

---

# I Dropped a Neural Net

---

Hyunwoo Park  
Carnegie Mellon University  
hp2@andrew.cmu.edu

## Abstract

A recent Dwarkesh Patel podcast with John Collison and Elon Musk featured an interesting puzzle from Jane Street: *they trained a neural net, shuffled all 96 layers, and asked to put them back in order.*

Given unlabelled layers of a Residual Network and its training dataset, we recover the exact ordering of the layers. The problem decomposes into pairing each block’s input and output projections (48! possibilities) and ordering the reassembled blocks (48! possibilities), for a combined search space of  $(48!)^2 \approx 10^{122}$ , which is more than the atoms in the observable universe. We show that stability conditions during training like dynamic isometry leave the product  $W_{\text{out}} W_{\text{in}}$  for correctly paired layers with a negative diagonal structure, allowing us to use *diagonal dominance ratio* as a signal for pairing. For ordering, we seed-initialize with a rough proxy such as delta-norm or  $\|W_{\text{out}}\|_F$  then hill-climb to zero mean squared error.

## 1 Introduction

Consider the following combinatorial puzzle: a 48-block ResNet has been “dropped,” scattering its 97 constituent linear layers (96 block layers plus one output projection) into an unlabelled collection. Given only these and 10,000 training datapoints, can we recover the original network?

Deep Residual Networks He et al. [2016] compute a sequence of near-identity transformations, where each block adds a small perturbation to its input via a residual branch. Veit et al. Veit et al. [2016] showed that this architecture behaves like an ensemble of relatively shallow networks, implying that individual blocks can be reordered with controlled impact on the output. A complementary perspective comes from dynamic isometry Saxe et al. [2014], Pennington et al. [2017], Xiao et al. [2018]: well-trained residual blocks have stable gradients, which constrains the structure of their weight products. We exploit these properties to *reconstruct* a dropped ResNet.

The problem decomposes into *pairing* (matching each input projection to its output projection) and *ordering* (sequencing the 48 reassembled blocks), yielding a combined search space of  $(48!)^2 \approx 10^{122}$ , more than the atoms in the observable universe, which makes brute force impossible.

1. **Pairing** (Section 3): Match each  $W_{\text{in}}$  to its  $W_{\text{out}}$  via a diagonal dominance ratio on  $W_{\text{out}} W_{\text{in}}$ .
2. **Ordering** (Section 4): Recover the block sequence via a seed initialization, refined by Bradley–Terry ranking, and finalized by MSE bubble repair.

## 2 The Puzzle

*Oh no! I dropped an extremely valuable trading model and it fell apart into linear layers! I need to rebuild it before anyone notices, but I can't remember how these pieces go together, or how it was trained. All I have left are the pieces of the model and some historical data. Can you help me figure out how to put it back together? Luckily I still have the source code of the layers that the neural network is made of. They look like this:*

```
class Block(nn.Module):
    def __init__(self, in_dim: int, hidden_dim: int):
        super().__init__()
        self.inp = nn.Linear(in_dim, hidden_dim)
        self.activation = nn.ReLU()
        self.out = nn.Linear(hidden_dim, in_dim)

    def forward(self, x):
        residual = x
        x = self.inp(x)
        x = self.activation(x)
        x = self.out(x)
        return residual + x

class LastLayer(nn.Module):
    def __init__(self, in_dim: int, out_dim: int):
        super().__init__()
        self.layer = nn.Linear(in_dim, out_dim)

    def forward(self, x):
        return self.layer(x)
```

The solution is a permutation of indices  $\{0, \dots, 96\}$ . We are also given :

```
historical_data_and_pieces
└── historical_data.csv
└── pieces
    ├── piece_0.pth
    ├── piece_1.pth
    :
    └── piece_96.pth
```

The CSV file contains 10,000 rows with 48 input features (`measurement_0` through `measurement_47`), the original model's predictions (`pred`), and the ground-truth target values (`true`).

## 2.1 Architecture

The network consists of 48 residual blocks followed by a final linear layer (Figure 1). Each block computes:

$$\text{Block}_k(x) = x + r_k(x), \quad r_k(x) := W_{\text{out}}^{(k)} \text{ReLU}(W_{\text{in}}^{(k)}x + b_{\text{in}}^{(k)}) + b_{\text{out}}^{(k)} \quad (1)$$

where  $W_{\text{in}}^{(k)} \in \mathbb{R}^{96 \times 48}$ ,  $W_{\text{out}}^{(k)} \in \mathbb{R}^{48 \times 96}$ , and the `in_dim=48` with `hidden_dim=96`. The full forward pass is:

$$f_\sigma(x) = W_{\text{last}} \text{Block}_{\sigma(48)} \circ \dots \circ \text{Block}_{\sigma(1)}(x) + b_{\text{last}} \quad (2)$$

where  $\sigma \in S_{48}$  is the block permutation and  $W_{\text{last}} \in \mathbb{R}^{1 \times 48}$  is the scalar output projection (`piece_85`, identified by its unique shape).

