

A Best Match Is Not a Twin: What Feature Similarity Between State Space Models and Transformers Certifies, Why the Null and the Shared Inputs Decide the Verdict, and the Controls That Would Show Two Architectures Learned the Same Thing

Pranay Mahendrakar

SONYTECH · ORCID 0009-0003-7224-029X

pranaymahendrakar@sonytech.in · mahendrakarpranay@gmail.com · +91 6361723454

Abstract. Selective state space models such as Mamba compress the past into a fixed-size state, while Transformers attend over the whole context, and a growing set of studies asks whether the two learn the same internal features. A 2026 study reports that 99.98% of a Mamba model's sparse-autoencoder features have a strong Transformer counterpart and reads this as support for the universality hypothesis. In this analysis the author reviews and synthesises published results to show that such verdicts are set less by the models than by three choices the measurement makes: the reference point a score is judged against, the inputs the two models share, and the instrument that extracts features. The author derives the expected Jaccard similarity of two binary features that share nothing and shows that the study's reported band, 0.234 to 0.411 for every feature's best match, is what unrelated features would score if they fired on roughly a quarter to three fifths of tokens; firing rates were not reported, so this cannot be checked. Against same-architecture skylines, the strongest design supports a narrower claim than universality: features recur across the two families nearly as well as across random seeds, most strongly for token-level features, while the documented differences sit in context mixing and recurrent state, which residual-stream dictionaries do not measure. Sixteen published findings are consolidated in a table, a seven-step procedure licenses a convergence claim at one of four levels, and five studies are proposed that would settle the open part.

Keywords: universality hypothesis; representational similarity; sparse autoencoders; state space models; Mamba; mechanistic interpretability; null calibration; Platonic representation hypothesis

SONYTECH ORCID: 0009-0003-7224-029X

1. Introduction

The universality hypothesis entered mechanistic interpretability as a deliberately speculative claim. Olah et al. (2020) offered it as the third of three claims about circuits in vision networks: that analogous features and circuits form across models and tasks. The claim has practical stakes. If different networks learn the same features, then a feature catalogued in one model is evidence about another, interpretability results transfer, and the effort spent on one architecture is not wasted when the field moves to the next. Sparse autoencoders (SAEs), which decompose a layer's activations into a wide dictionary of sparsely active latents on the premise that networks store more features than they have dimensions by overlapping them (Elhage et al., 2022; Cunningham et al., 2023), have made the hypothesis testable at scale, because they give each model a list of candidate features that can be matched against another model's list (Lan et al., 2024; Wang et al., 2024). In parallel, a broader literature on representational convergence has argued that networks trained on different data and modalities are coming to measure similarity between inputs in increasingly alike ways (Huh et al., 2024).

State space models are the sharpest test of the hypothesis available in language modelling. Mamba replaces

attention with a selective recurrence whose parameters depend on the current token and whose state has a fixed size (Gu and Dao, 2023). A Transformer can look back at every earlier position; Mamba must decide, token by token, what to keep. If features are shaped by the computation a model performs, the two families should differ. If they are shaped mainly by the statistics of the text, they should not. The question is live because the architectures are known to differ in capability in ways that a shared feature vocabulary would not obviously predict: Transformers copy and retrieve from context far better than state space models of similar size (Jelassi et al., 2024; Waleffe et al., 2024), even where the state space model has the lower perplexity.

A 2026 study answers the question directly. Nagaraj et al. (2026) trained an SAE on the middle layer of Mamba-130m, took a public SAE for the middle layer of Pythia-70m, ran both models over 10 million tokens of Wikipedia, binarised every latent's activations, and found for each Mamba latent the Pythia latent with the highest Jaccard overlap. Of 12,288 Mamba latents, 12,285 had a best match above 0.35, a threshold the authors placed from the observed distribution, and they conclude that the recurrent bottleneck leaves the core semantic vocabulary intact, with divergence confined to a handful of formatting features. An earlier study of Mamba-130M and Pythia-160M reached a similar verdict through a different instrument, the maximum Pearson correlation between latent activations, and added the comparison

that makes such a verdict interpretable: the same procedure run between two Pythia models that differ only in random seed (Wang et al., 2024).

Two positions now face each other. On one side, cross-model and cross-architecture feature matches are evidence that networks converge on shared representations, and the convergence literature supplies a theoretical reason to expect it (Huh et al., 2024; Ziyin and Chuang, 2025; Doshi et al., 2026). On the other, the similarity scores on which such claims rest have been shown to be inflated by model width and by the practice of reporting the best of many comparisons (Groger et al., 2026), to be reproducible by networks whose weights are random (Heap et al., 2025; Murphy et al., 2024), and to disagree with each other about the most basic comparisons (Ding et al., 2021; Klabunde et al., 2023a). Both sides are credible, both publish in the same venues, and they rarely engage the same data.

In this analysis the author reviews and synthesises published results to show that the cross-architecture verdict is set by three choices the measurement makes before any model is consulted: the reference point against which a similarity score is judged, the inputs that the two models share, and the instrument that turns activations into features. Where a study fixes those choices well, the evidence supports a narrower claim than universality. Where it does not, the score does not distinguish shared representation from shared input, and a high best match is not evidence of a twin.

The paper's contribution is fourfold. First, it separates four claims that "the same features" can mean, from co-activation through shared geometry and shared causal role to shared computation, and states what each requires (Section 2). Second, it derives the null expectation of the Jaccard statistic used in the 2026 study and shows that the study's reported band is uninterpretable without firing rates the study did not report (Section 4, Figure 1). Third, it consolidates sixteen published findings on cross-model and cross-architecture similarity, each beside the reference point its source reported (Table 1). Fourth, it gives a seven-step procedure for licensing a convergence claim (Algorithm 1) and proposes five studies that would settle what remains open (Section 11).

Three things the paper does not argue should be stated at the outset. It does not argue that Mamba and Transformers learn different features; on the strongest design available they share many, and Section 9 makes that case at full strength. It does not argue that SAEs are useless for cross-model work; several of the instruments discussed are SAE variants built for exactly this purpose. And it does not re-argue that SAE latents vary across training seeds, which is established (Paulo and Belrose, 2025) and enters here as an input, not a conclusion. The paper's object is narrower: what a cross-architecture feature-similarity score licenses, and what it would take

to license more.

Section 3 describes what the 2026 study measured. Sections 4 to 7 take the three choices in turn - the null, the shared inputs, the instrument - and then the choice of metric. Section 8 sets the similarity evidence beside the published differences between the two families. Section 9 states the strongest case for universality. Section 10 consolidates the evidence and gives the licensing procedure; Section 11 proposes studies; Sections 12 and 13 state what is not known and what limits this paper.

2. Four things "the same features" can mean

A claim that two models share a feature can be made at four levels, each stronger than the one before. The levels are the author's stipulation; they are introduced so that each piece of evidence can be assigned to the level it can reach.

Level 1: co-activation. A unit in model A and a unit in model B are active on the same inputs. This is what matching by activation correlation (Wang et al., 2024; Lan et al., 2024), by maximum correlation over neurons (Gurnee et al., 2024) or by Jaccard overlap of binarised activations (Nagaraj et al., 2026) measures. It is a statement about two functions of the input, not about the models' internal organisation. Two thermometers in the same room co-activate.

Level 2: shared geometry. The relations among units, or among inputs as each model represents them, agree up to a stated class of transformations. Centered kernel alignment (Kornblith et al., 2019), canonical correlation variants (Raghu et al., 2017; Morcos et al., 2018), representational similarity analysis applied to paired SAE features (Lan et al., 2024), shape metrics (Williams et al., 2021) and mutual nearest neighbours (Huh et al., 2024) operate here. Level 2 is not implied by Level 1: two dictionaries can contain co-activating pairs whose arrangement differs, and two networks can span common low-dimensional subspaces while their individual units differ (Li et al., 2015).

Level 3: shared use. The matched units play the same causal role in each model's computation: intervening on one changes behaviour in the way intervening on the other does, or a representation from one can be substituted into the other without loss. Model stitching (Lenc and Vedaldi, 2014; Bansal et al., 2021), cross-model transfer of steering and editing interventions (Paulo et al., 2024; Sharma et al., 2024) and crosscoder features validated by steering (Jiralerspong and Bricken, 2026) reach here.

