WW-PGD: WeightWatcher Projected Gradient Descent

Charles H. Martin

December 12, 2025

Abstract

We introduce WW-PGD (WeightWatcher Projected Gradient Descent), a spectral projection method designed as an add-on to standard gradient-based optimizers, with primary empirical validation using AdamW. WW-PGD applies structured projections in spectral space at batch or epoch boundaries using diagnostics computed by the WeightWatcher library. These projections explicitly drive neural network weight matrices toward the empirical renormalization group (ERG) fixed point identified in the Semi-Empirical Theory of (Deep) Learning (SETOL), characterized by a heavy-tailed spectral exponent $\alpha \approx 2$ and a vanishing trace–log condition over the spectral tail.

1 Background and Motivation

Empirical studies have shown that well-trained deep neural networks exhibit heavy-tailed empirical spectral distributions (ESDs) of their layer-wise weight correlation matrices [1, 2]. In such models, the tail of the eigenvalue distribution follows an approximate power law

$$\rho(\lambda) \sim \lambda^{-\alpha}, \tag{1}$$

with α typically in the range $(2, 4)$ and a strong empirical attraction toward $\alpha \approx 2$.

Heavy-Tailed Self-Regularization (HTSR) theory explains this behavior as an emergent consequence of stochastic optimization [1]. SETOL extends HTSR by identifying a universal empirical renormalization group (ERG) fixed point governing optimal generalization [4]. Related phenomena such as grokking and generalization collapse are explained within the same framework [3].

WW-PGD is designed to make this fixed point *explicitly reachable* during training by augmenting standard optimizers with principled spectral projections.

2 Weight Matrices and Spectral Normalization

Let $W \in \mathbb{R}^{N \times M}$ denote a layer weight matrix. Following standard practice, we define the layer-wise correlation matrix as

$$\boxed{X \;=\; \frac{1}{N} W^\top W,} \tag{2}$$

where $X \in \mathbb{R}^{M \times M}$.

This normalization ensures that X remains well-scaled as the layer width N changes, and corresponds to the usual mean-field normalization of correlation matrices.

Secondary normalization (WeightWatcher). In practice, WeightWatcher applies a second, lightweight normalization to X that rescales its eigenvalues so that their average magnitude is $\mathcal{O}(1)$. This removes an otherwise arbitrary global scale and makes spectral diagnostics (such as α, `num_pl_spikes`, and `detX_num`) directly comparable across layers of different sizes. This second normalization does *not* alter the shape of the spectrum, only its overall scale, and is essential for defining the ERG trace–log condition consistently. All spectral quantities used by WW-PGD are computed from the spectrum of this normalized matrix.

3 Critical Conditions from HTSR and SETOL

HTSR and SETOL identify two complementary critical conditions governing optimal generalization.

3.1 HTSR Critical Condition

$$\boxed{\alpha \approx 2} \tag{3}$$

This condition corresponds to scale invariance of the ESD tail and is observed empirically across architectures, datasets, and training regimes. Values $\alpha > 2$ correspond to progressively weaker correlations. By contrast, values $\alpha < 2$ are generally associated with excessive correlation and overfitting, although such regimes may be useful in specialized memorization-heavy settings; this remains under investigation.

3.2 SETOL (ERG) Critical Condition

Let $\{\lambda_i\}$ denote the normalized eigenvalues of X, ordered in descending magnitude, and let "tail" denote the heavy-tailed spectral subspace.

SETOL identifies the ERG fixed point via a vanishing trace–log condition:

$$\boxed{\sum_{i \in \text{tail}} \log \lambda_i = 0,} \tag{4}$$

or equivalently,

$$\det X_{\text{tail}} = 1. \tag{5}$$

This condition removes the remaining scale degree of freedom in the tail and defines a true ERG fixed point rather than a marginal flow.

4 WeightWatcher Diagnostics and the Midpoint Rule

At each projection step (performed at batch or epoch boundaries), WW-PGD runs WeightWatcher and extracts two complementary diagnostics for each layer:

- `num_pl_spikes`: the number of statistically significant power-law spikes in the ESD tail.
- `detX_num`: a trace–log based estimate of the effective tail size, i.e. the number of eigenvalues contributing to the ERG constraint.

These two quantities define an interval of uncertainty for the true tail size. WW-PGD resolves this uncertainty using a midpoint rule,

$$k_{\text{mid}} = \left\lfloor \frac{\texttt{num\_pl\_spikes} + \texttt{detX\_num}}{2} \right\rfloor, \tag{6}$$

and defines the spectral tail as the top k_{mid} eigenvalues.

Exactness at criticality. A key prediction of SETOL is that, as $\alpha \to 2$, the two diagnostics converge:

$$\boxed{\texttt{num\_pl\_spikes} = \texttt{detX\_num}.} \tag{7}$$

Consequently, the midpoint rule becomes exact at the ERG fixed point.

