \ell = 1, \dots, L,$$

  where σ is the fixed matrix appearing in the theoretical characterization (e.g. $\sigma = \Sigma_{XY} U_Q$ in the population setting, or an empirical analogue).

Energy formulation. To convert these matrix equalities into a scalar objective, we introduce the constraint energy

$$E(Z) \;=\; \frac{1}{2}\|\mu(Z)\|_F^2 \;+\; \frac{1}{2}\sum_{\ell=1}^{L} \|d\mu_\ell(Z)\|_F^2.$$

Then $E(Z) = 0$ if and only if all cyclic and fiber constraints are satisfied exactly.

Ensuring saddle points. A cheap sufficient construction for producing saddle points is to enforce two distinct residual blocks to be zero. Concretely, choose $a \neq b$ and set

$$Z_a = 0, \qquad Z_b = 0,$$

while sampling the remaining Z_ℓ. This automatically enforces the fiber constraint and typically satisfies all cyclic constraints.

Boltzmann sampling over constrained residuals. To sample many solutions rather than a single feasible point, we sample from a Boltzmann distribution concentrating near the constraint variety:

$$p_\beta(Z) \;\propto\; \exp\big(-\beta\, E(Z)\big)\, \pi(Z),$$

where $\beta > 0$ controls concentration and $\pi(Z)$ is a prior that prevents collapse to trivial solutions. Typical choices include Gaussian or Gaussian with weight decay

$$\log \pi(Z) = -\frac{\alpha}{2} \sum_{\ell=1}^{L} \|Z_\ell\|_F^2,$$

or norm-targeting penalties that keep $\|Z_\ell\|_F$ away from zero.

SGLD updates. Define the negative log-density (up to an additive constant)

$$\mathcal{L}_\beta(Z) := \beta\, E(Z) - \log \pi(Z).$$

Stochastic gradient Langevin dynamics (SGLD) then takes the form

$$Z_{t+1} = Z_t - \frac{\epsilon_t}{2} \nabla \mathcal{L}_\beta(Z_t) + \eta_t, \qquad \eta_t \sim \mathcal{N}(0, \epsilon_t I),$$

with stepsize schedule ϵ_t. When empirical analogues of the constraints are used, gradients can be estimated on minibatches.

Projection to true DLN critical points and Hessian certification. Comment: I'm not sure about the usefulness of doing this: Because the parameterization and the matrix σ may be approximate in finite-sample settings, after obtaining a low-energy configuration Z we assemble the full weights $W_1, \ldots, W_L$ and perform a short refinement step on the true DLN loss to reach $\|\nabla \mathcal{L}\| \approx 0$, while periodically re-balancing layer norms to control gauge drift. **Comment: this is going to be useful:** finally, saddle points are certified by estimating the smallest eigenvalues of the Hessian using Hessian-vector products; the presence of a negative eigenvalue indicates a strict saddle and a non strict saddle otherwise (or perhap§s local minimum in the underparametrize regime).