Level 4: shared computation. The models compute the matched features by the same mechanism, through analogous circuits. Circuit-level comparisons, such as the finding that induction behaviour in Mamba is implemented by a structure analogous to the

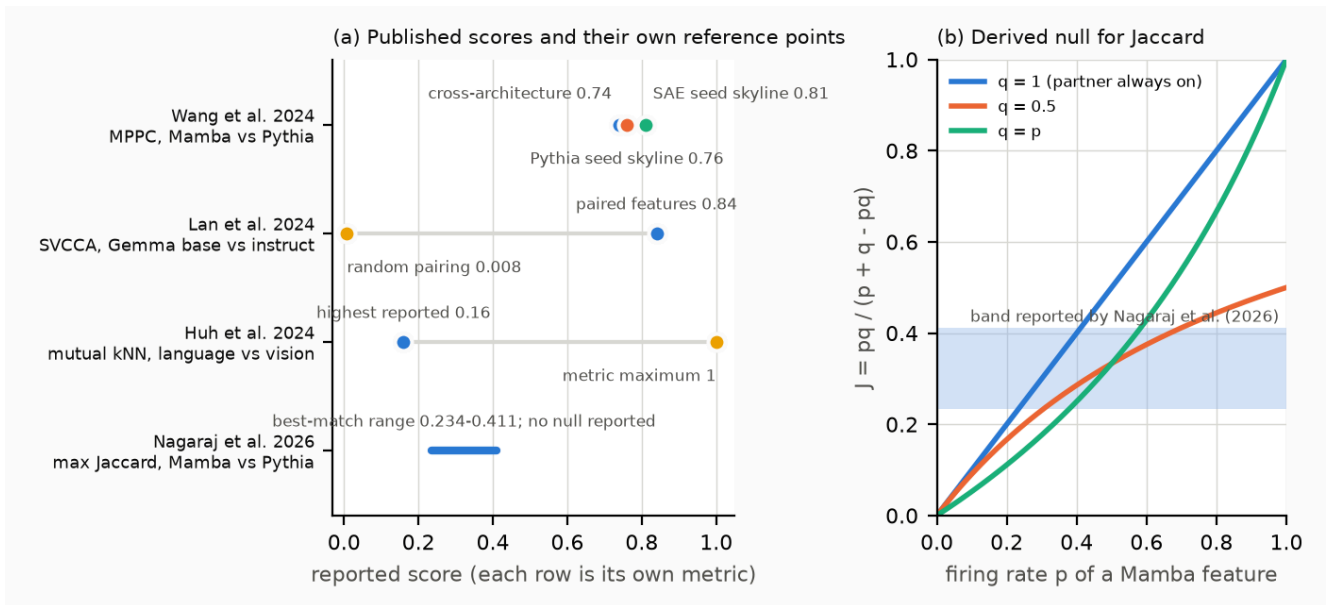


Figure 1. (a) Published cross-model similarity scores beside the reference point each source reported: Wang et al. (2024) cross-architecture MPPC against same-architecture seed and SAE-seed skylines; Lan et al. (2024) paired-feature SVCCA against random pairing; Huh et al. (2024) language-vision mutual nearest-neighbour alignment against the metric's maximum; Nagaraj et al. (2026) best-match Jaccard range, for which no reference was reported. Each row is its own metric and rows are not comparable with one another. (b) The author's derivation, not a measurement: the expected Jaccard of two features with no shared content, $J = pq / (p + q - pq)$, as a function of the firing rate p of one feature, for partner rates $q = 1$, $q = 0.5$ and $q = p$, with the band of best-match scores reported by Nagaraj et al. (2026) shaded.

Transformer's induction circuit but with an offset in where a token's information is written (Wang et al., 2024), operate at this level.

The universality hypothesis as Olah et al. (2020) stated it concerns features and circuits, so it is a Level 3 and Level 4 claim. The convergence hypothesis of Huh et al. (2024) is a Level 2 claim about how models measure distance between inputs. Most cross-architecture SAE evidence is Level 1. The gap between what is measured and what is concluded is the subject of the rest of this paper.

The levels also fix what a negative result would mean. A pair of models could share every Level 1 correspondence and differ at Level 4, if the same features are computed by different routes. The clock and pizza algorithms for modular addition are an existence proof at small scale: networks trained on the same task, differing only in hyperparameters or initialisation, compute the same function through qualitatively different internal procedures (Zhong et al., 2023). Group-composition networks show the converse pattern: the family of representations a network can learn is fixed by the mathematics, but which members a given network learns, and in what order, varies from run to run (Chughtai et al., 2023). Neither result concerns state space models, and neither shows that Mamba and Transformers diverge. They show that agreement at one level does not transfer to another without a separate test.

3. What the 2026 study measured

Nagaraj et al. (2026) chose Mamba-130m and Pythia-70m-deduped as models of comparable scale. Both were run on the same 10 million tokens of English Wikipedia, tokenised with the GPT-NeoX tokeniser into sequences of 1,024 tokens. Activations were taken from the residual stream at the midpoint of each model's depth: block 12 of Mamba's 24 and block 3 of Pythia's 6. For Pythia the authors used a public SAE with an expansion factor of 64 over a 512-dimensional residual stream, giving 32,768 latents. For Mamba they trained an SAE with an L1 sparsity penalty and an expansion factor of 16 over a 768-dimensional stream, giving 12,288 latents. Every latent's activation was then reduced to a binary mask: active wherever its value exceeded zero. For each Mamba latent, the Pythia latent with the highest Jaccard overlap of masks was called its twin.

The reported distribution of twin scores has a minimum of 0.234 and a maximum of 0.411. The authors partition it into an aligned class (above 0.35), a grey zone (0.28 to 0.35) and an alien class (below 0.25), and state that these thresholds were taken from the observed range rather than from theory and should be read as descriptive in the absence of a null distribution. Three latents fall below 0.35: two in the grey zone and one alien. The aligned class therefore holds 12,285 of 12,288 latents, which the paper reports as 99.98%. The authors characterise the three divergent latents qualitatively, including a Mamba latent that fires on several unrelated formatting tokens where its Pythia twin fires on one, and

present this as a hypothesis-generating observation. Their limitations section names the asymmetric dictionary sizes as the study's most significant weakness, noting that the larger Pythia dictionary gives every Mamba latent more candidates and may inflate alignment.

Several features of this design matter for what follows, and each is stated in the source.

The comparison is one-directional. Each Mamba latent is matched into the Pythia dictionary; the converse search, for Pythia latents with no Mamba counterpart, is not reported. A study looking for features unique to one architecture has searched for them on one side only. Wang et al. (2024) report both directions.

The scores are low in absolute terms. A Jaccard of about 0.41, the highest in the study, means that the union of the two latents' active tokens is about two and a half times the size of their intersection. No Mamba latent in the study has a Pythia partner that shares even half of their combined support. The label "aligned" is applied to latents whose best partner shares between 35% and 41% of it.

No reference distribution was computed. The authors say so, and suggest random feature pairs as future work. Without a reference, the statement that 99.98% of latents exceed 0.35 places the latents relative to one another, not relative to what unrelated latents would score.

Firing rates are not reported. Section 4 shows why that omission is decisive for a Jaccard statistic in particular.

The models were trained on nearly the same text with the same tokeniser. Section 5 shows why that matters for any Level 1 comparison.

There is also a small internal inconsistency in the source, noted here only so that it is not attributed to this paper: the divergent fraction is given as 0.02% in the abstract, introduction and conclusion and as 0.03% in one subsection. Three latents of 12,288 is 0.024%.

4. The reference point decides the verdict

A similarity score is evidence only relative to what the same score would be if the property in question were absent. The cross-model literature has used four kinds of reference point, and they lead to different verdicts on the same data.

4.1 What unrelated binary features score

The Jaccard statistic has a property that correlation does not: its expected value under independence depends strongly on how often the two features fire. The following derivation is the author's.

Let two binary features be active on fractions p and q of N tokens, and suppose their activations are independent. The expected size of their intersection is Npq and the

expected size of their union is $N(p + q - pq)$. For $N = 10$ million, both counts concentrate tightly around these values, so the Jaccard of two unrelated features is close to

$$J0 = pq / (p + q - pq) .$$

Three properties follow directly. First, $J0$ rises with both rates. When the partner fires on every token ($q = 1$), $J0 = p$: a feature active on 30% of tokens scores 0.3 against an always-on partner with which it shares nothing. When the two rates are equal, $J0 = p / (2 - p)$. Second, for any dependence whatever, J cannot exceed $\min(p, q) / \max(p, q)$, because the intersection is at most the smaller set and the union at least the larger. Jaccard therefore measures agreement in firing rate as well as agreement in content: two features with rates differing by a factor of three cannot score above one third however closely they track each other. Third, when each Mamba feature is matched to the best of many Pythia candidates and the candidates are unrelated to it, the maximum tends to be won by the candidate with the highest firing rate, because $J0$ increases with q .

Inverting the equal-rate expression locates the 2026 band. Unrelated features with equal firing rates produce $J0$ between 0.234 and 0.411 when they fire on between about 38% and 58% of tokens. Against an always-on partner, unrelated features produce the same band when they fire on between 23% and 41% of tokens. Figure 1(b) plots these two cases and a third, intermediate one: against a partner active on half of all tokens ($q = 0.5$), unrelated features fall in the band when they fire on between about 31% and 70% of tokens.