5 Projected Gradient Descent on the ERG Manifold

WW-PGD is a projected gradient descent (PGD) method [5], applied in *spectral space* rather than parameter space.

Conceptually, the ERG fixed point defines a *critical manifold* in the space of spectra: the set of weight matrices whose ESD tails satisfy

$$\alpha \approx 2, \qquad \sum_{i \in \text{tail}} \log \lambda_i = 0.$$

Standard optimizers do not explicitly enforce these conditions. WW-PGD augments them by projecting weights back toward this manifold during training.

5.1 Algorithmic Structure (high level)

Each training iteration consists of two stages:

1. **Base optimizer step.** A standard optimizer (e.g. AdamW) updates the weights using gradient descent:
$$W \leftarrow W - \eta \nabla_W \mathcal{L}.$$
2. **Spectral projection (WW-PGD).** At prescribed boundaries, the updated weights are projected onto the ERG constraint set by:
   - computing the SVD of W,
   - identifying the spectral tail via the midpoint rule,
   - reshaping the tail eigenvalues toward a power-law template with target exponent $\alpha \to 2$,
   - retracting to exactly satisfy the trace–log constraint,
   - reconstructing W and **blending** with the pre-projection weights.

This projection is implemented via a Cayley-style multiplicative update in log-eigenvalue space, which preserves positivity and allows controlled movement along the ERG manifold.

Homotopy behavior. Early in training, the projection is softened to avoid disrupting feature formation. As training progresses and α approaches 2, the projection is gradually hardened, yielding a homotopy toward the ERG fixed point rather than a hard constraint from initialization.

5.2 PGD as projection onto a spectral constraint set

Let $W^{(t)}$ denote the weights at the start of epoch t, and let the base optimizer (SGD/Adam/AdamW/Muon) produce an intermediate update

$$W^{(t+\frac{1}{2})} \;=\; W^{(t)} - \eta_t \nabla_W \mathcal{L}(W^{(t)}). \tag{8}$$

WW-PGD then applies a projection-like map

$$W^{(t+1)} \;=\; \Pi_{\mathcal{C}_t}\Big(W^{(t+\frac{1}{2})}\Big), \tag{9}$$

where the (estimated) constraint set $\mathcal{C}_t$ is defined implicitly by:

- a tail index set $\mathcal{T}_t$ (chosen by the midpoint rule from WeightWatcher),
- the HTSR criticality target $\alpha_t \downarrow 2$,
- the SETOL ERG condition $\sum_{i\in\mathcal{T}_t} \log \lambda_i = 0$.

Because $\mathcal{C}_t$ is defined through the spectrum of $X = (1/N)W^\top W$, the projection is most naturally implemented by modifying eigenvalues (or singular values) and reconstructing the matrix.

5.3 Spectral parameterization and tail extraction

Compute the SVD of the intermediate weights:

$$W^{(t+\frac{1}{2})} = U\Sigma V^\top, \qquad \Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_M), \tag{10}$$

and define the normalized correlation matrix

$$X = \frac{1}{N}\,(W^{(t+\frac{1}{2})})^\top W^{(t+\frac{1}{2})} = V\,\frac{1}{N}\Sigma^2\,V^\top. \tag{11}$$

Thus the eigenvalues of X are

$$\lambda_i = \frac{1}{N}\sigma_i^2. \tag{12}$$

WeightWatcher provides `num_pl_spikes` and `detX_num`, and we define the tail size

$$k_{\mathrm{mid}} = \left\lfloor \frac{\texttt{num\_pl\_spikes} + \texttt{detX\_num}}{2} \right\rfloor,$$

which induces a tail index set $\mathcal{T}_t = \{1, \ldots, k_{\mathrm{mid}}\}$ after ordering eigenvalues decreasingly.

5.4 Target template and ERG trace–log retraction

We define a rank-ordered power-law template on the tail:

$$\mu_i(q_t) \;=\; i^{-q_t}, \qquad i = 1, \ldots, k_{\mathrm{mid}}, \tag{13}$$

where $q_t = 1/(\alpha_t - 1)$ and $\alpha_t \downarrow 2$ (so $q_t \to 1$).

