Rotation in Language Models: Simple Structure, Changes of Basis, and Conceptors

What factor rotation can and cannot do for the residual stream of a transformer, an algebraic proof with numerical verification that an obliquely rotated model is the same model, and a reading of conceptors in rotated coordinates

Tyler M. Moore[1,2] · Lyle H. Ungar[3] · Ilias Triantafyllopoulos[4] · Ren Tao[3] · Neville G. Ryant[5] · Sunghye Cho[6] · João Sedoc[4]

[1] Department of Psychiatry, Perelman School of Medicine, University of Pennsylvania · [2] Lifespan Brain Institute, Penn Medicine and Children's Hospital of Philadelphia · [3] Department of Computer and Information Science, University of Pennsylvania · [4] Department of Technology, Operations, and Statistics, Stern School of Business, New York University · [5] Linguistic Data Consortium, University of Pennsylvania · [6] Department of Communication Sciences and Disorders, University of Cincinnati

Manuscript · October 2026 Correspondence: tymoore@pennmedicine.upenn.edu ORCID 0000-0002-1384-0151

DOI 10.5281/zenodo.23719033

Recommended citation: Moore, T. M., Ungar, L. H., Triantafyllopoulos, I., Tao, R., Ryant, N. G., Cho, S., & Sedoc, J. (2026). *Rotation in language models: Simple structure, changes of basis, and conceptors*. Manuscript retrieved from https://mooremetrics.com/rotation_in_language_models. https://doi.org/10.5281/zenodo.23719033

ABSTRACT

The residual stream of a transformer has no privileged coordinate system, so any description of it begins by choosing one; factor rotation is the discipline of choosing coordinates that can be read. This report applies that discipline at the four places it can be applied to a language model, using GPT-2 small, and asks what it buys for conceptors, the soft projection matrices used to read and steer a model's activations. (1) The 50,257 × 768 token lookup table, rotated to simple structure: varimax yields hundreds of nameable factors (word families, numbers, years, suffixes, formats) and one giant factor that is the table's dominant common direction; promax recovers the same group factors without reducing cross-loadings, and its factor correlation matrix has a single large eigenvalue, the signature of a general factor. An exploratory bifactor rotation (bi-quartimin) models that structure directly: a general factor loading positively on every token, associated with low word frequency ($r = -0.55$ with Zipf frequency over 18,770 words), and group factors whose poles are no longer mixed with the common direction. (2) The whole model: a change of basis of the residual stream, orthogonal or oblique, leaves the model exactly unchanged once the weights that read from and write to the stream are counter-transformed and the normalization layer carries the induced metric. The orthogonal case is known; the oblique case is a small extension with a useful reading, that the input and output embeddings then stand as pattern matrix to structure matrix. GPT-2 rewritten in promax coordinates reproduces its own logits to 10^{-10}; the same rewrite with ordinary LayerNorm is a different model. (3) Conceptors: they transform covariantly under any invertible change of basis, with gains, quota, action and Boolean-style operations invariant, but the aperture regularizer is metric-dependent, so a conceptor re-fitted naively in oblique coordinates is a different filter, wrong by tens of

percent. (4) A layer: rotating a layer's activations gives interpretable axes for that layer, and reading a concept on them needs baselines. Against random-direction chance, a sentiment direction is absent from the leading directions of a general corpus until the middle of the model; against a label-shuffled null, and against the corpus's own sentence-final states as the reference set, the depth trend disappears and the direction stands out at one layer only, the layer at which the sentences are also best separated. Rotation inside the concept's own subspace yields an unsupervised axis that separates held-out sentence pairs by sentiment, less sharply than the supervised difference-of-means direction does, and names the nuisance variation beside it. Proposals for using rotation in conceptor practice follow, with the caveat that they are untested.

Keywords: factor rotation; varimax; promax; bifactor rotation; simple structure; residual stream; privileged basis; conceptors; activation steering; interpretability; GPT-2

1 Introduction

A transformer language model stores each token of its vocabulary as a row of numbers, and processes text by repeatedly rewriting a table of such rows, one per token position, through a stack of layers. Every row lives in the same vector space, called the residual stream, and the coordinates of that space are arbitrary in one precise sense: a rotated copy of the model, with the weights rotated to match, computes exactly the same function. Any description of the stream therefore begins, whether it says so or not, by choosing coordinates. This is the first thing a psychometrician recognizes. A loading matrix is likewise defined only up to rotation, and a century of factor-analytic practice has been devoted to choosing the rotation that makes the loadings readable.

The question of this report is what happens when that practice (varimax, promax, bifactor rotation, the pattern/structure distinction, the general factor) is applied to a language model, at each of the four places it can be applied: to the *table* of token vectors the model starts from (Section 3); to the *model* as a whole, which raises the question of whether an oblique transformation, not a rotation but an arbitrary invertible change of coordinates, can be used without changing it (Section 4); to a *conceptor* (Jaeger, 2014), a soft projection matrix fitted to a layer's activations and used to read and steer what the model is doing (Section 5); and to a *layer*, the activations themselves (Section 6). The conceptor is the application that motivates the work, and it is also what ties the four places together, because a conceptor is a factor-analytic object: it is built from the covariance matrix of a layer's activations, and its eigen-decomposition is an unrotated principal-components solution, so everything that factor analysis knows about rotating such a solution applies to it. Section 7 turns the findings into proposals for conceptor practice.

The tone is deliberately didactic. Derivations are carried out in full, assumptions are stated, and every numerical claim comes from a computation on GPT-2 small (Radford et al., 2019) that can be re-run from the accompanying scripts given the public model weights (Appendix A). The intended reader

knows factor analysis and is willing to learn the two or three facts about transformers that matter here; Section 2 supplies them. Readers who know transformers but not factor analysis will find the rotation facts they need in the same section.

1.1 What is established, and what this report adds

Rotating a word-embedding table to interpretable axes is not new. Park, Bak and Oh (2017) applied the Crawford–Ferguson family, in orthogonal and oblique form, to full skip-gram and GloVe tables and scored the axes by an automated word-intrusion test; Hollis and Westbury (2016) and Westbury, Yang and Anderson (2025) ran principal components and factor analysis on word2vec from the psychology side; the ICA line (Yamagiwa, Oyama & Shimodaira, 2023; Musil & Mareček, 2024) finds interpretable axes by a different rotation criterion; Rothe, Ebert and Schütze (2016) and Dufter and Schütze (2019) learn orthogonal transformations that concentrate a labelled property on a few axes; and Rohe and Zeng (2023) supplied the statistical theory under which principal components followed by varimax recovers sparse latent factors. Sparse overcomplete codings of a table (Murphy, Talukdar & Mitchell, 2012; Faruqui et al., 2015; Subramanian et al., 2018) and sparse autoencoders for a layer's activations (Bricken et al., 2023; Huben et al., 2024) are the overcomplete, nonlinear descendants of the same idea, and Arora et al. (2018) model a word vector as a sparse sum of "discourse atoms", which is the cross-loading phenomenon of Section 3 seen from a generative model. That embedding spaces have a dominant common direction, and why, is also known: Gao et al. (2019) showed that likelihood training with tied embeddings drives the vocabulary into a narrow cone, and Biś, Podkorytov and Liu (2021) traced the drift to the gradient that every non-target token receives at every step; the outlier coordinates this produces are documented by Kovaleva et al. (2021), Timkey and van Schijndel (2021) and Puccetti et al. (2022), and the anisotropy of contextual representations by Ethayarajh (2019) and Mu and Viswanath (2018). Conceptors have been applied to embedding tables, notably by Liu, Ungar and Sedoc (2019a), whose "conceptor negation" removes a table's dominant directions, and to debiasing (Karve, Ungar & Sedoc, 2019; Yifei, Ungar & Sedoc, 2023). Section 3 is, in that light, a replication on a transformer's table with the full psychometric toolkit; its value is in the detail, and in modelling the general factor as one.

That a transformer is unchanged by an *orthogonal* rotation of its residual stream is also established. Elhage et al. (2021) note the freedom to reparameterize the stream and the related freedom inside each attention head (only the products $W_Q W_K^T$ and $W_O W_V$ are determined); Elhage, Lasenby and Olah (2023) state the residual-stream version as the absence of a privileged basis; SliceGPT (Ashkboos, Croci et al., 2024) proves it as a theorem, handles LayerNorm by converting it to RMSNorm, and uses it to compress models; and QuaRot (Ashkboos, Mohtashami et al., 2024) uses the same invariance with Hadamard rotations to remove outliers before quantization. Three things need keeping apart here: that a model admits equivalent coordinate descriptions (exact); that the coordinates it was trained in nevertheless carry empirical structure, which Elhage et al. (2023) document and Section 3.5 meets in the form of outlier coordinates; and that a different basis may be more useful for description. The first does not deny the second, and the third is what this report is about.

Against that background, what this report adds is modest in algebra and, we think, useful in practice:

1. **The oblique change of basis, stated and verified** (Section 4). Extending the orthogonal invariance to any invertible transformation is a small step once the normalization layer is given the induced metric; what the step buys is a reading, that the input and output tables then stand in the pattern-to-structure relation of oblique factor analysis, and a warning, that the rewrite which forgets the metric is a different model, which the verification on GPT-2 makes concrete.
2. **The transformation law of conceptors** (Section 5). Under a change of basis a conceptor transforms as $M^{-1}CM$; its gains, quota, action and Boolean-style operations are invariant; the aperture regularizer is not, and the fitting formula in oblique coordinates acquires the metric. The algebra is elementary; the consequence, that a conceptor fitted naively in oblique coordinates is wrong by tens of percent while looking reasonable, is the practical point.
3. **Conceptors read on rotated axes** (Section 6). A conceptor's gain along each axis of a rotated, interpretable basis of its layer is a labelled directional summary of what it keeps, and rotation inside the conceptor's own subspace yields an unsupervised axis that separates held-out sentences by the concept, though not as sharply as the supervised direction itself, and names the nuisance variation beside it. What rotation adds there is names, not separation, and the report says so.
4. **Where in the model a concept is, measured against baselines** (Section 6). Against random-direction chance, a sentiment concept is absent from the leading directions of a general corpus until the middle of the model; but a label-shuffled null rises with depth in the same way, and against a reference set matched to the concept's own kind of state the concept stands out at one layer only, the one at which the sentences are also best separated. The baselines, not the depth trend, are what bears on choosing a layer for conceptor fitting.

Section 7 turns these into proposals for conceptor practice. They are proposals; none has been tested beyond the demonstrations here.

2 Background: the residual stream, factor rotation, conceptors

This section fixes notation and collects the facts used later. Vectors are *rows* throughout, so that a linear map acts by right-multiplication, $h \to hM$; this is the convention of the model's own code and it keeps the factor-analytic formulas in their familiar shape.

2.1 The residual stream of a transformer

GPT-2 small has a vocabulary of $V = 50{,}257$ tokens and a width of $d = 768$. Its **lookup table** E is a $V \times d$ matrix, one row per token; a second table P holds one row per position. Reading a text of T tokens, the model forms the $T \times d$ matrix H_0 whose t-th row is $E_{\text{token}(t)} + P_t$, and then applies twelve blocks, each of which *adds* an update to every row:

$$H \leftarrow H + \mathrm{Attn}(\mathrm{LN}_1(H)), \quad H \leftarrow H + \mathrm{MLP}(\mathrm{LN}_2(H)). \tag{1}$$

The sequence of H matrices, from H_0 to H_{12}, is the **residual stream**; "layer ℓ" below means the state after ℓ blocks, so layer 12 is the state after the last block and before the final normalization. The attention and MLP sub-blocks each begin by *reading* the stream through weight matrices (the query, key and value maps W_Q, W_K, W_V, and the MLP's input map W_{in}, all of shape $d \times$ something) and end by *writing* to it through others (W_O and W_{out}, of shape something $\times d$). Everything in between, the softmax over attention scores and the GELU nonlinearity, acts on the *read-out* values, never on the stream's coordinates directly. The one exception is the normalization layer. GPT-2 uses LayerNorm (Ba, Kiros & Hinton, 2016):

$$\mathrm{LN}(h) = \gamma \odot (h - \mu(h)\mathbf{1}) \,/\, \sigma(h) + \beta, \quad \mu(h) = h\cdot\mathbf{1}/d, \quad \sigma(h)^2 = \|h - \mu\mathbf{1}\|^2/d + \varepsilon, \tag{2}$$

where $\mathbf{1}$ is the all-ones row vector, γ and β are learned elementwise gains and biases, and $\varepsilon = 10^{-5}$ is a stabilizer. LayerNorm therefore does two things that depend on the coordinate system: it subtracts the mean *along the all-ones direction*, and it divides by a length measured with the Euclidean metric. Modern models (Llama, Qwen, Gemma) use RMSNorm (Zhang & Sennrich, 2019), which omits the centring; the difference matters in Section 4. Finally the output is read through the **tied unembedding**: logits $= \mathrm{LN}_f(H_{12})\, E^{\mathrm{T}}$, the same table used backwards.