This does not show that the 2026 features fired that often. The study does not report firing rates, so the author cannot say, and neither can a reader. What it shows is that the reported numbers are uninterpretable without them. If the Mamba latents were sparse in the usual sense - active on a percent of tokens or less - then a best match of 0.35 would be far above the independence expectation, which for $p = q = 0.01$ is about 0.005, and the result would be strong Level 1 evidence. If the latents were active on a third or more of tokens, the same number would be what unrelated latents produce, and the result would be no evidence at all. Two features of the reported distribution make the second reading harder to dismiss than it might seem. One is the narrowness of the band: every one of 12,288 best matches falls between 0.234 and 0.411, a spread that a mixture of true twins, partial matches and orphans would not obviously produce, and that features of similar density matched against a pool of dense candidates would (the author's inference, offered as a hypothesis). The other is that dense latents are a documented and persistent part of SAE dictionaries trained on language models, not a rare artefact (Sun et al., 2025), and the binarisation threshold of zero counts every positive activation, however small, as firing.

The authors call for a null distribution and suggest random feature-pair sampling. The test that would settle it, which the author proposes, is cheap and also controls the maximum: compute, for each Mamba latent, the distribution of its best-match Jaccard against a Pythia dictionary whose activations have been shifted by a random offset in token position, which preserves every firing rate and destroys every correspondence. The study's twin scores mean something only to the extent that they exceed that distribution.

4.2 The maximum over candidates inflates the score

Taking the best of many comparisons raises the reported score even when nothing is shared, because the expected maximum of many draws exceeds their mean, and the inflation grows with the number of candidates. Groger et al. (2026) make this point about the common practice of comparing every layer of one model with every layer of another and reporting the best pair, and they show that it is one of two confounders, with model width, that can make larger models look more aligned for reasons unrelated to shared representation. Their remedy is to compute the null distribution of the reported statistic itself - the maximum, if a maximum is reported - by breaking the correspondence between the samples the two models see. The same logic applies to matching features, where the selection is over tens of thousands of candidates rather than tens of layers. Every SAE matching study discussed here reports a best match per feature: over 32,768 candidates in Nagaraj et al. (2026), over every feature in the Mamba dictionaries in Wang et al. (2024), and over the second model's dictionary in Lan et al. (2024).

Gurnee et al. (2024) give the design that corrects for this at the neuron level. For each neuron in one GPT-2 model they compute its maximum activation correlation with any neuron in another model trained from a different seed, averaged over the four other seeds, and subtract the maximum correlation it attains against a random rotation of the other model's neurons, averaged the same way. The difference, which they call excess correlation, is a maximum compared with a maximum. Only 1,253 of 98,304 neurons in one GPT-2 medium model have an excess correlation above 0.5, and the paper reports that 1 to 5% of neurons are universal across seeds on this standard. That is the figure the same-architecture, same-data, different-seed comparison yields when the selection step is controlled.

4.3 A random-pairing null is the weakest available

Lan et al. (2024) pair each feature in one model with its most correlated feature in another, measure how similar the resulting paired spaces are with SVCCA and RSA, and compare the paired score with the score obtained when features are paired at random. Against this reference their results are decisively significant: for almost all layer pairs past the first, most SVCCA p-values lie between 0.00 and 0.01, and in one base-versus-instruct

comparison the paired SVCCA is 0.84 against a random-pairing mean of 0.008. The comparison establishes that the pairing carries information. It does not establish that the information is a shared representation rather than shared input, because random pairing destroys both. A pairing chosen by maximum correlation will beat random pairing whenever the two models' features respond to the same tokens, which, for models that share a tokeniser and corpus, they will do for reasons Section 5 sets out. Lan et al. state that comparing features across models requires a common tokeniser for exactly this reason: the pairing is computed token by token.

Lan et al. also report a control that bears directly on the question. When the same pipeline is run between SAEs trained on Pythia-70m and SAEs trained on a copy of Pythia-70m with randomised weights but trained embeddings, the similarity scores are very low, and on average only about 1% of feature pairs survive their filters. That is the right kind of control, but its result is harder to read than it looks, because the non-random side is one of the authors' own SAEs. They report that their custom-trained SAEs keep only about 2 to 4% of pairs in any comparison, against 20 to 30% for the publicly hosted SAEs, and they attribute the gap to SAE quality: their SAEs saw 100 million training tokens, where SAEs like Eleuther's saw about 8 billion. It is also a reminder of how much the filters do: the headline scores are computed on the minority of pairs that survive filtering for shared top tokens, for one-to-one mappings and for a minimum correlation.

4.4 A same-architecture skyline is the reference that answers the question

The question the universality hypothesis poses about architectures is comparative: does changing the architecture change the features more than other changes do? The reference point that answers it is a skyline, the same procedure run between two models of the same architecture. Wang et al. (2024) run two. The first compares Pythia-160M with another Pythia model of the same architecture trained from a different seed, which changes initialisation and data order; the second compares two SAEs trained on the same Pythia model, which isolates the variability of the SAE itself. Their cross-architecture mean maximum pairwise Pearson correlation is 0.74, against 0.76 for the seed skyline and 0.81 for the SAE skyline (Figure 1a). For 90% of Pythia features, the cross-architecture score minus the seed-skyline score lies between -0.22 and 0.14, and more than 60% of features show near-zero difference. A neuron-basis comparison between the two architectures, by contrast, shows almost no similarity.

On this design the cross-architecture comparison loses little to the same-architecture one, and the authors' conclusion - that most SAE features are similar in the two families - follows. It is the strongest piece of evidence for cross-architecture universality in the

literature reviewed here. Its scope, set by the rest of the design, is examined in Sections 5 and 9.

4.5 What the four reference points imply together

The four reference points form a ladder. Random pairing (Lan et al., 2024) is beaten by any pairing that tracks the input. A random-weight model (Lan et al., 2024; Heap et al., 2025) separates what training produced from what the data and architecture produce untrained. A selection-aware null (Gurnee et al., 2024; Groger et al., 2026) removes the inflation that comes from searching. A same-architecture skyline (Wang et al., 2024) measures architecture against the variability that exists anyway. A cross-architecture claim is best supported when it passes all four, and the 2026 study reports none.

5. Shared inputs make agreement expected

Mamba was trained on the Pile with the GPT-NeoX tokeniser for 300 billion tokens, and its authors state that the Pythia models they compare against were trained with the same tokeniser, dataset and training length (Gu and Dao, 2023). The Pythia suite was trained in two copies, one on the Pile and one on a near-deduplicated Pile of about 207 billion tokens seen for roughly 1.5 epochs; the "-deduped" models used by Nagaraj et al. (2026) are the second copy (Biderman et al., 2023). Both families inherit the tokeniser built for GPT-NeoX-20B, itself trained on the Pile (Black et al., 2022; Gao et al., 2020). Wang et al. (2024) note the same shared tokeniser and corpus for their pair.

This makes Mamba and Pythia an unusually clean pair for isolating architecture, which is why they are chosen. It also means that any feature which is a function of the current token, or of the current token and its immediate neighbours, will appear in both models whatever their architectures do with it, because both must read the same tokens and predict the same next tokens from the same corpus statistics. At Level 1, such features are guaranteed to co-activate.

The published evidence indicates that such features account for much of the measured agreement. Wang et al. (2024) report that feature pairs with a maximum correlation near 1 are typically very simple, firing on individual tokens - parts of speech, prefixes and suffixes, letters, punctuation and digits - and that as feature complexity rises, most matched pairs fall between 0.3 and 0.8. More than 30% of Pythia features find a Mamba partner above 0.95. Lan et al. (2024) find that the first layer of every model compares poorly with every other layer, including the first layer of the other model, and suggest that it contains few meaningful comparable features; they also find that removing pairings based on non-concept tokens raises their scores. Latents that fire on a single vocabulary item concentrate in early layers (Cho et al., 2026).

Three further results show that agreement of this kind does not require the trained computation at all. In many settings, SAEs trained on Transformers whose weights are random reach auto-interpretability and reconstruction scores similar to those trained on the corresponding trained models across Pythia sizes (Heap et al., 2025); as Lan et al. (2024) summarise that result, the interpretable latents in the random case tend to be simple features tied to single tokens. Across dozens of vision models, nearly all the alignment two networks will eventually share is in place within the first epoch, well before accuracy plateaus, which the authors attribute to shared input statistics and architectural bias rather than to the solution of the task (Kapoor et al., 2025). And a saliency method can be insensitive both to the model's parameters and to the data-generating process while still producing plausible-looking output, which is the general reason randomisation tests are needed for any interpretability method (Adebayo et al., 2018).

