WEIGHTS HAVE TRAITS: ESTIMATING FINE-TUNING LINEAGE IN LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Over a million deep learning models are hosted online, and most are fine-tuned descendants of a few ancestors. Their lineage is rarely fully documented, even though provenance, auditing, and safe deployment depend on it. We hypothesize that fine-tuning acts as an evolutionary process, so that weights have traits inherited from their ancestors. We show that a model *phylogeny* (evolutionary tree) can be estimated from the weights of the observed leaf models alone, with all ancestors latent. On controlled LLM fine-tuning trees whose true structure is known, our estimates contain every true clade (a group of models descended from one ancestor) in 85–100% of trees, compared with 3% for a random tree. Treating each weight layer as a gene-like estimator of lineage, we find that a single weight matrix, such as one attention key matrix (0.1–0.2% of the weights), estimates the phylogeny as well as all the weights combined. Moreover, estimation accuracy is positively associated with two measures from phylogenetics, the Atteson margin and a tree-likeness score based on four-point additivity. The latter needs no true tree. We also report two secondary findings. Models closer in weight space often behave more similarly on held-out tasks, and weight-based estimation matches behavior-based estimation (PhyloLM) on two documented real families while outperforming it on the controlled Llama trees (100% of clades versus 35–56%).

1 INTRODUCTION

While there are over one million deep learning models available on Hugging Face and a handful of frontier labs constantly releasing updated versions, most models are fine-tuned descendants of a few ancestors. Very few of those models transparently document their full model lineage (Horwitz et al., 2025a). Benchmarks can reveal how a model performs on a task, but not how a model originated, how it differs from its sibling models, or how it may fail. In contrast, training lineage can reveal much about model provenance, auditing, and safety (Xu & Sheng, 2026; Nikolic et al., 2026; Wu et al., 2025).

We, like other prior works (Horwitz et al., 2025b; Yu & Wang, 2025; Wu et al., 2026; Gallagher et al., 2026), observe that fine-tuning is a path-dependent, branching process. Each child model inherits its parent's updated weights, nudged towards a certain loss and/or dataset. This is precisely an evolutionary process, as described in Yax et al. (2025), and moreover, the evolutionary path leaves signal in weight space. We treat a line or genealogy of fine-tuned models as a phylogeny (i.e., an evolutionary tree) and ask whether this phylogeny can be *estimated* from the model weights alone.

As motivation, consider a firm that forks a base model into finance, IT, and fraud variants, each fine-tuned until dozens of versions circulate with no recorded lineage (Fig. 1A). A practitioner needs to know which to build on, where two versions diverged, and where each fails. Phylogenetic algorithms were built for exactly this inference in biology (Saitou & Nei, 1987; Robinson & Foulds, 1981), and we adapt them to neural net model weights. We estimate a phylogeny from model weights and score it against a *known* true phylogeny (Fig. 1A), link the changes in weights to changes in behavior (Fig. 1B), and ask whether diagnostic measurements indicate when phylogeny estimation will succeed even without access to the true tree (Fig. 1C).

Our central finding is that weights have traits. In particular, fine-tuned models inherit a lineage signal in their weights that is strong enough to estimate their family tree. Moreover, similar model weights are indicative of similar model behavior on tasks the models were never fine-tuned on. Our

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.

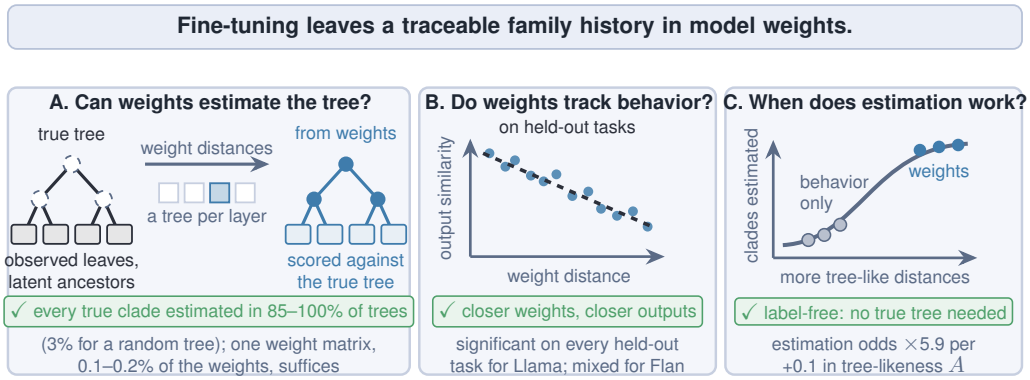


Figure 1: Overview. **(A)** Can the training tree be estimated from the leaf models' weights alone, with ancestors latent? **(B)** Do models closer in weight space behave alike on held-out tasks? **(C)** Does the label-free tree-likeness score signal when estimation succeeds?

primary contribution (P1) is a method and an evaluation framework for estimating phylogenies from weights. Two secondary findings (S1, S2) follow from these weight-based estimates.

- **P1 (primary). Phylogeny estimation from leaf-model weights with latent ancestors.** We estimate a model phylogeny from the weights of the observed leaf models alone and treat every ancestor as latent. Each weight layer acts as a gene-like estimator that yields its own tree, and an average of these trees gives the combined estimate. To evaluate the method, we introduce controlled LLM fine-tuning trees with known true trees for Flan-T5-base and Llama-3.2-1B under full fine-tuning and LoRA. Weights correctly estimate every true clade in 85–100% of trees and 92–100% of clades overall, compared with 3% and 9% for a random tree. Moreover, a single weight matrix, such as one attention key matrix (0.1–0.2% of the weights), estimates phylogenies just as well as all the weights combined. We also use two classical phylogenetic diagnostics, a tree-likeness score based on four-point additivity and a fitted Atteson margin. The former needs no true tree, and in our experiments higher scores were associated with more accurate estimates. On two documented Hugging Face LLM families, our weight-based method correctly estimates every documented clade.
- **S1 (secondary). Similar weights signal similar behavior.** Models that are closer in weight space produce more similar outputs on *held-out* tasks involving truthfulness, reasoning, knowledge, and common sense. Unlike tree estimation, this analysis uses every trained model, including internal checkpoints, and pools within-tree correlations across trees with a random-effects meta-analysis. For Llama, the pooled correlation is significant and negative on every held-out task ($|r_{\mathrm{DL}}| = 0.17$–0.23), whereas Flan's results are negative and significant in four of six tasks.
- **S2 (secondary). Weight-based estimation can outperform behavior-based estimation.** On the controlled Llama trees, weights correctly estimate 100% of clades, versus 35–56% for the behavior-only PhyloLM (Yax et al., 2025). The tree-likeness score from P1 signals this gap. We show that PhyloLM's behavioral distances are much less tree-like, and across our estimators the odds of correctly estimating a clade grow 4.5-fold per 0.10 increase in the score. On the same two documented families, the behavior-based method estimates all true clades, indicating that behavior-only estimation can perform just as well as weight-based estimation in some scenarios.

2 Related Work

Deep learning lineage studies broadly infer model relationships from either *model weights* or *behavior*. These methods have been driven by different applications including provenance and IP (Horwitz et al., 2025b; Zeng et al., 2024; Nikolic et al., 2026), model poisoning and misalignment (Cloud et al., 2025), trait inheritance (Cloud et al., 2025; Yang et al., 2026), and data auditing (Zhu et al., 2025).

Model weights: Several works trace weight changes from fine-tuning or reinforcement learning from human feedback (RLHF) back to parent models. MoTHer (Horwitz et al., 2025b) recon-

Table 1: We compare with the closest prior methods. Controlled trees are families the authors fine-tuned. Empirical confidence is a label-free score that is tested against estimation accuracy.

Method	Access	Latent ancestors	Controlled trees	Weight–behavior	Empirical confidence	Cross-arch.
MoTHer (Horwitz et al., 2025b)	weights	✗	vision	✗	✗	✗
Neural Phylogeny (Yu & Wang, 2025)	weights	✗	vision	✗	✗	✗
PhyloLM (Yax et al., 2025)	behavior	✓	✗	✗	✗	✓
LLM DNA (Wu et al., 2026)	behavior	✓	✗	✗	✗	✓
Ours	**weights**	✓	**LLM**	✓	✓	✗

structs directed model trees from weight similarity via a directed minimum spanning tree and evaluates against controlled, known model hierarchies; Yu & Wang (2025) identify parent–child and grandparent–child lineage with both a learning-free fine-tuning approximation and a learned detector ($> 99\%$ accuracy). Beyond direct lineage, spectral weight "fingerprints" identify common ancestry (Wang et al., 2026), and Spectral DeTuning (Horwitz et al., 2024) reconstructs base weights from shared LoRA variants. Unlike MoTHer, which reconstructs relationships among observed parent and descendant models, we observe only leaf models and infer a phylogeny with latent ancestral branching points. This setting is relevant to LLM family reconstruction, where intermediate or parent models may not be available.

REEF (Zhang et al., 2025) compares internal representations using centered kernel alignment (CKA; Kornblith et al., 2019) to identify model derivation, while TensorGuard (Wu et al., 2025) uses gradient-response fingerprints for pairwise model similarity and family classification.

Other works examine *why* lineage is detectable in models. Models fine-tuned from a shared initialization stay within a single low-error basin and are robust to SGD noise (Neyshabur et al., 2020; Wortsman et al., 2022; Frankle et al., 2020), so pairwise weight distance can approximate genealogical distance. Additionally, task vectors (Ilharco et al., 2023) are near-orthogonal across tasks, so distinct lineages contribute independent directions. LoRA (Hu et al., 2022) and intrinsic-dimensionality results (Aghajanyan et al., 2021) demonstrate that meaningful adaptation can occur in low-dimensional parameter subspaces. Together, these results motivate the possibility that lineage information remains detectable after fine-tuning.