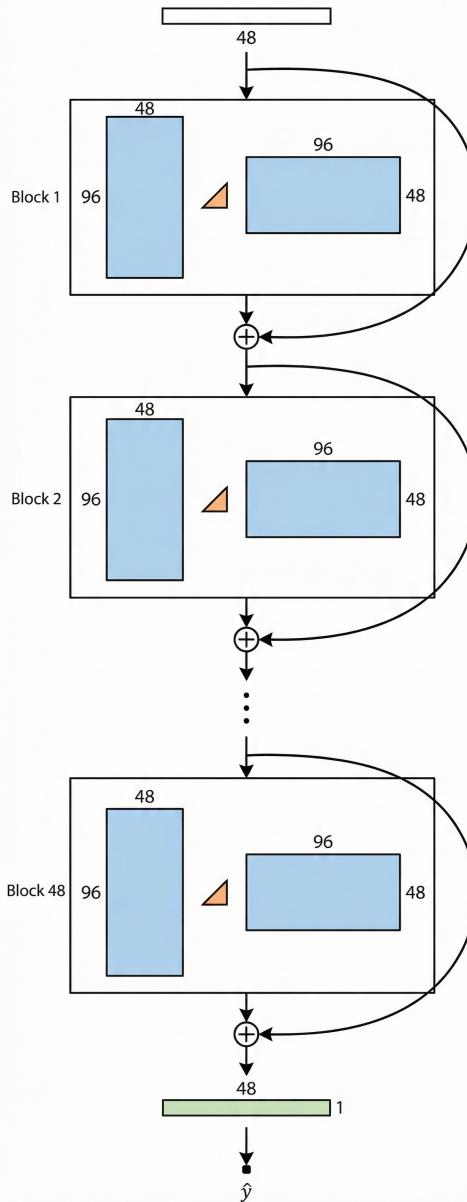


Figure 1: **Architecture.** The 48-block ResNet. Each block applies input layer, ReLU, output layer, then a residual connection. The final layer produces the scalar output  $\hat{y}$ .

## 2.2 Available Data

We are given:

- 97 weight matrices and bias vectors. The shapes are: 48 of  $(96, 48)$  (input projections), 48 of  $(48, 96)$  (output projections), and 1 of  $(1, 48)$  (last layer).
- $N = 10,000$  data points  $(x_i, \hat{y}_i, y_i)$  where  $x_i \in \mathbb{R}^{48}$  are input features,  $\hat{y}_i \in \mathbb{R}$  are the original model's predictions, and  $y_i \in \mathbb{R}$  are the ground-truth targets.

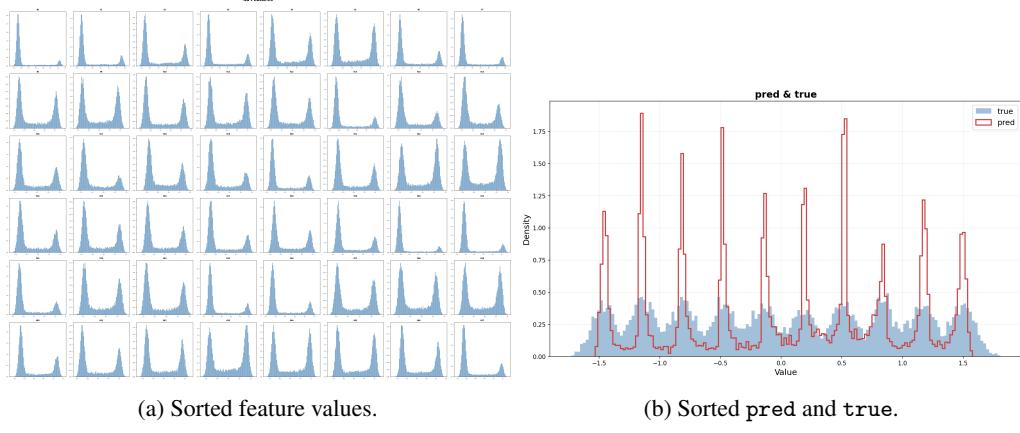


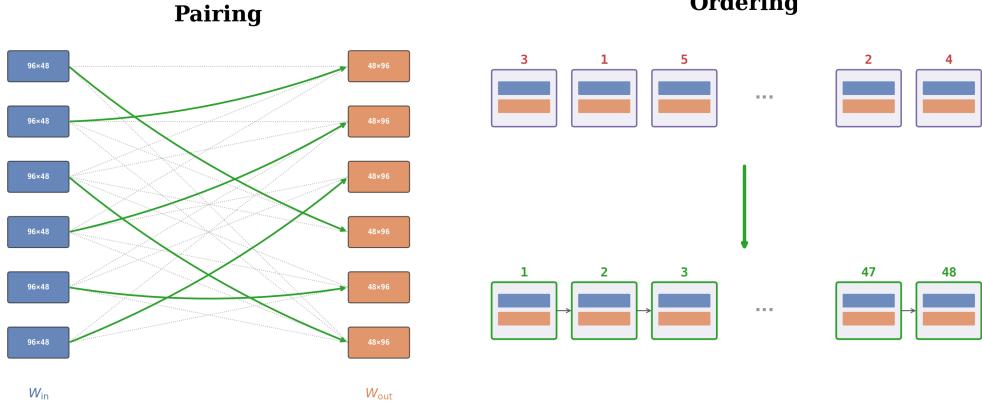
Figure 2: **Dataset overview.** Features and outputs are normalized.

## 2.3 Objective

Given the dataset  $\{(x_i, \hat{y}_i)\}_{i=1}^N$ , we define the mean-squared error against `pred`

$$\mathcal{L}(\sigma) := \frac{1}{N} \sum_{i=1}^N (f_\sigma(x_i) - \hat{y}_i)^2. \quad (3)$$

The correct permutation should have  $\mathcal{L}(\sigma) = 0$  up to floating-point error.



1. **Pairing:** Pair each  $W_{\text{in}} \in \mathbb{R}^{96 \times 48}$  to its  $W_{\text{out}} \in \mathbb{R}^{48 \times 96}$ . Bipartite matching with  $48!$  possibilities.
2. **Ordering:** Order the 48 paired blocks in the correct sequence. Permutation over another  $48!$  possibilities.

The combined search space of  $(48!)^2 \approx 10^{122}$  is more than the atoms in the observable universe, making brute force search impossible.