Nothing above names a coordinate. If every row of the stream is replaced by hQ for an orthogonal Q, and every weight that reads is replaced by $Q^{\mathrm{T}}W$, every weight that writes by WQ, the reads and writes are unchanged and so is the model; the only thing that could notice is LayerNorm, through the all-ones direction. Elhage, Lasenby and Olah (2023) call this the absence of a privileged basis, while showing that the coordinates a model is trained in do acquire empirical structure (a few of GPT-2's coordinates carry unusually large activations, a point Section 3.5 returns to); SliceGPT (Ashkboos, Croci et al., 2024) states the invariance as a theorem and uses it. The consequence for this report is modest and exact: a rotation chosen for interpretability is as legitimate a description of the model as the trained coordinates, in the way a rotated factor solution is as legitimate as the unrotated one.

2.2 Factor rotation in one page

Let Λ be a $p \times k$ loading matrix: p indicators, k factors. Any orthogonal $k \times k$ matrix Q gives an equally good solution ΛQ; rotation chooses Q to make the loadings readable, usually by Thurstone's (1947) *simple structure*: each indicator loads on few factors and each factor on few indicators. **Varimax** (Kaiser, 1958) maximizes, over orthogonal Q, the sum over factors of the variance of the squared loadings; with *Kaiser normalization* each row of Λ is first scaled to unit length so that long rows do not dominate. Oblique rotations allow correlated factors. Writing the transformation as an invertible T, the **pattern** matrix is $\Lambda_p = \Lambda T$ (regression weights of indicators on factors), the **factor correlation** matrix is $\Phi = (T^{\mathrm{T}}T)^{-1}$, and the **structure** matrix is $\Lambda_s = \Lambda_p\Phi = \Lambda T^{-\mathrm{T}}$ (correlations of indicators with factors). For orthogonal T the two coincide. The oblique method used here is **promax** (Hendrickson & White, 1964): raise the varimax loadings to a power (4) to sharpen them, fit that target by least squares, and rescale. (Direct oblimin, Jennrich & Sampson, 1966, was also run on the full table; it recovered the same factors with the same conclusions, and is omitted to keep the comparison to one oblique method.)

When a solution has a *general factor*, a direction almost every indicator shares, neither orthogonal nor oblique simple-structure rotation represents it well: an orthogonal rotation must put it in one column, which then dominates, and an oblique rotation disperses it into the factor correlations. The model built for this situation is the **bifactor** model (Holzinger & Swineford, 1937; Reise, Moore & Haviland, 2010): one general factor on which every indicator loads, plus group factors each indicator loads on at most one of, all mutually orthogonal. The exploratory version used here is the **bi-quartimin** rotation of Jennrich and Bentler (2011): rotate Λ orthogonally to minimize the quartimin criterion applied to all columns but the first, $\Sigma_i \Sigma_{1<r<s} \lambda_{ir}^2 \lambda_{is}^2$, which is zero exactly when the group columns have perfect cluster structure and leaves the first column free to hold whatever is common. It is computed by the gradient projection algorithm (Jennrich, 2001; Bernaards & Jennrich, 2005). Jennrich and Bentler show that the older Schmid–Leiman (1957) procedure, which first fits an oblique solution and then orthogonalizes it through a second-order factor, imposes proportionality constraints and can fail badly when the data have genuine bifactor structure (see also Reise, Moore & Haviland, 2010; Abad et al., 2017); it is not used here. As a check on the analytic rotation, which is known to be prone to local minima and to shift variance onto the general factor (Mansolf & Reise, 2016), a partially specified target rotation (Browne, 2001) was also run, with the general column unspecified and the group columns given zeros wherever the varimax loading fell below 0.2.

Simple structure is measured in three ways below: Hofmann's (1978) row complexity $(\Sigma\lambda^2)^2/\Sigma\lambda^4$, which is 1 for an indicator loading on one factor and k for one loading equally on all; the number of *salient* loadings per row ($|\lambda| \geq 0.3$); and the share of a row's squared coefficients carried by its largest coefficient. For an orthogonal solution of unit-length rows the last is the share of the row's squared length on one axis; for oblique pattern coefficients, whose squares do not sum to a length, it is a coefficient-concentration index and is labelled as such.

One more distinction is needed, because the matrices here are tall and nobody designed their columns. In **R-mode** analysis the columns are variables and the rows observations; factors are directions in variable space. In **Q-mode** analysis (Stephenson, 1935; Cattell, 1952) the roles are reversed: the rows are variables, the columns observations, and factors are clusters of rows. For a token table the rows are tokens and the columns are the model's 768 arbitrary coordinates, and Section 3 shows that only Q-mode produces anything readable.

2.3 Conceptors in one page

A conceptor (Jaeger, 2014, 2017) is fitted to a set of N activation rows X (N × d), here the residual-stream rows of example sentences at one layer. With c the mean row and $R = (X - c)^T(X - c)/N$ the covariance matrix of the examples (Jaeger calls it the correlation matrix; the coordinates are not standardized),

$$C = R(R + \lambda I)^{-1} = U \,\mathrm{diag}(s_j \,/\, (s_j + \lambda))\, U^T, \tag{3}$$

where $R = U\,\mathrm{diag}(s_j)U^T$ is the eigen-decomposition. The **gains** $s_j/(s_j + \lambda)$ lie in [0, 1); directions along which the examples vary a lot are kept, directions they do not use are muted. The **aperture** $\alpha = \lambda^{-1/2}$ sets how picky the filter is; this report uses $\lambda = \mathrm{tr}(R)/d$. The **quota** $\mathrm{tr}(C)/d$ is the mean gain, which

Triantafyllopoulos et al. (2026) use as an unsupervised diagnostic for choosing a layer. A conceptor is applied by multiplication, and the published steering rules differ in what is multiplied: Postmus and Abreu (2024) replace the activation by βhC; Triantafyllopoulos et al. (2026) use replacement or the interpolation $(1 - \beta)h + \beta hC$; Moore (2026), whose sentences and conceptors are reused here, centres and amplifies, $h + t(h - c)C$, and filters by $c + (h - c)C$. This report uses the last two. Conceptors can be combined by Boolean-style operations: NOT $C = I - C$; C_1 AND $C_2 = (C_1^{-1} + C_2^{-1} - I)^{-1}$ when the inverses exist, with Jaeger's (2014) general definition through pseudo-inverses otherwise; OR by De Morgan. These operations are not a Boolean algebra in the strict sense (AND is not idempotent, for instance), which is why "Boolean-style" is used here. The conceptors in this report are fitted to 96 or 16 centred rows in 768 dimensions and so have rank at most 95 or 15; wherever AND is computed, a ridge of $10^{-8}I$ is added before inversion, and all Boolean results below are for that regularized operation. Equation (3) is an unrotated principal-components solution: U is the principal-component basis of the examples, and the gains are a monotone function of the eigenvalues. Everything factor analysis says about rotating such a solution therefore has a conceptor reading, which is the subject of Section 5.

2.4 Why simple structure has a ceiling: superposition

Any single vector can be made one-factorial by choosing a basis with an axis along it. The difficulty is a *common* basis for 50,257 vectors at once. Elhage et al. (2022) showed that a network pressed to represent more features than it has dimensions stores them as nearly, but not exactly, orthogonal directions, so that a typical activation is a dense superposition of several features; and if each token's row is a sum of several features that are shared across tokens, then a basis that makes one token sparse will not make its neighbours sparse, and the best any rotation can do for the collection is to align axes with the features and accept that a token loads on as many axes as it has features. The token table is a case in point: *legends* is at once a plural noun and a word about myth, and GPT-2 writes both into one 768-vector. This sets expectations for Section 3: rotation will sharpen the table a great deal, and cross-loadings will remain. Whether a given cross-loading reflects a shared feature, a tokenization artefact, or imperfect optimization is not something the rotation itself can say; the examples in Section 3 are read as shared features because they are legible as such, which is an interpretation.

3 Rotating the lookup table

The lookup table E of GPT-2 small is a 50,257 × 768 matrix of real numbers; its rows have lengths between 2.5 and 6.3, and two tokens chosen at random have a mean cosine similarity of 0.27, the anisotropy first documented for contextual models by Ethayarajh (2019). This section treats E as a loading matrix and asks what rotation does to it.

3.1 Three ways to factor a token table, and which one to rotate

There are three natural analyses of a $V \times d$ table.

- **R-mode:** the 768 columns are variables, standardized across tokens; the tokens are observations. The correlation matrix is 768 × 768, extraction gives directions in the model's coordinate space, and the pattern matrix is 768 × k: how each *coordinate* loads on each factor.
- **Q-mode:** the tokens are variables, each standardized across its 768 coordinates; the coordinates are observations. The correlation between two tokens is then their cosine similarity after row-centring, so the analysis factors the token-similarity matrix; the pattern matrix is V × k: how each *token* loads on each factor.
- **Full rotation:** take E itself as the loading matrix (tokens = indicators, $k = d = 768$ factors), scale its rows to unit length, and rotate. This is Q-mode without the reduction to $k < d$ factors and with unit rows in place of standardized ones.

The three use different preprocessing (column standardization, row standardization, unit rows) and so are not literally one singular value decomposition seen from three sides, but their leading subspaces nearly coincide, and the point of listing them is which side receives the simple-structure criterion. In R-mode, rotation asks "which factor does coordinate 447 belong to?", a question with no answer because the coordinates are arbitrary (Section 2.1); the criterion is spent on noise. In Q-mode it asks "which cluster does the token *youth* belong to?", which is the question one wants answered. The empirical confirmation is in Section 3.5: an R-mode pattern matrix of GPT-2 is a wall of coefficients near 0.1, interpretable only through its first factor, which is a handful of outlier coordinates. Everything else in this section is Q-mode or the full rotation, and the two agree: the full rotation is the finest resolution (one factor per word family), the $k = 30$ Q-mode solution the coarsest (one factor per broad category).

3.2 Varimax on the full table

Rows were scaled to unit length (Kaiser normalization) and the 50,257 × 768 matrix was rotated by varimax for 600 iterations of the singular-value algorithm (42 minutes on two CPU cores; the criterion rose by 0.03% over the last thirty iterations, so the solution is stable rather than formally converged). Figure 1(a) shows what happened to six tokens: in the trained coordinates their largest coordinate carries 2–3% of their squared length; after rotation one coordinate carries 30–50%. Over all tokens the mean share of squared length in the largest coordinate rises from 0.022 to 0.166, and 58% of tokens have exactly one loading above 0.3 (49% of ordinary English words, defined throughout as lower-case alphabetic tokens with a leading space and a frequency in the `wordfreq` lexicon, $n = 18{,}770$). Table 1 shows what the factors are. A factor is, most often, a *word family*: by a crude rule (the five highest-loading tokens share their first three letters after stripping the leading space) 343 of the 768 are, and the examples show the rule under-counts, since *youth*, *youngsters*, *teenagers*, *young*, *teen*, *kids*, *children* is a family the rule misses. The orthographic variants of a word (capitalized, with and without a leading space, plural) cluster together, which says that the table stores "which word" strongly and its surface form as a small offset. The remainder are *format* factors (three-digit numbers, years, HTML brackets, ALL-CAPS words, sentence-initial capitalized words, two-letter fragments), *grammatical* factors (past-tense verbs, -ing forms, prepositions), and *semantic* factors broader than a family (European nationalities; medical vocabulary; superlative adjectives).