Behavior: Behavioral approaches infer model relationships using only API access or model outputs, without requiring access to model weights (Yax et al., 2025; Wu et al., 2026). PhyloLM (Yax et al., 2025) builds LLM phylogenies from output similarity and predicts benchmark performance, while LLM DNA (Wu et al., 2026) defines a functional "DNA" representation carrying lineage information. Both methods estimate phylogenies over hundreds of models. Other output-only methods test pairwise provenance from next-token distributions (Nikolic et al., 2026) or output independence (Zhu et al., 2025). A key advantage of behavioral methods is that they can compare models across architectures and sizes, whereas weight-based comparisons generally require compatible parameterizations. Conversely, weight-based lineage estimates do not depend on the choice of prompts or test tasks.

Our Approach: We compare weight- and behavior-based approaches, determine where phylogenetic signal can be detected, and evaluate the estimated phylogenies against known true trees. Our findings are consistent with prior work showing that inherited behavioral signals can be preserved under sufficient training signal (Cloud et al., 2025), as well as phylogenetic results showing that the underlying tree can be correctly estimated when certain conditions on the estimated distances are satisfied (Atteson, 1999). As Table 1 shows, ours is the only work that combines weight-based estimation, latent ancestors (which allow for unobserved intermediate models), controlled LLM fine-tuning trees, a link between weights and behavior, and a label-free tree-likeness score that serves as an empirical measure of confidence in the estimated phylogeny.

3 Framework

We first define our terms, notation, and statistical framework. A *phylogeny* $\mathcal{P}$ is a tree whose leaves are the observed models and whose internal nodes are latent common ancestors; a *clade* is the group of leaves descended from one such ancestor. We analyze single-root model families but observe only

their leaf models. We form a pairwise distance matrix and apply neighbor-joining (NJ; Saitou & Nei, 1987), a standard algorithm that builds a tree from pairwise distances, to estimate the phylogeny. NJ returns an unrooted tree, so we estimate the unrooted tree over the leaf models, with internal ancestors latent.

Phylogeny estimation. Let a model's weights θ be composed of layers θ_ℓ. To estimate $\widehat{\mathcal{P}}$: (1) choose a set of layers $\theta_{\mathcal{L}}$ and a distance metric d; (2) for each layer ℓ compute $D_{i,j}^{(\ell)} = d(\theta_i^{(\ell)}, \theta_j^{(\ell)})$ between models i, j; and (3) apply NJ to each selected layer's matrix to obtain a layer-level tree. We either analyze these layer-level trees separately or combine them into one tree, as described below.

Step (3) lets different layers yield different estimates of $\widehat{\mathcal{P}}$, which we use to assess each layer's lineage signal. Each internal edge of a tree splits the leaves into two groups, and each such split of the true tree corresponds to a clade. We compare estimated and true phylogenies using Robinson–Foulds (RF) distance (Robinson & Foulds, 1981), which counts disagreeing splits as false positives (FP) and false negatives (FN). Because NJ returns binary trees while our true trees may contain *polytomies* (nodes with more than two children), NJ must add splits to resolve them, and RF counts these as FPs even though they do not contradict the known lineage. We therefore focus on false negatives (FN), the true clades missing from the estimated tree, and report two summaries: *clade estimation*, $1 - \mathrm{FN}/|\text{true clades}|$, the fraction of true clades that appear in the estimated tree; and *full estimation*, the fraction of trees in which every true clade appears ($\mathrm{FN} = 0$). RF distances are reported in App. A. We score only trees with at least one clade that can be identified without the root, i.e., at least one internal edge that splits the leaves into two groups of at least two; other trees have no clade to estimate. Separately, we report the within-tree branch correlation r_{branch} between weight distance and a same-root-branch indicator (§4) to assess whether weight distance reflects lineage relationships.

Distance metric. We flatten each weight layer to a vector and use cosine distance (1 minus cosine similarity) throughout, for three reasons. It is bounded and scale-invariant across layers, so no single high-magnitude layer dominates the aggregate, unlike ℓ_1 or ℓ_2, the latter of which is used by Horwitz et al. (2025b). It is the prevailing choice in recent provenance and fingerprinting work (Zeng et al., 2024; Xu & Sheng, 2026; Zhu et al., 2025), keeping our distances comparable to that literature. Finally, cosine distance has a simple geometric interpretation: descendant models whose weight vectors remain similarly oriented in parameter space are considered closer. For LoRA-trained trees, we compare merged weights (base plus adapters) rather than the adapters themselves.

Per-layer estimated phylogenies. We treat each weight layer as an independent estimator of lineage, analogous to using individual genes to estimate biological phylogenies (Maddison, 1997). For a given training tree and layer, we compute the cosine distance matrix across its leaf models and apply NJ, producing one estimated phylogeny for that layer. The per-layer profile (App. B) scores each layer's tree and averages the scores across trees. We combine only *trained layers*, whose weights differ across models; a layer whose pairwise cosine distances are all zero carries no lineage information under this metric.

To combine layer estimates into one tree, we compute the Billera–Holmes–Vogtmann (BHV) Fréchet mean of the layer-level trees (Billera et al., 2001; Willis, 2019), the average tree in the space of phylogenetic trees; we call this the *all-weight estimate* when every trained layer is used. For analyses that require a single pairwise distance matrix (shared lineage, the tree-likeness score, the Atteson margin, and weight–behavior associations), we average the trained layers' distance matrices. Behavioral similarity is likewise the mean of prompt-paired output-embedding cosine similarities.

When estimation succeeds. We assess lineage signal at two levels. First, we test whether leaf models on the same root branch are closer in weight space than leaf models on different root branches (the shared-lineage analysis, §4). This measures pairwise lineage signal without estimating the full phylogeny.

Second, we ask when the full phylogeny can be estimated from the weight distances. The NJ method is exact when the distance matrix is *additive*, i.e., when the distances are exactly path lengths on some tree, a condition model-weight distances satisfy only approximately. Atteson's theorem makes the condition precise (Atteson, 1999). Let D_T denote the additive leaf-distance matrix of a binary tree T with positive edge lengths L_s, and let D be an observed distance matrix. If

$$\|D - D_T\|_\infty < \frac{1}{2} \min_s L_s,$$

then NJ applied to D estimates the topology of T.

Equivalently, define

$$m := \frac{\min_s L_s}{2\|D - D_T\|_\infty},$$

the *Atteson margin*. Then $m > 1$ is sufficient for exact estimation. This condition is conservative, as NJ may still estimate the correct topology when $m \leq 1$.

In our controlled experiments the true topology is known, but its edge lengths are not. We fit nonnegative edge lengths on that topology to obtain D_{fit} and report the *fitted Atteson margin* $\widetilde{m} = \min_s \widehat{L}_s/(2\|D - D_{\text{fit}}\|_\infty)$. Because D_{fit} is estimated from D and some true trees contain polytomies, $\widetilde{m}$ approximates rather than checks the Atteson condition. Because $\widetilde{m}$ also requires the true tree, we additionally use a label-free score computed from D alone, based on the classical four-point condition for tree metrics (Semple & Steel, 2003).

Tree-likeness score (A). We measure how far a distance matrix departs from additivity with the quartet δ-score of Holland et al. (2002). For four leaves i, j, k, l, the three pairwise-distance sums $d_{ij} + d_{kl}$, $d_{ik} + d_{jl}$, and $d_{il} + d_{jk}$, sorted $s_1 \geq s_2 \geq s_3$, give

$$\delta_q = \frac{s_1 - s_2}{s_1 - s_3} \in [0, 1],$$

which is 0 exactly when the four-point condition $s_1 = s_2$ holds (the additive, tree-like case; Semple & Steel, 2003) and 1 for the least tree-like quartet (the two *smallest* sums equal, the largest strictly greater). We report its complement, averaged over the Q quartets,

$$A = \frac{1}{Q}\sum_{q=1}^{Q}(1 - \delta_q) = \frac{1}{Q}\sum_{q=1}^{Q}\frac{s_2^{(q)} - s_3^{(q)}}{s_1^{(q)} - s_3^{(q)}} \in [0, 1],$$

which we call the *tree-likeness score*. $A = 1$ means the distances are perfectly additive. A high A shows that the distances are close to *some* tree, not that they match the true history. NJ returns the correct tree when the distances are close to the *true* tree's distances (the Atteson condition above). Distances can also be perfectly additive for the wrong tree. Because A requires no labels, we test empirically in §5 whether it and the fitted Atteson margin are associated with estimation accuracy.

4 Experiments

We structure our experiments around three questions: whether the model lineage or phylogeny can be estimated from weight changes; whether those changes predict held-out behavior; and how behavior-based phylogeny estimation differs from weight-based estimation.

Controlled and real training trees. To create a controlled experiment, we generate 50 random model training trees, each starting from a root model and ending in 4–10 leaf models (process detailed in App. A). We assign a random task/dataset pair to each edge and fine-tune each child model from its parent on its edge's pair. Four of the 50 trees have no clade that can be identified once the root is removed, so tree-estimation results use the remaining 46. The behavior analysis uses all 50. For LoRA, each child's adapter is merged into its parent's weights, so every LoRA leaf is a full set of weights (one variant, marked † in Table 2, instead compares the summed adapter updates; App. B). We then estimate a phylogeny from the leaf models and compare it to the known training tree, testing both shared lineage and phylogeny estimation.