### 3 Pairing

We train a small model with the same architecture on the same data. Upon examining the product  $W_{\text{out}}^{(k)} W_{\text{in}}^{(k)} \in \mathbb{R}^{48 \times 48}$  we observe a negative diagonal structure. (Proposition 1). Which motivates the metric:

#### 3.1 Diagonal Dominance Ratio

For each candidate pair of input layer  $i$  and output layer  $j$ , we form the product  $W_{\text{out}}^{(j)} W_{\text{in}}^{(i)}$  and compute the **diagonal dominance ratio**:

$$d(i, j) = \frac{|\text{tr}(W_{\text{out}}^{(j)} W_{\text{in}}^{(i)})|}{\|W_{\text{out}}^{(j)} W_{\text{in}}^{(i)}\|_F} \quad (4)$$

For correctly paired blocks, the negative diagonal structure (Proposition 1) contributes to the trace magnitude, giving  $d \in [1.76, 3.28]$ . For incorrect pairings, the product has no structure, giving  $d \in [0.00, 0.58]$ .

We compute  $d(i, j)$  for all  $48 \times 48$  candidate pairs and solve the optimal bipartite assignment via the Hungarian algorithm Kuhn [1955] in  $O(n^3)$ , recovering all 48 pairs perfectly. In practice row-wise greedy argmax is enough; the Hungarian algorithm simply provides the optimality guarantee.

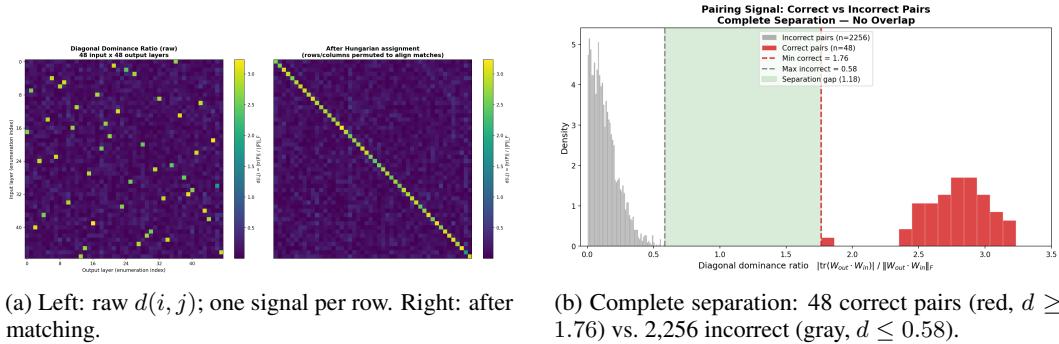


Figure 3: **Pairing via diagonal dominance.** The diagonal dominance ratio achieves complete separation between correct pairs ( $d \in [1.76, 3.28]$ ) and incorrect pairs ( $d \in [0.00, 0.58]$ ), with a gap of 1.18.

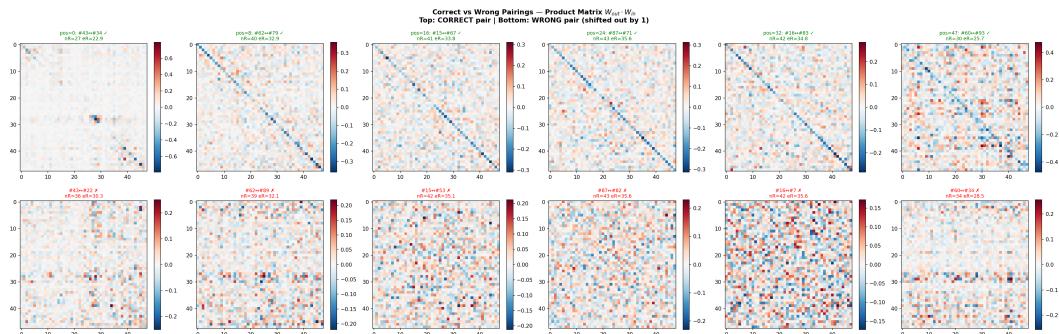


Figure 4: **Correct vs. incorrect pairings.** Top: Correctly paired matrices show negative diagonal structure. Bottom: Incorrect pairings with no structure.

