DOES THE MASK KNOW WHERE TO CUT? SEPARATING TOPOLOGY FROM WEIGHT PLACEMENT IN REWIRING NULLS

Anonymous authors
Paper under double-blind review

ABSTRACT

Rewiring null models are widely used to test whether the pattern of connections in a weighted network matters. Yet rewiring changes both the connections and the weights assigned to them, so the resulting score combines topology with weight placement. We show that this score decomposes into wiring and placement terms, provide an exact finite-sample test for placement, and separate placement variance from rewiring variance. The construction applies the null's weight rule to the original topology as well as to rewired topologies, allowing wiring to be compared under the same treatment of weights. In neural-network pruning, placement is significant in all 36 high-sparsity iterative-magnitude-pruning comparisons across 3 initializations. In connectome reservoirs, it accounts for up to 99.8% of the null variance; removing it from the score difference reduces the statistic's magnitude, whereas removing its contribution to the variance increases it. A conventional rewiring score is therefore not a biased topology score but a different quantity that combines wiring and weight placement.

1 INTRODUCTION

A recurring question about a network is whether its function depends on which nodes are connected, or only on coarser properties such as how many connections each node has. The standard way to answer it is a rewiring null model (Rubinov & Sporns, 2011; Váša & Mišić, 2022). The connections are rewired at random while the chosen properties, such as the degree sequence, are preserved, and the observed network's score is reported as a z-score against the scores of the rewired networks, where a large z-score is interpreted as evidence that the wiring itself matters. The same design is used in connectome reservoir computing, where a brain network serves as the recurrent weights of a reservoir (Suárez et al., 2021), and in pruning, where the layerwise rearranging ablation of Frankle et al. (2021) permutes a sparse mask within each layer, retrains the network, and compares its accuracy with that of the original mask.

In a weighted network, rewiring a connection also changes its weight. A permuted mask retains a different subset of the initial weights, and a rewired connectome edge is assigned a different weight. The reported statistic therefore compares both the wiring and the placement of weights on it with their null counterparts.

We separate the two effects by also scoring the observed topology with its weights placed by the null's own weight rule. Compared with these matched draws, the reported statistic splits exactly into a wiring term, which compares the observed topology with rewired ones when both carry weights placed by the same rule, and a placement term, which compares the observed placement with placements drawn from that rule. We give an exact test for the placement term. The spread of the null also mixes variation due to rewiring with variation due to placement, so we split the null variance as well and bound the statistic that remains once placement is removed from both. Figure 1 shows both corrections on pruning masks and on connectome reservoirs.

Contributions and scope. The scope is weighted structures, such as pruned networks and connectomes with weighted edges, since without weights there is nothing for the null to place and the placement term is undefined. Our contributions are:

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.

- An exact decomposition of the reported statistic into wiring and placement terms, with a paired estimator and a pilot-based choice of K (Proposition 1, Corollary 1).
- An exact finite-sample test for the placement term (Proposition 2).
- A decomposition of the null variance into placement and rewiring components, with a finite bound on the corrected statistic (Proposition 3).
- Case studies that re-examine the rearranging ablation and separate topology from placement in connectome reservoirs, and a domain-agnostic package implementing all of the above.

Existing controls for weights are applied separately from the rewired null. Our work is novel in pairing the two: we apply the null's own weight rule to the observed topology, which separates the reported statistic into the part due to wiring and the part due to the placement of weights, and shows how much each contributes. The pairing also admits an exact test for the placement term and allows the denominator to be decomposed.

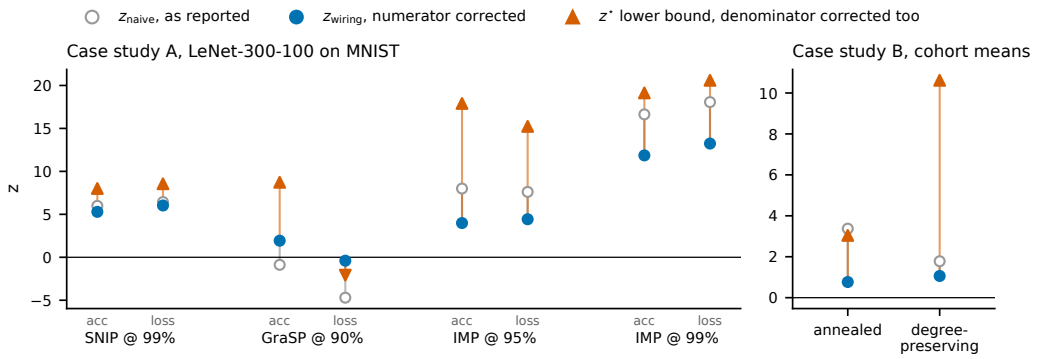


Figure 1: Reported and corrected statistics. Open circles show the reported statistic z_{naive}, and filled circles show the wiring term z_{wiring}, which removes placement from the numerator. Triangles show the bound on $z^\star$, which also removes placement variance from the denominator; they point up for a positive bound and down for a negative one.

2 Related work

Degree-preserving rewiring originates in the edge swaps of Maslov & Sneppen (2002), and Váša & Mišić (2022) classify network nulls by what each preserves and randomizes. In a weighted network the null also needs a rule for the weights on the rewired edges. When the swaps are applied to a weighted network, each weight usually moves with its edge (Milisav et al., 2025); Rubinov & Sporns (2011) instead reassign the weights to approximate the empirical node strengths, and Milisav et al. (2025) place them by simulated annealing to match those strengths. A fixed-topology control keeps the observed topology and randomizes only the weights. Ansmann & Lehnertz (2011; 2012) apply such controls to complete graphs, which have no topology to rewire, and standardize network measures against them. Alstott et al. (2014) show that whether a rich club is found in the topology or in the weights depends on which of the two the null randomizes, and Liu et al. (2018) compare a rewiring null, a structure shuffle, and a weight shuffle on the fixed topology to separate the effects of topology and weights on link prediction.

In pruning, a winning ticket loses its advantage when its mask is kept and its initial weights are redrawn (Frankle & Carbin, 2019), and the layerwise rearranging ablation moves the mask over fixed weights (Frankle et al., 2021). Su et al. (2020) run both kinds of control as separate checks of three trials each: rearranging leaves initialization-time methods and tickets rewound to initialization unaffected, whereas shuffling the weights on the fixed mask harms tickets obtained with learning-rate rewinding. Rearranging also reduces the accuracy of tickets reset to an early iteration at high sparsity (Morcos et al., 2019). The signs of the initial weights can suffice for a ticket (Zhou et al., 2019), whereas permuting rewound weights within layers and sign classes shows that magnitudes

carry the state of early training in deeper networks (Frankle et al., 2020). Recent work attributes the benefit of learning-rate rewinding to early sign changes (Gadhikar & Burkholz, 2024) or to a favorable sign configuration (Oh et al., 2025).

Studies of connectome reservoirs compare the empirical wiring with degree-preserving nulls (Suárez et al., 2021; 2024). In the reference implementation of Suárez et al. (2021) the weights move with the swapped edges, so the null changes both the wiring and the placement of weights. Other studies keep the connectome's topology and change its weights. Costi et al. (2025) find that shuffling the connectome's weights on its own topology affects performance as much as randomizing the topology, and Morra & Daley (2023) find that a reservoir with the connectome's weights permuted or redrawn can perform worse than the unmodified one. Damicelli et al. (2022) draw new weights in every condition, including the empirical topology, so their comparison concerns wiring alone and uses the empirical weights at most through their rank order.

These studies use a rewired null, a fixed-topology control, or both, as separate comparisons. Appendix Table 9 lists the weight rules and draw counts of each control; in the pruning studies, the fixed-topology control is run once or a few times per replicate, or its count is not reported.