None of this shows that the cross-architecture agreement is only token identity. Wang et al. (2024) find many matched pairs among more complex features, and their subtraction of the seed skyline is designed to cancel the effect of feature complexity, since simple features match well across seeds too. What it shows is that a single aggregate - a mean correlation, a percentage above a threshold - mixes a component that any two models with the same tokeniser and corpus would share with a component that would bear on universality, and that the aggregate cannot be read without separating them. The informative part of the cross-architecture evidence is in the tail of complex, context-dependent features, and that is where the published scores are lowest.

The point extends beyond language. Across many vision models, the consistency of representational similarity from one stimulus set to another depends on the training objective, and the link between representational similarity and task behaviour holds most strongly on single-domain datasets (Ciernik et al., 2024). A theory of cross-lingual similarity in language models derives the familiar mid-depth peak from data with shared abstract structure and language-specific surface structure, and separates similarity of neighbourhoods from alignment of coordinates, the latter appearing only when code-switched data are plentiful (Doshi et al., 2026). In both cases what two models share is a function of what their inputs share, which is the variable cross-architecture comparisons on the Pile hold fixed.

6. The instrument contributes its own variance

A feature-matching study measures the models through two dictionaries, and each dictionary is an estimate with its own variance. That variance is large enough to matter.

SAEs trained on the same activations from the same model, differing only in the random seed of the SAE, recover different sets of latents. Paulo and Belrose (2025) match latents between seeds one-to-one with the Hungarian algorithm and call a latent shared when its matched partner has cosine similarity above 0.7 in both encoder and decoder weights; for a 131K-latent SAE on a feedforward layer of Llama 3 8B, about 30% of latents are shared. Smaller models and smaller SAEs share more, and ReLU SAEs trained with an L1 penalty are more stable across seeds than TopK SAEs. Wang et al.'s SAE-seed skyline of 0.81 (Section 4.4) is the same phenomenon measured by activation correlation: two SAEs on one model do not agree perfectly, so the ceiling for any two-model comparison through SAEs is below 1.

Dictionary size and family change what is found. Latents from a larger SAE divide into those that replace latents of a smaller SAE and those that add information the smaller one lacked, and larger-SAE latents often decompose into combinations of smaller-SAE latents, so neither size gives a complete, atomic list (Leask et al., 2025). As dictionaries grow, sparsity pressure splits general features into specific ones and can absorb a general feature into its children, so that a latent that appears monosemantic fails to fire on some inputs it should cover (Chanin et al., 2024); nested dictionaries reduce this (Bussmann et al., 2025). The causal role of single-token latents differs between SAE families trained on the same base model, by more than it differs across scale within a family (Cho et al., 2026).

These results bear on cross-architecture comparison in a specific way. When two dictionaries of different sizes and families are compared, a mismatch can arise because the models differ or because one dictionary has split, absorbed or omitted a feature that the other kept. The 2026 study compares a 12,288-latent L1 dictionary on one model with a 32,768-latent dictionary on the other, and its authors acknowledge the asymmetry. In the direction they report, the larger Pythia dictionary offers more candidates and more chances of a high maximum (Section 4.2). In the direction they do not report, Pythia latents with no Mamba counterpart could reflect nothing more than Pythia's dictionary being nearly three times as wide.

Crosscoders, which learn one dictionary over both models' activations, avoid separate dictionaries but bring priors of their own. A standard crosscoder has a prior toward shared latents, as Jiralerspong and Bricken (2026) note, and the L1-trained version can misattribute concepts as unique to one model when they exist in both (Minder et al., 2025). Dedicated feature crosscoders partition the dictionary into shared and exclusive sets by design, and the size of the exclusive partition, between 1% and 5% in the reported runs, is set by the experimenter (Jiralerspong and Bricken, 2026). Applied across architectures - Llama, Qwen, GPT-OSS - they find interpretable model-exclusive features whose presence is

validated by steering. A universal SAE that reconstructs every model's activations from one shared code finds concepts common to several vision models (Thasarathan et al., 2025). Both instruments are valuable, and both make the fraction of features called shared depend in part on a design choice.

The consequence is not that SAE-based comparisons are uninformative. It is that the cross-architecture score must be read against the instrument's own variability, measured by the same procedure, which is what the SAE-seed skyline does and what a study that trains dictionaries of different sizes on the two models cannot do.

7. The metric changes the verdict

Even with the reference point, inputs and dictionaries fixed, different similarity measures can disagree. The literature on this is mature and points one way.

Canonical correlation methods are invariant to invertible linear transformations, which makes them unable to measure meaningful similarity when representations have more dimensions than there are data points; centered kernel alignment avoids this and identifies corresponding layers across networks trained from different initialisations (Kornblith et al., 2019). Yet CKA is sensitive to outliers and to transformations that preserve linear separability, and can be changed substantially without changing what the network does (Davari et al., 2022). In the regime of many dimensions and few samples, the biased estimator of CKA reports high similarity for pairs of independent random matrices and can be driven to its maximum value by them, while shuffling real data barely changes it; only the debiased estimator detects stimulus-driven alignment (Murphy et al., 2024). Widely used measures disagree on whether networks that differ only in initialisation learn similar representations, and when measures are graded on whether they respond to changes that affect behaviour and ignore changes that do not, each has different blind spots and a classical baseline does surprisingly well (Ding et al., 2021). A survey of these measures and their properties reaches the same conclusion in general form (Klabunde et al., 2023a), and a study of 7-billion-parameter language models finds that some are substantially different from others while warning that the scores need careful study to avoid false conclusions (Klabunde et al., 2023b).

Stitching, which tests whether one model's representation can feed another's later layers through a learned map, was introduced to measure equivalence directly (Lenc and Vedaldi, 2014) and can show that differently trained networks of one architecture are interchangeable without loss (Bansal et al., 2021). But when the map is trained on the task loss, stitching can rate distant layers as more similar than corresponding ones and even rate a layer as more similar to another

than to itself, because the trained map can produce out-of-distribution inputs that happen to perform well; matching representations directly avoids this (Balogh and Jelasity, 2024). Stitching layers with different shapes across architectures produces its own surprises (Hernandez et al., 2023).

The convergence literature has recently been re-examined with these problems in view. Huh et al. (2024) report that language and vision models align more as they improve, measured by mutual nearest neighbours, and they are explicit about the size of the effect: alignment rises but reaches 0.16 on a metric whose maximum is 1, and they leave open whether that is strong alignment with residual noise or weak alignment with major differences. When Groger et al. (2026) calibrate a range of metrics against permutation nulls that account for width and for selection over layers, the convergence trend in global spectral measures largely disappears, while agreement in local neighbourhoods survives and agreement in local distances does not; they propose that what converges is neighbourhood structure. You et al. (2026) then separate which samples are related from how far apart they are, and find that relational structure converges robustly at both local and global scales after calibration, while agreement in metric distances weakens as it is required to be more exact. A study of 36 astronomy models finds significant neighbourhood alignment somewhere in the network in every one of 139 comparisons, and every final-layer model pair across two surveys aligned, yet every transfer of redshift prediction has negative R squared (Duraphe et al., 2026). Geometry can be shared while the information a downstream task needs is not.

Two architecture-level comparisons outside language reinforce the point that the answer depends on what is measured. Vision Transformers and convolutional networks reach comparable accuracy, yet their internal representations differ strikingly by CKA, with the Transformer's more uniform across depth, a difference traced to early global aggregation by attention and to strong residual propagation (Raghu et al., 2021). Conversely, units that respond to the same visual patterns can be found across very different vision architectures and training objectives (Dravid et al., 2023), and relative representations, built from similarities to a set of anchor inputs, exploit the observation that angles between encodings stay stable across training runs to enable zero-shot stitching between latent spaces (Moschella et al., 2022). Both are true: there is shared structure, and there are architectural differences, and which one a study reports depends on the measure and level it chooses.

For the Mamba question, the implication is that a verdict from one metric at one layer at Level 1 should not be generalised to the others. Wang et al. (2024) use correlation and Nagaraj et al. (2026) use Jaccard of binarised activations; the two studies compare different

Pythia sizes at different layers and are not directly comparable. But it is worth noting that Wang et al. find more than 30% of features with a partner above 0.95 by correlation, while no feature in Nagaraj et al. has a partner above 0.412 by Jaccard. The two instruments, applied to the same model families, describe agreement of very different strength.

8. Where the two families are known to differ, and where a residual-stream dictionary looks

If Mamba and Transformers learned the same features and differed only in efficiency, their behaviour would be expected to match at equal perplexity. It does not, and the published differences are concentrated in a particular place.