We vary the root model, task types, and trained layers. Our primary evaluation uses an encoder–decoder model (Flan-T5-base, 250M) and a decoder-only model (Llama-3.2-1B), fine-tuning on QA, classification, translation, and summarization. For Flan-T5, we run full fine-tuning and four LoRA (Hu et al., 2022) scopes: key only; query, key, and value (q/k/v); q/k/v plus the attention output (q/k/v/o); and all attention and feed-forward projections. For Llama-3.2-1B, we run full fine-tuning and q/k/v LoRA at ranks 8 and 64. For each base model, we use the same task instructions. We also validate our method on two documented model families from Hugging Face, Mistral-7B and Llama-2-7B. We summarize them in the main text and show both in App. D.

Shared lineage. MoTHer (Horwitz et al., 2025b) assembles pairwise relatedness and direction scores into a directed model tree, while Neural Phylogeny (Yu & Wang, 2025) identifies parent–child relationships. Our shared-lineage analysis instead works at the level of root branches among leaf models. Within each training tree, two leaves are on the "same root branch" when their first ancestor below the root is the same; all other leaf pairs are "cross branch." We test whether cross-branch distances are larger with a one-sided Mann–Whitney U test and report the corresponding rank-biserial effect $r_{\mathrm{rb}} = 2U/(n_{\mathrm{cross}}n_{\mathrm{same}}) - 1$, along with the point-biserial correlation r_{branch} between weight distance and a same-branch indicator. Across trees, we report the mean and sample SE of r_{rb}, and the Fisher z-average of r_{branch}[1].

Phylogeny estimation. Besides shared lineage, we test whether our method estimates the phylogeny. From §3, accurate estimation should be possible when the distance matrix is sufficiently additive, for which the tree-likeness score A is a practical, label-free proxy. For each tree we report FN, clade estimation, and full estimation ($\mathrm{FN} = 0$), relating them to the fitted Atteson margin $\widetilde{m}$ and A in §5. The per-layer trees also let us identify which layers are strong or weak carriers of lineage signal.

Predicting held-out behavior. As a secondary analysis, we test whether weight distance predicts behavioral similarity. Unlike tree estimation, this analysis uses every trained model, including internal ancestors (641 models across the 50 trees: 365 leaves and 276 internal nodes; App. A), because it asks how weights relate to behavior rather than whether latent ancestors can be inferred. For each task, we embed each output with a small sentence-transformer (`all-MiniLM-L6-v2`, 384-d), compute the cosine similarity between two models' responses to the same prompt, and average these similarities across prompts. Decoding differs by task: translation uses one greedy response per prompt, while HellaSwag, ARC-Challenge, MMLU, and TruthfulQA use three sampled responses per prompt (temperature 1), whose similarities are averaged over prompts and draws. Within each tree, we correlate weight distance with output similarity, with both quantities z-scored within the tree, and pool the tree-level correlations with a DerSimonian–Laird random-effects meta-analysis (DerSimonian & Laird, 1986), giving r_{DL}. Other generation details are reported in App. A.

Comparison to behavior-only estimation. We compare our weight-based phylogeny estimation with PhyloLM (Yax et al., 2025), which estimates model phylogenies from behavior alone and therefore does not require access to model weights. Because our two documented Hugging Face families have known structure, we can score behavior-only estimation against them. We run PhyloLM with its released prompt set and similarity measure on the same model and compare it with our weight-based estimate (details in App. C). Our controlled comparison runs PhyloLM on every tree in the Llama rank-8 LoRA, rank-64 LoRA, and full fine-tuning conditions and scores the same 46 trees. PhyloLM does not apply to the encoder–decoder Flan models.

5 Results

Shared lineage. We first test whether leaf models on the same root branch are closer in weight space than leaf models on different root branches. There are 26 eligible trees for this analysis, those with at least one same-branch pair and at least one cross-branch pair of leaves. Table 2 reports the Mann–Whitney rank-biserial effect and the within-tree correlation r_{branch} between cosine distance and a same-branch indicator. Every mean rank-biserial effect is positive and every r_{branch} is negative. Leaf models on the same root branch are consistently closer in weight space than leaf models on different root branches, implying that lineage is passed down.

Phylogeny estimation. Beyond shared lineage, the BHV Fréchet means estimate the full phylogeny (Table 2). In the controlled experiments, clade estimation for our method ranges from 92–100% and full estimation from 85–100%. The LoRA variants correctly estimate 92–99% of clades for Flan and 100% for Llama. Full fine-tuning estimates 100% of clades for both. On the same 46 trees, weights correctly estimate 100% of clades versus PhyloLM's 35% under rank-8 LoRA, 100% versus 41% at rank 64, and 100% versus 56% under full fine-tuning.

[1] For each tree we compute $z = \operatorname{arctanh}(r)$, average on the z scale, and back-transform with $r = \tanh(\bar{z})$.

Table 2: Shared lineage and tree estimation by training regime. Tree estimation (clade, full estimation, FN) uses 46 trees; shared lineage (rank-biserial, r_{branch}) uses the 26 trees with both same- and cross-branch leaf pairs. Trees are BHV Fréchet means of per-layer NJ trees; pairwise statistics use the mean trained-layer distance matrix. Full estimation: every true clade estimated (FN $= 0$). PhyloLM does not apply to Flan. †Compares summed adapter updates rather than merged weights (App. B). *117/117 true clades, 46/46 trees. ‡Separately trained conditions whose statistics coincide at display precision (App. A). Chance level for a random binary tree: 8.7% clade estimation, 3.2% full estimation (App. A). RF distances are in App. A.

Variant	Rank-bis. (SE)	r_{branch} (SE)	Weight clade (SE)	PhyloLM clade (SE)	Weight full estimation (SE)	Weight FN (SE)
Encoder–decoder: Flan-T5-base ($d = 768$)						
LoRA adapter scope						
LoRA: key only	0.52 (0.09)	−0.53 (0.07)	**92%** (3%)	n/a	**85%** (5%)	0.17 (0.06)
LoRA: q/k/v‡	0.56 (0.09)	−0.62 (0.07)	**99%** (1%)	n/a	**98%** (2%)	0.02 (0.02)
LoRA: q/k/v/o (all attention)‡	0.56 (0.09)	−0.62 (0.07)	**99%** (1%)	n/a	**98%** (2%)	0.02 (0.02)
LoRA: q/k/v/o + FFN projections†	1.00 (0.00)	−0.91 (0.02)	**96%** (2%)	n/a	**89%** (5%)	0.11 (0.05)
Full fine-tuning						
Full fine-tuning	0.69 (0.08)	−0.71 (0.06)	**100%***	n/a	**100%***	0.00 (0.00)
Decoder-only: Llama-3.2-1B ($d = 2048$)						
LoRA: q/k/v (r=8)	0.63 (0.08)	−0.64 (0.07)	**100%***	35% (5%)	**100%***	0.00 (0.00)
LoRA: q/k/v (r=64)	0.65 (0.08)	−0.67 (0.07)	**100%***	41% (5%)	**100%***	0.00 (0.00)
Full fine-tuning	0.64 (0.08)	−0.66 (0.07)	**100%***	56% (5%)	**100%***	0.00 (0.00)

Treating each layer as a gene-like estimator reveals that a single weight matrix can estimate the phylogeny as well as the all-weight estimate under full fine-tuning. On the same 46 trees, each of the 12 Flan encoder self-attention key matrices on its own (0.24% of the weights) correctly estimates 100% of clades with 100% full estimation, as do 10 of the 16 Llama key matrices (0.085% each), matching the all-weight estimate (App. B; Table 6 reports the corresponding layer subsets). In Flan, every individual encoder attention and feed-forward matrix does the same. The signal also survives low-rank training: q/k/v LoRA adapters that train only 0.54% of Flan's parameters give 99% clade estimation and 98% full estimation, and rank-8 q/k/v LoRA, which trains 0.10% of Llama's parameters, gives 100% and 100% (Table 2). The per-layer profile in App. B shows which layers carry the strongest signal, and this depends on the architecture.

Finally, we ask whether the tree likeness scores of §3 signal when estimation succeeds. Figure 2 plots clade estimation against the fitted Atteson margin $\widetilde{m}$ and the tree-likeness score A for 228 weight-based estimators on the same 46 trees (the seven all-weight conditions, 17 attention-layer subsets, and 204 single query, key, or value matrices) and the three PhyloLM conditions. In our experiments, both scores signal success. Clade estimation rises steeply as the distances become more tree-like and then levels off near 100%. We fit a fractional-logit model to clade estimation p_c, the fraction of true clades correctly estimated, with $\text{logit}(p_c) = \beta_0 + \beta_1 x$. The odds of correctly estimating a clade grow 4.5-fold per 0.10 increase in A (95% CI 3.9–5.8, paired bootstrap over trees) and 15.2-fold per tenfold increase in $\widetilde{m}$ (95% CI 8.5–68.0). Every weight-based estimator in Figure 2 has $A \geq 0.82$ and correctly estimates at least 84% of clades, and the all-weight conditions and layer subsets, all of which have $\widetilde{m} > 1$, estimate at least 90%. In contrast, PhyloLM's behavioral distances have $A = 0.56$–0.60 and $\widetilde{m} < 0.25$, and PhyloLM correctly estimates 35–56% of clades. Much of the range in Figure 2 comes from this gap between weight- and behavior-based distances, but the association also holds among weight-based estimators alone (tree-level fit, odds ratio 5.7 per 0.10, 95% CI 4.0–9.3). Because A needs no labels, it offers an empirical guide to how far to trust an estimated phylogeny, although a high score shows tree-like distances rather than a correct history.