Varimax factor	Highest-loading tokens (unit-row loadings; ␣ marks a leading space)	Reading
F96	␣youth .57, ␣youngsters .55, ␣teenagers .53, ␣young .53, ␣teen .53, ␣teenager .52, ␣children .51, ␣kids .51	a word family (semantic)
F301	␣dead .64, ␣Dead .58, dead .51, Dead .50, ␣DEAD .45, ␣deceased .33, ␣corpse .26	one lemma, all surface forms
F393	ifying .69, ification .67, ified .66, ify .64, ifies .61, ifier .59, ifications .58	a suffix family
F643	␣both .71, ␣Both .58, both .55, Both .54, ␣neither .50, ␣Neither .41, ␣either .40	a function-word family
F20	␣1974 .69, ␣1978 .69, ␣1977 .69, ␣1972 .68, ␣1976 .68, ␣1982 .68, ␣1971 .68	years (format)
F2	␣156 .70, ␣153 .69, ␣234 .69, ␣139 .69, ␣167 .69, ␣258 .69, ␣176 .69	three-digit numbers (format)
F53	␣German .62, ␣Italian .60, ␣French .58, ␣Dutch .55, ␣Swedish .55, ␣Spanish .55, ␣Polish .54	a semantic category
F22	␣medical .62, ␣doctors .56, ␣hospital .55, ␣patients .54, ␣clinical .54, ␣Medical .54, ␣physician .53	a semantic field
F4	␣incredible .66, ␣amazing .64, ␣fantastic .62, ␣wonderful .60, ␣magnificent .59, ␣remarkable .59	evaluative adjectives
F412	> .68, ␣> .57, >< .54, >, .52, >. .49, ></ .46, < .43	HTML (format)
F12	␣IN .53, ␣THE .53, ␣FOR .52, ␣AND .50, ␣TO .48, ␣YOU .47, ␣RE .47	ALL-CAPS (format)
F11	␣It .69, ␣This .68, ␣There .68, ␣In .67, ␣If .67, ␣While .67, ␣When .66	sentence-initial words
F5	␣made .55, ␣brought .54, ␣pulled .52, ␣kept .52, ␣picked .52, ␣pushed .51, ␣moved .51	past-tense verbs (grammar)
F7	␣putting .62, ␣making .60, ␣giving .60, ␣creating .59, ␣providing .59, ␣taking .59, ␣bringing .59	-ing forms (grammar)

Table 1. Fourteen of the 768 varimax factors of the GPT-2 lookup table, chosen by seed token. Loadings are coordinates of the unit-length token vector in the rotated basis. The full sorted pattern matrix and a 30-token list for every factor are in the accompanying files; the readings in the last column are the authors'.

Cross-loadings remain, as Section 2.4 predicted, and the legible ones pair a form with a topic: *legends* loads on the plural-noun factor and on the mythology factor; *failed* on past tense and on collapse; *percentages* on plurals and on statistical vocabulary. Roughly half the factors describe *form* (number, tense, capitalization, format) and half describe *topic*, and a typical word has one loading of each kind. To the extent this holds, the table is close to additive in form and meaning, which is why one-factor structure is unreachable.

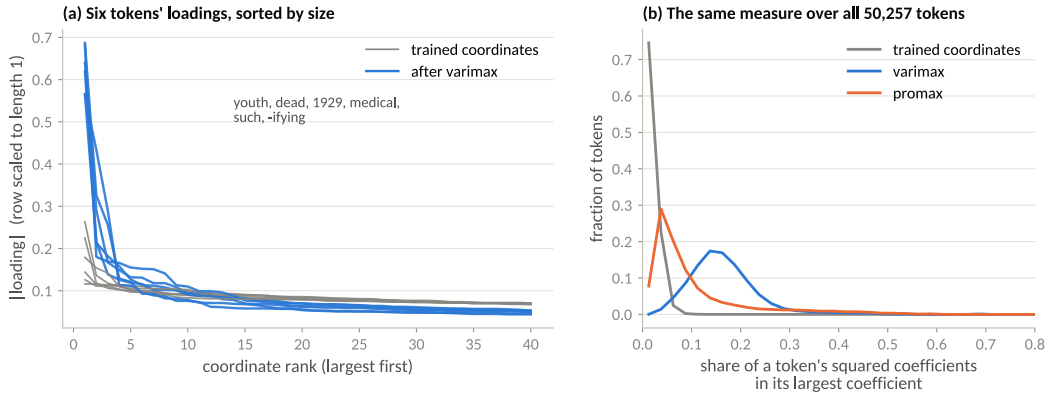


Figure 1. Simple structure of the lookup table. (a) For six tokens, the absolute loadings sorted by size, in the trained coordinates (grey) and after varimax (blue). (b) The share of each token's squared coefficients carried by its single largest coefficient, over all 50,257 tokens, in the trained coordinates and after the two rotations; promax spreads coefficients over more factors than varimax does.

3.3 Promax does not reduce cross-loadings, and its factor correlations hide a general factor

Promax was applied to the varimax solution with the usual power of 4. Table 2 gives the simple-structure indices. On every measure varimax is the sharper: it puts the larger share of each token's coefficients on one factor, and 58% of tokens have exactly one salient loading against 32% for promax, which lets 57% of tokens cross-load. The two recover the *same* factors where they were compared (the youth family is varimax F96 and promax F29 with the same ten tokens at the top; likewise *dead*, the *-ify* suffix, *both/neither*, years, medical, negative affect), so the difference is in the bookkeeping of cross-loadings, not in what the axes mean; the comparison was made on these examples, not on all 768 axes.

Solution	share of a token's squared coefficients in its largest	exactly one salient loading	two or more salient	none salient	median Hofmann complexity	factor correlations
trained coordinates	0.022	0.1%	0%	99.9%	240	—
varimax	**0.166**	**58%**	31%	11%	**27**	0 (orthogonal)
promax	0.103	32%	57%	12%	98	mean $\lvert\varphi\rvert$ 0.065; eigenvalues of Φ 28.4, 18.6, 17.7, …

Table 2. Simple-structure indices for the full 50,257 × 768 rotation (all tokens; "salient" = |loading| ≥ 0.3). Restricted to ordinary English words the ordering is the same (largest-coefficient share 0.161 vs 0.103; one salient loading 49% vs 27%).

The factor correlation matrix is the diagnostic that decides what to do next. Pairwise, the promax factors barely correlate (mean $|\varphi| = 0.065$), which agrees with Park et al. (2017), who found orthogonal rotations better than oblique ones for interpretability of word-vector axes, and with what superposition predicts: when every row is a sum of several nearly orthogonal features, the sparsest

coordinate system is itself nearly orthogonal. But a correlation matrix can be close to the identity pairwise and far from it collectively, and this one is: its largest eigenvalue is 28.4 against 18.6 for the next and a smallest near 10^{-6} (promax split a tight core of some sixty unusual byte tokens into two near-collinear factors with $r = -0.98$, an ill-conditioned solution rather than a Heywood case). A dominant eigenvalue of Φ is what a general factor looks like after an oblique rotation has dispersed it into the factor correlations. That is the cue to stop comparing simple-structure rotations and fit the model built for a general factor.

3.4 The general factor: varimax, promax, and the bifactor rotation

Embedding spaces have a dominant common direction, and the mechanism is understood: under likelihood training with a tied output embedding, every token that is not the target at a given step receives a gradient in the same direction, and the accumulated drift pushes the vocabulary, the rarely-targeted tokens most of all, into a narrow cone (Gao et al., 2019; Biś, Podkorytov & Liu, 2021; see also Mu & Viswanath, 2018). In the Q-mode analysis of this table the first eigenvalue of the token-similarity matrix is 13,625 against 443 for the second: 27% of the variance in one component, 0.9% in the next (Figure 2(a)). A spike of that size is a near-rank-one component that nearly every token shares with the same sign, and it behaves exactly as a general factor behaves in psychometrics. Three rotations treat it three ways.

Varimax, being orthogonal, must put the common direction somewhere, and puts it in one factor: in the full solution the largest factor is the primary factor for 71% of all tokens and 65% of ordinary words, and in the $k = 30$ Q-mode solution one factor is primary for 82% of tokens. **Promax** partly disperses it (20% of tokens remain on its largest factor) but leaves it visible in two places at once: in the 30-factor Q-mode promax solution, factor 1 is primary for 18,573 tokens *and* correlates 0.5–0.66 with the next six (Figure 2(b)). A power-4 target built from a varimax start cannot parcel out a direction that almost every row genuinely contains, which is the situation Jennrich and Bentler (2011) describe. **Bi-quartimin** gives the general factor a column of its own. On the 30 leading Q-mode components the rotation converged to the same criterion from the principal-components start and from ten random orthogonal starts (criterion 0.01079 to 0.01083, a spread of 0.4%), so the local-minimum concern of Mansolf and Reise (2016) did not materialize here; the target rotation described in Section 2.2 recovered the same general factor ($r = 0.99$ between the two general columns) and group factors that match the bi-quartimin ones at a mean absolute column correlation of 0.75. Table 3 summarizes the solution.

Solution (30 token-side components)	general factor: share of common variance	general loading, mean (5th–95th percentile)	tokens with a positive general loading	largest group factor, share of tokens for which it is primary	tokens with exactly one salient group loading
varimax (largest factor read as general)	—	—	—	82%	—
bi-quartimin	71%	0.52 (0.43–0.60)	100%	10.6%	4.8%

Solution (30 token-side components)	general factor: share of common variance	general loading, mean (5th–95th percentile)	tokens with a positive general loading	largest group factor, share of tokens for which it is primary	tokens with exactly one salient group loading
target rotation (varimax-defined zeros, free general column)	70%	0.52	100%	8.3%	5.4%
Full table (768 columns)					
varimax (largest factor read as general)	—	—	—	71%	58% (any factor)
bi-quartimin, from the varimax solution	27% of all squared length	0.52 (0.14–0.90 range)	100%	2.5%	26%

Table 3. The general factor under the three treatments. "Share of common variance" is the general factor's sum of squared loadings over the total for the $k = 30$ solution (the explained common variance; Reise, Moore & Haviland, 2010); for the full table the comparable figure is the share of the tokens' total squared length. "Salient" = |loading| ≥ 0.3 on a group factor; group loadings are smaller in a bifactor solution because the general factor has taken its share of each row.

The bifactor general factor loads positively on every one of the 50,257 tokens, with a mean of 0.52 and a narrow spread (0.43 to 0.60 between the 5th and 95th percentiles); the full-table version has the same mean and the same ordering of tokens (range 0.14 to 0.90). At the top are byte-level and non-Latin fragments and code-like fragments (*externalToEVA*, *quickShip*); at the bottom are the most frequent function words (*the* 0.14, *and* 0.16, *a* 0.18, *in* 0.19). Over 18,770 ordinary words the general loading correlates −0.55 with Zipf frequency (Figure 2(c)); the relation is flat below Zipf 4 and steep above it, so it is the frequent words that have moved away from the common direction rather than the rare ones that have moved toward it, which is what the gradient account of Gao et al. (2019) and Biś et al. (2021) predicts: frequently targeted tokens are the ones whose embeddings are repeatedly pulled off the common drift. The correlation is with an external frequency estimate, not with the model's training frequencies, which are unknown, and the tight cluster of byte tokens at the top is better described as a set of unusual tokens whose embeddings sit close together than as untrained, since even a token never seen as input receives gradient through the tied unembedding. It is the same structure Puccetti et al. (2022) and Timkey and van Schijndel (2021) describe from the dimension side, and it connects to conceptors: Liu, Ungar and Sedoc's (2019a) conceptor negation fits a conceptor to the whole table and applies NOT, which softly removes exactly these dominant directions, the "all-but-the-top" post-processing of Mu and Viswanath (2018) done with a conceptor.

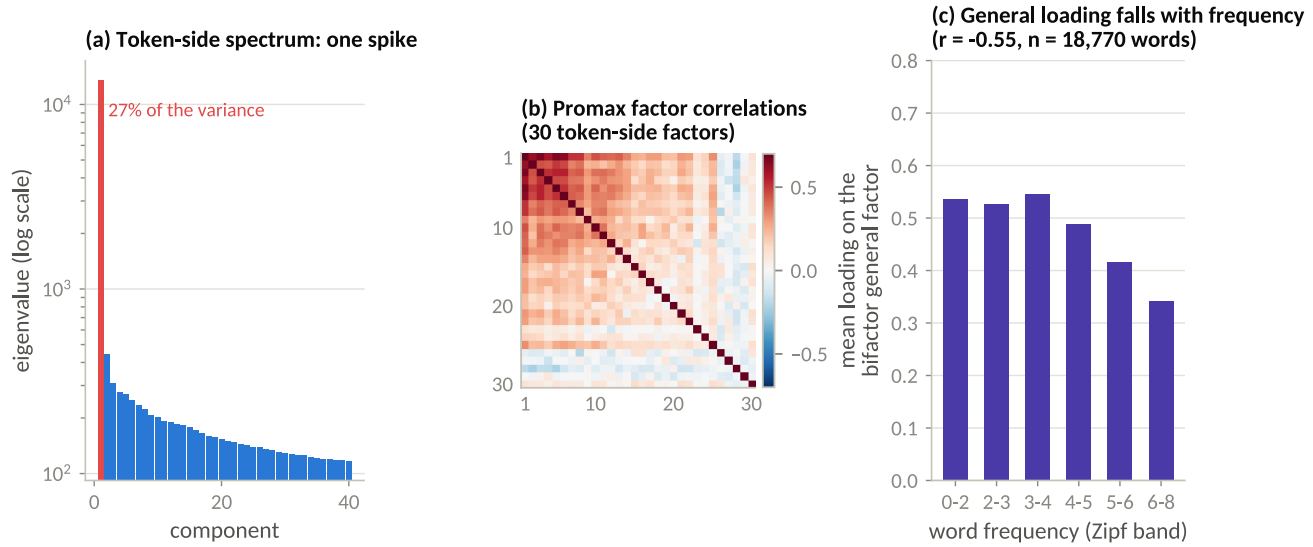


Figure 2. The general factor of the lookup table. (a) Eigenvalues of the token-side (Q-mode) correlation matrix, log scale. (b) The factor correlation matrix of the 30-factor promax solution; factor 1 correlates 0.5–0.66 with the next six. (c) Mean loading of ordinary words on the bi-quartimin general factor, by frequency band.