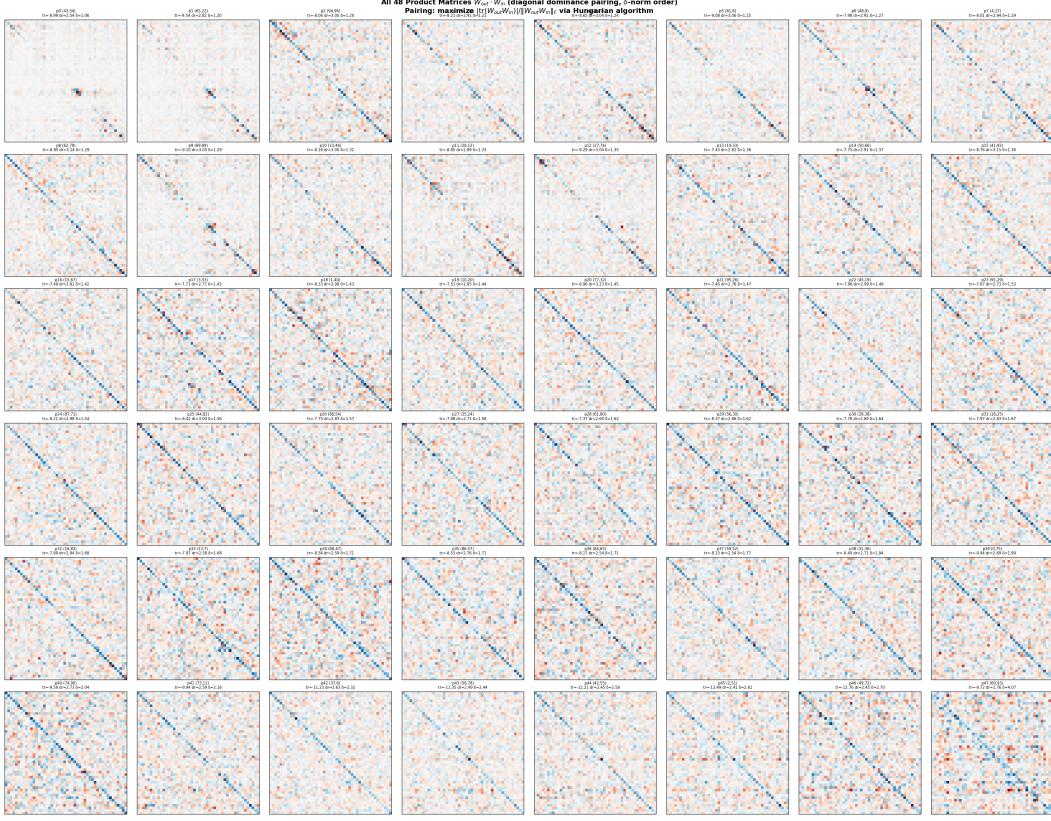


Figure 5: **All 48 product matrices from the original model**, ordered by delta-norm  $\delta_k$  (Section 4.1). Every block exhibits a negative diagonal structure. Earlier blocks (top-left) and later blocks (bottom-right). Traces range from  $-13.5$  to  $-7.4$ . Compare with Figure 7.

## 4 Ordering

With all 48 correctly paired blocks, we now need to recover the correct order. We start by seed ordering with a rough proxy that achieves low MSE.

### 4.1 Seed Ordering

**Delta-norm (data-dependent).** We compute each block’s **delta-norm**, the mean magnitude of its residual contribution when applied to the raw input:

$$\delta_k := \mathbb{E}_{x \sim \mathcal{D}} [\|r_k(x)\|_2] = \mathbb{E}_{x \sim \mathcal{D}} \left[ \left\| W_{\text{out}}^{(k)} \text{ReLU}(W_{\text{in}}^{(k)} x + b_{\text{in}}^{(k)}) + b_{\text{out}}^{(k)} \right\|_2 \right] \quad (5)$$

Sorting by ascending  $\delta_k$  places blocks with smaller perturbations first, motivated by the empirical observation that residual perturbation magnitude increases with depth in trained networks, inspired by the rank-diminishing principle Feng et al. [2022]. This achieves  $\text{MSE} = 0.036794$ .

**$\|W_{\text{out}}\|_F$  (data-free).** We find that sorting blocks by the Frobenius norm of their output projection  $\|W_{\text{out}}^{(k)}\|_F$  is effective and data-free. Sorting by ascending  $\|W_{\text{out}}\|_F$  achieves  $\text{MSE} = 0.075851$ .

### 4.2 Near-Commutativity of Residual Blocks

We must examine the noise in ordering before we proceed with sorting. Because each block computes  $x \mapsto x + r_k(x)$  where  $r_k$  is small relative to  $x$ , the composition of two blocks is approximately commutative:

$$\text{Block}_A \circ \text{Block}_B(x) \approx \text{Block}_B \circ \text{Block}_A(x) + O(\|r_A\| \|r_B\|) \quad (6)$$

Therefore pairwise ordering preferences are only approximately **transitive**: if  $A \prec B$  and  $B \prec C$ , then  $A \prec C$  with high probability since the residual perturbations are small relative to the identity path. We can't rely on sorting algorithm alone, therefore we look to methods that are robust to noisy ordering.

### 4.3 Bradley–Terry Ranking

For all  $\binom{48}{2} = 1,128$  pairs  $(A, B)$ , we swap their positions in the seed ordering and compute the change in MSE:

$$g_{AB} = \mathcal{L}(\sigma_{\text{swap}(A,B)}) - \mathcal{L}(\sigma_{\text{seed}}) \quad (7)$$

If  $g_{AB} > 0$ , then  $A$  should precede  $B$ ; if  $g_{AB} < 0$ , the reverse. Each comparison is done on one forward pass with just  $N_{\text{cmp}} = 2,000$  points for efficiency.

Rather than counting wins, we fit a margin-weighted Bradley–Terry model Bradley and Terry [1952] to extract strength parameters. Each pairwise MSE gap  $g_{AB}$  is converted to a soft win probability via a sigmoid with temperature  $T$ :

$$p_{A \prec B} = \frac{1}{1 + \exp(-g_{AB}/T)} \quad (8)$$

The latent strengths  $\{s_k\}$  are estimated by maximizing the Bradley–Terry likelihood via the MM algorithm of Hunter [2004]:

$$s_i^{(t+1)} = \frac{\sum_{j \neq i} w_{ij}}{\sum_{j \neq i} \frac{w_{ij} + w_{ji}}{s_i^{(t)} + s_j^{(t)}}} \quad (9)$$

where  $w_{ij} = p_{i \prec j}$ , converging in  $\sim 280$  iterations ( $T = 0.001$ ).

**Transitivity violations.** Out of  $\binom{48}{3} = 17,296$  directed triples, only **66** (0.38%) exhibit a transitivity violation, forming 22 unique 3-cycles. These violations are concentrated among a small number of blocks, spatially local (spanning  $< 15$  ground-truth positions), and weak (median margin  $1.8 \times 10^{-4}$  vs. overall median 0.041). Sorting by BT strength yields MSE = 0.00299, a  $12\times$  improvement over the delta-norm seed.

### 4.4 Bubble Repair

We finally perform local correction by greedy adjacent-swap hill climbing on MSE: sweep through all 47 adjacent pairs, swap if it reduces MSE, undo otherwise, and repeat until a full sweep produces zero swaps.

Table 1: **Bubble repair convergence from BT ordering.** Converges to MSE = 0 in 5 rounds with 37 swaps.