Predicting held-out behavior. While estimating phylogeny from weights is useful for safety and provenance, it is even more useful if similarity in weights signals similarity in behavior. Inspired by Cloud et al. (2025)'s subliminal-learning experiment, we test whether models closer in weight space produce more similar outputs on *held-out* tasks, using translation (on a separate set of trees that never trained on it) and four test tasks that are not in the fine-tuning data (HellaSwag, ARC-Challenge, MMLU, and TruthfulQA). Because r_{DL} correlates weight *distance* with output *similarity*, a negative value means that models closer in weight space give more similar outputs, and a positive value means the reverse. Table 3 shows a significant negative correlation on every held-out task in Llama. Some

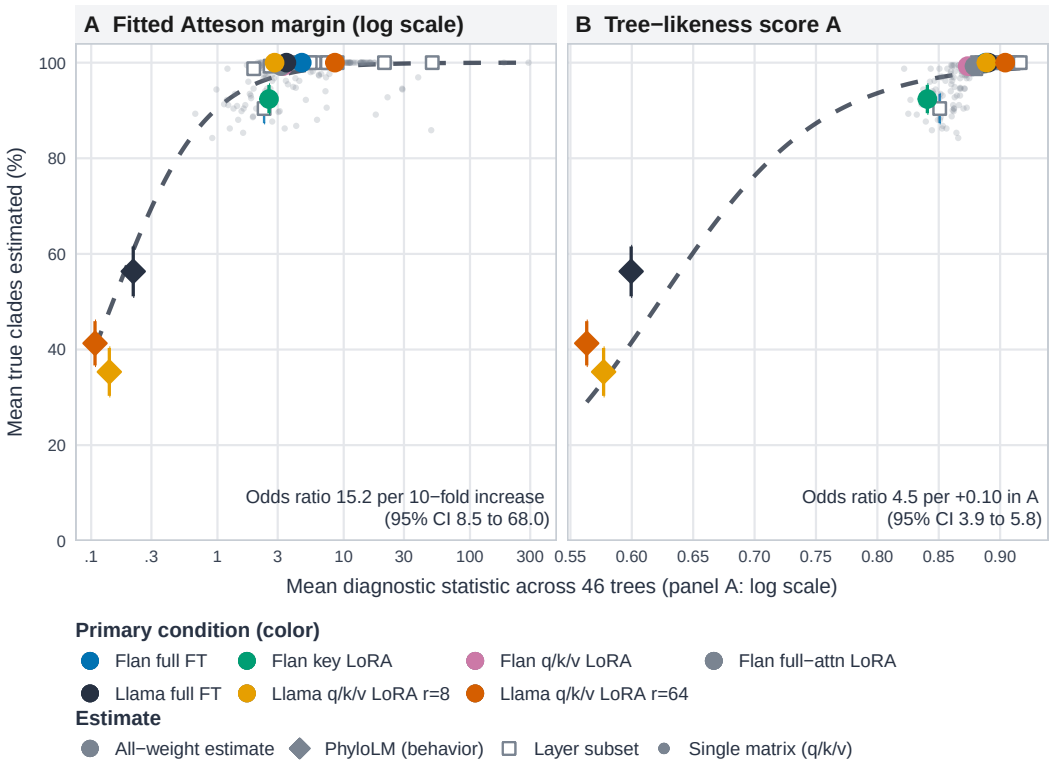


Figure 2: Mean clade estimation (% of true clades correctly estimated) against **(A)** the fitted Atteson margin $\widetilde{m}$ on a logarithmic x-axis and **(B)** the tree-likeness score A. Each mark averages over the 46 trees used in Table 2: seven all-weight conditions, 17 attention-layer subsets, 204 single q, k, or v matrices, and three PhyloLM conditions. Vertical intervals on all-weight, PhyloLM, and subset points are ± 1 SE. Dashed curves are fractional-logit fits. Odds ratios have 95% paired-bootstrap CIs over the 46 trees.

Table 3: Correlation between weight distance and output similarity under full fine-tuning: random-effects pooled r_{DL} with 95% CI over 50 trees, using all trained models. Negative values mean that models closer in weight space give more similar outputs. Empty responses are kept as outputs (App. F); many Flan outputs are fine-tuning labels, and heterogeneity is detailed in App. E.

r_{DL} [95% CI]	Translation		HellaSwag	ARC-C	MMLU	TruthfulQA
Model	trained	held-out	held-out	held-out	held-out	held-out
Llama-3.2-1B	-0.20 $[-.26, -.13]$	-0.23 $[-.29, -.17]$	-0.17 $[-.24, -.10]$	-0.20 $[-.26, -.13]$	-0.19 $[-.26, -.12]$	-0.22 $[-.27, -.16]$
Flan-T5-base	-0.10 $[-.16, -.04]$	$+0.04$ $[-.01, +.09]$	$+0.09$ $[+.05, +.14]$	-0.14 $[-.19, -.09]$	-0.16 $[-.20, -.11]$	-0.17 $[-.22, -.12]$

responses are empty because the model ends its output immediately. We keep these as valid (empty) outputs. In the Llama full fine-tuning trees, 39% of HellaSwag leaf outputs were empty, versus at most 0.02% for the other held-out tasks. For HellaSwag, keeping empty responses gives $r_{\mathrm{DL}} = -0.17$ $[-0.24, -0.10]$, whereas excluding them strengthens the estimate to -0.26 $[-0.34, -0.19]$ and leaves the other tasks essentially unchanged (App. F). Thus, models closer in weight space behave more similarly on held-out tasks. The effects are modest ($|r_{\mathrm{DL}}| \approx 0.2$) but significant. Flan has inconsistent results. Its HellaSwag correlation is significantly positive, and its held-out translation correlation is near zero. Part of this reflects degenerate outputs after repeated Flan fine-tuning: 15–30% of Flan outputs are only a label from a fine-tuning task, such as `entailment`. Flan's output similarity partly reflects shared collapse onto fine-tuning labels rather than shared task behavior (App. E).

Weight- versus behavior-based estimation. When weights are unavailable or architectures differ, lineage must be inferred from behavior alone. Table 2 already compares our estimates with the

behavior-only PhyloLM (Yax et al., 2025) on the controlled trees. We also score both approaches against two documented Hugging Face families, Mistral-7B and Llama-2-7B. Our weight-based estimates contain every documented clade, and so do PhyloLM's, although the weight-based distances are more tree-like ($A = 0.86$ vs. 0.68 for Mistral and 0.82 vs. 0.64 for Llama-2; App. D).

The Mistral forks are large RLHF or direct preference optimization (DPO) updates that shift behavior substantially. Our controlled fine-tunings are smaller, and there PhyloLM correctly estimates only 35–56% of clades versus 100% for weights (Table 2). Behavior thus carries some, but not all, of the lineage signal in the weights, and weight-based estimation does not depend on a choice of prompts. Overall, weight-based estimation matches behavior-based estimation on these real families and substantially outperforms it in the controlled experiments.

6 Discussion

We show that model family phylogeny can be estimated from weights alone, and that a single weight matrix, such as one attention key matrix (0.1–0.2% of the weights), estimates training trees as well as all trained layers together. We also find that clade estimation works across encoder–decoder and decoder-only models and survives LoRA training at low rank ($r = 8$). Across our controlled conditions, full estimation ($\mathrm{FN} = 0$) ranges from 85–100%. The label-free tree-likeness score A is associated with higher clade estimation in our experiments. In addition to estimating phylogeny and shared lineage from weights, we show that models closer in weight space tend to have more similar outputs on held-out tasks. The effect depends on the task and ranges from $r_{\mathrm{DL}} = -0.23$ to -0.17 across the held-out Llama tasks. Finally, weight- and behavior-based estimation both correctly estimate the documented Mistral-7B and Llama-2-7B families, while weight-based estimation is substantially stronger in all three controlled Llama conditions and requires no prompts.

Implications. Our results suggest three uses beyond estimating controlled trees. First, weight-based phylogenies can fingerprint models and help chart the largely undocumented atlas of fine-tuned models on public hubs (Horwitz et al., 2025a), at least within a shared architecture. Second, the tree-likeness score offers a label-free measure of empirical confidence. In our experiments, clade estimation rose sharply with A (odds ratio 4.5 per 0.10), so a practitioner without a true tree can use A as a heuristic for how far to trust an estimated lineage, which brings a classical idea from phylogenetics to model auditing. Third, a tiny, architecture-dependent fraction of the weights suffices. A single weight matrix, such as one attention key matrix (0.1–0.2% of the weights), matches the all-weight estimate, which makes lineage estimates affordable even for large models.

Limitations. One limitation is that our methods for phylogeny estimation using weights currently only work when the underlying architecture is identical, i.e., the weight space is the same. This means that Llama cannot directly be compared to Mistral, for example. Moreover, unlike Horwitz et al. (2025b), our method treats every observed model as a leaf. Another issue is that our controlled training tree generation results from a fairly simple process that may not result in the most realistic training trees. We attempt to mitigate that with two documented real-world families. Future directions include better relating models across different architectures such as with CKA and further exploring the mechanism by which lineage is inherited.

In summary, we hope that our work continues to illuminate the need to consider models holistically rather than only as a collection of benchmark scores, and shows how analyzing weight space can shed light on model behavior, especially traits passed down through fine-tuning.