3 METHODS

3.1 SETUP AND NOTATION

Let A be an empirical weighted structure and $s(A)$ a score oriented so that larger values are better. A null family $N = (R, \omega)$ consists of a rewiring rule R, which acts on the binary support, and a weight rule ω, which places the empirical multiset of weights on a support. A null draw therefore changes both the support and the placement, whereas a matched draw $\omega(A)$ applies the same weight rule to the empirical support. All expectations below are conditional on A. We write

$$\mu = \mathbb{E}_{R,\omega}[s(\omega(R(A)))]\,, \qquad \sigma^2 = \mathrm{Var}_{R,\omega}[s(\omega(R(A)))]\,, \tag{1}$$

$$\bar{s}_\omega(A) = \mathbb{E}_\omega[s(\omega(A))]\,. \tag{2}$$

The null ensemble contains n draws, and the matched mean in Equation 2 is estimated from K matched draws. The denominator decomposition uses T independently rewired supports with K placements each.

3.2 DECOMPOSITION AND INFERENCE

Proposition 1 (Numerator decomposition)**.** *For any null family with $\sigma > 0$,*

$$z_{\mathrm{naive}} = \frac{s(A) - \mu}{\sigma} = \underbrace{\frac{\bar{s}_\omega(A) - \mu}{\sigma}}_{z_{\mathrm{wiring}}} + \underbrace{\frac{s(A) - \bar{s}_\omega(A)}{\sigma}}_{\Delta_\omega} = z_{\mathrm{wiring}} + \Delta_\omega. \tag{3}$$

Proof. Add and subtract $\bar{s}_\omega(A)$ in the numerator, then divide both terms by σ. □

Propositions 1 and 3 condition on the chosen support, so neither requires the support to be independent of the weights. For a mask pruned by weight magnitude, this dependence appears in Δ_ω.

Proposition 2 (Exact placement test)**.** *Let $S_0 = s(A)$ and $S_k = s(\omega_k(A))$ for independent matched draws $k = 1, \ldots, K$. Let $\bar{S} = (K+1)^{-1} \sum_{i=0}^{K} S_i$, define $d_i = |S_i - \bar{S}|$, and set*

$$p_{\mathrm{placement}} = \frac{1 + \#\{k \geq 1 : d_k \geq d_0\}}{K+1}. \tag{4}$$

Under the null hypothesis that the empirical placement is itself a draw from ω, the scores and hence the deviations are exchangeable. Therefore $\Pr(p_{\mathrm{placement}} \leq \alpha) \leq \alpha \;\forall\; \alpha$. The test is exact and two-sided, and its smallest attainable value is $1/(K+1)$.

Proof. Because centering by the pooled mean is permutation-equivariant, exchangeability of the scores implies exchangeability of the d_i. Every rank of d_0 is therefore equally likely up to ties, and the $1 + \#\{\cdot\}$ construction preserves the bound under ties. □

The test requires no distributional assumption. It is needed because Δ_ω compares a single fixed score with an estimated mean. Larger n or K makes the mean more precise but does not replicate the empirical structure, so a large placement term can arise from the variability of one structure alone.

Proposition 3 (Denominator decomposition and estimator)**.** *Conditioning on the rewired support gives*

$$\sigma^2 = \underbrace{\mathbb{E}_R[\mathrm{Var}_\omega(s \mid R)]}_{\sigma^2_{\text{placement}}} + \underbrace{\mathrm{Var}_R[\mathbb{E}_\omega(s \mid R)]}_{\sigma^2_{\text{rewiring}}}, \tag{5}$$

$$z^\star = \frac{\bar{s}_\omega(A) - \mu}{\sigma_{\text{rewiring}}}. \tag{6}$$

For support t, let $\bar{S}_t$ and v_t be the mean and sample variance of its K placement scores. Let $V_{\bar{S}}$ be the sample variance of the T support means and set $\widehat{\sigma}^2_{\text{placement}} = T^{-1}\sum_t v_t$. Then

$$\mathbb{E}[V_{\bar{S}}] = \sigma^2_{\text{rewiring}} + \frac{\sigma^2_{\text{placement}}}{K}, \tag{7}$$

$$\widehat{\sigma}^2_{\text{rewiring}} = \left[V_{\bar{S}} - \frac{\widehat{\sigma}^2_{\text{placement}}}{K} \right]_+ . \tag{8}$$

where $[x]_+ = \max(x, 0)$. Each support mean averages K placements, so the T means are close to normal, and $(T-1)V_{\bar{S}}$ divided by its expectation is close to χ^2_{T-1}. The resulting chi-square one-sided 95% upper confidence limit on $\sigma^2_{\text{rewiring}}$ is

$$U^2_R = \left[\frac{(T-1)\, V_{\bar{S}}}{\chi^2_{0.05,\, T-1}} - \frac{\widehat{\sigma}^2_{\text{placement}}}{K} \right]_+ , \tag{9}$$

where the pooled within-support variance, which has $T(K-1)$ degrees of freedom, is treated as known. Whenever U_R is at least σ_{rewiring},

$$|z^\star| \geq \left| \frac{\bar{s}_\omega(A) - \mu}{U_R} \right|. \tag{10}$$

In practice, we compute the bound as $|z_{\text{wiring}}|\, \widehat{\sigma}_{\text{split}}/U_R$, where $\widehat{\sigma}_{\text{split}}$ is the standard deviation of the $T \times K$ split draws.

Proof. Equation 5 is the law of total variance. Averaged over supports, the mean of K conditional draws contributes $\sigma^2_{\text{placement}}/K$ to the variance of the support means, so subtracting the estimate of this term removes the bias of $V_{\bar{S}}$. Dividing by an upper confidence limit for σ_{rewiring} instead of its point estimate gives a bound on $|z^\star|$ that is conservative with respect to the denominator. □

Equation 9 rests on two approximations. The T support means are treated as normal, because each averages K placements, and the pooled within-support variance is treated as known, since it has $T(K-1)$ degrees of freedom. The limit is zero only when the variance of the support means falls in the lowest five percent of its distribution under zero rewiring variance. In simulations at both designs, with normal scores and the placement variance estimated rather than known, the coverage of the limit matched its nominal level within simulation error. The bound treats the numerator as known.

Corollary 1 (Matched-draw complexity)**.** *The contribution of the matched draws to the standard error of the estimated wiring term, and the number of matched draws needed to reduce it to ϵ, are*

$$\mathrm{SE}[\widehat{z_{\text{wiring}}}] = \frac{\mathrm{SD}_\omega[s(\omega(A))]}{\sigma\sqrt{K}}, \qquad K \geq \left(\frac{\mathrm{SD}_\omega[s(\omega(A))]}{\sigma\epsilon} \right)^2 . \tag{11}$$

The variance ratio in Equation 11 can be measured in a pilot run, so K can be fixed before the full run. The matched-draw component is the whole error only when μ and σ are known. Both are

estimated from the n null scores, and to first order the delta method adds three terms: $1/n$ from the null mean, $z^2_{\text{wiring}}(\kappa-1)/(4n)$ from the null standard deviation, where κ is the kurtosis of the null scores, and $z_{\text{wiring}}\gamma/n$ from their covariance, where γ is their skewness. Altogether,

$$\widehat{\text{Var}[z_{\text{wiring}}]} \approx \frac{\text{Var}_\omega[s(\omega(A))]}{\sigma^2 K} + \frac{1 + z_{\text{wiring}}\gamma + z^2_{\text{wiring}}(\kappa-1)/4}{n}, \tag{12}$$

which for normal null scores reduces to $1/n + z^2_{\text{wiring}}/(2n)$ in the second term. The standard errors of z_{wiring} in Section 4 are the square root of Equation 12, evaluated with the sample moments of the null scores. Because the terms from the null grow with z^2_{wiring}, the matched-draw component alone understates the error when z_{wiring} is large.