Round	Swaps	MSE	Cum. swaps
0 (BT init)	–	0.002989	0
1	21	0.001025	21
2	7	0.000553	28
3	5	0.000200	33
4	3	0.000052	36
5	1	<b>0.000000</b>	37

## 4.5 Visual Progression

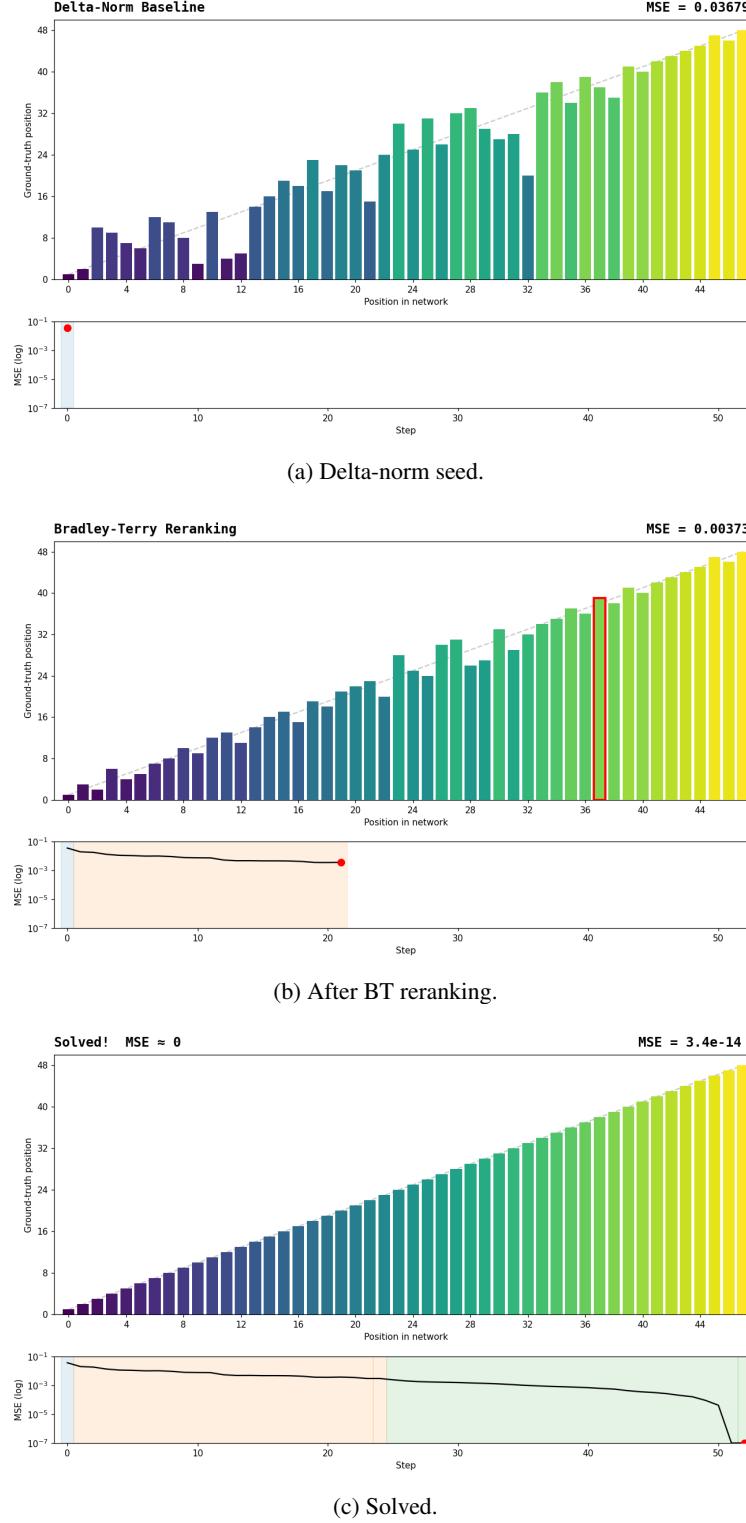


Figure 6: **Ordering pipeline progression.** Bar height = ground-truth position (1–48)

## 5 Analysis

### 5.1 Bradley–Terry Is Not a Necessity

In practice, we find that transitivity violations are negligible and, due to strong initializations, hill-climbing alone could recover the solution. **Direct MSE hill-climbing from the delta-norm seed converges to  $\text{MSE} = 0$  on its own**, without pairwise comparisons or Bradley–Terry (BT) ranking. Starting from  $\text{MSE} = 0.036794$ , bubble repair converges in 13 rounds with 122 total swaps (Table 2).

Table 2: **Hill-climb from delta-norm seed (no BT)**. Converges to exact solution in 13 rounds / 122 swaps.

Round	Swaps	MSE	Cum. swaps
0 (seed)	–	0.036794	0
1	23	0.021690	23
2	20	0.012919	43
3	21	0.008559	64
4	15	0.005873	79
5	11	0.004097	90
6	9	0.002745	99
7	12	0.001012	111
8	5	0.000380	116
9	2	0.000247	118
10	1	0.000182	119
11	1	0.000118	120
12	1	0.000052	121
13	1	<b>0.000000</b>	122

### 5.2 Hill-climb from $\|W_{\text{out}}\|_F$ seed has better convergence

Starting from a *data-free* seed ordering based on  $\|W_{\text{out}}\|_F$ , bubble repair reaches the solution in 6 rounds with 72 swaps (Table 3). Despite the higher initial MSE (0.075851 vs. 0.036794), convergence is  $2\times$  faster in rounds and requires  $1.7\times$  fewer swaps.

Table 3: **Hill-climb from  $\|W_{\text{out}}\|_F$  seed (data-free, no BT)**. Starting from  $\text{MSE} = 0.075851$ , converges in 6 rounds / 72 swaps. Despite higher initial MSE, convergence is  $2\times$  faster in rounds and  $1.7\times$  fewer swaps, and is *data-free*.