Reproducibility statement

All controlled experiments follow seeded, fully specified procedures. Appendix A describes the tree-generation process (Poisson branching, node budget, depth limit, and acceptance rule), the 36 task/dataset pairs and their assignment, the training configuration of both base models (pinned Hugging Face revisions, learning rates, batching, schedules, precision, sequence limits, and LoRA hyperparameters; Table 5), and the behavioral test-task protocol (prompt counts, decoding settings, generation seeds, and the pinned sentence-embedding model). Appendix B defines every layer subset and LoRA variant (Tables 6 and 7); Appendix C gives the PhyloLM configuration and pinned commit used for the behavior-only comparison; and Appendix D lists the Hugging Face repository

of every checkpoint in the two documented families (Table 8). The scoring and statistical procedures (RF, FN, clade estimation, and full estimation, the BHV Fréchet mean, the tree-likeness score and fitted Atteson margin, and jackknife variances pooled by DerSimonian–Laird) are specified in Sections 3–4 and Appendix E. Code for tree generation, training, distance computation, reconstruction, scoring, and analysis, together with the tree manifests, true trees, and per-tree distance matrices needed to regenerate every table and figure, will be released publicly upon publication.

AI USE STATEMENT

Generative AI assistants were used to write and debug software across the experimental pipeline (data processing, weight-distance computation, tree estimation and scoring, statistical analysis, figures, and tables) and to review and edit the manuscript, including checking consistency between numerical results and prose, improving readability, searching and summarizing relevant literature, and formatting citations. The research questions, experimental design, hypotheses, conceptual framework, and scientific interpretations are the authors' own. The authors reviewed and tested AI-assisted code and independently verified all results, figures, numerical claims, citations, and manuscript text. The authors take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Annual Meeting of the Association for Computational Linguistics*, 2021.

Kevin Atteson. The performance of neighbor-joining methods of phylogenetic reconstruction. *Algorithmica*, 25(2):251–278, 1999.

Louis J. Billera, Susan P. Holmes, and Karen Vogtmann. Geometry of the space of phylogenetic trees. *Advances in Applied Mathematics*, 27(4):733–767, 2001. doi: 10.1006/aama.2001.0759.

Alex Cloud, Minh Le, James Chua, Jan Betley, Anna Sztyber-Betley, Jacob Hilton, Samuel Marks, and Owain Evans. Subliminal learning: Language models transmit behavioral traits via hidden signals in data. *arXiv preprint arXiv:2507.14805*, 2025.

Rebecca DerSimonian and Nan Laird. Meta-analysis in clinical trials. *Controlled Clinical Trials*, 7 (3):177–188, 1986. ISSN 0197-2456. doi: 10.1016/0197-2456(86)90046-2.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carlin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, 2020.

Shannon K. Gallagher, Swati Rallapalli, Tyler Brooks, Chuck Loughin, Michele Sezgin, and Ronald Yurko. Analysis and explainability of LLMs via evolutionary methods, 2026. URL https://arxiv.org/abs/2605.02930.

Julian P. T. Higgins, Simon G. Thompson, Jonathan J. Deeks, and Douglas G. Altman. Measuring inconsistency in meta-analyses. *BMJ*, 327(7414):557–560, 2003. doi: 10.1136/bmj.327.7414.557.

Barbara R. Holland, Katharina T. Huber, Andreas Dress, and Vincent Moulton. Delta plots: a tool for analyzing phylogenetic distance data. *Molecular Biology and Evolution*, 19(12):2051–2059, 2002. doi: 10.1093/oxfordjournals.molbev.a004030.

Eliahu Horwitz, Jonathan Kahana, and Yedid Hoshen. Recovering the pre-fine-tuning weights of generative models. In *International Conference on Machine Learning (ICML)*, 2024. arXiv:2402.10208; "Spectral DeTuning".

Eliahu Horwitz, Nitzan Kurer, Jonathan Kahana, Liel Amar, and Yedid Hoshen. We should chart an atlas of all the world's models, 2025a. URL https://arxiv.org/abs/2503.10633.

Eliahu Horwitz, Asaf Shul, and Yedid Hoshen. Unsupervised model tree heritage recovery. In *International Conference on Learning Representations*, 2025b. OpenReview: QVj3kUvdvl.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2023.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, 2019.

Wayne P. Maddison. Gene trees in species trees. *Systematic Biology*, 46(3):523–536, 1997.

Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Ivica Nikolic, Teodora Baluta, and Prateek Saxena. Model provenance testing for large language models. *Advances in Neural Information Processing Systems*, 38:34126–34153, 2026.

David F Robinson and Leslie R Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53(1-2):131–147, 1981.

Naruya Saitou and Masatoshi Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425, 1987.

Charles Semple and Mike Steel. *Phylogenetics*, volume 24 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, 2003.

Suqing Wang, Ziyang Ma, Xinyi Li, and Zuchao Li. Ghost in the transformer: Detecting model reuse with invariant spectral signatures. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026. Oral; arXiv:2511.06390.

Amy Willis. Confidence sets for phylogenetic trees. *Journal of the American Statistical Association*, 114(525):235–244, 2019. doi: 10.1080/01621459.2017.1395342.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carlin, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, 2022.

Zehao Wu, Yanjie Zhao, and Haoyu Wang. TensorGuard: Gradient-based model fingerprinting for LLM similarity detection and family classification. In *2025 40th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 445–456, 2025. doi: 10.1109/ASE63991.2025.00044.

Zhaomin Wu, Haodong Zhao, Ziyang Wang, Jizhou Guo, Qian Wang, and Bingsheng He. LLM DNA: Tracing model evolution via functional representations. In *International Conference on Learning Representations*, 2026. Oral presentation. OpenReview: UIxHaAqFqQ.

Zhenyu Xu and Victor S. Sheng. A behavioral fingerprint for large language models: Provenance tracking via refusal vectors. *arXiv preprint arXiv:2602.09434*, 2026.

Chenghao Yang, Yuning Zhang, Zhoufutu Wen, Tao Gong, Jiaheng Liu, Qi Chu, and Nenghai Yu. When agents look the same: Quantifying distillation-induced similarity in tool-use behaviors. *arXiv preprint arXiv:2604.21255*, 2026. Accepted to ACL 2026 Main Conference.

Nicolas Yax, Pierre-Yves Oudeyer, and Stefano Palminteri. PhyloLM: Inferring the phylogeny of large language models and predicting their performances in benchmarks. In *International Conference on Learning Representations*, 2025.

Runpeng Yu and Xinchao Wang. Neural phylogeny: Fine-tuning relationship detection among neural networks. In *International Conference on Learning Representations*, 2025.

Boyi Zeng, Lizheng Wang, Yuncong Hu, Yi Xu, Chenghu Zhou, Xinbing Wang, Yu Yu, and Zhouhan Lin. HuRef: HUman-REadable fingerprint for large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2312.04828.

Jie Zhang, Dongrui Liu, Chen Qian, Linfeng Zhang, Yong Liu, Yu Qiao, and Jing Shao. REEF: Representation encoding fingerprints for large language models. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.14273.

Sally Zhu, Ahmed Ahmed, Rohith Kuditipudi, and Percy Liang. Independence tests for language models. In *International Conference on Machine Learning*, 2025.

A Experimental Details

Tree Generation

Training trees are generated via a Poisson branching process as follows:

1. The root (the untrained base model) draws $k \sim \mathrm{Poisson}(\lambda = 1.5)$ children, with a minimum of 1. We instantiate the same accepted topologies and task/data assignments with either Flan-T5-base or Llama-3.2-1B.
2. Each queued node below the depth limit draws $k \sim \mathrm{Poisson}(\lambda = 1.5)$ children while node budget remains. Nodes that draw $k = 0$ naturally become leaves; no separate pruning step is applied.
3. Expansion is breadth-first, with a budget of at most $n_{\mathrm{nodes}} = 14$ training nodes (not counting the root) and maximum depth 4. Reaching depth 4 on one branch prevents only those depth-4 nodes from expanding; shallower nodes already in the queue can still produce children while budget remains. Growth ends when the queue is empty or the node budget is exhausted. Candidate trees with fewer than four leaves are rejected, and seeded draws continue until 50 trees are accepted.
4. Each non-root node is assigned one of 36 task/dataset pairs (9 summarization, 10 classification, 8 QA, and 9 translation). The pairs are sampled without replacement within a tree, so datasets do not repeat but task families can. Topology generation and task/dataset assignment are seeded separately.
5. Tree estimation uses only accepted trees with at least one clade that can be identified without the root, i.e., an internal edge that splits the leaves into two groups of at least two. Four accepted trees have four leaves but, once unrooted, reduce to stars with only one-versus-three splits; this leaves 46 trees for tree estimation.

This procedure produces trees with 4–10 leaves and variable depth and branching structure. Across the 46 trees used for tree estimation, there are 609 trained nodes (349 leaves and 260 internal nodes), in addition to 46 untrained roots. The true topology is fully known by construction, enabling exact evaluation.

Chance baseline. The chance level quoted in the caption of Table 2 is an exact expected value rather than a simulation. There are $(2n-5)!!$ unrooted binary topologies on n labeled leaves (Semple & Steel, 2003). A fixed split whose sides contain a and $b = n-a$ leaves is present in $(2a-3)!!\,(2b-3)!!$ of them, because contracting the split edge leaves a rooted binary tree on each side and there are $(2k-3)!!$ rooted binary trees on k leaves. A uniformly random binary tree therefore contains the split with probability $(2a-3)!!\,(2b-3)!!/(2n-5)!!$, one third for a four-leaf quartet, and 0.5% for a balanced $5\,|\,5$ split of ten leaves. Averaging this probability over the true clades of each of the 46 trees, and then over trees exactly as clade estimation is averaged in Table 2, gives an expected chance clade estimation of 8.7% (SE 1.1 across trees). A random tree contains every clade of a true tree T (full estimation) only if it refines T. A multifurcating T has $\prod_v (2d_v-5)!!$ binary refinements, the product running over its internal vertices v of degree d_v, so the chance full estimation is the mean over trees of $\prod_v (2d_v-5)!!/(2n-5)!!$, which is 3.2%.