The group factors are the second dividend, and the claim for them is descriptive rather than metric: with the common direction removed, the 29 bi-quartimin group factors are the promax factors with their poles unmixed. Function words and punctuation, which under promax were the *negative* pole of the general factor, become a group factor of their own; the rest are word-internal fragments, administrative vocabulary, three-digit numbers, plural nouns, evaluative intensifiers, sentence-initial words, male first names, capitalized game and weapon nouns, violence and crime, past-tense verbs, Asian nationalities, chemistry and biomedicine, gender and sexuality, entertainment events, finance, ALL-CAPS, computing, the divine, -ing forms, geography, quantitative decline, research, soldiers and warriors, programming identifiers, innovation, political rhetoric, anxiety and distress, and years (Appendix B lists all 29 with their top tokens). Group loadings are modest (0.25 to 0.69 at the top of each) because the general factor has taken 71% of the common variance, and only 4.8% of tokens have a group loading above 0.3; the factors are read from their highest-loading tokens, which are unambiguous, rather than from salience counts, and no word-intrusion or other quantitative interpretability test was run on them. On the full table the picture is the same at word-family resolution: the general factor keeps 27% of the tokens' squared length and no group factor is primary for more than 2.5% of tokens, against 71% for varimax's giant factor.

3.5 The dimension side

For completeness, the R-mode analysis (768 coordinates as variables, promax, $k = 30$) was also run. Its pattern matrix is unreadable: the median coordinate's largest loading is 0.18 and only 7% of coordinates load above 0.4 on anything. The one exception is factor 1, which is carried by about 35 coordinates including the known outlier dimensions of GPT-2 (coordinates 138 and 447 in the usual zero-based numbering; Timkey & van Schijndel, 2021; Kovaleva et al., 2021) and which, read through its token scores, is the common direction again. Rotating the dimension side spends the simple-

structure criterion on a question that has no answer; the coordinates are unprivileged in the sense of Section 2.1, and the factor analysis knows it. That the common direction nevertheless concentrates on a few trained coordinates is the empirical privilege Elhage et al. (2023) describe; the two statements are compatible.

4 A change of basis leaves the model unchanged

Section 3 chose new coordinates for the lookup table. This section shows that the whole model can be carried into those coordinates, orthogonal or oblique, without changing what it computes, and says exactly what has to be done to the weights. The orthogonal case is established: Elhage et al. (2021) note the freedom, SliceGPT (Ashkboos, Croci et al., 2024) proves it for any orthogonal Q after converting LayerNorm to RMSNorm by absorbing the centring into the write matrices, and QuaRot (Ashkboos, Mohtashami et al., 2024) applies it with Hadamard rotations. Inside an attention head the freedom is larger still, since only the products $W_Q W_K^{\mathrm{T}}$ and $W_O W_V$ are determined (Elhage et al., 2021), and reparameterization symmetries of network weights are a subject of their own. The oblique case, which is what a promax solution calls for, is the residual-stream symmetry written out with LayerNorm left in place. It needs two things: the normalization layer must carry the induced metric, and the tied unembedding must be transformed by the inverse-transpose. Neither is deep. What they buy is a reading, that the input and output tables then stand in the pattern-to-structure relation of oblique factor analysis (in plain linear algebra, coordinates in a basis and inner products with it, the primal and dual descriptions of one vector), and a warning, that a rewrite which forgets the metric is a different model.

4.1 The transformation rules

Let M be any invertible $d \times d$ matrix and write the residual stream in new coordinates, $h' = hM$. A read, hW, is preserved by replacing W with $M^{-1}W$, since $hMM^{-1}W = hW$. A write of a vector u into the stream is preserved by writing uM instead, so a write matrix W becomes WM and a bias b becomes bM. The lookup and position tables are writes: $E' = EM$, $P' = PM$. Everything between a read and a write (attention scores, softmax, GELU) sees the same numbers as before and is untouched. That leaves LayerNorm and the unembedding.

LayerNorm. Equation (2) centres h along $\mathbf{1}$ and divides by a Euclidean length. In the new coordinates the all-ones direction is $\mathbf{1}' = \mathbf{1}M$. The Euclidean length of a vector v whose new coordinates are $v' = vM$ is $\|v\|^2 = v'M^{-1}M^{-\mathrm{T}}v'^{\mathrm{T}} = v'\Phi v'^{\mathrm{T}}$ with

$$\Phi = (M^{\mathrm{T}}M)^{-1}, \tag{4}$$

which is the factor-correlation matrix of Section 2.2 when M is a rotation's pattern transform, and the identity when M is orthogonal. The mean is $\mu = h\cdot\mathbf{1}/d = h'c/d$ with $c = M^{-1}\mathbf{1}^{\mathrm{T}}$. The normalized vector in new coordinates is therefore

$$n' = (h' - \mu\mathbf{1}') \,/\, \sigma, \quad \sigma^2 = (h' - \mu\mathbf{1}')\,\Phi\,(h' - \mu\mathbf{1}')^{\mathrm{T}} \,/\, d + \varepsilon, \tag{5}$$

a LayerNorm that centres along the image of the ones vector and measures length with the metric Φ. The stabilizer ε is added to the same variance as before and needs no change. The elementwise gain γ and bias β are folded into the read that follows: $\mathrm{LN}(h)W + b = n\,\mathrm{diag}(\gamma)W + (\beta W + b)$. The read therefore becomes $M^{-1}\mathrm{diag}(\gamma)W$ with bias $b + \beta W$, and the new LayerNorm has unit gain and zero bias.

Two special cases are worth naming. If M is orthogonal *and* fixes the ones direction ($M^{\mathrm{T}}\mathbf{1}^{\mathrm{T}} = \mathbf{1}^{\mathrm{T}}$), then $\Phi = I$, $\mathbf{1}' = \mathbf{1}$, $c = \mathbf{1}^{\mathrm{T}}$, and equation (5) *is* ordinary LayerNorm: nothing in the architecture changes. If M is orthogonal but moves the ones direction, as a varimax rotation generally does, the metric is still the identity but the centring direction has moved, and ordinary LayerNorm is wrong. SliceGPT's route around this is to make the stream mean-free by projecting every write onto the complement of $\mathbf{1}$, after which LayerNorm is RMSNorm and any orthogonal Q is exact. Equation (5) instead keeps the stream as it is and moves the centring. For RMSNorm models there is no centring, so every orthogonal rotation is exact with the standard layer, and only the metric is needed for an oblique one.

The tied unembedding. The logits are $\mathrm{LN}_{\mathrm{f}}(h)E^{\mathrm{T}} = n\,\mathrm{diag}(\gamma_{\mathrm{f}})E^{\mathrm{T}} + \beta_{\mathrm{f}}E^{\mathrm{T}}$. The unembedding is a read, so in the new coordinates it is $U' = M^{-1}\,\mathrm{diag}(\gamma_{\mathrm{f}})\,E^{\mathrm{T}}$, plus a constant logit offset $\beta_{\mathrm{f}}E^{\mathrm{T}}$. Set the final gain aside for a moment and compare the two tables. The input table is EM. The transposed output table is $EM^{-\mathrm{T}} = EM(M^{\mathrm{T}}M)^{-1} = (EM)\Phi$. In the language of Section 2.2, with $\Lambda = E$ and $T = M$, the input table is the **pattern matrix** ΛT and the output table is the **structure matrix** $\Lambda T^{-\mathrm{T}} = \Lambda_{\mathrm{p}}\Phi$. For an orthogonal M the two coincide. For an oblique M the model needs both faces of the same solution, pattern coordinates to read a token in and structure coordinates to score a token out, and the tie between the tables is broken. The final gain γ_{f} is a separate diagonal factor that sits between the stream and the unembedding in the original model as well. Folding it into the output table, as the implementation below does for convenience, makes that table the structure counterpart of $E\,\mathrm{diag}(\gamma_{\mathrm{f}})$ rather than of E, and breaks the literal tie even for orthogonal M; keeping it as its own layer preserves the tie exactly in the orthogonal case. This answers a natural question (can the structure matrix be used in place of the pattern matrix?) precisely: one goes in, the other comes out, and the relation between them is the factor correlation matrix.

> **Proposition.** For any invertible M, the model with tables EM and PM, reads $M^{-1}\mathrm{diag}(\gamma)W$ (biases $b + \beta W$), writes WM (biases bM), LayerNorm (5) with unit gain and zero bias, and unembedding $M^{-1}\mathrm{diag}(\gamma_{\mathrm{f}})E^{\mathrm{T}}$ with logit offset $\beta_{\mathrm{f}}E^{\mathrm{T}}$, computes exactly the same logits as the original model for every input. Proof: each read and write is preserved by construction, equation (5) reproduces equation (2) in the new coordinates (Appendix C.2), and the composition of preserved maps is preserved.

Object	Orthogonal rotation Q	Oblique transform M
residual stream	hQ	hM
lookup / position tables	EQ, PQ	EM, PM (pattern coordinates)
weights that read	$Q^{\mathrm{T}}\mathrm{diag}(\gamma)W$	$M^{-1}\mathrm{diag}(\gamma)W$

Object	Orthogonal rotation Q	Oblique transform M
weights that write	WQ, bQ	WM, bM
LayerNorm	unchanged if Q fixes $\mathbf{1}$; else centre along $\mathbf{1}Q$ (or make the stream mean-free first, as in SliceGPT)	centre along $\mathbf{1}M$, length via $\Phi = (M^{\mathsf{T}}M)^{-1}$
RMSNorm	unchanged	length via Φ
unembedding	$Q^{\mathsf{T}}\mathrm{diag}(\gamma_f)E^{\mathsf{T}}$ (tied to the input table up to the final gain)	$M^{-1}\mathrm{diag}(\gamma_f)E^{\mathsf{T}}$ (structure coordinates; untied)
attention patterns, MLP activations	unchanged	unchanged

Table 4. Transformation rules for carrying a transformer into new coordinates of its residual stream.

4.2 Verification on GPT-2

The rules were implemented as a subclass of the NumPy GPT-2 used throughout this report, in double precision, and tested with three transforms: a random orthogonal rotation constructed to fix the ones direction; the varimax rotation of Section 3.2 (orthogonal, but it moves the ones direction); and the promax transform (oblique; condition number 2,439 because of the near-collinear doublet of Section 3.3, so that the induced metric Φ has condition number 6×10^6, a severe test). Each was tried with ordinary LayerNorm, equation (2) applied as if the coordinates were the old ones, and with equation (5). For three prompts the logits at every position were compared with the original model's, and twenty tokens were generated greedily from each prompt.