Round	Swaps	MSE	Cum. swaps
0 (seed)	–	0.075851	0
1	30	0.005936	30
2	16	0.003397	46
3	12	0.001405	58
4	9	0.000568	67
5	4	0.000223	71
6	1	<b>0.000000</b>	72

The BT step mainly improves the *starting point*: from a BT ordering, bubble repair converges in 5 rounds (Table 1) rather than 13. However, this comes at the cost of 1,128 pairwise swap evaluations (forward passes). Overall, direct hill-climbing and BT+repair have comparable end-to-end cost in this setting.

## 6 Discussion

### 6.1 What is this data?

I have spent some time trying to figure out what this data could be and how these features were processed. This turned out to be very difficult, if not impossible. I am very curious to find out (see Figure 2a). It might just be synthetic.

### 6.2 Unreasonable effectiveness of delta norm

Why does the delta norm work so well as a seed initialization? Are there better seed orderings? what are their theoretical underpinnings?

### 6.3 Convergence

Why does  $\|W_{\text{out}}\|_F$  seed has better convergence despite being a worse initialization compared to the delta norm? Is the geometry of the loss landscape nice? Are there local minima where hill-climbing would get stuck?

## References

- Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.
- Ruili Feng, Kecheng Zheng, Yukun Huang, Deli Zhao, Michael Jordan, and Zheng-Jun Zha. Rank diminishing in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 35, pages 33054–33065, 2022.
- 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*, pages 770–778, 2016.
- David R Hunter. MM algorithms for generalized Bradley–Terry models. *The Annals of Statistics*, 32(1):384–406, 2004.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.
- Harold W Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.
- Jeffrey Pennington, Samuel S Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. In *Advances in Neural Information Processing Systems*, volume 30, 2017.
- Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear networks. *International Conference on Learning Representations*, 2014.
- Andreas Veit, Michael J Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. In *Advances in Neural Information Processing Systems*, volume 29, 2016.
- Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel S Schoenholz, and Jeffrey Pennington. Dynamical isometry and a mean field theory of CNNs: How to train 10,000-layer vanilla convolutional neural networks. In *International Conference on Machine Learning*, pages 5393–5402. PMLR, 2018.

## Appendix

### A Engineer's Induction

We train our own 48-block network with identical architecture on the same data (Adam Kingma and Ba [2015],  $1r = 1e-4$ , 200 epochs,  $MSE \approx 0.0006$ ). We observe a strong negative diagonal structure. (Figure 7), matching the original model (Figure 5).

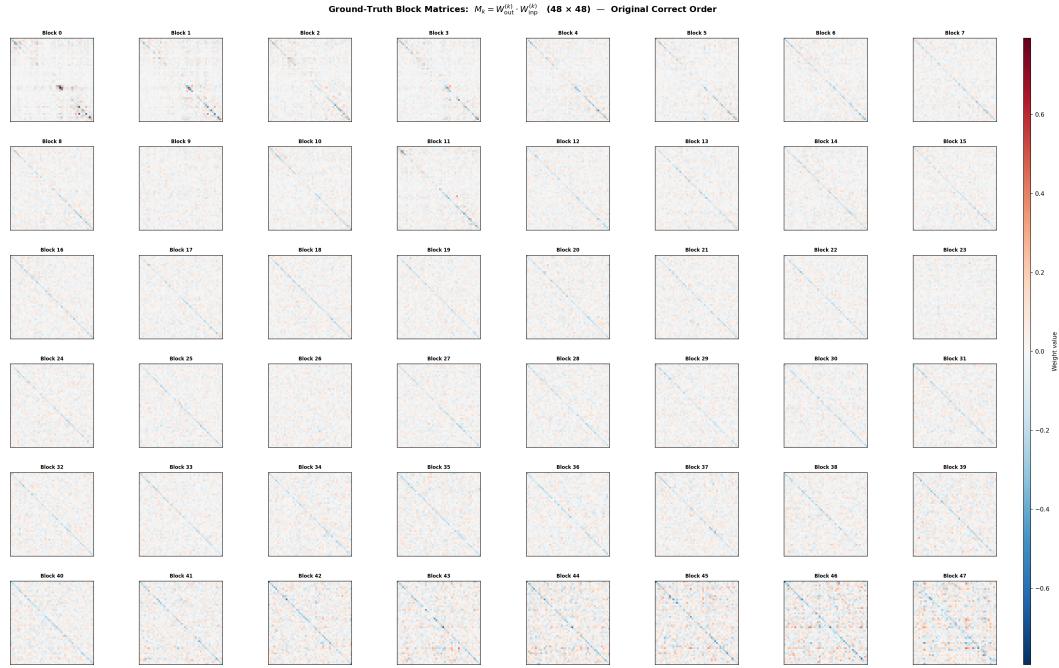


Figure 7: **Product matrices from a freshly trained model.** All 48 matrices  $W_{\text{out}}^{(k)} W_{\text{in}}^{(k)}$ , sorted by Frobenius norm.

### B Why Negative Diagonal Structure?

We derive the negative diagonal structure of  $W_{\text{out}}^{(k)} W_{\text{in}}^{(k)}$  from dynamic isometry Saxe et al. [2014], Pennington et al. [2017], Xiao et al. [2018], then show why it provides a signal for correct pairings.

#### B.1 Setup

Recall each residual block (Eq. 1):  $f(x) = x + r(x)$  with  $r(x) = W_{\text{out}} \text{ReLU}(W_{\text{in}} x + b_{\text{in}}) + b_{\text{out}}$ , where  $W_{\text{in}} \in \mathbb{R}^{h \times d}$ ,  $W_{\text{out}} \in \mathbb{R}^{d \times h}$  ( $d=48$ ,  $h=96$ ).