Experiment structure. All tree-estimation comparisons use the same 46 tree topologies, with node identities aligned across conditions. In the standard Flan and Llama cohorts, every aligned node also retains the same task/dataset label across root models and fine-tuning interventions. The matched no-translation cohorts preserve those topologies and node identities, but replace translation assignments with unused classification, QA, or summarization datasets. Thus, comparisons are paired by tree and, except for the intended translation-removal intervention, by node label.

Different analyses use fixed eligible subsets of these shared trees. Tree estimation uses the common 46 trees with at least one clade that can be identified without the root (excluding topology IDs 015, 020, 037, and 047), while the shared-lineage analysis uses the common 26 trees with both same- and cross-branch leaf pairs. PhyloLM produces a behavioral tree for every tree in each matched Llama condition, and its summaries and the matched weight values in Table 2 use the same 46 trees. The Flan q/k/v and q/k/v/o LoRA cohorts are trained and analyzed separately (separate adapters, checkpoints, and distance matrices). Their all-weight (BHV) trees differ on 8 of the 46 scored trees, and the NJ trees on their averaged distance matrices differ on 5 of the 50 trees. In every case the differing splits only resolve polytomies in the true tree: both conditions miss no true clade on those trees and add the same number of extra splits, so their Table 2 rows coincide at display precision. The Flan and Llama per-layer profiles in Fig. 3 likewise use their matched 46-tree full fine-tuning cohorts.

Robinson–Foulds distances. Table 4 reports the RF distance and FN for each condition in Table 2. Because RF also counts the splits that NJ adds to resolve polytomies in the true tree, it stays near 2 even when every true clade is estimated (FN $= 0$).

Table 4: Robinson–Foulds (RF) distance and false negatives (FN) on the 46 trees, mean (SE). PhyloLM does not apply to Flan.

Model	Variant	Weight RF	Weight FN	PhyloLM RF	PhyloLM FN
Flan	LoRA: key only	2.39 (0.25)	0.17 (0.06)	n/a	n/a
	LoRA: q/k/v	2.09 (0.21)	0.02 (0.02)	n/a	n/a
	LoRA: q/k/v/o	2.09 (0.21)	0.02 (0.02)	n/a	n/a
	LoRA: q/k/v/o + FFN†	2.26 (0.21)	0.11 (0.05)	n/a	n/a
	Full fine-tuning	2.04 (0.19)	0.00 (0.00)	n/a	n/a
Llama	LoRA: q/k/v (r=8)	2.04 (0.19)	0.00 (0.00)	5.52 (0.37)	1.74 (0.16)
	LoRA: q/k/v (r=64)	2.04 (0.19)	0.00 (0.00)	5.09 (0.34)	1.52 (0.14)
	Full fine-tuning	1.83 (0.17)	0.00 (0.00)	4.26 (0.36)	1.11 (0.14)

TRAINING CONFIGURATION

Table 5 summarizes the audited configurations for `google/flan-t5-base`[2] and the base, non-Instruct `meta-llama/Llama-3.2-1B`[3]. Each child is initialized from its parent and trained under the configuration for its cohort.

[2]Pinned revision: `7bcac572ce56db69c1ea7c8af255c5d7c9672fc2`.
[3]Pinned revision: `4e20de362430cd3b72f300e6b0f18e50e7166e08`.

Table 5: Training configuration for the controlled Flan and Llama cohorts.

Setting	Flan-T5-base	Llama-3.2-1B
Objective	Sequence-to-sequence	Completion-only causal language modeling
Training length	2,000 steps; no early stopping	2,000 steps; no early stopping
Learning rate	3×10^{-4} (full FT and LoRA)	2×10^{-5} (full FT); 3×10^{-4} (LoRA)
Batching	Batch size 8; gradient accumulation 4; effective batch 32	Batch size 8; gradient accumulation 4; effective batch 32
Schedule	200 warmup steps; weight decay 0.01	200 warmup steps; weight decay 0.01
Precision	bfloat16	bfloat16
Sequence limits	512 input tokens; 128 target tokens	1,024 tokens total for prompt plus completion
Padding	Right	Right
LoRA	$r = 8$, $\alpha = 16$, dropout 0.05; module scopes in Table 7	q/k/v; $(r, \alpha) = (8, 16)$ or $(64, 128)$; dropout 0.05
Data sample	At most 10,000 train and 1,000 evaluation examples per dataset	At most 10,000 train and 1,000 evaluation examples per dataset

Training and dataset subsampling are seeded and held fixed across matched conditions and across nodes assigned the same dataset. Topology generation and task/dataset assignment use separate seed streams, as described above.

Behavioral Test Tasks and Analysis Population

Within each behavioral cohort, the primary analysis uses all 641 trained, non-root checkpoints across 50 trees: 276 trained internal nodes and 365 leaves. Each internal node is an actual checkpoint saved during training, not a latent ancestor inferred from its descendants; we evaluate its weights and outputs directly, just as we do for a leaf. For each within-tree checkpoint pair, weight distance is the mean trained-layer cosine distance described in §3. Translation uses 200 frozen prompts with one deterministic greedy completion per checkpoint (up to 128 new tokens for Flan and 64 for Llama). HellaSwag, ARC-Challenge, MMLU, and TruthfulQA each use 100 frozen prompts and three sampled completions per prompt, with a distinct generation seed for each draw (temperature 1, $p = 1$; up to 64 new tokens). Dolly, used only in the empty-response sensitivity analysis (App. F), uses 30 prompts, three draws, and up to 96 new tokens.

Outputs are embedded with `all-MiniLM-L6-v2`[4]. We average prompt-paired cosine similarity (and prompt-by-draw similarity for the three-draw tasks) before correlating it with weight distance within each tree. Tree-level correlations are then pooled across trees as described in App. E.

B Layer Variant Definitions

Two kinds of variants appear in the paper. *Layer subsets* (Table 6) restrict the all-weight distance to a chosen set of layers, testing which layers carry the lineage signal. For each subset, we average the selected per-layer cosine-distance matrices before NJ. "SA.k" denotes the self-attention key projection, and "pre-attention LayerNorm" denotes `layer.0.layer_norm` in Flan-T5-base and `input_layernorm` in Llama-3.2-1B. Counts and percentages are relative to the 282 Flan layers or 146 Llama layers in the analysis. The `low_signal` control contains post-attention, cross-attention, and post-FFN LayerNorms for Flan and post-attention LayerNorms for Llama.

LoRA variants (Table 7) instead change which modules are fine-tuned. All use dropout 0.05; the Flan variants and Llama rank-8 variant use $\alpha = 16$, while the Llama rank-64 variant uses $\alpha = 128$. Their controlled-tree analysis merges the full root-to-node adapter chain into the base weights, so distances are computed between full sets of weights. The all-projections row (marked † in Table 2) instead compares the summed root-to-node adapter updates without the shared base weights. It is therefore excluded from the Figure 2 fit. In Table 7, q, k, v, o denote the attention projections; $\mathrm{wi}_{0,1}$ and wo denote the two gated FFN inputs and the FFN output.

[4]Pinned revision: `1110a243fdf4706b3f48f1d95db1a4f5529b4d41`.

Table 6: Layer-subset estimation under full fine-tuning on the matched 46-tree cohorts (cosine). Values are percentages reported as mean (SE); full estimation is the percentage of trees in which every true clade is estimated.

Model	Variant	Layers included	Count	% layers	Clade (SE)	Full est. (SE)
Flan	`full`	all layers	282	100.0	100.0 (0.0)	100.0 (0.0)
	`high_signal`	SA.k + pre-attn LN	48	17.0	100.0 (0.0)	100.0 (0.0)
	`sak`	SA.k only	24	8.5	100.0 (0.0)	100.0 (0.0)
	`encoder_h`	encoder SA.k + pre-attn LN	24	8.5	100.0 (0.0)	100.0 (0.0)
	`enc_sak`	encoder SA.k only	12	4.3	100.0 (0.0)	100.0 (0.0)
	`low_signal`	LayerNorm control	36	12.8	100.0 (0.0)	100.0 (0.0)
Llama	`full`	all layers	146	100.0	100.0 (0.0)	100.0 (0.0)
	`high_signal`	SA.k + pre-attn LN	32	21.9	100.0 (0.0)	100.0 (0.0)
	`sak`	SA.k only	16	11.0	100.0 (0.0)	100.0 (0.0)
	`low_signal`	LayerNorm control	16	11.0	53.3 (5.0)	21.7 (6.1)

The zero SEs are exact rather than rounded: every Flan subset and each of the three signal-bearing Llama subsets estimates every true clade in all 46 trees. In Flan, even the LayerNorm control does so, consistent with the per-layer profile below, in which every encoder layer in a transformer block carries nearly the full signal. In Llama, the LayerNorm control does not (53% of clades), showing that the ceiling is not an artifact of the scoring.

Table 7: LoRA configurations used in the fixed-2,000-step cohorts. Percentages compare trainable LoRA parameters with the analyzed base-model weights.

Model	Variant	Target modules	r	α	Trainable params	% weights
Flan	`lora_k`	k	8	16	442,368	0.18
	`lora_qkv`	q, k, v	8	16	1,327,104	0.54
	`lora_qkvo`	q, k, v, o	8	16	1,769,472	0.71
	`lora_all_proj`	$q, k, v, o, \mathrm{wi}_{0,1}, \mathrm{wo}$	8	16	3,391,488	1.37
Llama	`lora_qkv`	q, k, v	8	16	1,179,648	0.10
	`lora_qkv`	q, k, v	64	128	9,437,184	0.76

Per-layer Estimation

Treating each layer as an independent estimator, Fig. 3 reports the percentage of true clades each layer's tree misses. On the Flan trees the signal is diffuse across individual layers. Every encoder attention and feed-forward matrix, including each self-attention key matrix, estimates every true clade in all 46 trees, while individual decoder layers miss up to 19% of clades (decoder self-attention keys, 9–16%). In Llama, 10 of the 16 self-attention key matrices likewise estimate every clade in all 46 trees, and the other trained projections miss about 1% on average. This is consistent with Table 6, where averaging only the 12 encoder self-attention key matrices matches the all-weight estimate. LayerNorm layers are the weakest carriers of lineage signal in both architectures.