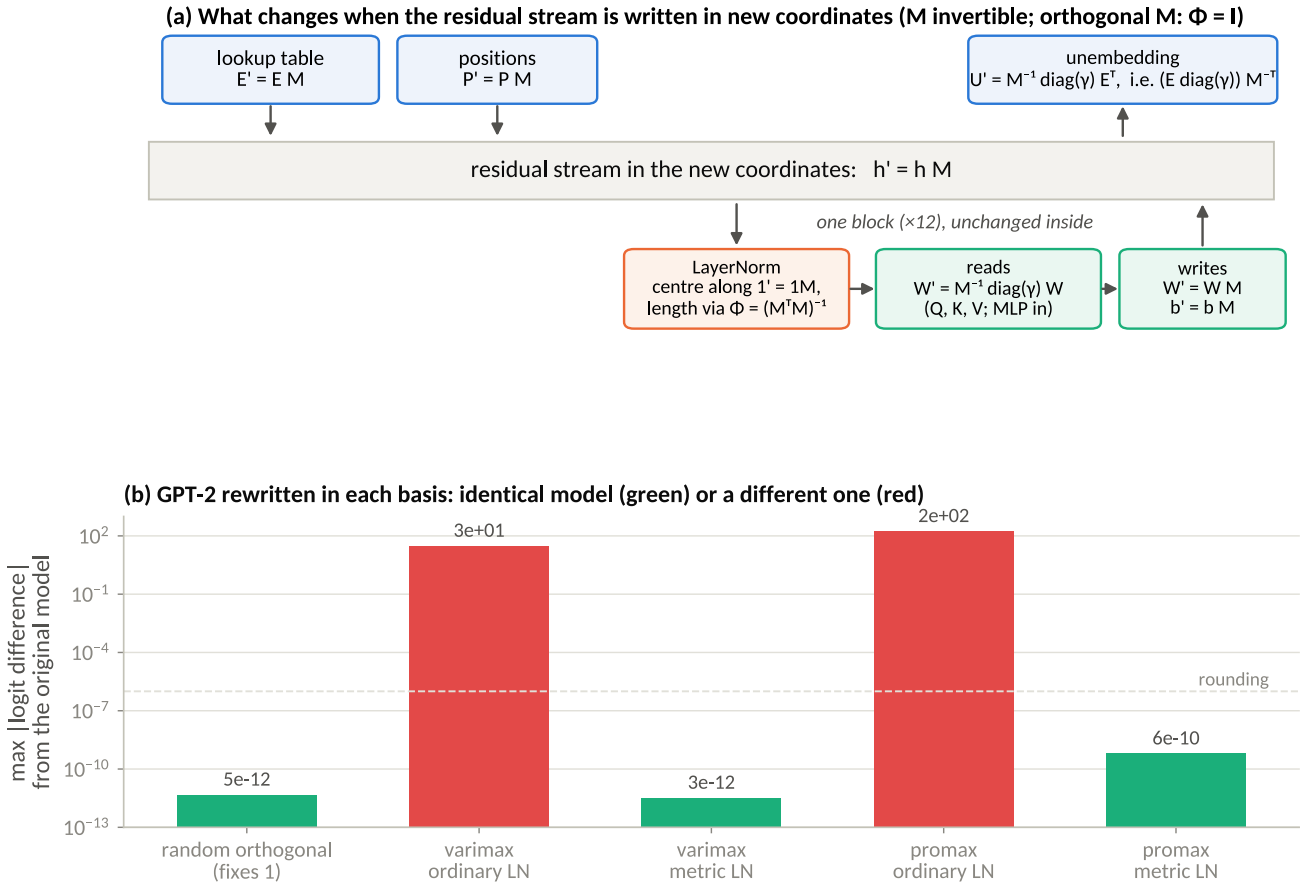


Figure 3. (a) The pieces of a transformer that change under a change of basis of its residual stream. (b) GPT-2 rewritten in each basis: the maximum absolute difference in logits from the original model over three prompts (log scale). Green bars are at floating-point rounding and produced identical generations; red bars are a different model.

Transform	LayerNorm	cond(*M*)	max |Δ logit|	top-1 agreement (worst prompt)	20-token generations
random orthogonal, fixes **1**	ordinary	1	4.7×10^{-12}	100%	identical
varimax (moves **1**)	ordinary	1	30	0%	different
varimax	eq. (5)	1	3.2×10^{-12}	100%	identical
promax (oblique)	ordinary	2,439	172	0%	different
promax (oblique)	eq. (5)	2,439	6.2×10^{-10}	100%	identical

Table 5. GPT-2 small rewritten in new coordinates. "Ordinary" LayerNorm is equation (2) applied to the new coordinates; equation (5) is the metric form. Top-1 agreement is the share of positions at which the most probable next token agrees with the original model's, reported for the prompt on which it was lowest. (The oblimin transform mentioned in Section 2.2, condition number 30, gave the same pattern: 156 with ordinary LayerNorm, 10^{-11} with equation (5).)

The identity of Section 4.1 was also checked directly: the output table of the promax model equals the structure counterpart of the gain-scaled input table, $(E\,\mathrm{diag}(\gamma_f))M(M^TM)^{-1}$, to 3×10^{-12}. Two features of Table 5 deserve emphasis. First, the naive rewrites are not slightly wrong; they are a different

model, with no agreement on the most probable token at any position of the worst prompt and logit errors of order 10^2. The only place a transformer can notice its coordinate system is the normalization layer, and it notices emphatically. Second, the varimax row is a warning for the orthogonal case too: a rotation chosen for interpretability will in general move the all-ones direction, and a model rewritten with it needs either the centring of equation (5) or SliceGPT's mean-free conversion, unless it uses RMSNorm. The random-orthogonal row shows that the issue is the ones direction and nothing else.

4.3 What the result is good for

The proposition says that the model is indifferent to the coordinates its residual stream is written in, and Section 3 says that some coordinate systems are far more readable than the trained one. Together they license a programme: rewrite the model in a rotated basis once, and then read its *weights*, not only its activations, in interpretable terms. A rotated reading weight $M^{-1}\mathrm{diag}(\gamma)W_Q$ says which interpretable axes an attention head's queries attend to; a rotated writing weight W_OM says which axes it writes to; the rotated unembedding says which axes raise which tokens' logits. None of this changes the model's behaviour, so it is pure description, in the way that choosing a rotation is pure description of a factor solution. The oblique extension matters for one reason: the best descriptive basis may be oblique (Section 7 argues that it is, for correlated concepts), and the proposition says an oblique basis can be used at the price of carrying Φ through the normalization layer, which is a one-line change to the code. Sparse autoencoders (Bricken et al., 2023) pursue the same descriptive goal with an overcomplete, nonlinear dictionary that cannot be folded back into the model; a rotation can.

5 Conceptors under a change of basis

A conceptor is a matrix that acts on the residual stream, so Section 4's change of basis acts on it too. This section works out how, separates what is invariant from what is not, and finds that the one non-invariant piece, the aperture regularizer, is exactly the piece a practitioner would get wrong by fitting in rotated coordinates without thinking.

5.1 The transformation law and its invariants

Let a conceptor C act on centred rows by $v \to vC$. In new coordinates $v' = vM$ the same operation must send v' to $(vC)M = v'M^{-1}CM$, so the transported conceptor is

$$C' = M^{-1}CM, \tag{6}$$

a similarity transform. Everything that is preserved by similarity is therefore basis-independent: the **gains** (eigenvalues of C), the **quota** $\mathrm{tr}(C)/d$, the **action** on any row, and the Boolean-style operations, since $M^{-1}(I - C)M = I - C'$, $M^{-1}(C_1^{-1} + C_2^{-1} - I)^{-1}M = (C'_1{}^{-1} + C'_2{}^{-1} - I)^{-1}$, and the ridge of Section 2.3 transports with them ($M^{-1}(C + \varepsilon I)M = C' + \varepsilon I$). Overlaps between conceptors measured as traces of

products are likewise invariant, and so is the quota-by-layer diagnostic of Triantafyllopoulos et al. (2026). What changes is the *representation*: the eigenvectors move with the basis, and for an oblique M the transported matrix is no longer symmetric (its largest asymmetry in the example below is 12), though it is self-adjoint with respect to the metric, $C'\Phi = (C'\Phi)^{\mathrm{T}}$ (Appendix C.4; verified to 4×10^{-12}). For an orthogonal Q, $C' = \mathrm{Q}^{\mathrm{T}}C\mathrm{Q}$ is symmetric and everything is as before.

5.2 Fitting in rotated coordinates: the aperture carries a metric

The subtlety is in equation (3). Written in new coordinates, the covariance matrix of the examples is $R' = M^{\mathrm{T}}RM$, and the obvious thing to do, $R'(R' + \lambda I)^{-1}$, is not C'. The term λI is a ridge penalty that is isotropic in the *old* coordinates, "the same unit length in every direction"; the identity operator is still the identity under similarity, but an isotropic penalty in the old metric is represented in the new coordinates by $\lambda M^{\mathrm{T}}M = \lambda\Phi^{-1}$. Substituting $R = M^{-\mathrm{T}}R'M^{-1}$ into (3) and (6) gives the correct fitting formula in oblique coordinates,

$$C' = \Phi\, R'\, (R' + \lambda\Phi^{-1})^{-1}\, \Phi^{-1}, \quad \text{with the default} \quad \lambda = \mathrm{tr}(\Phi R')/d, \tag{7}$$

which reduces to (3) when $\Phi = I$. The default rule for λ must be transported too, because $\mathrm{tr}(R) = \mathrm{tr}(\Phi R') \neq \mathrm{tr}(R')$; for the promax transform below, $\mathrm{tr}(R')/d$ would give $\lambda = 943$ where the original is 1.385, and $\mathrm{tr}(\Phi R')/d$ gives 1.385 exactly.

Table 6 and Figure 4 test this on the sentiment conceptor of Moore (2026), fitted at layer 8 to the final-token rows of 48 positive and 48 negative sentences ($\lambda = 1.385$, quota 0.076), transported by the promax transform of Section 3 (Φ far from the identity, condition number 6×10^{6}). The outputs of the filters are compared on 24 evaluation rows: the 16 neutral sentences, which are outside the fitting set, and 8 of the fitting sentences. The transported operator (6) reproduces the original filter's output to 3×10^{-14}; the metric formula (7) reproduces it to 7×10^{-10}, the residue of inverting an ill-conditioned M. The naive refit with the same λ is a different filter: its output differs from the original by 36% of the centred output vector on average (6.5% at the worst coordinate), its quota is 0.108 rather than 0.076, and its gains, in Figure 4, are uniformly too high; the naive refit with the untransported default rule errs the other way (quota 0.013, gains collapsed, mean output error 77%). For an orthogonal rotation the naive refit coincides with the transported operator to 4×10^{-15}, as it must. The regularized AND and NOT were checked as well: computed in the new coordinates they agree with the transported originals to 10^{-12}.

Conceptor in the new coordinates	max relative error of the filtered output	mean relative error	quota	gains vs original
transported, eq. (6)	3×10^{-14}	2×10^{-13}	0.0760	identical (3×10^{-14})
re-fitted with the metric, eq. (7)	7×10^{-10}	2×10^{-9}	0.0760	identical to rounding
re-fitted without the metric, same λ	6.5%	36%	0.108	all too high
re-fitted without the metric, λ = $\mathrm{tr}(R')/d$	7.3%	77%	0.013	collapsed

Conceptor in the new coordinates	max relative error of the filtered output	mean relative error	quota	gains vs original
orthogonal rotation, naive refit	4×10^{-15}	—	0.0760	identical

Table 6. The layer-8 sentiment conceptor carried into promax coordinates. Errors are relative to the original filter's output on 24 evaluation rows, 16 of them outside the fitting set (max: largest coordinate error over the largest output coordinate; mean: norm of the error over the norm of the centred output). The comparison tests an algebraic identity, so held-out rows are not needed for it; they are reported for completeness.

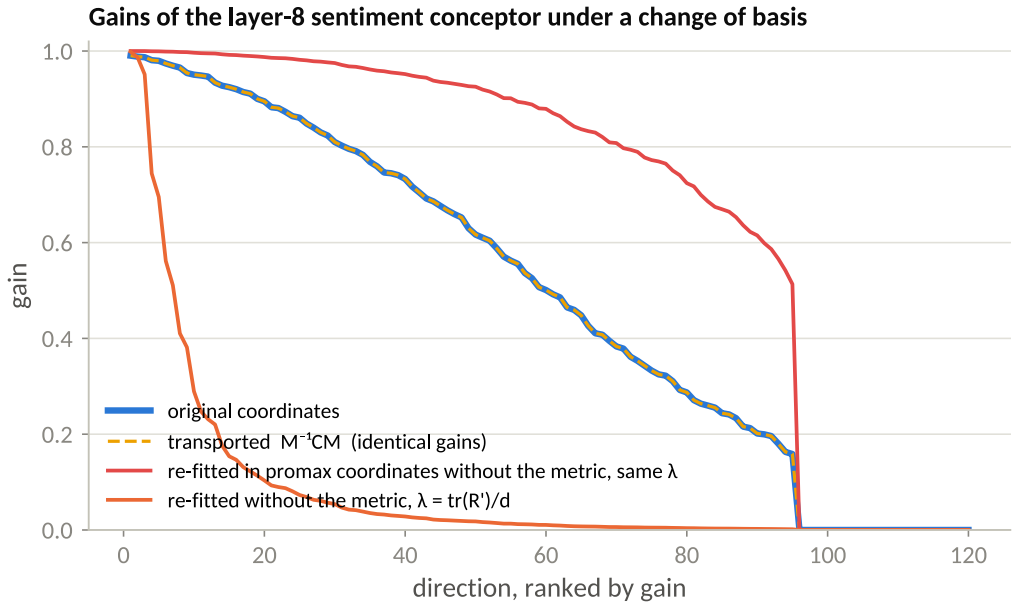


Figure 4. Gains of the sentiment conceptor, ranked, in the original coordinates and after three ways of moving it into promax coordinates. Transport by equation (6) leaves the gains unchanged; refitting without the metric changes every one of them.