3.3 Sign–magnitude split

When the placement test rejects, the same construction can divide the placement term along a property that ω randomizes. The weight rule ω_{sign} permutes weights within sign classes, preserving the signed multiset and the sign at every position. With $\bar{s}_{\text{sign}}(A) = \mathbb{E}_{\omega_{\text{sign}}}[s(\omega_{\text{sign}}(A))]$,

$$\Delta_\omega = \underbrace{\frac{s(A) - \bar{s}_{\text{sign}}(A)}{\sigma}}_{\Delta_{\text{magnitude}}} + \underbrace{\frac{\bar{s}_{\text{sign}}(A) - \bar{s}_\omega(A)}{\sigma}}_{\Delta_{\text{sign}}}, \tag{13}$$

where the two terms are the magnitude and sign components. The split costs K further scores and no second null ensemble. It is ordered. The magnitude component measures the effect of the magnitudes given the empirical signs, and the sign component the effect of the signs once the magnitudes are permuted. The reverse order would assign their interaction differently, but it is not an exact weight rule, because permuting signs over fixed magnitudes can change the signed multiset (for example, from $\{+1, -2\}$ to $\{-1, +2\}$). Section 5 returns to the interaction.

3.4 Experimental setup

Case A: pruning masks. We train LeNet-300-100 on MNIST for 40 epochs with the `open_lth` settings (Frankle, 2020). The grid covers random pruning, magnitude pruning, SNIP, GraSP, SynFlow, and iterative magnitude pruning (IMP) (Lee et al., 2019; Wang et al., 2020; Tanaka et al., 2020; Frankle & Carbin, 2019) at 90%, 95%, 98%, and 99% sparsity; one method at one sparsity is a cell. IMP rewinds to iteration zero. The literal family reproduces the rearranging ablation, with null draws that shuffle each layer's mask over the fixed initial weights and matched draws that permute each layer's initial weights under the empirical mask. The exactly paired family also permutes the weights on the null side, so both sides apply one weight rule. Each cell uses 100 null and 50 matched draws per family. The denominator decomposition uses 30 rewired supports with 40 placements each, on 4 cells chosen after the main grid had been run (IMP at 95% and 99%, GraSP at 90%, and SNIP at 99%), which illustrate the decomposition and are not a sample of the grid. We record test accuracy, which Frankle et al. (2021) report for the rearranging ablation, and test cross-entropy (loss in the tables), which uses the full predictive distribution and resolves differences about an order of magnitude smaller, both from the same runs over 3 initializations. Within an initialization, training seeds and data order are fixed, so the score is a deterministic function of the mask and its weights.

Case B: connectome reservoirs. We use 5 adult connectomes and the NARMA-10 task (Atiya & Parlos, 2000) on an echo-state reservoir built from the connectome as in Suárez et al. (2021). Both null families rewire the support with the same degree-preserving Maslov–Sneppen swaps (Maslov & Sneppen, 2002). The degree-preserving family then assigns the empirical weights to the rewired edges uniformly at random, whereas the annealed family places them by the simulated-annealing rule of Milisav et al. (2025), which reproduces the empirical strength sequence. Each graph uses 200 null and 50 matched draws, and the denominator decomposition uses 40 rewired supports with 50 placements each. Reservoir, task, and readout seeds are fixed, so the score is a deterministic function of the weighted graph and the spread of the null ensemble reflects structural variation only.

Table 1: Case study A on LeNet-300-100/MNIST under the exactly paired family. Each entry gives the range over 3 initializations, and "sig." counts the initializations with $p_{\text{placement}} < 0.05$. At high sparsity, IMP attributes the placement term to the signs of the initial weights at every initialization, whereas the cells at 90% vary across initializations.

cell	readout	z_{naive}	z_{wiring}	Δ_ω	sig.	$\Delta_{\text{magnitude}}$	Δ_{sign}
IMP @ 95%	acc.	+7.27 to +8.00	+3.83 to +4.16	+3.44 to +4.02	3/3	-0.24 to +0.34	+3.67 to +3.68
	loss	+6.89 to +7.68	+4.17 to +4.56	+2.72 to +3.18	3/3	+0.01 to +0.73	+2.45 to +2.92
IMP @ 98%	acc.	+12.22 to +13.31	+7.45 to +8.54	+4.48 to +4.77	3/3	-0.03 to +0.43	+4.23 to +4.80
	loss	+11.10 to +14.42	+8.59 to +10.14	+2.51 to +4.27	3/3	-0.76 to -0.15	+3.27 to +4.42
IMP @ 99%	acc.	+14.29 to +16.64	+10.49 to +11.86	+3.80 to +4.78	3/3	+0.25 to +0.42	+3.55 to +4.37
	loss	+15.61 to +18.07	+12.08 to +13.24	+3.53 to +4.83	3/3	-0.08 to +0.10	+3.62 to +4.73
IMP @ 90%	acc.	+5.35 to +6.59	+2.25 to +2.60	+2.91 to +3.99	3/3	-0.00 to +0.42	+2.91 to +3.57
	loss	+3.05 to +6.15	+2.30 to +2.77	+0.63 to +3.38	1/3	-0.87 to +0.61	+1.49 to +2.77
GraSP @ 90%	acc.	-2.76 to +0.49	+1.19 to +1.98	-3.95 to -1.50	2/3	-2.84 to -0.40	-1.73 to -1.09
	loss	-4.70 to -0.70	-0.90 to -0.37	-4.30 to -0.33	2/3	-3.48 to -0.52	-0.82 to +0.19

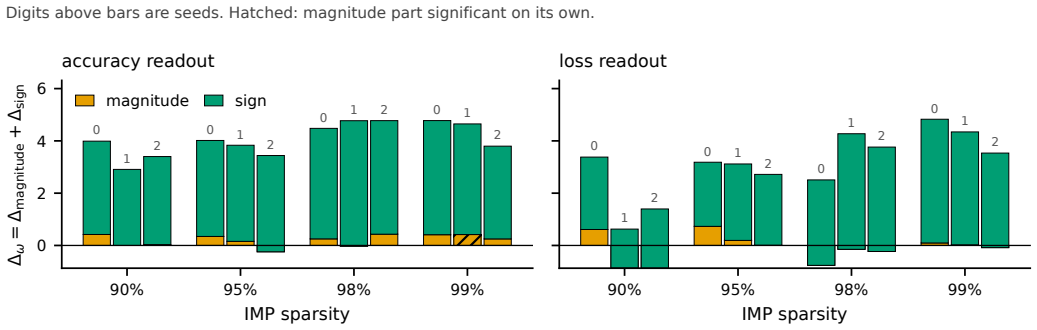


Figure 2: Sign magnitude split of the IMP placement term for both readouts and all initializations. Hatching marks a significant magnitude component.

4 Results

4.1 Pruning masks

On LeNet-300-100, the statistic reported by the rearranging ablation for IMP at 95%, 98%, and 99% sparsity ranged from +6.58 to +18.07 across both null families, both readouts, and all 3 initializations (Table 1). The placement term ranged from +2.51 to +4.83, and the placement test rejected in all 18 cases, one for each combination of sparsity, readout, and initialization. The test uses only the observed score and the matched draws, which both null families share, so it is the same test under both families; the two families also give nearly the same placement term (Appendix Table 7). The sign–magnitude split attributes this placement term to the signs of the initial weights (Figure 2). The magnitude component never exceeded 0.84 in absolute value, whereas the sign component ranged from +2.45 to +4.80, under both readouts.