On copying and retrieval from context, the gap is large and consistent. Jelassi et al. (2024) prove that a two-layer Transformer can copy strings of length exponential in its size while a model with a fixed-size state cannot copy strings carrying more information than the state can hold, and they find that pretrained Pythia models substantially outperform Mamba models of similar size at looking up a number in an in-context phone book, across sizes, despite Mamba's lower perplexity on the Pile. Mamba-130M's reported perplexity on the Pile validation split is 10.56 against 29.64 for Pythia-160M, and its average across six zero-shot tasks is 44.7 against 40.6 (Gu and Dao, 2023). In a controlled comparison at 8 billion parameters on the same data, pure Mamba and Mamba-2 models match or exceed a Transformer on many tasks but lag on tasks needing strong copying or in-context learning, such as five-shot MMLU and phone-book lookup, while a hybrid with a small share of attention layers exceeds the Transformer on all twelve standard tasks evaluated, by 2.65 points on average (Waleffe et al., 2024). In a suite of 17 pretrained models, attention-free gated-convolution models trailed attention by up to 2.1 perplexity points on the Pile, 82% of that gap was attributable to recalling previously seen tokens, and a 70M attention model beat a 1.4B gated-convolution model at that recall (Arora et al., 2023). On synthetic in-context learning tasks, Mamba matches Transformers on standard regression and beats them on sparse parity but falls short on tasks needing non-standard retrieval (Park et al., 2024). On two-hop position retrieval, Transformers keep an advantage over state space and hybrid models, and state space models develop locality-aware token embeddings that Transformers do not need (Pantazopoulos et al., 2026).

Not every expected difference appears. State space models of the Mamba type are bounded by the same circuit-complexity class as Transformers and cannot express simple state-tracking problems such as permutation composition, so the recurrence does not buy the expressive advantage it appears to offer (Merrill et al., 2024). Mamba can be written as a form of attention through its input-dependent matrices (Ali et al., 2024),

and a whole family of attention variants and state space layers can be related through structured semiseparable matrices (Dao and Gu, 2024). The families are mathematically closer than their names suggest, which is part of the reason to expect shared structure.

At the mechanistic level the evidence is mixed in an informative way. Factual recall in Mamba localises to middle-layer components at the last subject token, as it does in Transformers, rank-one editing inserts facts at specific locations, and the authors conclude that the two families share many similarities in factual recall (Sharma et al., 2024). Knockout methods adapted from attention to Mamba-1 and Mamba-2 find some information-flow phenomena that differ between the families and others common to all models examined (Endy et al., 2025). Contrastive activation steering, the tuned lens and latent-knowledge elicitation mostly transfer from Transformers to Mamba and RWKV, and some work better when they exploit the recurrent state (Paulo et al., 2024). Mamba's induction mechanism is structurally analogous to the Transformer's, with the difference that a token's information is written into the state at the next position (Wang et al., 2024). The relation between which layers a model cannot do without and which layers absorb fine-tuning updates runs opposite in the two families: anti-aligned across depth in every residual Transformer evaluated, overlapping in the Mamba-style models, with consequences for forgetting (Abbes et al., 2026). Mamba models contain activation subspaces whose rescaling improves performance across benchmarks by 8.27% on average (Mohan et al., 2026).

The pattern across these results is that the documented differences live in how information moves between positions: what can be copied, retrieved and kept over a sequence, and how the state is written. A residual-stream SAE at one layer decomposes the representation of each token position separately. It sees the output of the context-mixing operator at that position, not the operator, and not the recurrent state. For recurrent layers, each token writes a matrix-shaped update into the state that a vector-shaped residual-stream dictionary cannot represent directly; a dictionary of rank-one matrix atoms built for those writes can replace the model's own write and come closer to the original output distribution than deleting it on 92.4% of evaluated positions at one Gated DeltaNet layer-head, and on 88.1% when the same test is applied to Mamba-2-370M (Young, 2026). That is an instrument built for the place where the families differ. Residual-stream comparisons, by construction, sample the place where both families must represent the same tokens.

The author's reading, offered as an inference from these results and not as a finding, is that per-position residual-stream feature matching is the measurement least likely to detect an architectural difference of the kind the capability literature documents. High agreement there is what both the universality hypothesis and its denial

predict for features of the current token; the hypotheses diverge in the context-dependent tail, in the state, and in what can be recovered from far back in the sequence. A study designed to find what the recurrent bottleneck removes would look for features that depend on distant context and test whether they survive in Mamba at the distances where retrieval accuracy falls.

9. The strongest case for universality

The argument so far has been critical, and it is right to state the opposing case at full strength.

First, the best-designed cross-architecture study supports a substantial degree of shared features. Wang et al. (2024) compare Mamba with Pythia against a seed skyline, report both directions, show that matched pairs fall mainly at corresponding depths, repeat the experiment on RWKV, and find the cross-architecture score only slightly below the same-architecture one. That design meets the skyline standard of Section 4.4. If architecture strongly reshaped features, it should have shown a larger gap.

Second, the agreement is not only at Level 1. Interventions developed for Transformers - causal tracing, rank-one editing, steering by activation addition, the tuned lens - work on Mamba with modest changes (Sharma et al., 2024; Paulo et al., 2024), and induction behaviour is implemented by an analogous circuit (Wang et al., 2024). Those are Level 3 and Level 4 observations, and they agree with the Level 1 ones.

Third, the same-architecture baseline is itself low. If only 1 to 5% of GPT-2 neurons are universal across seeds (Gurnee et al., 2024) and only about 30% of a large SAE's latents recur across SAE seeds (Paulo and Belrose, 2025), then perfect cross-architecture agreement was never the right expectation, and a cross-architecture score close to the seed skyline is about as much universality as can be shown with these instruments. Universal neurons, though rare, tend to be interpretable and fall into a few functional families (Gurnee et al., 2024), which is what the hypothesis predicts about the features worth finding.

Fourth, there are theoretical reasons to expect convergence. In embedded deep linear networks trained by stochastic gradient descent, networks of different widths and depths trained on different data provably reach representations that agree up to rotation, even though most global minima of the loss do not (Ziyin and Chuang, 2025). Data with shared abstract structure produce shared mid-depth representations in models trained on it (Doshi et al., 2026). Across scales, Gemma-2 models four times apart in size share matched SAE feature spaces most strongly in middle layers (Son et al., 2025). And after calibration, neighbourhood and relational structure do converge across models and modalities (Groger et al., 2026; You et al., 2026).

The case is strong at the level it reaches. What it supports is that Mamba and Transformers trained on the same corpus with the same tokeniser share a large set of per-token and short-context features at corresponding depths, at about the rate two Transformers trained from different seeds share them, and that several Transformer interpretability methods transfer. It does not yet support the further claims sometimes drawn from it: that the architecture has negligible effect on what a model represents, that the recurrent bottleneck removes nothing of semantic importance, or that representational vocabulary is fixed by data alone. Those claims concern the context-dependent features and state dynamics that Section 8 shows are where the families differ, and that the evidence in this section does not measure. The case would be stronger still if the matching studies stratified their scores by feature complexity and context dependence and reported the skyline gap within each stratum; Wang et al.'s analysis of complexity points in this direction but does not report the gap by stratum in numbers.

10. Consolidated evidence and a licensing procedure

Table 1 sorts into three groups. Rows 1 to 7 are feature-level comparisons, and their verdicts track their reference points: the row with no reference (1) reports the strongest claim; the row with a same-architecture skyline (2) reports agreement close to what same-architecture variation already produces, and rows 6 and 7 are the same-architecture baselines against which it is read; the row with the weakest null (4) reports the largest margin. Rows 8 to 13 concern the measures themselves, and each shows a score that is high or rising for reasons other than shared representation, or a shared geometry that does not carry task information. Rows 14 to 16 locate the behavioural and mechanistic differences between the families in context use and recurrent state.

Algorithm 1 states the decision structure the author proposes for licensing a cross-architecture convergence claim from feature matching. It is meant to be applied to a methods section before a percentage of aligned features is repeated.

Applied to the 2026 study, Algorithm 1 stops at Step 2: no null was computed, so the reported distribution is descriptive. Applied to Wang et al. (2024), it finds no shifted null at Step 2, but the skylines at Step 3 run the same best-match search on both arms, so the maximum is applied equally and the study passes Step 3. It also passes part of Step 4 (both directions are reported) and reaches Level 1 relative to a skyline. Its induction-circuit result is a qualitative Level 4 observation without the seed-variant control that Table 2 requires; the authors write that they "have not yet quantitatively measured the similarity with some automated tools" (Wang et al., 2024). It does not pass Step 5 with numbers per stratum. Applied to Lan et al. (2024), it passes a random-weight

version of the control at Step 6 in spirit and a weak form of Step 2, and does not apply to architecture at all, because every pair compared is a Transformer pair.

11. Studies that would settle the open part

Five studies would answer most of what this paper leaves open. All are feasible with public models and dictionaries.