5.3 What this means for practice

Three rules follow. (i) A conceptor fitted in the trained coordinates can be carried into any rotated basis by equation (6), and nothing about what it does changes; rotation is a reading device, never a modelling decision. (ii) If a conceptor is to be *fitted* in rotated coordinates, as it would be if a model were stored in an interpretable basis (Section 4.3), an orthogonal basis needs no change to the fitting code and an oblique one needs equation (7) with its transported default; forgetting the metric produces a filter that is wrong by tens of percent while looking perfectly reasonable. (iii) The quantities one reports about a conceptor, its gains, its quota, its overlaps with other conceptors, are coordinate-free once the conceptor is transported correctly, so published results that report them do not depend on the basis the model was trained in; quantities computed *in* a basis, such as Euclidean distances between transformed rows, a naive refit, or the choice of axes to read, are not invariant, as this section's own demonstrations show. The only coordinate-dependent thing about a correctly transported conceptor is the *description* of what it keeps, which is exactly the thing rotation is for, and which Section 6 takes up.

6 Rotating a layer

The lookup table enters the model only at the start, as one of the two terms of the initial residual state. By layer 8 a row no longer says which token it is; it says something about the sentence so far, and a conceptor fitted there is a filter on *that*. Rotating the table says nothing about it. But the technique transfers: collect the rows a layer produces over a corpus, and rotate those. This section does so at five depths, labels the resulting axes, and then reads the sentiment arrow and conceptor of Moore (2026) on them. The question in the background is the one every conceptor practitioner faces: which layer, and why. The answer this section reaches is less than the question hoped for, and the baselines that cut it down are the useful part.

6.1 Method

GPT-2 was run over 41,984 tokens from six public-domain books (Austen's *Emma*, Carroll's *Alice*, Chesterton's *The Man Who Was Thursday*, Melville's *Moby-Dick*, Bryant's *Stories to Tell to Children*, Whitman's *Leaves of Grass*), in windows of 128 tokens, and the residual-stream row of every token was kept at layers 2, 5, 8, 11 and 12 (layer 12 being the state after the last block, before the final normalization). At each depth the rows were centred on the corpus mean, scaled to unit length, projected on their 60 leading principal directions (53–68% of their squared length, depending on depth) and varimax-rotated with the rows as indicators, a Q-mode rotation in which the "variables" are tokens-in-context. Each rotated axis was labelled by the fourteen contexts scoring highest on it and the six scoring lowest; the labels are qualitative readings of those lists. Separately, at the same depth, the sentiment arrow (difference of the mean final-token rows of 48 positive and 48 negative sentences) and the sentiment conceptor (fitted to those 96 rows) were computed exactly as in Moore (2026), together with a second conceptor fitted to 16 neutral sentences.

Three quantities were read off. For a unit arrow u and orthonormal axes A, the squared loading on an axis a_j decomposes as $(a_j{\cdot}u)^2 = \|\mathrm{A}^\mathrm{T}u\|^2 \times (a_j{\cdot}u)^2/\|\mathrm{A}^\mathrm{T}u\|^2$: the share of the arrow *captured* by the retained subspace, times its *concentration* on one axis within it. The conceptor's gain along an axis is aCa^T, and the share of its trace inside the subspace is $\mathrm{tr}(\mathrm{A}^\mathrm{T}C\mathrm{A})/\mathrm{tr}(C)$. Each statistic is reported against a baseline, because each has a chance level that is not zero. A random unit direction in $d = 768$ dimensions puts k/d of its squared length into any k orthonormal directions, so capture by 60 directions is 0.078 at chance (0.10 at the 95th percentile) and by 300 directions 0.39 (0.43); the concentration of a random direction on the largest of 60 axes is 0.11 (0.17). A second, stricter null controls for what the arrow is made of rather than for its direction: the sentiment labels of the 96 sentences were permuted 500 times and the difference of means recomputed, giving arrows that are differences of review-sentence final states with no sentiment in them. And because the arrow is read from sentence-final states while the corpus directions are fitted to every token of the novels, the capture statistic was computed a second time with the corpus's own sentence-final states (the 1,133 period tokens) as the reference set.

6.2 Where the concept sits in a layer's leading directions depends on what they are compared with

Three readings of Table 7 are possible, and they do not agree, which is the finding. Read against random-direction chance, the sentiment arrow is at chance in the leading directions of a general corpus at layer 2 (8% against 7.8%) and layer 5 (10% against a 95th percentile of 10%), above chance from layer 8 (15%), and well above it at layers 11 and 12 (34%, 51%); with 300 retained directions the figures are 43%, 52%, 68%, 66% and 72% against a chance level of 39% (Figure 5(a)). That reading says that the concept enters the layer's leading directions only with depth. Read against the label-shuffled null, it says something else. An arrow made of the same sentence-final states with its sentiment removed is captured *more* than the sentiment arrow at layers 2, 5 and 8 (15%, 17%, 20% against 8%, 10%, 15%), rises with depth in the same way (28% at layer 11, 34% at layer 12), and is exceeded by the sentiment arrow beyond its 95th percentile only at layer 12. Most of the depth trend in the second column is therefore a trend in where sentence-final states of short modern sentences sit relative to the rest of the layer, not in where sentiment sits. The third reading uses the matched reference set, and inverts the first: against the leading directions of the corpus's own sentence-final states the arrow is inside the leading 60 directions at every depth (44–71%), as any difference of sentence-final states is (the shuffled null is 45–58%), and it stands out from that null at one layer only, layer 8, where 71% of it is captured against a null 95th percentile of 62%. Layer 8 is also where Moore (2026) found the positive and negative sentences best separated. The concentration columns tell the same story at the level of single axes: at layer 2 the arrow's largest axis share is at chance whether or not the axes are rotated (0.10 and 0.12 against 0.11), rotation raises it above chance at layers 5 and 8 (0.23, 0.19), and only at layer 12 does one axis carry more than half of the captured arrow (0.54).

What survives is modest and, we think, right. A general corpus does not vary much in sentiment, so sentiment is a low-variance direction of the layer, and an unsupervised reduction of all the layer's states discards most of it at every depth; that is a fact about corpora and about what "leading directions" means, not about where the model keeps the concept. The depth trend that the first reading suggests is carried by a confound the second reading exposes: the sentence-final states of the review sentences move into the corpus's leading directions with depth whether or not they carry sentiment. Once the reference set is matched to the kind of state the arrow is made of, the picture is flat except at layer 8, where the sentiment direction occupies more of the leading sentence-final directions than a relabelled arrow does. That coincidence with the separability result is the one piece of layer-choice information the analysis yields, and Section 7.1 proposes it in that form: a matched reference set, a label-shuffled null, and a layer chosen where the concept stands out from the null, not where its raw capture is highest. The rotated axes themselves remain interpretable at every depth; what they name is the dominant variation in *this* corpus, literary register, punctuation and syntax, sentence position, and Austen's social vocabulary, and only late in the model is any of it readable as sentiment.

layer	variance kept by 60 directions	captured by the 60 leading directions of all corpus tokens		captured by the 60 leading directions of the corpus's sentence-final states		concentration on one axis		most aligned varimax axis, read from its contexts
		sentiment arrow	label-shuffled arrow (95th pct.)	sentiment arrow	label-shuffled arrow (95th pct.)	unrotated	varimax	
2	66%	8%	15% (26%)	57%	58% (69%)	0.10	0.12	none readable as sentiment
5	60%	10%	17% (24%)	44%	49% (60%)	0.10	0.23	none readable as sentiment
8	53%	15%	20% (27%)	**71%**	50% (62%)	0.10	0.19	formal register vs *never / not / no*
11	57%	34%	28% (38%)	48%	45% (58%)	0.22	0.18	*well-informed, affectionate, civil* vs *trap, vulgar, noise*
12	68%	**51%**	34% (47%)	60%	51% (64%)	0.36	0.54	*I wouldn / I don / it doesn* (an - *n't* follows) vs *idea, then*

Table 7. The sentiment arrow read on each layer's leading directions, against two baselines. Capture is $\|A^{T}u\|^2$ for the 60 leading principal directions of the reference set; a random direction scores 0.078 (95th percentile 0.10). The label-shuffled arrow is the difference of means of the same 96 sentence-final states under 500 random relabellings (mean, and 95th percentile). Concentration is the largest single-axis share within the captured part, for which a random direction scores 0.11 (95th percentile 0.17). Bold marks the two cells in which the sentiment arrow exceeds the 95th percentile of the shuffled null.

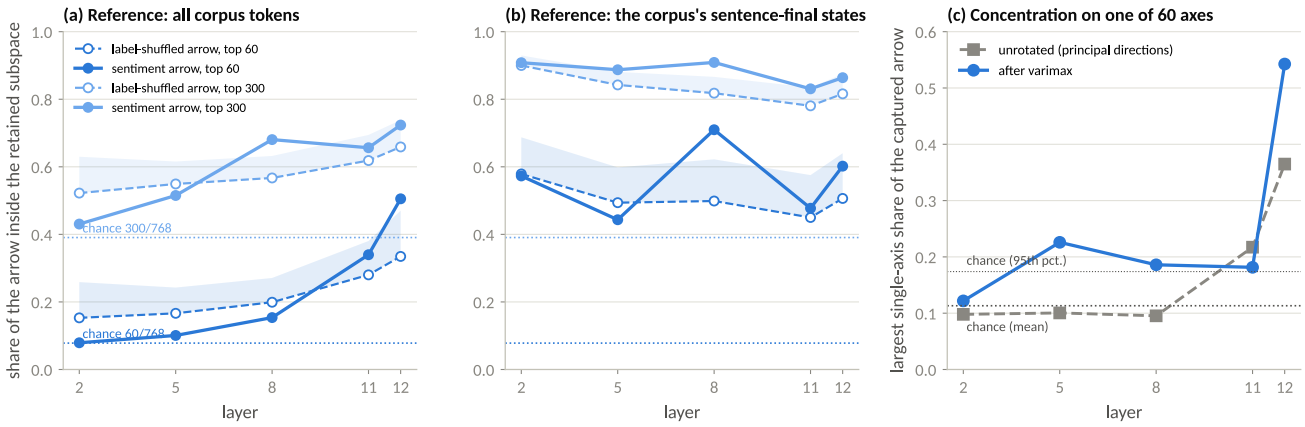


Figure 5. Rotating a layer, with baselines. (a) The share of the sentiment arrow captured by the 60 and 300 leading principal directions of all corpus tokens at each depth, with the chance level *k/d* (dotted) and the label-shuffled null (open markers, dashed; the shaded band reaches its 95th percentile). (b) The same with the corpus's sentence-final states as the reference set. (c) Within the 60-direction subspace, the largest single-axis share of the captured arrow before and after varimax, against the chance level for a random direction.

The last layer's most aligned axis is instructive and should be read as a hypothesis from context inspection rather than a mechanism. Its contexts are *I wouldn*, *I don*, *it doesn*, *he didn*, the stems of negated contractions whose next token is *'t*. The output layer represents the next token, and at the

output layer "negative sentiment" overlaps with "a negation is coming", which is why the sentiment arrow is most captured there and why the capture is not the same thing as the concept. The second-most aligned axis at layer 12 (loading −0.32) is sentiment proper, *hopeless*, *suffering*, *danger*, *terrify* against *nice*, *right*, *good*. The lesson for layer choice is not "use the last layer"; steering at the last layer is steering the next token directly, which is not the point of a conceptor. The lesson is that *presence of the concept in a layer's leading directions, and its alignment with an unsupervised rotation of them, are different properties from separability of the concept*, and that the first two depend on the reference set in a way the third does not. A conceptor needs separability. An account of what the conceptor keeps needs the axes, and Section 6.4 shows how to get them at any layer by rotating in the right subspace.