On ResNet-20 with CIFAR-10, IMP at 95% sparsity was run with three rewinding points, which set the weights that the surviving connections have: the initialization, the weights after 1000 steps, or the fully trained weights with the learning-rate schedule restarted (learning-rate rewinding, Renda et al., 2020; Table 2). The reported statistic is larger with trained than with initial weights, and the wiring term is consistent with zero only after full training. At the rewinding point at iteration 1000 (seed 0, accuracy), the standard error of the wiring term is 0.36, of which the matched-draw component alone is 0.21.

Under learning-rate rewinding, the reported statistic ranged from +11.47 to +18.76. The wiring terms, from -0.16 to +0.29, were each within two standard errors of zero (largest standard error 0.26), so placement accounts for almost the entire statistic (+98% to +101%), and in every case

Table 2: IMP on ResNet-20/CIFAR-10 at 95% sparsity with three rewinding points. Ranges span both readouts and, respectively, 2, 3, and 3 initializations; "sig." counts the cases in which the placement test rejects.

surviving weights	z_{naive}	z_{wiring}	Δ_ω	sig.
initialization	+1.56–+3.25	+1.52–+1.81	-0.10–+1.73	0/4
1000 steps	+13.31–+16.07	+2.35–+2.78	+10.75–+13.28	6/6
full training	+11.47–+18.76	-0.16–+0.29	+11.51–+18.47	6/6

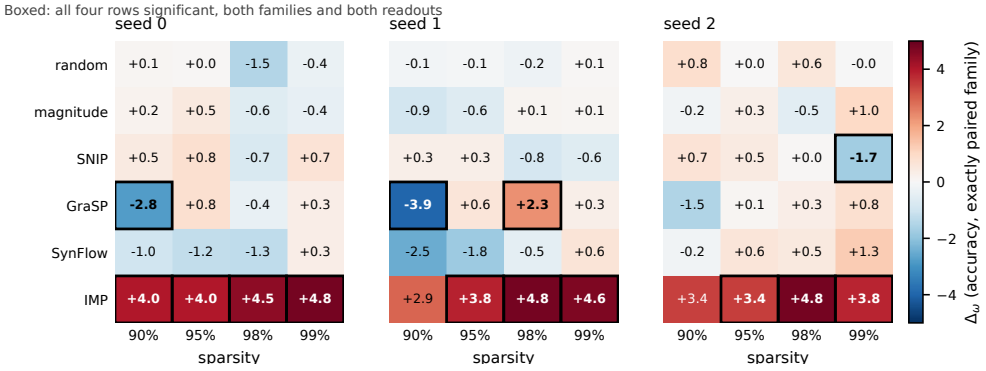


Figure 3: Placement term Δ_ω for every method, sparsity, and initialization. A box marks a cell in which the placement test rejects under both readouts (the test is the same under both null families); only IMP at high sparsity is boxed at every initialization.

the empirical score lies outside all 40 matched scores. The sign component ranged from +12.01 to +18.33, whereas the magnitude component ranged from -0.50 to +0.59 and was significant in only 1 of 6 cases.

At 98% sparsity with the rewinding point at iteration 1000, a single initialization gave wiring terms of +4.82 (accuracy) and +5.71 (cross-entropy), larger than at 95%, and placement shares of +42% and +40%, smaller than at 95%. Under both readouts the empirical score again lies outside all 40 matched scores, and the placement term is attributed to the signs, with a magnitude component that is not significant.

The initialization-time methods did not produce a reproducible placement effect (Figure 3). Of the 16 cells of magnitude pruning, SNIP, GraSP, and SynFlow, only 3 were significant in all four tests (two families by two readouts) at any initialization, and none at more than 2 of the 3 initializations. For GraSP at 90%, the test rejected under both readouts at 2 of 3 initializations, and at seed 0 the wiring terms under the two readouts had opposite signs. IMP at 90% was significant under both readouts at only 3 were significant under both readouts at any initialization of 3 initializations.

Neither a large placement term nor an atypical weight multiset made the test reject. In the random-pruning control, which has no placement structure by construction, $|\Delta_\omega|$ reached 1.85, yet no cell was significant under both readouts at any initialization. Magnitude pruning retained the most atypical weight multiset in the grid (closed-form placement diagnostic $z = +323.9$ at 90% sparsity), yet the placement test never rejected for any of its cells.

4.2 Connectome Reservoirs

Under degree-preserving rewiring, the placement test rejected for 0 of 5 connectomes, so the placement term was not detected in any of the individual graphs (Table 3). Under the annealed family, the test rejected for 4 of 5 graphs.

Table 3: Case study B: cohort means of the NARMA-10 decomposition over 5 adult connectomes, with $n = 200$ null draws and $K = 50$ matched draws under each null family. The placement share is $\Delta_\omega / z_{\text{naive}}$; the last column counts the graphs for which the placement test rejects.

null family	z_{naive}	z_{wiring}	Δ_ω	placement share	graphs with $p_{\text{placement}} < 0.05$
degree-preserving	+1.78	+1.06	+0.72	40.4%	0 of 5
annealed	+3.37	+0.77	+2.61	77.3%	4 of 5

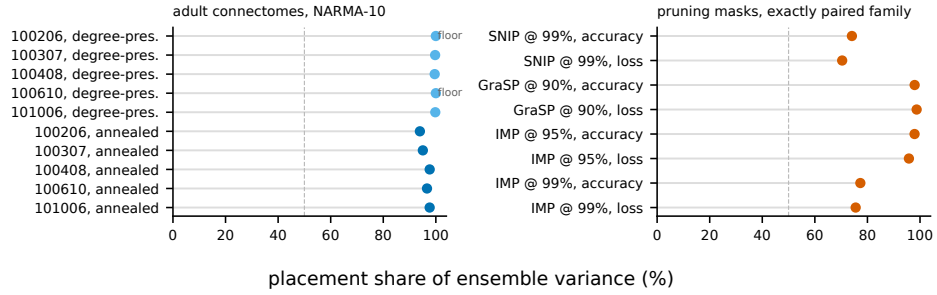


Figure 4: Share of the null variance due to placement, for each connectome and null family and for each pruning cell in which the variance was decomposed. "Floor" marks cases in which the estimated rewiring variance is zero.

4.3 Denominator decomposition

On the connectomes, placement accounted for 96.2% of the variance of the annealed ensemble (Figure 4); the estimated rewiring variance was positive for every graph, and $z^\star$ was at least +3.04. Under degree-preserving rewiring, placement accounted for 99.8%, the estimated rewiring variance was zero for 2 of 5 graphs, and the bound on $z^\star$ was at least +10.62. Propagating the standard error of the numerator through the same rescaling does not move any bound across zero.

In the four pruning cells, placement accounted for 58.7% to 96.4% of the null variance (Table 4), less under IMP at 99% than at 95%. For IMP at 95% under accuracy, $z_{\text{naive}} = +8.00$, $z_{\text{wiring}} = +3.99$, and $z^\star$ is at least +17.91, so correcting the numerator moves the statistic toward zero and correcting the denominator moves it away from zero. For GraSP at 90%, the correction reverses the sign under accuracy, from $z_{\text{naive}} = -0.88$ to a bound of +8.73; under cross-entropy the sign is unchanged ($z_{\text{naive}} = -4.70$, bound -2.08).

5 Limitations

The main pruning grid uses a single architecture and dataset, LeNet-300-100 on MNIST, a setting of the ablation under study that its authors describe as "not representative of larger-scale settings" (Frankle et al., 2021, Appendix K). The decomposition holds for the statistic in general, but the sign result and the negative result for initialization-time methods are findings about this network. The CIFAR study is replicated over initializations only at 95% sparsity; other sparsities have one initialization each (Appendix Table 8), so it is not a sparsity grid. The placement test cannot return a p-value below $1/(K+1)$; with $K = 40$, a p-value at this minimum shows that the empirical score lies outside every matched score, not by how much. The connectome study covers 5 adult graphs at one parcellation and one reservoir task, so it demonstrates the decomposition without establishing how common the effect is across cohorts, parcellations, or scores.