To enforce the SETOL ERG condition, we rescale the template by a factor A_t so that the *tail trace–log is zero*:

$$\sum_{i\in\mathcal{T}_t} \log\big(A_t \mu_i(q_t)\big) = 0 \quad\Longrightarrow\quad A_t = \exp\left(-\frac{1}{k_{\mathrm{mid}}}\sum_{i=1}^{k_{\mathrm{mid}}} \log \mu_i(q_t)\right). \tag{14}$$

We then set the target tail eigenvalues

$$\lambda_i^\star \;=\; A_t\,\mu_i(q_t), \qquad i \in \mathcal{T}_t. \tag{15}$$

5.5 Cayley-style update in log-spectrum space

Rather than a hard replacement $\lambda_i \leftarrow \lambda_i^\star$, we use a stable multiplicative (Cayley-style) update in log-space:

$$g_i \;=\; \log \lambda_i - \log \lambda_i^\star, \qquad \lambda_i^{\text{new}} \;=\; \lambda_i \, \frac{1 - \eta_c g_i}{1 + \eta_c g_i}, \qquad i \in \mathcal{T}_t, \tag{16}$$

with $\eta_c > 0$ a projection step-size, and with ratio clamping in implementation for numerical stability. This update has three useful properties: (i) it is scale-invariant in λ, (ii) it preserves positivity, and (iii) it becomes the identity map as $\eta_c \to 0$.

After the Cayley update, we apply a final retraction so that the ERG condition is satisfied exactly:

$$\lambda_i^{\text{proj}} \;=\; e^{\delta} \lambda_i^{\text{new}}, \qquad \delta \;=\; -\frac{1}{k_{\text{mid}}} \sum_{i \in \mathcal{T}_t} \log \lambda_i^{\text{new}}, \qquad i \in \mathcal{T}_t, \tag{17}$$

so that $\sum_{i \in \mathcal{T}_t} \log \lambda_i^{\text{proj}} = 0$.

Non-tail eigenvalues are left unchanged:

$$\lambda_i^{\text{proj}} = \lambda_i, \qquad i \notin \mathcal{T}_t.$$

5.6 Reconstruction and blend step

Convert eigenvalues back to singular values:

$$\sigma_i^{\text{proj}} = \sqrt{N \, \lambda_i^{\text{proj}}}. \tag{18}$$

Let $\Sigma^{\text{proj}} = \text{diag}(\sigma_i^{\text{proj}})$ and define the projected matrix

$$W_{\text{proj}}^{(t+\frac{1}{2})} = U \Sigma^{\text{proj}} V^\top. \tag{19}$$

Finally, WW-PGD applies a convex blend between the intermediate weights and the projected weights:

$$\boxed{W^{(t+1)} \;=\; (1 - \beta_t) \, W^{(t+\frac{1}{2})} + \beta_t \, W_{\text{proj}}^{(t+\frac{1}{2})}.} \tag{20}$$

Here $\beta_t \in [0, 1]$ controls the projection strength: $\beta_t = 1$ is a hard projection (full replacement), while $\beta_t \ll 1$ yields a soft projection.

This blend can be interpreted as a proximal step:

$$W^{(t+1)} = \arg\min_W \left\{ \frac{1}{2} \|W - W^{(t+\frac{1}{2})}\|_F^2 + \frac{\beta_t}{2(1 - \beta_t)} \|W - W_{\text{proj}}^{(t+\frac{1}{2})}\|_F^2 \right\}, \tag{21}$$

i.e. it trades off staying close to the base-optimizer iterate versus moving toward the ERG manifold estimate.

6 Practical Considerations

Because WW-PGD requires spectral decompositions and explicit projections, it is inherently more computationally expensive than pure gradient descent. If applied too aggressively or too early, it may slow convergence or interfere with early representation learning.

For this reason, practical implementations typically include:

- warmup epochs with no projection,
- reduced projection frequency (e.g. per epoch),
- softened projections early in training (small η_c and/or small β_t).

Performance optimizations and tighter integrations with base optimizers are active areas of ongoing work. User feedback and empirical results on a broader range of architectures and optimizers are strongly encouraged.

7 Conclusion

WW-PGD combines empirical spectral diagnostics from WeightWatcher with classical projected gradient descent to explicitly guide neural networks toward the HTSR and SETOL ERG fixed point. By enforcing $\alpha \approx 2$, a vanishing tail trace–log condition, and a dynamically exact midpoint rule, WW-PGD provides a principled mechanism for controlling spectral geometry during training.

References

[1] C. H. Martin and M. W. Mahoney. Implicit Self-Regularization in Deep Neural Networks. *Journal of Machine Learning Research*, 22(213):1–40, 2021. https://jmlr.org/papers/v22/20-410.html

[2] C. H. Martin, T. Peng, and M. W. Mahoney. Predicting trends in the quality of state-of-the-art neural networks. *Nature Communications*, 12:3892, 2021. https://www.nature.com/articles/s41467-021-24025-8

[3] H. K. Prakash and C. H. Martin. Grokking and Generalization Collapse: Insights from HTSR Theory. *arXiv:2506.04434*, 2025. https://arxiv.org/abs/2506.04434

[4] C. H. Martin and C. Hinrichs. SETOL: Semi-Empirical Theory of (Deep) Learning. *arXiv:2507.17912*, 2025. https://arxiv.org/abs/2507.17912

[5] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[6] WeightWatcher. https://weightwatcher.ai/