Let  $D(x) := \text{diag}(\mathbf{1}[W_{\text{in}} x + b_{\text{in}} > 0]) \in \{0, 1\}^{h \times h}$  be the ReLU gating matrix. The Jacobians are

$$J_r(x) = W_{\text{out}} D(x) W_{\text{in}}, \quad J_f(x) = I_d + J_r(x). \quad (10)$$

#### B.2 Main Result

**Proposition 1** (Negative diagonal structure). *If the block satisfies dynamic isometry Pennington et al. [2017] in expectation,  $\mathbb{E}_x[J_f(x)^\top J_f(x)] = I_d$ , with non-trivial residual, and the ReLU activation probability is  $\frac{1}{2}$  per unit (i.e.  $\mathbb{E}_x[D(x)] = \frac{1}{2} I_h$ ), then  $\text{tr}(W_{\text{out}} W_{\text{in}}) < 0$ .*

*Proof.* **Step 1 (Isometry constraint).** Expand  $\mathbb{E}_x[J_f^\top J_f] = I_d$ . Since  $J_f = I + J_r$ :

$$\mathbb{E}_x[(I + J_r)^\top (I + J_r)] = I \implies \mathbb{E}_x[J_r + J_r^\top + J_r^\top J_r] = 0. \quad (11)$$

**Step 2 (Trace).** Take the trace. Using  $\text{tr}(J_r^\top) = \text{tr}(J_r)$  and  $\text{tr}(J_r^\top J_r) = \|J_r\|_F^2$ :

$$2\mathbb{E}_x[\text{tr}(J_r)] + \mathbb{E}_x[\|J_r\|_F^2] = 0 \implies \mathbb{E}_x[\text{tr}(J_r)] = -\frac{1}{2}\mathbb{E}_x[\|J_r\|_F^2]. \quad (12)$$

**Step 3 (Evaluate the left-hand side).** Since  $W_{\text{out}}$  and  $W_{\text{in}}$  are fixed and only  $D(x)$  depends on  $x$ :

$$\mathbb{E}_x[\text{tr}(J_r(x))] = \text{tr}(W_{\text{out}} \mathbb{E}_x[D(x)] W_{\text{in}}). \quad (13)$$

Each diagonal entry of  $D(x)$  is either zero or one. By the activation-probability assumption ( $\mathbb{E}[D(x)] = \frac{1}{2}I_h$ ; verified empirically in Figure 8 and Figure 9):

$$\mathbb{E}_x[\text{tr}(J_r(x))] = \frac{1}{2} \text{tr}(W_{\text{out}} W_{\text{in}}). \quad (14)$$

**Step 4 (Combine).** Substituting into Eq. (12):

$$\frac{1}{2} \text{tr}(W_{\text{out}} W_{\text{in}}) = -\frac{1}{2}\mathbb{E}_x[\|J_r(x)\|_F^2] \implies \boxed{\text{tr}(W_{\text{out}} W_{\text{in}}) = -\mathbb{E}_x[\|J_r(x)\|_F^2] < 0.} \quad (15)$$

Strict inequality holds because  $J_r(x) \neq 0$  (the residual is non-trivial), so  $\mathbb{E}_x[\|J_r(x)\|_F^2] > 0$ .  $\square$

Empirically,  $\text{tr}(W_{\text{out}}^{(k)} W_{\text{in}}^{(k)}) \in [-13.5, -7.4]$  across all 48 blocks (Figure 5).

### B.3 Why Pairing Works

**Correct pair:  $-\epsilon I$  structure.** Proposition 1 guarantees  $\text{tr}(M_{ii}) < 0$ , so we can decompose a correctly paired product as

$$M_{ii} := W_{\text{out}}^{(i)} W_{\text{in}}^{(i)} = -\epsilon I_d + E, \quad \epsilon := \frac{1}{d} |\text{tr}(M_{ii})| > 0, \quad (16)$$

where  $E := M_{ii} + \epsilon I_d$  captures off-diagonal structure and diagonal variation. By construction  $\text{tr}(E) = 0$ , so  $\text{tr}(M_{ii}) = -\epsilon d$ . For the Frobenius norm, expand using  $(M_{ii})_{lm} = -\epsilon (I_d)_{lm} + E_{lm}$ :

$$\|M_{ii}\|_F^2 = \sum_{l,m} (-\epsilon (I_d)_{lm} + E_{lm})^2 = \underbrace{\epsilon^2 \sum_{l,m} (I_d)_{lm}^2}_{=d} - 2\epsilon \underbrace{\sum_l E_{ll}}_{=\text{tr}(E)=0} + \|E\|_F^2 = \epsilon^2 d + \|E\|_F^2. \quad (17)$$

The diagonal dominance ratio for a correct pair is therefore

$$d(i, i) = \frac{|\text{tr}(M_{ii})|}{\|M_{ii}\|_F} = \frac{\epsilon d}{\sqrt{\epsilon^2 d + \|E\|_F^2}} = \frac{\sqrt{d}}{\sqrt{1 + \|E\|_F^2 / (\epsilon^2 d)}} \leq \sqrt{d}. \quad (18)$$

The upper bound  $\sqrt{d} \approx 6.93$  is attained only for scaled identity. Empirically the correct-pair ratios reach  $d(i, i) \in [1.76, 3.28]$ , reflecting non-negligible off-diagonal energy  $\|E\|_F$ .