The sign–magnitude split is ordered, and the reverse order would assign the interaction of sign and magnitude differently. This matters little on MNIST, where one component is close to zero, but more for the CIFAR cells at 95% sparsity rewound to iteration 1000, where both are large. The variance estimator can return zero, which leaves only the bound on $z^\star$; under degree-preserving rewiring this happens for 2 of 5 connectomes even with $K = 50$.

Table 4: Denominator correction: the share of the null variance due to placement, a lower bound on $\sigma/\sigma_{\text{rewiring}}$, and the resulting bound on $z^\star$. Case A uses seed 0, with T rewired supports and K placements each (Appendix C); case B gives cohort means.

cell	readout	z_{naive}	z_{wiring}	placement share of σ^2	$\sigma/\sigma_{\text{rewiring}} \geq$	$z^\star$ bound
IMP @ 95%	acc.	+8.00	+3.99	95.2%	4.49	+17.91
	loss	+7.61	+4.43	91.9%	3.44	+15.24
IMP @ 99%	acc.	+16.64	+11.86	66.8%	1.61	+19.14
	loss	+18.07	+13.24	64.7%	1.56	+20.60
SNIP @ 99%	acc.	+5.99	+5.29	62.9%	1.51	+8.01
	loss	+6.43	+6.03	58.7%	1.42	+8.56
GraSP @ 90%	acc.	-0.88	+1.93	95.2%	4.51	+8.73
	loss	-4.70	-0.40	96.4%	5.22	-2.08
connectomes, annealed	NARMA-10	+3.37	+0.77	96.2%	3.99	+3.04
connectomes, degree-preserving	NARMA-10	+1.78	+1.06	99.8%	9.83	+10.62

At the connectome scale studied here the annealed weight rule is close to the identity, which does not hold on smaller graphs. Finally, the placement term is defined only for weighted structures.

6 Discussion

We introduce a decomposition of rewiring-null statistics on weighted structures, to separate the effect of wiring from the effect of weight placement, which a rewiring null changes together. We obtain it by applying the null's own weight rule to the observed topology, which splits the reported statistic exactly into a wiring term and a placement term, admits an exact test for the placement term, and splits the null variance into placement and rewiring components. For IMP masks on LeNet-300-100, the placement term is significant at every high sparsity and initialization and is carried by the signs of the initial weights, and on ResNet-20 with learning-rate rewinding it accounts for almost the entire statistic. In connectome reservoirs, placement accounts for most of the null variance, and removing it from the denominator increases the magnitude of the statistic.

For IMP on LeNet-300-100, the advantage of the mask over its rearrangements is therefore partly an advantage in which initial weights it keeps. For the initialization-time methods, which showed no reproducible placement term, the reported statistic reflects wiring. On ResNet-20, the part of the statistic due to wiring depends on the weights the mask carries. The learning-rate rewinding result confirms earlier work. The fixed-mask shuffle of Su et al. (2020) showed that trained weights matter for tickets obtained with this schedule, and Gadhikar & Burkholz (2024) and Oh et al. (2025) attribute its benefit to the signs; the decomposition adds a standard error for the wiring term and an exact test for the placement term. At the intermediate rewinding point and 95% sparsity, the magnitudes account for most of the placement term, consistent with Frankle et al. (2020).

Placement accounts for most of the null variance wherever variance was decomposed. Because the rewiring variance is only part of the null variance, removing placement from the denominator can only increase the magnitude of the wiring term, whereas removing the placement term from the numerator moves the statistic toward zero when the two terms have the same sign. The two corrections can therefore move the statistic in opposite directions, as they do for IMP at 95% sparsity.

Topology claims based on a weighted rewiring null should state the rewiring and weight rules and report the number K of matched draws with the number of null draws. They should give z_{wiring} with its standard error and Δ_ω with $p_{\text{placement}}$, because the random-pruning control shows that a large placement term can arise from the variability of a single structure. They should also decompose the denominator and, when the estimated rewiring variance is zero, report the bound on $z^\star$. When the placement test rejects, a split along a property that the weight rule randomizes, such as sign and magnitude, shows what the placement term consists of. The released package implements each of these steps.

A replicated series of sparsities on ResNet-20, larger architectures, one-shot pruning of large language models, and weighted graph-learning applications remain to be examined.

AI USE STATEMENT

Generative AI tools were used to aid and polish the writing of this paper. In particular, they were used to polish the wording of the Related Works section, Methods section, Results section and of the Appendices, and to help with LaTeX formatting and the alignment of equations and tables. We reviewed every resulting change and take responsibility for the final text.

REPRODUCIBILITY STATEMENT

The supplementary material contains all code, including the reservoir scorer of case study B, the cache of trained scores, the pruning masks, and every results file used in this paper. Every training run in Section 3.4 and every reservoir score in Section 3.4 is a deterministic function of the structure it scores and of fixed seeds. The five adult connectomes are included under their terms of use, which ship with the data. Section 3 states every assumption.

REFERENCES

Jeff Alstott, Pietro Panzarasa, Mikail Rubinov, Edward T. Bullmore, and Petra E. Vértes. A unifying framework for measuring weighted rich clubs. *Scientific Reports*, 4:7258, 2014. doi: 10.1038/srep07258.

Gerrit Ansmann and Klaus Lehnertz. Constrained randomization of weighted networks. *Physical Review E*, 84(2):026103, 2011.

Gerrit Ansmann and Klaus Lehnertz. Surrogate-assisted analysis of weighted functional brain networks. *Journal of Neuroscience Methods*, 208(2):165–172, 2012.

Amir F. Atiya and Alexander G. Parlos. New results on recurrent network training: Unifying the algorithms and accelerating convergence. *IEEE Transactions on Neural Networks*, 11(3):697–709, 2000. doi: 10.1109/72.846741.

Leone Costi, Alexander Hadjiivanov, Dominik Dold, Zachary F. Hale, and Dario Izzo. The Drosophila connectome as a computational reservoir for time-series prediction. *Biomimetics*, 10(5):341, 2025. doi: 10.3390/biomimetics10050341.

Fabrizio Damicelli, Claus C. Hilgetag, and Alexandros Goulas. Brain connectivity meets reservoir computing. *PLoS Computational Biology*, 18(11):e1010639, 2022.

Jonathan Frankle. OpenLTH: A framework for lottery tickets and beyond. `https://github.com/facebookresearch/open_lth`, 2020.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019.

Jonathan Frankle, David J. Schwab, and Ari S. Morcos. The early phase of neural network training. In *International Conference on Learning Representations*, 2020.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Pruning neural networks at initialization: Why are we missing the mark? In *International Conference on Learning Representations*, 2021.

Advait Gadhikar and Rebekka Burkholz. Masks, signs, and learning rate rewinding. In *International Conference on Learning Representations*, 2024.

Csaba Kerepesi, Balázs Szalkai, Bálint Varga, and Vince Grolmusz. The braingraph.org database of high resolution structural connectomes and the brain graph tools. *Cognitive Neurodynamics*, 11 (5):483–486, 2017. doi: 10.1007/s11571-017-9445-1.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip H. S. Torr. SNIP: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*, 2019.

Bo Liu, Shuang Xu, Ting Li, Jing Xiao, and Xiao-Ke Xu. Quantifying the effects of topology and weight for link prediction in weighted complex networks. *Entropy*, 20(5):363, 2018. doi: 10.3390/e20050363.

Sergei Maslov and Kim Sneppen. Specificity and stability in topology of protein networks. *Science*, 296(5569):910–913, 2002.