C Behavior-Only (PhyloLM) Details

PhyloLM reproduction. For the behavior-only comparison on the documented Mistral-7B (Fig. 4) and Llama-2-7B (Fig. 5) families, we reproduce PhyloLM (Yax et al., 2025) using its released `math` prompt set (its "gene bank")[5] and its published Nei-similarity genetic distance. We sample $G = 128$ genes and, for each, draw $N = 32$ raw completions of 4 tokens per model (temperature 1, no chat template, matching PhyloLM's design), take the first 4 characters as the allele, form the per-gene allele populations, compute the Nei similarity S between every pair of the ten checkpoints, convert it to $-\log \max(S, 10^{-3})$, and build the tree by neighbor-joining. The resulting tree is scored against the same documented tree, with the same scoring used for the weight-based estimates. The Llama-2-7B family uses the identical configuration with two loading changes. Four of its checkpoints are stored in fp32 and do not fit on a 24 GB GPU, so we load them in fp16, the precision of the other checkpoints. Vicuna-7B-v1.5 is distributed only as PyTorch `.bin` files, which our pinned software stack will not deserialize, so we load an exact safetensors copy (all 323 tensors verified identical).

[5] Pinned PhyloLM revision: `8c70edf062a0adce2a3e6c8c79cd23a645fd0905`.

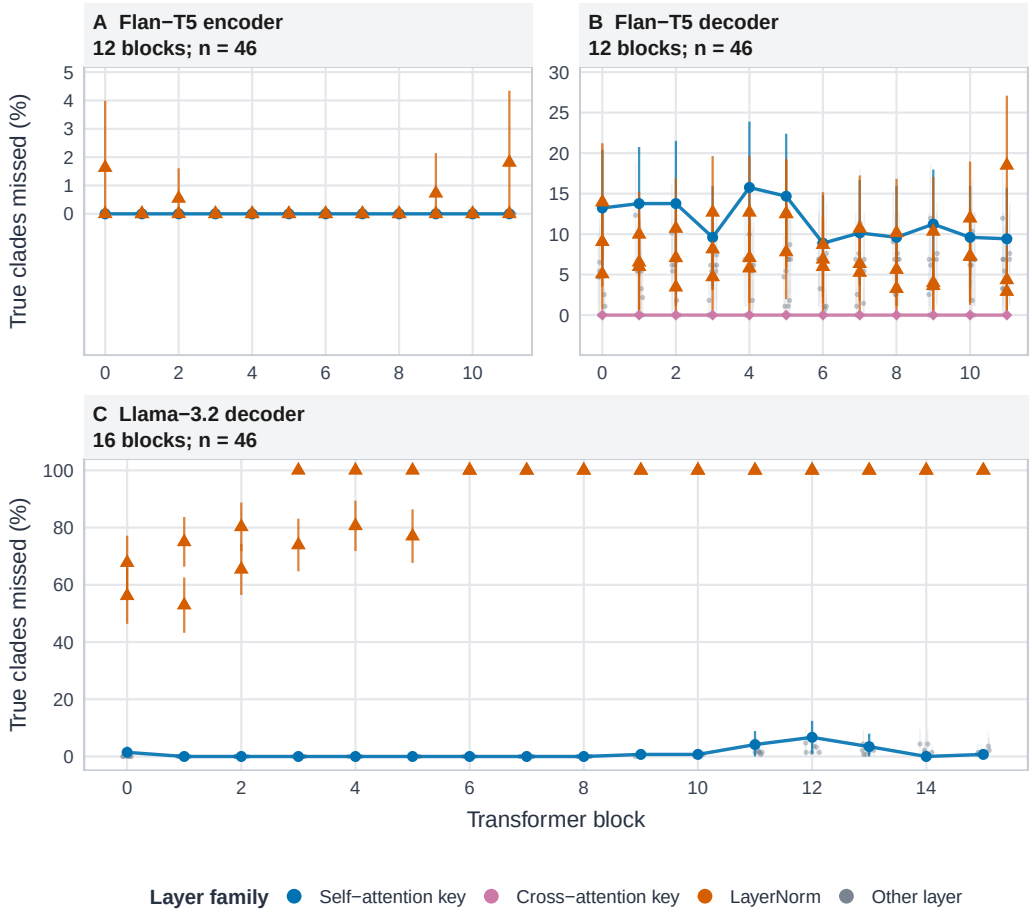


Figure 3: Block-aligned per-layer clades missed (% of true clades missed; mean ± 2 SE, cosine, full fine-tuning) for the encoder and decoder of Flan-T5-base (d=768, $n = 46$ trees) and the decoder of the matched Llama-3.2-1B cohort (d=2048, $n = 46$). Each transformer block occupies one x-position; gray points show the remaining layers without connecting unrelated layer types. In Llama, 24 RMSNorm layers never change during bfloat16 training, so all of their pairwise distances are zero; they carry no lineage information and do not affect any combined estimate. Twenty-three of them are the LayerNorm triangles fixed at 100% missed in blocks 3–15, plotted at 100% by construction; the 24th is the final norm. Layers outside the transformer blocks are omitted: for Flan, the two block-0 relative-attention-bias layers (32.8% encoder; 55.4% decoder), the shared embedding (0%), the encoder and decoder final LayerNorms (12.0% and 31.2%), and the output head (44.6%); for Llama, the embedding (31.5%) and the final norm.

The controlled comparison uses the same PhyloLM configuration ($G = 128$, $N = 32$, four-token continuations, temperature 1) with seeded sampling on all 50 trees in the Llama rank-8 LoRA, rank-64 LoRA, and full fine-tuning conditions. For LoRA leaves, we apply the full root-to-leaf adapter chain and query the resulting model; full fine-tuning leaves are loaded directly. Each condition contains generated populations for all 365 leaves across the 50 trees, and all generated alleles are nonempty. The four trees with no clade to estimate are not scored, leaving the same 46 trees in every row of Table 2.

D Documented Hugging Face Families

The paper evaluates two documented Hugging Face families (Table 8 lists every checkpoint). The ten-checkpoint Mistral-7B family contains three documented clades (Fig. 4), and weights and PhyloLM both estimate all three. In the ten-checkpoint Llama-2-7B family, the weight-based estimate (cosine BHV Fréchet mean) likewise contains the only documented clade, the seven chat-descended models (FN $= 0/1$); 95.2% of the per-layer trees contain this clade. PhyloLM also estimates this

clade (FN $= 0/1$), but its distances are much less tree-like than the weight distances ($A = 0.64$ vs. 0.82). The documented tree specifies no structure within the chat clade, so we claim one documented clade, not its full descendant history. Figure 5 shows the documented family alongside both estimates.

Table 8: Checkpoints in the two documented families, with their Hugging Face repositories.

Mistral-7B family		Llama-2-7B family	
Name	Repository	Name	Repository
Mistral-7B	`mistralai/Mistral-7B-v0.1`	Llama-2-7B (base)	`meta-llama/Llama-2-7b-hf`
Mistral-Instruct	`mistralai/Mistral-7B-Instruct-v0.1`	Llama-2-7B-chat	`meta-llama/Llama-2-7b-chat-hf`
Zephyr-SFT	`alignment-handbook/zephyr-7b-sft-full`	Nous-Hermes	`NousResearch/Nous-Hermes-llama-2-7b`
Zephyr-beta	`HuggingFaceH4/zephyr-7b-beta`	Vicuna-7B-v1.5	`lmsys/vicuna-7b-v1.5`
OpenChat-3.5	`openchat/openchat_3.5`	Indian-Constitution-chat	`Atulit23/meta-llama-indian-constitution-chat`
Starling-alpha	`berkeley-nest/Starling-LM-7B-alpha`	PubMedQA-211k	`BLACKBUN/llama-2-7b-pubmed-qa-211k`
OpenChat-0106	`openchat/openchat-3.5-0106`	EYE-Llama-gqa	`QIAIUNCC/EYE-Llama_gqa`
Starling-beta	`Nexusflow/Starling-LM-7B-beta`	EYE-Llama-qa	`QIAIUNCC/EYE-Llama_qa`
OpenHermes-2	`teknium/OpenHermes-2-Mistral-7B`	NorskGPT-Llama	`bineric/NorskGPT-Llama-7B-v0.1`
OpenHermes-2.5	`teknium/OpenHermes-2.5-Mistral-7B`	chat-oasst1-ft	`whatdhack/Llama-2-7b-chat-hf-oasst1-ft-sg`

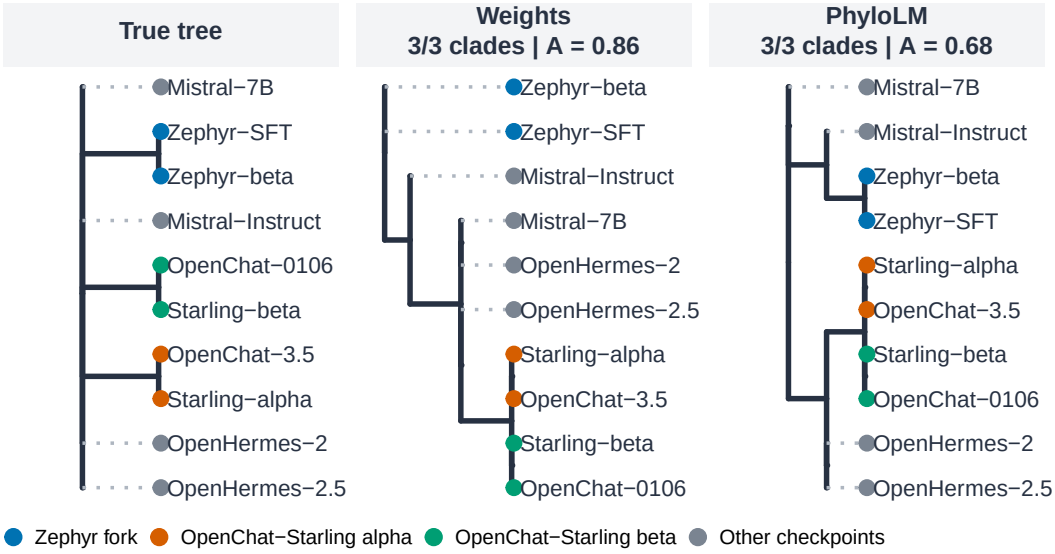


Figure 4: Documented Mistral-7B family: true tree (left), our weight-based estimate (middle), and PhyloLM's output-based estimate (right) (Yax et al., 2025). Panel headers give the documented clades estimated and the tree-likeness score A. Both estimate all three documented clades.