**Incorrect pair: random matrix baseline.** For an incorrect pairing  $(i, j)$  with  $i \neq j$ , the matrices  $W_{\text{out}}^{(j)}$  and  $W_{\text{in}}^{(i)}$  come from independently trained blocks. Model their entries as independent zero-mean random variables with variances  $\sigma_{\text{out}}^2$  and  $\sigma_{\text{in}}^2$ . Each entry of  $M_{ij} = W_{\text{out}}^{(j)} W_{\text{in}}^{(i)}$  is a sum of  $h$  independent zero-mean products:

$$(M_{ij})_{lm} = \sum_{k=1}^h (W_{\text{out}}^{(j)})_{lk} (W_{\text{in}}^{(i)})_{km}, \quad \text{Var}((M_{ij})_{lm}) = h \sigma_{\text{out}}^2 \sigma_{\text{in}}^2. \quad (19)$$

The trace sums  $d$  diagonal entries. Entry  $(M_{ij})_{ll}$  depends on row  $l$  of  $W_{\text{out}}^{(j)}$  and column  $l$  of  $W_{\text{in}}^{(i)}$ ; different values of  $l$  use disjoint rows and columns, so the diagonal entries are mutually independent. Therefore

$$\mathbb{E}[\text{tr}(M_{ij})] = 0, \quad \text{sd}(\text{tr}(M_{ij})) = \sigma_{\text{out}} \sigma_{\text{in}} \sqrt{dh}. \quad (20)$$

The expected squared Frobenius norm sums all  $d^2$  entries:  $\mathbb{E}[\|M_{ij}\|_F^2] = d^2 h \sigma_{\text{out}}^2 \sigma_{\text{in}}^2$ . The typical diagonal dominance ratio scales as

$$\frac{\text{sd}(\text{tr}(M_{ij}))}{\sqrt{\mathbb{E}[\|M_{ij}\|_F^2]}} = \frac{\sigma_{\text{out}} \sigma_{\text{in}} \sqrt{dh}}{\sigma_{\text{out}} \sigma_{\text{in}} d \sqrt{h}} = \frac{1}{\sqrt{d}} \approx 0.14. \quad (21)$$

**Separation.** The correct-pair score (Eq. 18) is around  $\sqrt{d} \approx 6.93$ , while the incorrect-pair score (Eq. 21) is  $\sim 1/\sqrt{d} \approx 0.14$ . Empirically:  $d(i, i) \geq 1.76$  vs.  $d(i, j) \leq 0.58$  (Figure 3b).

## C Additional Figures

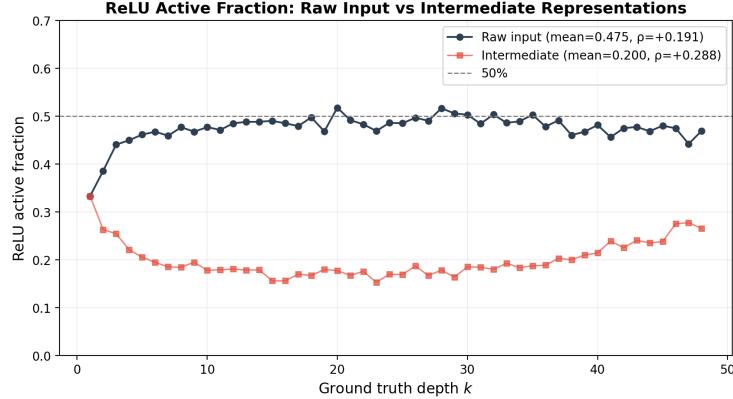


Figure 8: **ReLU activation fraction across blocks.** Per-block ReLU activation probability

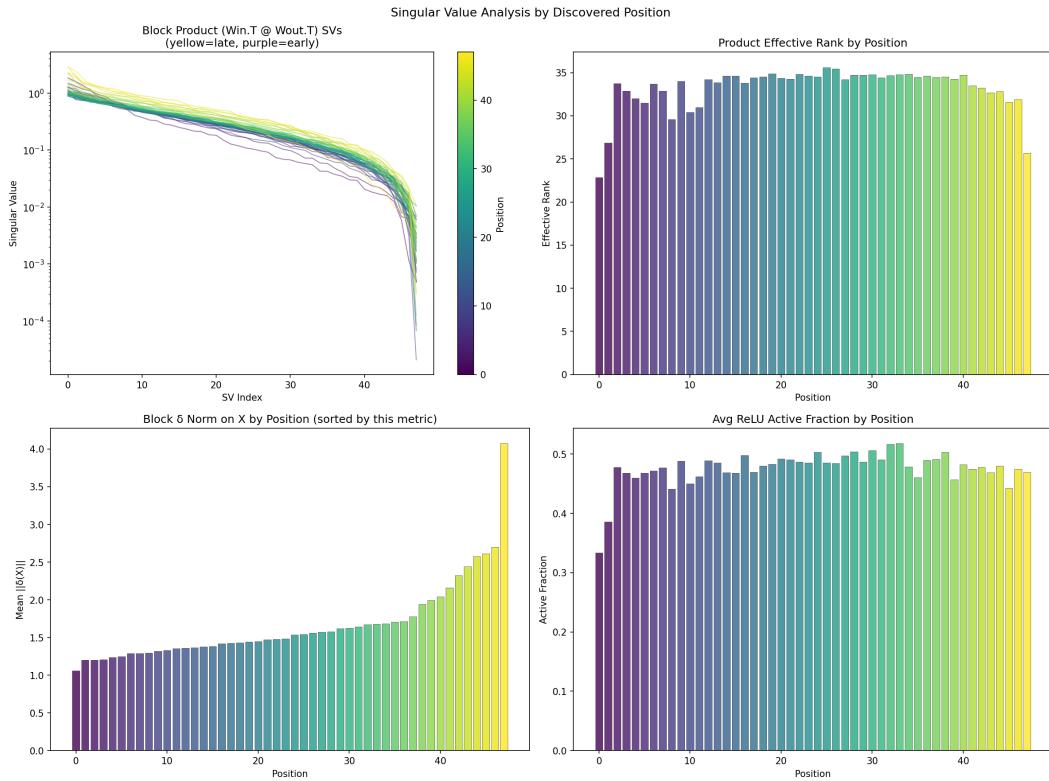


Figure 9: **Block properties by recovered position.** Top-left: singular value spectra of  $W_{\text{out}} W_{\text{in}}$ , colored by position (purple = early, yellow = late). Top-right: effective rank vs. position. Bottom-left: delta-norm increases with depth. Bottom-right: ReLU active fraction