Filip Milisav, Vincent Bazinet, Richard F. Betzel, and Bratislav Mišić. A simulated annealing algorithm for randomizing weighted networks. *Nature Computational Science*, 5(1):48–64, 2025.

Ari S. Morcos, Haonan Yu, Michela Paganini, and Yuandong Tian. One ticket to win them all: Generalizing lottery ticket initializations across datasets and optimizers. In *Advances in Neural Information Processing Systems*, volume 32, pp. 4933–4943, 2019.

Jacob Morra and Mark Daley. Using connectome features to constrain echo state networks. In *International Joint Conference on Neural Networks*, 2023. doi: 10.1109/IJCNN54540.2023.10191832.

Junghun Oh, Sungyong Baik, and Kyoung Mu Lee. Find a winning sign: Sign is all we need to win the lottery. In *International Conference on Learning Representations*, 2025.

Alex Renda, Jonathan Frankle, and Michael Carbin. Comparing rewinding and fine-tuning in neural network pruning. In *International Conference on Learning Representations*, 2020.

Mikail Rubinov and Olaf Sporns. Weight-conserving characterization of complex functional brain networks. *NeuroImage*, 56(4):2068–2079, 2011.

Jingtong Su, Yihang Chen, Tianle Cai, Tianhao Wu, Ruiqi Gao, Liwei Wang, and Jason D. Lee. Sanity-checking pruning methods: Random tickets can win the jackpot. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Laura E. Suárez, Blake A. Richards, Guillaume Lajoie, and Bratislav Mišić. Learning function from structure in neuromorphic networks. *Nature Machine Intelligence*, 3(9):771–786, 2021.

Laura E. Suárez, Agoston Mihalik, Filip Milisav, Kenji Marshall, Mingze Li, Petra E. Vértes, Guillaume Lajoie, and Bratislav Mišić. Connectome-based reservoir computing with the conn2res toolbox. *Nature Communications*, 15:656, 2024.

Hidenori Tanaka, Daniel Kunin, Daniel L. K. Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

František Váša and Bratislav Mišić. Null models in network neuroscience. *Nature Reviews Neuroscience*, 23(8):493–504, 2022.

Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. In *International Conference on Learning Representations*, 2020.

Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. Deconstructing lottery tickets: Zeros, signs, and the supermask. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

A Full results for case study A

Tables 5 and 6 give every cell of the grid at every initialization under the exactly paired family: the reported statistic, the wiring term, the placement term, and the placement test. Table 7 gives, for every cell, the largest difference between that family and the literal reproduction of the ablation over both readouts and every seed. The literal family's own rows are in the released results files. All three tables are generated from those files by the same script that produces every number in the main text.

Table 8 lists every ResNet-20 cell. With $K = 40$ matched draws, the smallest p-value the placement test can return is $1/(K+1)$. At 95% sparsity, all 6 rows rewound to iteration 1000 and all 6 rows with

Table 5: Case study A, every cell, exactly paired family, accuracy readout, one block per initialization seed. Columns within a block: z_{naive}, z_{wiring}, Δ_ω, $p_{\text{placement}}$. Generated from the results files.

cell	seed 0				seed 1				seed 2			
	z_{naive}	z_{wiring}	Δ_ω	p	z_{naive}	z_{wiring}	Δ_ω	p	z_{naive}	z_{wiring}	Δ_ω	p
random @ 90%	-0.06	-0.17	+0.11	0.980	-0.11	+0.02	-0.13	0.941	+0.79	-0.04	+0.83	0.510
random @ 95%	+0.15	+0.11	+0.05	0.706	+0.06	+0.14	-0.08	0.510	+0.08	+0.08	+0.00	1.000
random @ 98%	-1.13	+0.32	-1.45	0.078	-0.15	+0.09	-0.23	0.804	+0.60	+0.04	+0.56	0.431
random @ 99%	+0.64	+0.99	-0.35	0.686	+1.33	+1.27	+0.06	0.941	+0.71	+0.73	-0.02	0.980
magnitude @ 90%	+0.31	+0.14	+0.17	0.863	-0.88	+0.05	-0.93	0.392	-0.33	-0.17	-0.16	0.843
magnitude @ 95%	+0.61	+0.11	+0.50	0.549	-0.54	+0.10	-0.64	0.490	+0.23	-0.05	+0.28	0.804
magnitude @ 98%	-0.05	+0.55	-0.60	0.510	+0.28	+0.14	+0.14	0.804	-0.83	-0.29	-0.54	0.647
magnitude @ 99%	-0.00	+0.43	-0.43	0.608	-0.34	-0.44	+0.10	0.902	+0.89	-0.10	+0.99	0.235
SNIP @ 90%	+2.47	+2.00	+0.46	0.667	+1.78	+1.48	+0.29	0.784	+2.45	+1.77	+0.68	0.510
SNIP @ 95%	+2.70	+1.91	+0.79	0.608	+1.37	+1.08	+0.30	0.784	+1.90	+1.43	+0.46	0.667
SNIP @ 98%	+4.34	+5.08	-0.74	0.569	+2.64	+3.47	-0.83	0.373	+3.16	+3.12	+0.04	0.961
SNIP @ 99%	+5.99	+5.29	+0.70	0.196	+5.37	+5.99	-0.62	0.353	+2.53	+4.26	-1.74	0.039
GraSP @ 90%	-0.88	+1.93	-2.81	1/51	-2.76	+1.19	-3.95	1/51	+0.49	+1.98	-1.50	0.216
GraSP @ 95%	+2.74	+1.95	+0.79	0.549	+1.98	+1.38	+0.59	0.627	+1.81	+1.72	+0.09	0.922
GraSP @ 98%	+3.35	+3.73	-0.38	0.667	+4.71	+2.42	+2.29	1/51	+4.35	+4.02	+0.33	0.627
GraSP @ 99%	+4.73	+4.41	+0.32	0.392	+3.57	+3.27	+0.30	0.549	+5.90	+5.10	+0.79	0.078
SynFlow @ 90%	-0.99	+0.01	-1.00	0.451	-2.15	+0.33	-2.48	0.039	-0.09	+0.07	-0.16	0.922
SynFlow @ 95%	-0.90	+0.33	-1.23	0.235	-1.29	+0.53	-1.82	0.078	+1.16	+0.54	+0.62	0.510
SynFlow @ 98%	+2.97	+4.25	-1.28	0.196	+3.29	+3.79	-0.51	0.549	+3.62	+3.14	+0.49	0.667
SynFlow @ 99%	+7.07	+6.73	+0.34	0.647	+5.67	+5.04	+0.63	0.431	+8.50	+7.23	+1.26	0.176
IMP @ 90%	+6.59	+2.60	+3.99	1/51	+5.35	+2.44	+2.91	1/51	+5.65	+2.25	+3.40	1/51
IMP @ 95%	+8.00	+3.99	+4.02	1/51	+7.99	+4.16	+3.83	1/51	+7.27	+3.83	+3.44	1/51
IMP @ 98%	+12.33	+7.86	+4.48	1/51	+13.31	+8.54	+4.77	1/51	+12.22	+7.45	+4.77	1/51
IMP @ 99%	+16.64	+11.86	+4.78	1/51	+15.95	+11.30	+4.65	1/51	+14.29	+10.49	+3.80	1/51

Table 6: Case study A, every cell, exactly paired family, test cross-entropy readout, one block per initialization seed. Columns within a block: z_{naive}, z_{wiring}, Δ_ω, $p_{\text{placement}}$. Generated from the results files.