A rate-matched null for the 2026 data. Recompute every Mamba latent's best-match Jaccard against the Pythia dictionary after shifting Pythia's activation masks by a random offset within each document, repeated enough times to estimate the distribution of the maximum. Report the firing rate of every latent alongside its observed and null best match. If observed scores exceed the null by a wide margin for most latents, the 2026 conclusion stands at Level 1; if they do not, it does not. The source's authors call for a null distribution; this design supplies one that also controls the maximum and the firing rates, and it needs no new training.

A stratified skyline comparison. Take the Wang et al. (2024) design - Mamba, Pythia, a Pythia seed variant, a second SAE on Pythia - and report the cross-architecture gap from the seed skyline separately for features whose activation is determined by the current token, for features that depend on the previous few tokens, and for features whose activation changes when tokens more than, say, a hundred positions back are ablated. The universality hypothesis predicts a small gap in all three strata; the bottleneck hypothesis predicts a gap that grows with the distance of the context a feature depends on. Neither prediction has been tested.

A distance-resolved retrieval probe. At the context lengths where Mamba's phone-book and copying accuracy falls behind Pythia's (Jelassi et al., 2024), train probes and dictionaries for features that encode specific earlier tokens or entities, and measure how their recoverability decays with distance in each model. This converts the capability gap into a representational measurement at the place the gap occurs, and tests whether the "same features" exist in Mamba but cannot be read out, or do not exist at long range.

State-level dictionaries on both sides. Train dictionaries on the recurrent writes of Mamba-2 (Young, 2026) and on the attention outputs or key-value contents of a matched Transformer, and attempt matching at that level with the controls of Algorithm 1. If the residual-stream agreement reflects shared features of the current token, agreement at the level of what is written to memory should be lower.

A different-corpus control. Repeat a skyline comparison with a Mamba and a Transformer trained on different corpora with different tokenisers, matching features at the level of words or spans rather than tokens. If agreement survives a change of inputs, the

Table 1. Published findings on cross-model and cross-architecture similarity, each with the reference point its source reported, as read by the author. Values are as stated in each source's text or tables.

#	Source	Comparison	Statistic	Reported value	Reference point
1	Nagaraj et al. 2026	Mamba-130m vs Pythia-70m, mid layer	best-match Jaccard, binarised	0.234-0.411; 12,285 of 12,288 above 0.35	none; thresholds post hoc
2	Wang et al. 2024	Mamba-130M vs Pythia-160M SAEs	mean max Pearson (MPPC)	0.74	seed skyline 0.76; SAE skyline 0.81
3	Wang et al. 2024	same	features with MPPC > 0.95	over 30%, mostly single-token	neuron basis: near zero
4	Lan et al. 2024	Gemma-1-2B vs instruct, layer 12	paired-feature SVCCA	0.84	random pairing 0.008
5	Lan et al. 2024	Pythia-70m vs random-weight copy	pairs kept after filters	about 1%	custom SAEs 2-4% in any pair; hosted SAEs 20-30%
6	Gurnee et al. 2024	GPT-2 medium, five seeds	excess max correlation > 0.5	1,253 of 98,304 neurons	random rotation
7	Paulo and Belrose 2025	Llama 3 8B, two SAE seeds	latents shared (cos > 0.7)	about 30%	same model, same data
8	Huh et al. 2024	language vs vision models	mutual nearest neighbours	rises to 0.16	metric maximum 1
9	Groger et al. 2026	vision-language, many metrics	null-calibrated similarity	global trend largely gone	permutation null
10	Kapoor et al. 2025	dozens of vision models	alignment over training	nearly all by first epoch	final alignment
11	Heap et al. 2025	Pythia, trained vs random	SAE auto-interpretability	similar in many settings	random-weight model
12	Murphy et al. 2024	random matrices	biased CKA	can reach maximum	debiased CKA
13	Duraphe et al. 2026	36 astronomy models	neighbourhood alignment vs transfer	all 139 aligned somewhere; redshift R2 < 0	label-selected stages
14	Jelassi et al. 2024	Pythia vs Mamba, 3 sizes	phone-book lookup	Pythia ahead at every size	Pile perplexity favours Mamba
15	Waleffe et al. 2024	8B Mamba, Mamba-2, Transformer	12 standard tasks	pure SSMs lag on copying, ICL	hybrid + 2.65 over Transformer
16	Young 2026	Mamba-2-370M recurrent writes	rank-one atom replacement	closer than deletion on 88.1%	deleting the write

Table 2. Four levels of a "same features" claim, the instruments that reach each, and the control that each requires (the author's synthesis).

Level	Claim	Instruments	Required control
1 Co-activation	units fire on the same inputs	max correlation, Jaccard	rate-matched shifted null; seed skyline
2 Geometry	relations among units agree	CKA, SVCCA, RSA, mutual kNN	width- and selection-calibrated null
3 Use	matched units play the same causal role	stitching, transferred steering, crosscoders	direct matching; random-weight model
4 Computation	features computed by analogous circuits	circuit analysis, state-level dictionaries	same analysis on a seed variant

data-driven reading of convergence gains a test it has not yet had; if it falls to the level of a random-weight control, the cross-architecture agreement on the Pile is evidence about the Pile.

In all five, the reference distribution should be computed for the statistic actually reported, the same pipeline should be run on a same-architecture pair, and results should be broken down by feature type before any aggregate is given.

12. What is not known

Whether the 2026 study's best-match Jaccard scores exceed what unrelated latents with the same firing rates would produce is not known, because firing rates and a null were not reported.

Whether Mamba and Transformers trained on the same corpus differ more in context-dependent features than in token-level features is not known. No study has reported,

in numbers, the cross-architecture gap from a same-architecture skyline separately by feature type.

What the recurrent bottleneck removes from a model's representation, if anything, at the distances where its retrieval accuracy falls, is not known. The capability evidence shows a behavioural gap; no representational study has measured at those distances.

Whether cross-architecture feature agreement survives a change of training corpus and tokeniser is not known. Every Mamba-Transformer feature comparison reviewed here holds both fixed.

Which similarity measure, if any, should be treated as the reference for cross-architecture claims is not settled. Calibrated neighbourhood and relational measures are the best-supported candidates for Level 2 (Groger et al., 2026; You et al., 2026); for Level 1 there is no agreed standard beyond the skyline.

Algorithm 1. Licensing a cross-architecture "same features" claim.

```

Input: models A, B of different architectures; dictionaries D_A, D_B;
      a similarity statistic S and a reported aggregate (e.g. % above t).
Step 1 INPUTS. Do A and B share tokeniser and training corpus?
  yes -> per-token features will co-activate regardless of architecture;
        go to Step 5 before reading any aggregate.
Step 2 NULL. Compute the null of the reported statistic itself: for each
  feature, best-match S against D_B with token correspondence broken
  (random circular shift), firing rates preserved. Report observed
  minus null maximum per feature.  pass -> Level 1 against chance.
  For a geometric score over matched pairs (SVCCA, RSA, CKA), use a
  width- and selection-calibrated null.  pass -> Level 2.
  no null -> go to Step 3; if it is not run either, the score is
  descriptive only and no level is licensed.
Step 3 SKYLINE. Run the identical pipeline between A and a seed variant
  of A, and between two dictionaries on A. The same best-match search
  then applies the same maximum to both arms, so the skyline can stand
  in for the null of Step 2. Report the cross-architecture score as a
  gap from these skylines.  pass -> Level 1 relative to the skyline.
  gap not reported -> claim co-activation, not universality.
Step 4 SYMMETRY. Match in both directions with dictionaries of equal size
  and family; repeat with a second width or family.
  asymmetric -> divergence in the unreported direction is unmeasured.
Step 5 STRATIFY. Split features into single-token and context-dependent
  (e.g. by top-activating contexts or ablation of prior tokens). Report
  Steps 2-3 within each stratum.
Step 6 FUNCTION. For matched pairs, test whether intervening on the pair
  transfers through a directly matched map; compare with a random-weight
  control.  pass -> Level 3 for that stratum.
Step 7 LOCATION. Where the architectures differ (attention vs state),
  repeat Steps 2-6 with an instrument for that component. For a claim
  about mechanism, compare the circuits computing matched features and
  run the same circuit analysis on a seed variant of A.
  circuit gap within the seed-variant range -> Level 4.
Output: the highest level passed, scoped to stratum, layer, corpus,
  tokeniser, metric and dictionary family.