6.3 Reading two conceptors on the same axes

Figure 6 is the payoff of having interpretable axes: a conceptor becomes a labelled profile. The profile is a *directional summary*: aCa^{T} is the component the filter keeps along a from an input along a, and a conceptor can also move an input along a into other axes, which the diagonal profile does not show; it describes a conceptor the way the diagonal of a covariance matrix describes a distribution. On the twelve layer-12 axes most aligned with the arrow, the sentiment conceptor keeps 0.86 of the "negation coming" axis and the neutral-sentence conceptor 0.76, so that axis is shared, as it should be, since both sets of example sentences are ordinary clauses that may or may not be negated. On the sentiment-proper axis (51: *hopeless*, *suffering* vs *nice*, *good*) the sentiment conceptor keeps 0.26 and the neutral conceptor 0.03. On the remaining axes, which are about syntax and punctuation, the sentiment conceptor keeps 0.1–0.25 and the neutral conceptor almost nothing. The comparison between the two conceptors is confounded by their sample sizes (96 rows against 16, hence rank 95 against 15, with the default λ set separately for each), so the neutral conceptor's low gains on most axes say as much about its small sample as about neutrality; the contrast that survives this caveat is between the two axes it does keep and the one it does not. The point of the figure is the form of the description: two 768 × 768 matrices read axis by axis, which is what the conceptor literature's overlap statistics summarize in one number. The gains are modest in absolute terms because most of each conceptor's trace lies outside the 60-axis subspace (19% of the sentiment conceptor's is inside), which is why the next subsection rotates inside the conceptor instead.

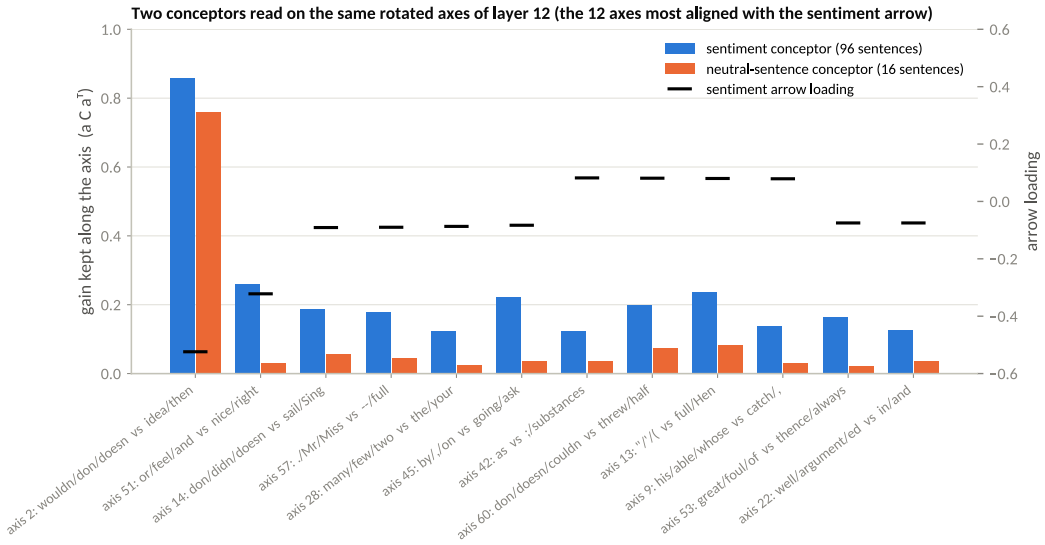


Figure 6. The gain the sentiment conceptor and the neutral-sentence conceptor keep along each of the twelve layer-12 axes most aligned with the sentiment arrow, with the arrow's loading (black ticks) and a label from the tokens at the two ends of each axis.

6.4 Rotating inside the conceptor's own subspace

A conceptor's eigenvectors span the directions its examples vary along; its top ten, at layer 8, carry gains of 0.95–0.99, contain 99.3% of the sentiment arrow, and account for 17% of the conceptor's trace (the other 83% is spread over the remaining 85 directions at lower gains). What follows describes that leading ten-dimensional subspace, not the whole conceptor. Rotating the ten eigenvectors by varimax, with the 42,000 corpus rows as the indicators, re-expresses the same subspace in axes along which contexts have simple structure. The result is in Figure 7 and Table 9, and the summary has three parts. First, the arrow is mostly one direction in either basis: on the full 96 sentences one eigenvector carries 64% of it and the top rotated axis 73%, and in two fits on 36 of the 48 pairs one eigenvector carries 82–86% and the top rotated axis 68–70%. That eigenvector is the second or third by gain, not the first; the first eigenvector, the direction along which the example sentences vary most, carries 6–10% of the arrow. Rotation does not concentrate the arrow beyond what the eigenbasis already does. Second, the top rotated axis separates held-out sentences by sentiment: fitting the conceptor and the rotation on pairs 1–36 and scoring the 12 held-out pairs, it separates positive from negative with $d = 3.6$ (the next axis 1.3, no other above 1.0), and on a second split (pairs 25–36 held out) with $d = 4.7$ (next 2.0). In sample, on all 96, the figures are 3.8 and 1.0.

Third, and this is the comparison that decides what the rotation is for, the same held-out pairs are separated by the arrow itself, the supervised difference of means of the 36 fitting pairs, with $d = 7.4$ and 6.4 on the two splits; by the arrow's projection into the ten-dimensional subspace with $d = 7.0$ and 6.2; by the single most aligned eigenvector with $d = 4.7$ and 5.6; by the first eigenvector with $d = 0.4$ and 0.8; and by a random direction with mean $|d| = 0.6$ (95th percentile 1.5), or 0.9 (2.3) for a random direction inside the subspace (Table 8). The rotated axis is well above chance and at the level of the best unrotated eigenvector, and well below the arrow. Rotation inside the subspace therefore does not improve separation, and nothing in it could: the arrow is fitted to the labels and the rotation is not, and an unsupervised axis that happens to align with a supervised direction (cosine 0.83 on

both splits) separates about as well as the alignment allows. What the rotation adds is names. The other nine axes are what else the example sentences vary in, and the top axis's own context list (Table 9) shows how weak those names can be: its highest corpus contexts are proper names from *Emma* and a bench, alongside *joy* and *inexpressibly*, and the label "sentiment" rests on the held-out separation statistic, not on the homogeneity of the list.

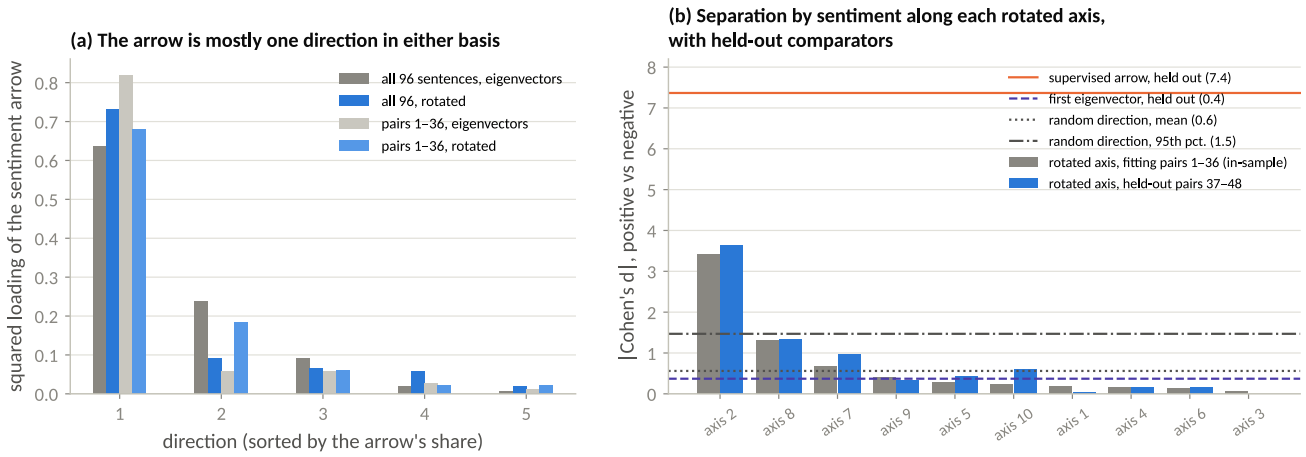


Figure 7. Rotation inside the sentiment conceptor's leading 10-direction subspace at layer 8. (a) The arrow's squared loadings on the conceptor's eigenvectors and on the varimax-rotated axes of the same subspace, sorted, for the fit on all 96 sentences and for the fit on pairs 1–36. (b) Separation of positive from negative sentences along each rotated axis of the 36-pair fit, in sample and on the 12 held-out pairs, with the held-out separation along the supervised arrow, the first eigenvector, and a random direction (mean and 95th percentile) for comparison.

direction (fitted on 36 pairs, scored on 12 held-out pairs)	$\|d\|$, split 1 (pairs 37–48 held out)	$\|d\|$, split 2 (pairs 25–36 held out)	what it is
sentiment arrow (difference of means)	7.4	6.4	supervised; the reference
arrow projected into the conceptor's 10-direction subspace	7.0	6.2	99% of the arrow lies in the subspace
most aligned eigenvector of the conceptor	4.7	5.6	unsupervised; the 2nd and 3rd eigenvector by gain
top varimax axis of the subspace	3.6	4.7	unsupervised; cosine 0.83 with the arrow
first eigenvector of the conceptor	0.4	0.8	the direction of largest variance among the sentences
random direction inside the subspace	0.9 (2.3)	0.9 (2.4)	mean (95th percentile) over 1,000 draws
random direction in the full space	0.6 (1.5)	0.6 (1.5)	mean (95th percentile) over 2,000 draws

Table 8. Held-out separation of positive from negative sentences at layer 8 along the rotated axis and its comparators. Cohen's d with sample variances, 12 pairs per split, so the sampling error of each entry is of order 0.5.

axis	arrow loading	gain	*d*, all 96 (in sample)	contexts at the high end (bracketed token is the row)	low end
3	0.86	0.98	3.8	`Donwell[ Abbey] · not wished you[ joy] · Mr. Knight[ley] · an inexpress[ibly]`	`could not taste theirs,[ and] · hoarse and strange,[ and]`
9	0.30	0.96	1.0	`a happy circumstance,[ and] · entertaining, as aff[ording] · unpretentious[ exterior]`	`said Alice to[ herself] · Donwell[ Abbey]`
4	0.25	0.98	0.7	`in the world, and[ have] · Mr. Woodhouse's[ family] · among the[ crew]`	`for it now, I[ suppose] · way you can;--[but]`
8	0.24	0.98	0.6	`* * * * *[ *] (section breaks) · on to the shelf[ and]`	`raise the[ siege] · of the world![ land]`
10	0.14	0.96	0.4	`didn't much[ matter] · nothing more[ happened] · It smelt so[ good]`	`Donwell[ Abbey] · I[ suppose]`
5	0.11	0.97	0.3	`Cheerful[,] · an intelligent[,] · so artful,[ bold]`	`Stop this[ moment] · winds,[ tumultuous]`
6, 2, 7, 1	≤ 0.10	0.97–0.98	≤ 0.2	`word-fragment, proper-name and sentence-final axes with no sentiment content`	

Table 9. The ten varimax-rotated axes of the sentiment conceptor's leading subspace at layer 8 (fit on all 96 sentences), ordered by the arrow's loading. The top axis's highest contexts include proper names and a bench as well as *joy* and *inexpressibly*; the label "sentiment" rests on the separation statistic, not on the homogeneity of the context list.

The practical point is the contrast with Section 6.2, restated with the baselines in view. Rotating a whole layer answers "what does this layer represent, in this corpus?", and a specific concept may be largely outside the answer, or inside it for reasons that have nothing to do with the concept. Rotating inside a conceptor's leading subspace answers "what did my examples make this conceptor keep?": the concept is one of the axes it finds, and the other nine are what else the examples share, a "happy circumstance / entertaining" axis, a narrative-continuation axis, a punctuation axis, an exclamation axis, each with gains near 1. That restates the caution of Moore (2026) in interpretable terms. A conceptor fitted to these 96 sentences keeps sentiment *and* nine other things the sentences share, and the rotation says what they are; it does not sharpen the concept, and should not be reported as if it did. Both computations are cheap; the second is a varimax of a 42,000 × 10 matrix.

7 Rotation in the conceptor workflow: proposals

The preceding sections establish what rotation does to the objects a conceptor practitioner handles. This section says how it might be used. Everything here is a proposal, grounded in the demonstrations above but not tested beyond them, and the order is from most to least conservative.