cell	seed 0				seed 1				seed 2			
	z_{naive}	z_{wiring}	Δ_ω	p	z_{naive}	z_{wiring}	Δ_ω	p	z_{naive}	z_{wiring}	Δ_ω	p
random @ 90%	+1.56	+0.12	+1.43	0.216	-0.08	+0.12	-0.20	0.843	+1.83	-0.02	+1.85	0.059
random @ 95%	+0.40	+0.08	+0.32	0.333	-0.25	+0.08	-0.33	0.314	-0.04	-0.04	-0.00	1.000
random @ 98%	-1.40	+0.41	-1.81	1/51	-0.06	+0.13	-0.20	0.784	+0.48	+0.05	+0.43	0.549
random @ 99%	+0.79	+1.44	-0.64	0.588	+2.19	+1.82	+0.37	0.627	+1.07	+1.05	+0.02	1.000
magnitude @ 90%	-1.25	+0.28	-1.53	0.078	-2.15	+0.06	-2.20	0.059	-1.21	-0.22	-0.99	0.373
magnitude @ 95%	+0.85	+0.15	+0.69	0.373	+1.04	+0.32	+0.72	0.431	+0.24	+0.16	+0.08	0.902
magnitude @ 98%	-0.04	+0.45	-0.49	0.549	+0.61	+0.26	+0.35	0.529	-0.69	-0.14	-0.55	0.608
magnitude @ 99%	+1.01	+0.48	+0.53	0.471	-0.22	-0.41	+0.19	0.882	+1.41	-0.32	+1.73	0.059
SNIP @ 90%	-0.88	-1.38	+0.49	0.765	-1.15	-1.65	+0.50	0.804	-1.49	-2.04	+0.55	0.608
SNIP @ 95%	+0.22	-1.47	+1.69	0.255	-1.49	-1.35	-0.14	0.980	-1.72	-1.27	-0.44	0.804
SNIP @ 98%	+5.20	+5.23	-0.03	0.961	+2.67	+4.02	-1.34	0.098	+4.48	+3.34	+1.13	0.216
SNIP @ 99%	+6.43	+6.03	+0.40	0.529	+5.86	+6.20	-0.34	0.627	+3.33	+5.00	-1.67	0.039
GraSP @ 90%	-4.70	-0.40	-4.30	1/51	-4.24	-0.90	-3.35	1/51	-0.70	-0.37	-0.33	0.745
GraSP @ 95%	+0.49	-0.70	+1.19	0.373	-0.38	-0.54	+0.16	0.784	+0.37	-0.39	+0.76	0.510
GraSP @ 98%	+4.54	+4.11	+0.43	0.667	+4.75	+2.75	+2.00	1/51	+5.64	+4.70	+0.95	0.373
GraSP @ 99%	+5.50	+4.96	+0.54	0.118	+4.14	+3.79	+0.36	0.412	+6.44	+5.53	+0.90	0.039
SynFlow @ 90%	-3.14	-2.59	-0.55	0.667	-2.98	-2.50	-0.47	0.706	-2.62	-2.60	-0.02	0.980
SynFlow @ 95%	-1.86	-2.12	+0.25	0.843	-5.35	-2.19	-3.16	0.039	-1.34	-2.00	+0.65	0.529
SynFlow @ 98%	+2.72	+4.57	-1.85	1/51	+3.12	+3.38	-0.26	0.784	+3.81	+3.33	+0.48	0.667
SynFlow @ 99%	+6.93	+7.44	-0.50	0.529	+5.86	+6.08	-0.22	0.784	+9.44	+8.23	+1.20	0.137
IMP @ 90%	+6.15	+2.77	+3.38	1/51	+3.05	+2.43	+0.63	0.471	+3.70	+2.30	+1.40	0.098
IMP @ 95%	+7.61	+4.43	+3.18	1/51	+7.68	+4.56	+3.12	1/51	+6.89	+4.17	+2.72	1/51
IMP @ 98%	+11.10	+8.59	+2.51	1/51	+14.42	+10.14	+4.27	1/51	+12.81	+9.05	+3.76	1/51
IMP @ 99%	+18.07	+13.24	+4.83	1/51	+16.91	+12.56	+4.34	1/51	+15.61	+12.08	+3.53	1/51

Table 7: The two pruning families agree. For every cell, the largest absolute difference between the exactly paired family and the literal reproduction of the ablation, over both readouts and every seed, in z_{naive} and in Δ_ω. Generated from the results files.

cell	$\max \lvert\Delta z_{\text{naive}}\rvert$	$\max \lvert\Delta\Delta_\omega\rvert$
random @ 90%	0.27	0.09
random @ 95%	0.11	0.04
random @ 98%	0.15	0.19
random @ 99%	0.57	0.07
magnitude @ 90%	0.33	0.14
magnitude @ 95%	0.28	0.08
magnitude @ 98%	0.34	0.12
magnitude @ 99%	0.37	0.41
SNIP @ 90%	0.36	0.07
SNIP @ 95%	0.34	0.13
SNIP @ 98%	0.98	0.15
SNIP @ 99%	1.26	0.16
GraSP @ 90%	0.67	0.51
GraSP @ 95%	0.42	0.10
GraSP @ 98%	0.73	0.11
GraSP @ 99%	1.35	0.17
SynFlow @ 90%	0.29	0.14
SynFlow @ 95%	0.27	0.04
SynFlow @ 98%	0.62	0.30
SynFlow @ 99%	1.14	0.15
IMP @ 90%	0.78	0.54
IMP @ 95%	0.91	0.50
IMP @ 98%	1.76	0.48
IMP @ 99%	1.40	0.45

Table 8: Every ResNet-20/CIFAR-10 cell: method, target sparsity, rewind step, initialization seed, the accelerator the whole cell was scored on, and under each readout the reported statistic, the wiring term with its standard error, the placement term, its share, the exact placement test, and the ordered split. Exactly paired family throughout. Generated from the results files.

method	sparsity	rewind	seed	device	readout	z_{naive}	z_{wiring} (SE)	Δ_ω	share	$p_{\text{placement}}$	Δ_{mag}	Δ_{sign}
IMP	90%	1000	0	RTX 5070 Laptop GPU	accuracy	+13.05	+1.18 (0.23)	+11.87	+91%	1/41	+9.71	+2.16
					loss	+6.79	+1.06 (0.23)	+5.73	+84%	1/41	+3.50	+2.22
IMP	95%	0	0	mps	accuracy	+1.56	+1.65 (0.28)	-0.10	-6%	0.951	+0.03	-0.13
					loss	+1.77	+1.63 (0.27)	+0.13	+7%	0.951	+0.10	+0.03
IMP	95%	0	2	A100-SXM4-80GB	accuracy	+3.25	+1.52 (0.34)	+1.73	+53%	0.195	+1.40	+0.32
					loss	+2.15	+1.81 (0.30)	+0.34	+16%	0.659	+0.29	+0.06
IMP	95%	1000	0	RTX 5070 Laptop GPU	accuracy	+13.31	+2.57 (0.36)	+10.75	+81%	1/41	+7.69	+3.06
					loss	+13.86	+2.43 (0.34)	+11.43	+82%	1/41	+8.32	+3.10
IMP	95%	1000	1	RTX 5070 Laptop GPU	accuracy	+14.21	+2.42 (0.32)	+11.79	+83%	1/41	+9.31	+2.49
					loss	+13.36	+2.35 (0.30)	+11.01	+82%	1/41	+8.40	+2.61
IMP	95%	1000	2	RTX 5070 Laptop GPU	accuracy	+16.07	+2.78 (0.35)	+13.28	+83%	1/41	+10.14	+3.14
					loss	+15.03	+2.43 (0.46)	+12.60	+84%	1/41	+9.41	+3.20
IMP	95%	LR	0	RTX 5070 Laptop GPU	accuracy	+18.71	+0.24 (0.21)	+18.47	+99%	1/41	+0.59	+17.87
					loss	+16.45	+0.29 (0.21)	+16.16	+98%	1/41	-0.06	+16.22
IMP	95%	LR	1	mps	accuracy	+13.26	-0.16 (0.20)	+13.42	+101%	1/41	-0.37	+13.79
					loss	+11.47	-0.04 (0.20)	+11.51	+100%	1/41	-0.50	+12.01
IMP	95%	LR	2	A100-SXM4-80GB	accuracy	+18.76	+0.29 (0.23)	+18.47	+98%	1/41	+0.13	+18.33
					loss	+16.73	+0.17 (0.26)	+16.56	+99%	1/41	+0.21	+16.35
IMP	98%	1000	0	mps	accuracy	+8.25	+4.82 (0.47)	+3.42	+42%	1/41	+1.02	+2.40
					loss	+9.55	+5.71 (0.67)	+3.84	+40%	1/41	+0.80	+3.04