```

13. Limitations

The paper reads published results and is limited by what they report. Its central quantitative argument, the independence null for Jaccard, is a derivation; it shows what the 2026 numbers would mean under different firing rates and cannot show which rates obtained. The claim that the narrow reported band is more consistent with dense features than with a mixture of twins and orphans is the author's inference and is offered as a hypothesis for the null study of Section 11 to test. The four levels of Section 2 and the procedure of Algorithm 1 are the author's synthesis; other partitions are possible, and the procedure has not been applied by anyone to new data. Several of the sources are 2026 preprints without peer review, including the anchor study; their numbers are reported as their authors state them, from text and tables rather than from figures. The comparison between the Jaccard and correlation results in Section 7 is between studies that differ in Pythia size, layer and corpus, and is offered as an illustration of instrument dependence rather than as a measured contrast. The literature search covered arXiv and the venues cited by the sources through mid-October 2026; work outside it, including unpublished replications, may change parts of the account.

14. Conclusion

A study that finds a counterpart for 99.98% of one architecture's features in another has answered a question, and the question matters. But a best match is a maximum, and a maximum always exists. Whether it is a twin depends on what unrelated features would score, on whether the two models had any choice but to agree, and on whether the instrument used to find features would find the same ones twice in one model.

On the evidence reviewed here, Mamba and Transformers trained on the same text with the same tokeniser share many features of the current token and short context, at close to the rate that two Transformers trained from different seeds do, and several interpretability methods transfer between them. That is a real and useful result. It is not yet evidence that architecture leaves representation untouched, because the comparisons were made where both families must agree and not where they are known to differ. The 2026 study's own numbers cannot decide even the narrower question until its firing rates and a null are reported. The honest summary is the one the strongest design supports: many features recur across the two families, the reference point decides how many count, and what the recurrent state removes has not yet been measured.

References

- Abbes, I., Islah, N., Rish, I., & Chandar, S. (2026). What Does Layer-Importance Reveal About Transformers and State-Space Models? arXiv:2609.16537. doi:10.48550/arXiv.2609.16537. [PREPRINT]
- Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., & Kim, B. (2018). Sanity Checks for Saliency Maps. arXiv:1810.03292. doi:10.48550/arXiv.1810.03292.
- Ali, A., Zimerman, I., & Wolf, L. (2024). The Hidden Attention of Mamba Models. arXiv:2403.01590. doi:10.48550/arXiv.2403.01590. [PREPRINT]
- Arora, S., Eyuboglu, S., Timalsina, A., Johnson, I., Poli, M., Zou, J., Rudra, A., & Re, C. (2023). Zoology: Measuring and Improving Recall in Efficient Language Models. arXiv:2312.04927. doi:10.48550/arXiv.2312.04927. [PREPRINT]
- Balogh, A., & Jelasity, M. (2024). How not to Stitch Representations to Measure Similarity: Task Loss Matching versus Direct Matching. AAAI 2025. arXiv:2412.11299. doi:10.48550/arXiv.2412.11299.
- Bansal, Y., Nakkiran, P., & Barak, B. (2021). Revisiting Model Stitching to Compare Neural Representations. arXiv:2106.07682. doi:10.48550/arXiv.2106.07682.
- Biderman, S., Schoelkopf, H., Anthony, Q., Bradley, H., O'Brien, K., Hallahan, E., et al. (2023). Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling. arXiv:2304.01373. doi:10.48550/arXiv.2304.01373.
- Black, S., Biderman, S., Hallahan, E., Anthony, Q., Gao, L., Golding, L., et al. (2022). GPT-NeoX-20B: An Open-Source Autoregressive Language Model. ACL Workshop on Challenges and Perspectives in Creating Large Language Models. arXiv:2204.06745. doi:10.48550/arXiv.2204.06745.
- Bussmann, B., Nabeshima, N., Karvonen, A., & Nanda, N. (2025). Learning Multi-Level Features with Matryoshka Sparse Autoencoders. arXiv:2503.17547. doi:10.48550/arXiv.2503.17547. [PREPRINT]
- Chanin, D., Wilken-Smith, J., Dulka, T., Bhatnagar, H., Golechha, S., & Bloom, J. (2024). A is for Absorption: Studying Feature Splitting and Absorption in Sparse Autoencoders. NeurIPS 2025. arXiv:2409.14507. doi:10.48550/arXiv.2409.14507.
- Cho, S., Wu, Z., Da Costa, K., Kalra, R., Wicaksono, I., & Koshiyama, A. (2026). Are Single-Token Sparse Autoencoder Features Causally Necessary? Layer-Depth and SAE-Family Effects. arXiv:2607.20596. doi:10.48550/arXiv.2607.20596. [PREPRINT]
- Chughtai, B., Chan, L., & Nanda, N. (2023). A Toy Model of Universality: Reverse Engineering How Networks Learn Group Operations. arXiv:2302.03025. doi:10.48550/arXiv.2302.03025.
- Ciernik, L., Linhardt, L., Morik, M., Dippel, J., Kornblith, S., & Muttenthaler, L. (2024). Objective drives the consistency of representational similarity across datasets. arXiv:2411.05561. doi:10.48550/arXiv.2411.05561. [PREPRINT]
- Cunningham, H., Ewart, A., Riggs, L., Huben, R., & Sharkey, L. (2023). Sparse Autoencoders Find Highly Interpretable Features in Language Models. arXiv:2309.08600. doi:10.48550/arXiv.2309.08600.
- Dao, T., & Gu, A. (2024). Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality. ICML 2024. arXiv:2405.21060. doi:10.48550/arXiv.2405.21060.
- Davari, M., Horoi, S., Natik, A., Lajoie, G., Wolf, G., & Belilovsky, E. (2022). Reliability of CKA as a Similarity Measure in Deep Learning. arXiv:2210.16156. doi:10.48550/arXiv.2210.16156.
- Ding, F., Denain, J.-S., & Steinhardt, J. (2021). Grounding Representation Similarity with Statistical Testing. NeurIPS 2021. arXiv:2108.01661. doi:10.48550/arXiv.2108.01661.
- Doshi, D., Zhou, W., Wegner, C. E., Korchinski, D. J., Acevedo, S., & Wyart, M. (2026). A theory of platonic representations in language models. arXiv:2610.07168. doi:10.48550/arXiv.2610.07168. [PREPRINT]
- Dravid, A., Gandelman, Y., Efros, A. A., & Shocher, A. (2023). Rosetta Neurons: Mining the Common Units in a Model Zoo. arXiv:2306.09346. doi:10.48550/arXiv.2306.09346.
- Duraphe, K., Kannappan, A., Chan, D. L., Famutimi, O., Ejagiri, Y. S., Smith, M. J., & Wu, J. F. (2026). Shared Geometry Is Not Shared Physics: A Layerwise Test of the Platonic Representation Hypothesis in Astronomy. arXiv:2610.04130. doi:10.48550/arXiv.2610.04130. [PREPRINT]
- Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., et al. (2022). Toy Models of Superposition. arXiv:2209.10652. doi:10.48550/arXiv.2209.10652.
- Endy, N., Grosbard, I. D., Ran-Milo, Y., Slutzky, Y., Tshuva, I., & Giryres, R. (2025). Mamba Knockout for Unraveling Factual Information Flow. ACL 2025. arXiv:2505.24244. doi:10.48550/arXiv.2505.24244.
- Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., et al. (2020). The Pile: An 800GB Dataset of Diverse Text for Language Modeling. arXiv:2101.00027. doi:10.48550/arXiv.2101.00027.
- Groger, F., Wen, S., & Brbic, M. (2026). Revisiting the Platonic Representation Hypothesis: An Aristotelian View. ICML 2026. arXiv:2602.14486. doi:10.48550/arXiv.2602.14486.
- Gu, A., & Dao, T. (2023). Mamba: Linear-Time Sequence Modeling with Selective State Spaces. arXiv:2312.00752. doi:10.48550/arXiv.2312.00752.
- Gurnee, W., Horsley, T., Guo, Z. C., Kheirhah, T. R., Sun, Q., Hathaway, W., Nanda, N., & Bertsimas, D. (2024). Universal Neurons in GPT2 Language Models. arXiv:2401.12181. doi:10.48550/arXiv.2401.12181.
- Heap, T., Lawson, T., Farnik, L., & Aitchison, L. (2025). Automated Interpretability Metrics Do Not Distinguish Trained and Random Transformers. arXiv:2501.17727. doi:10.48550/arXiv.2501.17727. [PREPRINT]
- Hernandez, A., Dangovski, R., Lu, P. Y., & Soljacic, M. (2023). Model Stitching: Looking For Functional Similarity Between Representations. arXiv:2303.11277. doi:10.48550/arXiv.2303.11277.
- Huh, M., Cheung, B., Wang, T., & Isola, P. (2024). The Platonic Representation Hypothesis. ICML 2024. arXiv:2405.07987. doi:10.48550/arXiv.2405.07987.
- Jelassi, S., Brandfonbrener, D., Kakade, S. M., & Malach, E. (2024). Repeat After Me: Transformers are Better than State Space Models at Copying. arXiv:2402.01032. doi:10.48550/arXiv.2402.01032.
- Jiralspong, T., & Bricken, T. (2026). Cross-Architecture Model Diffing with Crosscoders: Unsupervised Discovery of Differences Between LLMs. arXiv:2602.11729. doi:10.48550/arXiv.2602.11729. [PREPRINT]
- Kapoor, C., Srivastava, S., & Khosla, M. (2025). Bridging Critical Gaps in Convergent Learning: How Representational Alignment Evolves Across Layers, Training, and Distribution Shifts. arXiv:2502.18710. doi:10.48550/arXiv.2502.18710. [PREPRINT]
- Klabunde, M., Schumacher, T., Strohmaier, M., & Lemmerich, F. (2023a). Similarity of Neural Network Models: A Survey of Functional and Representational Measures. ACM Computing Surveys 57(9), 2025. arXiv:2305.06329. doi:10.48550/arXiv.2305.06329.
- Klabunde, M., Ben Amor, M., Granitzer, M., & Lemmerich, F. (2023b). Towards Measuring Representational Similarity of Large Language Models. UniReps Workshop at NeurIPS 2023. arXiv:2312.02730. doi:10.48550/arXiv.2312.02730.
- Kornblith, S., Norouzi, M., Lee, H., & Hinton, G. (2019). Similarity of Neural Network Representations Revisited. ICML 2019. arXiv:1905.00414. doi:10.48550/arXiv.1905.00414.
- Lan, M., Torr, P., Meek, A., Khakzar, A., Krueger, D., & Barex, F. (2024). Quantifying Feature Space Universality Across Large Language Models via Sparse Autoencoders. arXiv:2410.06981. doi:10.48550/arXiv.2410.06981. [PREPRINT]