E Behavioral Meta-Analysis Diagnostics

Pooling estimator. Because each tree has its own weight-space scale, a raw pooled slope is not meaningful across trees. We standardize within tree and estimate the variance of each Fisher-z correlation with a delete-one-trained-node jackknife, then pool tree-level effects using a DerSimonian–Laird random-effects meta-analysis. Pair counts are descriptive and are never treated as independent observations.

Per-cell heterogeneity. Table 9 gives, for every cell of Table 3, r_{DL} with its 95% CI, Higgins' I^2 (Higgins et al., 2003), and the number of trees k. The statistic estimates the proportion of observed effect variation attributable to between-tree heterogeneity rather than sampling error; here it ranges from none to moderate (I^2 0–36%). All cells except Flan held-out translation exclude zero. The Llama cells are negative, Flan HellaSwag is significantly positive, the other significant Flan cells are negative, and Flan held-out translation is near zero.

Flan output audit. We audited the full fine-tuning responses: 641 trained models in the standard Flan cohort and a separate 641 in the matched no-translation cohort. ARC-C and MMLU prompts display answer choices, so valid option-only answers are not counted as contamination; HellaSwag is rendered as a free completion and TruthfulQA as a concise free response. Even after separating

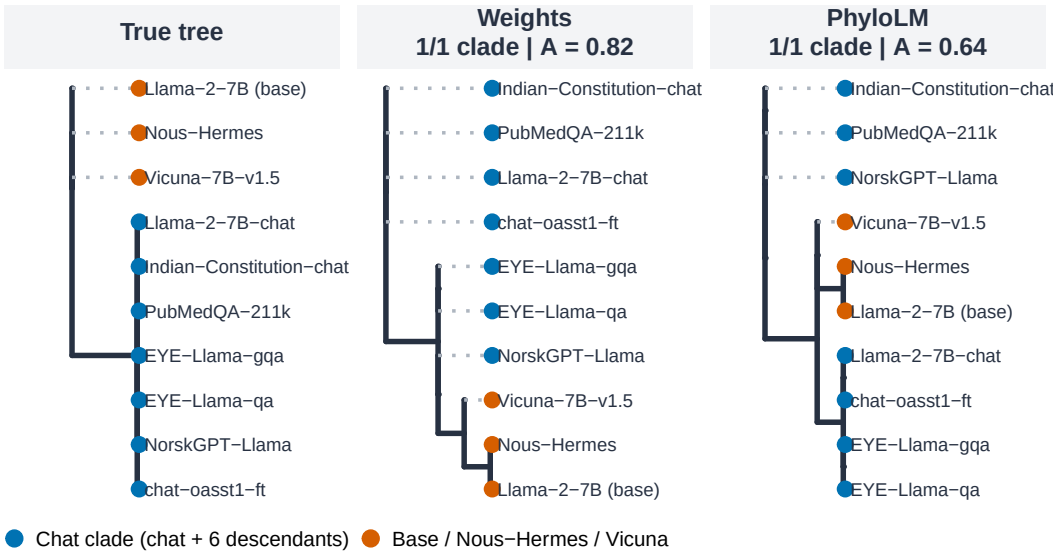


Figure 5: Documented Llama-2-7B family: true tree (left), our weight-based estimate (middle), and PhyloLM's output-based estimate (right). Panel headers as in Fig. 4. Both estimate the documented chat clade; the documented tree specifies no structure within that clade.

valid option labels, outputs that exactly match unrelated fine-tuning-task labels (e.g., `entailment`, `positive`, or `sci/tech`) comprise 20.4% of ARC-C outputs and 22.9% of MMLU outputs. The same contamination appears throughout the Flan results: 14.7% for trained translation, 22.6% for held-out translation, 21.3% for HellaSwag, and 30.3% for TruthfulQA. Because the problem spans every task, flagging individual tasks would be misleading. We report every cell but interpret the Flan row as associations among generated outputs, not as evidence of task performance.

Table 9: Per-cell heterogeneity for the weight–behavior correlations of Table 3 (full fine-tuning, empty responses kept). DerSimonian–Laird random-effects r_{DL} with 95% CI, Higgins' I^2, and number of trees k. Bold rows have 95% CIs excluding zero.

Model	Task	r_{DL} [95% CI]	I^2	k
Llama-3.2-1B	**Translation (trained)**	$\mathbf{-0.20}$ $\mathbf{[-0.26, -0.13]}$	**13%**	**50**
Llama-3.2-1B	**Translation (held-out)**	$\mathbf{-0.23}$ $\mathbf{[-0.29, -0.17]}$	**0%**	**50**
Llama-3.2-1B	**HellaSwag (held-out)**	$\mathbf{-0.17}$ $\mathbf{[-0.24, -0.10]}$	**26%**	**50**
Llama-3.2-1B	**ARC-C (held-out)**	$\mathbf{-0.20}$ $\mathbf{[-0.26, -0.13]}$	**33%**	**50**
Llama-3.2-1B	**MMLU (held-out)**	$\mathbf{-0.19}$ $\mathbf{[-0.26, -0.12]}$	**36%**	**50**
Llama-3.2-1B	**TruthfulQA (held-out)**	$\mathbf{-0.22}$ $\mathbf{[-0.27, -0.16]}$	**19%**	**50**
Flan-T5-base	**Translation (trained)**	$\mathbf{-0.10}$ $\mathbf{[-0.16, -0.04]}$	**0%**	**50**
Flan-T5-base	Translation (held-out)	$+0.04$ $[-0.01, +0.09]$	0%	50
Flan-T5-base	**HellaSwag (held-out)**	$\mathbf{+0.09}$ $\mathbf{[+0.05, +0.14]}$	**0%**	**50**
Flan-T5-base	**ARC-C (held-out)**	$\mathbf{-0.14}$ $\mathbf{[-0.19, -0.09]}$	**0%**	**50**
Flan-T5-base	**MMLU (held-out)**	$\mathbf{-0.16}$ $\mathbf{[-0.20, -0.11]}$	**0%**	**50**
Flan-T5-base	**TruthfulQA (held-out)**	$\mathbf{-0.17}$ $\mathbf{[-0.22, -0.12]}$	**3%**	**50**

F Empty-Response Sensitivity and Depth

Some responses are empty because the model emits its end-of-sequence (EOS) token immediately. An empty response is an observed output rather than a missing one, so the primary analysis keeps it as an (empty) output. Table 10 reports a sensitivity analysis that removes pairs involving an empty response. The choice matters only for HellaSwag, where empty responses are common.

Table 11 reports the same empty-output rate by leaf depth. HellaSwag empties increase from 16.72% at depth 1 to 41.13% at depth 3, then remain essentially level at 40.25% at depth 4. Empty responses

Table 10: Llama-3.2-1B weight–behavior correlations under full fine-tuning, over all trained models, with empty responses kept (primary, as in Table 3) or excluded (sensitivity). Values are r_{DL} [95% CI]. Empty rates pool all leaf outputs and all three draws.

Task	Empty kept	Empty excluded	Empty outputs
ARC-Challenge	-0.20 $[-0.26, -0.13]$	-0.20 $[-0.26, -0.13]$	0.00%
Dolly open-ended	-0.08 $[-0.13, -0.03]$	-0.08 $[-0.14, -0.03]$	0.16%
HellaSwag	-0.17 $[-0.24, -0.10]$	-0.26 $[-0.34, -0.19]$	38.64%
MMLU	-0.19 $[-0.26, -0.12]$	-0.19 $[-0.26, -0.12]$	0.00%
TruthfulQA	-0.22 $[-0.27, -0.16]$	-0.22 $[-0.28, -0.16]$	0.02%

therefore become more common with more fine-tuning; excluding them nevertheless strengthens the negative association, as shown in Table 10.

Table 11: Empty-response rate by leaf depth in the Llama-3.2-1B full fine-tuning trees. Rates pool outputs across all leaves observed at each depth and all three draws.

Task	Depth 1	Depth 2	Depth 3	Depth 4
ARC-Challenge	0.00%	0.00%	0.00%	0.00%
Dolly open-ended	1.45%	0.34%	0.10%	0.06%
HellaSwag	16.72%	30.86%	41.13%	40.25%
MMLU	0.00%	0.00%	0.00%	0.00%
TruthfulQA	0.10%	0.04%	0.01%	$< 0.01\%$