learning-rate rewinding reach this minimum, whereas none of the 4 rows rewound to initialization rejects. The magnitude component exceeds the sign component in all 6 rows rewound to iteration 1000 and in none of the rows with learning-rate rewinding, where the sign component ranges from +12.01 to +18.33. Because the split is ordered, the attribution of any interaction between sign and magnitude depends on that order.

B CONTROLS IN PRIOR WORK

Table 9 lists, for each study discussed in Section 2, the rewiring rule, the weight rule on the null side, the fixed-topology control, and the numbers of null and matched draws it reports.

Table 9: Controls that move weights, with their draw counts. n is the number of null draws and K the number of weight-rule draws on the fixed empirical topology. N/A: no separate weight rule. NR: draws implied but not reported.

work	rewiring	null weights	fixed-topology surrogate	n	K
Frankle et al. (2021)	mask shuffle	moved mask selects new values	real-mask reinitialization	one shuffle per replicate implied	NR
Su et al. (2020)	mask rearrange	moved mask selects new values	kept weights shuffled within layer on the real mask	one per trial, three trials	one per trial, three trials
Frankle & Carbin (2019)	none	N/A	real-mask reinitialization	five ticket trials	three per ticket
Zhou et al. (2019)	none	N/A	real-mask reshuffle	none	one per run
Frankle et al. (2020)	none	N/A	θ_k permuted within layer on the IMP mask, also within sign	none	one per replicate, five replicates
Suárez et al. (2021)[a]	degree preserving	carried in reference implementation	no	1,000	N/A
Suárez et al. (2024)	degree and density preserving	not stated	no	500	N/A
Costi et al. (2025)	matched sparsity	random draws or empirical multiset	random weights; shuffled empirical weights	no topology count	NR
Damicelli et al. (2022)	density, fixed k, or full	uniform draws	random or rank-matched weights	100 per condition	100
Morra & Daley (2023)	none	N/A	nonzero weights permuted, or redrawn, on the fixed connectome	N/A	NR
Milisav et al. (2025)	degree swaps, then annealing	multiset permuted to match strengths	no	10,000	N/A
Rubinov & Sporns (2011)[a]	signed-degree switching	iteratively rank-matched to strengths	no	NR	N/A
Ansmann & Lehnertz (2011; 2012)	none; complete graphs	shuffled weights or strength-preserving chain	fixed-topology permutation	4,096	4,096

[a] Entries for these works follow open preprints, bioRxiv 10.1101/2020.11.10.350876 and arXiv 1103.5112 respectively.

C PROTOCOL

C.1 CASE STUDY A

Network and training. LeNet-300-100 on MNIST, with the hyperparameters of the reference implementation of the paper under study (Frankle, 2020): stochastic gradient descent at learning rate 0.1 without momentum or weight decay, batches of 128, 40 epochs, Kaiming normal initialization. The test set is the standard 10,000 images. Accuracy and mean test cross-entropy are both recorded at the end of every run.

What varies and what does not. A structure is a mask together with the weights under it. Biases, batch-normalization parameters, and the order of the training data are drawn from two seeds that are shared by the real structure, every null draw, and every matched draw, so that two runs on the same structure return the same score. Redrawing the initialization (seeds 1 and 2) changes every method's mask as well as the weights, and each seed has its own cache, so no run is shared across seeds.

Pruning methods. SNIP, GraSP and SynFlow follow the unified implementations of Tanaka et al. (2020), which are the versions every benchmark runs and which differ from the original papers' prose in places; GraSP uses temperature 200 and keeps the largest scores, SynFlow is iterative and linearized in double precision. Magnitude pruning and random pruning are the two controls. Iterative magnitude pruning removes a fifth of the surviving weights by trained magnitude at each round, rewinds the survivors to the initialization, and repeats; all four target sparsities come off one pruning path, and the list of targets is part of the schedule, because stopping on an intermediate target changes the densities the path visits. Every mask hits its target sparsity exactly.

Null families and draws. The literal family shuffles each layer's mask and leaves the weights in place; its matched side keeps the mask and permutes each layer's initial weights. The exactly paired family permutes the weights on the null side as well, so both sides run one rule. Each cell uses 100 null draws per family and 50 matched draws, plus 50 draws of the sign-preserving rule for the sign–magnitude split. The denominator split was measured on 4 cells with 30 rewired masks and 40 placements on each. Every scored run is cached by the content of the structure and the training configuration, so a repeated or extended run retrains only what it adds.

Execution. Structures are trained in stacked groups, the whole cell as one tensor per layer; the arithmetic is the same and the summation order differs, so the execution path and group size are part of the cache key and a decomposition never mixes the two. A structure placed at different positions in a group returns bit-identical scores, which a test enforces.

The CIFAR study. ResNet-20 on CIFAR-10 uses the settings of the reference implementation (Frankle, 2020): 160 epochs, learning rate 0.1, momentum 0.9, weight decay 10^{-4}, and learning-rate drops at epochs 80 and 120. IMP targets 95% sparsity with three rewinding points, which set the weights that the surviving connections carry: the initialization, the weights at iteration 1000 (the late-rewinding setting of Frankle et al. (2021)), or the weights at the end of training, with the learning-rate schedule restarted (Renda et al., 2020). The cells at other sparsities use one initialization and the rewinding point at iteration 1000. In every setting, the null draws shuffle the mask over the resulting weights, and the matched draws permute those weights within each layer. We use the exactly paired family, both readouts, 40 null draws, and 40 matched draws, over 2, 3, and 3 initializations for the three rewinding points, respectively. Each run is trained separately, and its cache key records the execution path.

C.2 Case study B

Graphs and score. Five adult connectomes at the 83-region parcellation, streamline-count weighted, the redistributable graphs of the braingraph.org database (Kerepesi et al., 2017), computed from Human Connectome Project data; their terms of use ship with the data. Data were provided in part by the Human Connectome Project, WU-Minn Consortium (Principal Investigators: David Van Essen and Kamil Ugurbil; 1U54MH091657) funded by the 16 NIH Institutes and Centers that support the NIH Blueprint for Neuroscience Research; and by the McDonnell Center for Systems Neuroscience at Washington University. The score is the NARMA-10 error (Atiya & Parlos, 2000) of an echo-state reservoir built on the weighted graph, in the manner of Suárez et al. (2021), with spectral radius 0.95, leak rates in $[0.85, 1.0]$, input scale 0.1, washout 100, 2,000 samples, ridge readout, every score a mean over five reservoir seeds and three input draws.

Null families and draws. Both families rewire with Maslov–Sneppen swaps. The degree-preserving family re-deals the empirical weights uniformly onto the rewired edges; the annealed family places them by simulated annealing onto the empirical strength sequence (Milisav et al., 2025). 200 null draws and 50 matched draws per graph and family for the numerator; 40 rewired supports with 50 placements each for the denominator.