- Leask, P., Busmann, B., Pearce, M., Bloom, J., Tigges, C., Al Moubayed, N., Sharkey, L., & Nanda, N. (2025). Sparse Autoencoders Do Not Find Canonical Units of Analysis. ICLR 2025. arXiv:2502.04878. doi:10.48550/arXiv.2502.04878.
- Lenc, K., & Vedaldi, A. (2014). Understanding image representations by measuring their equivariance and equivalence. arXiv:1411.5908. doi:10.48550/arXiv.1411.5908.
- Li, Y., Yosinski, J., Clune, J., Lipson, H., & Hopcroft, J. (2015). Convergent Learning: Do different neural networks learn the same representations? ICLR 2016. arXiv:1511.07543. doi:10.48550/arXiv.1511.07543.
- Merrill, W., Petty, J., & Sabharwal, A. (2024). The Illusion of State in State-Space Models. ICML 2024. arXiv:2404.08819. doi:10.48550/arXiv.2404.08819.
- Minder, J., Dumas, C., Juang, C., Chugtai, B., & Nanda, N. (2025). Overcoming Sparsity Artifacts in Crosscoders to Interpret Chat-Tuning. NeurIPS 2025. arXiv:2504.02922. doi:10.48550/arXiv.2504.02922.
- Mohan, V. S., Gupta, K., Das, A., & Singh, C. (2026). Interpreting and Steering State-Space Models via Activation Subspace Bottlenecks. arXiv:2602.22719. doi:10.48550/arXiv.2602.22719. [PREPRINT]
- Morcos, A. S., Raghu, M., & Bengio, S. (2018). Insights on representational similarity in neural networks with canonical correlation. NeurIPS 2018. arXiv:1806.05759. doi:10.48550/arXiv.1806.05759.
- Moschella, L., Maiorca, V., Fumero, M., Norelli, A., Locatello, F., & Rodola, E. (2022). Relative representations enable zero-shot latent space communication. ICLR 2023. arXiv:2209.15430. doi:10.48550/arXiv.2209.15430.
- Murphy, A., Zylberberg, J., & Fyshe, A. (2024). Correcting Biased Centered Kernel Alignment Measures in Biological and Artificial Neural Networks. ICLR 2024 Re-Align Workshop. arXiv:2405.01012. doi:10.48550/arXiv.2405.01012.
- Nagaraj, R., Oruganti, R. L., Kunder, P. S., & Joshi, A. M. (2026). Comparing Latent Concept Formation in State Space Models and Transformers via Sparse Autoencoders. arXiv:2609.24440. doi:10.48550/arXiv.2609.24440. [PREPRINT]
- Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., & Carter, S. (2020). Zoom In: An Introduction to Circuits. Distill. doi:10.23915/distill.00024.001.
- Pantazopoulos, G., Nikandrou, M., Konstantas, I., & Suglia, A. (2026). Retrievit: In-context Retrieval Capabilities of Transformers, State Space Models, and Hybrid Architectures. arXiv:2603.02874. doi:10.48550/arXiv.2603.02874. [PREPRINT]
- Park, J., Park, J., Xiong, Z., Lee, N., Cho, J., Oymak, S., Lee, K., & Papailiopoulos, D. (2024). Can Mamba Learn How to Learn? A Comparative Study on In-Context Learning Tasks. arXiv:2402.04248. doi:10.48550/arXiv.2402.04248.
- Paulo, G., & Belrose, N. (2025). Sparse Autoencoders Trained on the Same Data Learn Different Features. arXiv:2501.16615. doi:10.48550/arXiv.2501.16615. [PREPRINT]
- Paulo, G., Marshall, T., & Belrose, N. (2024). Does Transformer Interpretability Transfer to RNNs? arXiv:2404.05971. doi:10.48550/arXiv.2404.05971. [PREPRINT]
- Raghu, M., Gilmer, J., Yosinski, J., & Sohl-Dickstein, J. (2017). SVCCA: Singular Vector Canonical Correlation Analysis for Deep Learning Dynamics and Interpretability. NIPS 2017. arXiv:1706.05806. doi:10.48550/arXiv.1706.05806.
- Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., & Dosovitskiy, A. (2021). Do Vision Transformers See Like Convolutional Neural Networks? arXiv:2108.08810. doi:10.48550/arXiv.2108.08810.
- Sharma, A. S., Atkinson, D., & Bau, D. (2024). Locating and Editing Factual Associations in Mamba. COLM 2024. arXiv:2404.03646. doi:10.48550/arXiv.2404.03646.
- Son, D., Rathore, S., Rufail, A., Simon, A., Zhang, D., Dave, S., Blondin, C., Zhu, K., & O'Brien, S. (2025). Semantic Convergence: Investigating Shared Representations Across Scaled LLMs. arXiv:2507.22918. doi:10.48550/arXiv.2507.22918. [PREPRINT]
- Sun, X., Stolfo, A., Engels, J., Wu, B., Rajamanoharan, S., Sachan, M., & Tegmark, M. (2025). Dense SAE Latents Are Features, Not Bugs. NeurIPS 2025. arXiv:2506.15679. doi:10.48550/arXiv.2506.15679.
- Thasarthan, H., Forsyth, J., Fel, T., Kowal, M., & Derpanis, K. G. (2025). Universal Sparse Autoencoders: Interpretable Cross-Model Concept Alignment. arXiv:2502.03714. doi:10.48550/arXiv.2502.03714. [PREPRINT]
- Waleffe, R., Byeon, W., Riach, D., Norick, B., Korthikanti, V., Dao, T., Gu, A., et al. (2024). An Empirical Study of Mamba-based Language Models. arXiv:2406.07887. doi:10.48550/arXiv.2406.07887.
- Wang, J., Ge, X., Shu, W., Tang, Q., Zhou, Y., He, Z., & Qiu, X. (2024). Towards Universality: Studying Mechanistic Similarity Across Language Model Architectures. arXiv:2410.06672. doi:10.48550/arXiv.2410.06672. [PREPRINT]
- Williams, A. H., Kunz, E., Kornblith, S., & Linderman, S. W. (2021). Generalized Shape Metrics on Neural Representations. NeurIPS 2021. arXiv:2110.14739. doi:10.48550/arXiv.2110.14739.
- You, J., Jang, M., Mo, S., & Jung, J.-H. (2026). What Converges in the Platonic Representation Hypothesis? Structure over Geometry. arXiv:2609.27252. doi:10.48550/arXiv.2609.27252. [PREPRINT]
- Young, J. (2026). WriteSAE: Sparse Autoencoders for Recurrent State. arXiv:2605.12770. doi:10.48550/arXiv.2605.12770. [PREPRINT]
- Zhong, Z., Liu, Z., Tegmark, M., & Andreas, J. (2023). The Clock and the Pizza: Two Stories in Mechanistic Explanation of Neural Networks. NeurIPS 2023. arXiv:2306.17844. doi:10.48550/arXiv.2306.17844.
- Ziyin, L., & Chuang, I. (2025). Proof of a perfect platonic representation hypothesis. arXiv:2507.01098. doi:10.48550/arXiv.2507.01098. [PREPRINT]