7.1 Choosing a layer with two criteria instead of one

The current practice is to fit at several depths and keep the one that works, guided by separability of the examples (Cohen's *d*, classification accuracy) or by the quota (Triantafyllopoulos et al., 2026). Section 6 shows that a second, different property is available at the same cost, provided it is measured properly: how much of the concept lies inside the leading directions of a reference set *matched to the kind of state the concept is read from* (sentence-final states for a sentence-level concept), compared with a label-shuffled null built from the same example states, and how concentrated it is on one interpretable axis once the conceptor's own subspace is rotated. Measured against all tokens of a general corpus and against random-direction chance, the statistic mostly tracks where the example states sit, as Section 6.2 found; measured against the matched reference and the shuffled null it picked out layer 8 for sentiment, the layer at which the examples were also best separated. The first property says where the examples are separable; the second says where the concept occupies more of the layer's own directions than an arbitrary difference of example states does, and where it is one nameable thing rather than a mixture. Reporting both, layer by layer, would turn "blindly trying layers" into a two-dimensional choice with a reason attached: a layer where separability is high and the concept is one axis is the natural place to steer, while a layer where separability is high but the concept is spread across several axes is a warning that the conceptor will carry those other axes along.

7.2 The conceptor profile as a standard output

A fitted conceptor is reported today by its quota, its gain spectrum and its overlaps with other conceptors. Section 6.3 adds a *profile*: the gain kept along each axis of an interpretable basis, with the axis's label, understood as a directional summary with the caveat stated there. Two bases are useful and both are cheap. The layer's own rotated basis (Section 6.2) places the conceptor among what the layer represents in general, which is what one needs to anticipate side effects of steering; the conceptor's own rotated subspace (Section 6.4) names what the examples made it keep, which is what one needs to judge the examples. Overlaps between conceptors become axis-by-axis statements ("both keep the negation axis; only the opinion conceptor keeps the valence axis"), which is more useful than a scalar, provided the off-diagonal part of the rotated operator is also reported when axes are claimed to be independently kept or muted.

7.3 Diagnosing the example set before fitting

The caution of Moore (2026), that a conceptor captures everything its examples share, becomes operational with rotation. Before fitting, project the example rows on their leading directions and rotate; the rotated axes list the factors present in the example set, and Table 9 is what the list looks like: one axis of sentiment, one of "entertaining / happy circumstance", one of narrative continuation, one of punctuation. A practitioner can then rewrite or rebalance the examples so that the nuisance axes vary independently of the concept, or build a conceptor on the nuisance axes alone and remove it with NOT before steering. Because the Boolean-style operations are basis-invariant (Section 5.1),

this can be done in whichever coordinates are convenient, with the centre convention fixed in advance: the operations act on matrices, and two conceptors fitted around different means must be referred to one centre before they are combined.

7.4 Oblique and bifactor rotation for correlated concepts

Section 3 found nothing for an oblique rotation to do in the token table by the pairwise criterion, and then found, through the spectrum of Φ, that what the oblique rotation had done was to disperse a general factor; the bifactor rotation was the right model, and gave the general factor a column of its own while leaving the group factors with their poles unmixed. Concepts are the same situation in miniature. Triantafyllopoulos et al. (2026) report overlaps near 0.3 between sentiment and expressions of depression across five models; "negative affect" is plausibly a general factor above both, and the unique part of each is what a NOT operation is meant to isolate. The proposal is to fit conceptors to the correlated concepts separately, pool the leading subspaces, and rotate them with contexts as indicators, either obliquely (reading pattern coefficients, which are the partial contributions of each axis holding the others constant) or, when a general factor is evident, by bi-quartimin, which gives the shared "negative affect" axis its own column and leaves the group axes to carry what is specific to depression and to sentiment. A conceptor built on the specific axes of concept A is then a candidate operational form of "A with the common part removed"; it is not the same operation as the conceptor A AND NOT B, which intersects subspaces rather than residualizing one direction against another, and whether the two agree in practice is an empirical question. Two cautions. Fitting in oblique coordinates must use equation (7) with its transported default, or fit in the trained coordinates and transport. And oblique axes are not orthonormal, so the gain along an axis is aCa^{T} only for axes expressed in the trained coordinates and normalized there.

7.5 Three further possibilities

Three more uses follow from Sections 4–6 and are recorded briefly, since none has been tried. *Conceptors by design:* if a rotated basis contains an axis to keep and an axis to mute, a conceptor can be written down rather than fitted, $C = A\,\mathrm{diag}(g)A^{\mathrm{T}}$ with orthonormal axes A and chosen gains $0 \le g_j \le 1$ (a gain of exactly 1 being the hard-projector limit of equation (3)), and composed with fitted conceptors through the Boolean-style operations; the layer-12 profile of Section 6.3 suggests the first use, a sentiment steer that keeps the valence axis and mutes the "negation coming" axis. *Models kept in interpretable coordinates:* Section 4 licenses storing a model in a rotated basis, so that fitting, reading and steering can all be done in coordinates whose axes have names, and the weights that read from and write to the stream become readable in the same terms (which heads read the valence axis, which MLP writes to it); an orthogonal basis needs only the centring correction of Section 4.1, an oblique one carries Φ through the normalization layer and the fitting formula, and bases at different depths can be aligned by orthogonal Procrustes (Schönemann, 1966; Smith et al., 2017) so that a concept's axis is followed through the layers rather than refitted at each. *Sparse autoencoders* (Bricken et al., 2023; Huben et al., 2024) are, in the terms of this report, a rotation to simple structure with more axes than dimensions and a nonlinear encoder; the profile of Section 7.2 can be computed on dictionary directions exactly as on rotated axes, since aCa^{T} needs only a unit direction, and what

a rotation offers that a dictionary does not is that it folds back into the model (Section 4) and preserves the conceptor operations exactly (Section 5): a dictionary for resolution, a rotation for exactness.

8 Limitations

Everything numerical here is GPT-2 small, one model of one family, with LayerNorm rather than RMSNorm; the depth results in particular should be re-run on a modern model, and the accompanying table-rotation script works for any Hugging Face model at reduced rank (the layer analyses require the model's activations and are not covered by it). The concept studied is sentiment, from 96 short sentences written for Moore (2026), and the neutral-sentence conceptor is fitted on 16, which is too few to define a region well and makes its comparison with the sentiment conceptor a comparison of sample sizes as much as of content. The corpus used to rotate layers is nineteenth-century literary prose, and the rotated axes reflect it; the sentiment arrow, by contrast, is read from the sentence-final states of short modern review sentences, so the capture statistic of Section 6.2 confounds concept with register and token position; the two controls reported there (a label-shuffled null and the corpus's own sentence-final states as the reference set) address token position and the composition of the arrow, and the depth trend did not survive them, but register remains uncontrolled, since the review sentences are modern and the corpus is not. Context snippets are decoded from the token stream around each position and can cross the 128-token window the model saw. No comparison was made with an ICA basis or with the unrotated principal directions beyond the concentration statistic of Table 7, so the claim that varimax axes are more readable than alternatives rests on inspection, not on a benchmark. Extraction in the $k < d$ analyses is by principal components, chosen as a descriptive method; the choices $k = 30$, 60 and 10 were made for readability rather than by a stopping rule. The full-table varimax was stopped at 600 iterations with the criterion changing by 0.03% per thirty iterations; the bi-quartimin runs converged by their gradient criterion and agreed across eleven starts. Interpretability of axes was judged by reading top tokens and contexts, not by the automated word-intrusion test of Park et al. (2017) or by human raters, and the "word family" count rests on a three-letter prefix rule. The held-out separation statistics of Section 6.4 are from two splits of 48 pairs, with 12 pairs held out in each, so their sampling error is large; the in-sample ones are summaries. On those splits the supervised difference-of-means direction separates the held-out pairs better than any rotated axis, and the result of Section 6.4 is correspondingly modest: rotation names the directions in the conceptor's subspace, it does not improve on the arrow as a classifier. Superposition means that rotated axes remain polysemantic to a degree; the claim is that they are far more readable than the trained ones, not that they are

monosemantic. The change-of-basis result of Section 4 is exact and does not share these limitations, but its usefulness depends on finding bases worth changing to, which is where the limitations above live.

9 Conclusion

Factor rotation and the residual stream are made for each other: the one is a method for choosing readable coordinates, the other a space whose coordinates are arbitrary by construction. The contributions here are correspondingly plain: a change of basis stated in the generality a psychometric rotation needs, the transformation law of the conceptor under it, and two ways of reading a conceptor on rotated axes, each measured against a baseline. Applied to GPT-2's lookup table, varimax finds hundreds of nameable factors and one giant factor; promax recovers the same group factors without reducing cross-loadings, and its factor correlations have a single large eigenvalue; the bifactor rotation that this calls for yields a general factor on which every token loads, associated with low word frequency in the way the gradient account of Gao et al. (2019) and Biś et al. (2021) predicts, and group factors whose poles are no longer mixed with the common direction. Applied to the model, a change of basis of any kind, orthogonal or oblique, leaves it exactly unchanged once the reads, writes, normalization and unembedding are transformed according to Table 4; the orthogonal case was known, and in the oblique case the input and output tables stand as pattern matrix to structure matrix. Applied to conceptors, a change of basis is a similarity transform that preserves gains, quota, action and Boolean-style operations and changes only the description, while the aperture regularizer and its default rule acquire the metric. And applied to a layer, rotation yields interpretable axes on which a conceptor can be read as a labelled profile; where a sentiment concept sits in a layer's leading directions depends on the reference set and the null it is compared with, and once both are matched to the concept's own kind of state it stands out at one layer only, the layer at which the examples are also best separated; and rotation inside the conceptor's own leading subspace yields an unsupervised axis that separates held-out sentences by sentiment, less sharply than the supervised direction does, and names the nuisance variation beside it. The proposals of Section 7 are best guesses at how to use this in practice; the demonstrations of Sections 3–6 are what they rest on.

References

Abad, F. J., Garcia-Garzon, E., Garrido, L. E., & Barrada, J. R. (2017). Iteration of partially specified target matrices: Application to the bi-factor case. *Multivariate Behavioral Research*, 52(4), 416–429. https://doi.org/10.1080/00273171.2017.1301244

Arora, S., Li, Y., Liang, Y., Ma, T., & Risteski, A. (2018). Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6, 483–495. https://doi.org/10.1162/tacl_a_00034

Ashkboos, S., Croci, M. L., Nascimento, M. G. do, Hoefler, T., & Hensman, J. (2024). SliceGPT: Compress large language models by deleting rows and columns. In *International Conference on Learning Representations*. https://arxiv.org/abs/2401.15024

Ashkboos, S., Mohtashami, A., Croci, M. L., Li, B., Cameron, P., Jaggi, M., Alistarh, D., Hoefler, T., & Hensman, J. (2024). QuaRot: Outlier-free 4-bit inference in rotated LLMs. In *Advances in Neural Information Processing Systems 37*. https://arxiv.org/abs/2404.00456

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). *Layer normalization*. arXiv. https://arxiv.org/abs/1607.06450

Biś, D., Podkorytov, M., & Liu, X. (2021). Too much in common: Shifting of embeddings in transformer language models and its implications. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 5117–5130). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.naacl-main.403

Bernaards, C. A., & Jennrich, R. I. (2005). Gradient projection algorithms and software for arbitrary rotation criteria in factor analysis. *Educational and Psychological Measurement*, 65(5), 676–696. https://doi.org/10.1177/0013164404272507

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., Lasenby, R., Wu, Y., Kravec, S., Schiefer, N., Maxwell, T., Joseph, N., Hatfield-Dodds, Z., Tamkin, A., Nguyen, K., … Olah, C. (2023). Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. https://transformer-circuits.pub/2023/monosemantic-features/index.html

Browne, M. W. (2001). An overview of analytic rotation in exploratory factor analysis. *Multivariate Behavioral Research*, 36(1), 111–150. https://doi.org/10.1207/S15327906MBR3601_05

Cattell, R. B. (1952). The three basic factor-analytic research designs: Their interrelations and derivatives. *Psychological Bulletin*, 49(5), 499–520.

Dufter, P., & Schütze, H. (2019). Analytical methods for interpretable ultradense word embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (pp. 1185–1191). Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1111

Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., DasSarma, N., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., … Olah, C. (2021). A mathematical framework for transformer circuits. *Transformer Circuits Thread*. https://transformer-circuits.pub/2021/framework/index.html

Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., Grosse, R., McCandlish, S., Kaplan, J., Amodei, D., Wattenberg, M., & Olah, C. (2022). Toy models of superposition. *Transformer Circuits Thread*. https://transformer-circuits.pub/2022/toy_model/index.html

Elhage, N., Lasenby, R., & Olah, C. (2023). Privileged bases in the transformer residual stream. *Transformer Circuits Thread*. https://transformer-circuits.pub/2023/privileged-basis/index.html

Ethayarajh, K. (2019). How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (pp. 55–65). Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1006

Faruqui, M., Tsvetkov, Y., Yogatama, D., Dyer, C., & Smith, N. A. (2015). Sparse overcomplete word vector representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing* (Vol. 1) (pp. 1491–1500). Association for Computational Linguistics. https://doi.org/10.3115/v1/P15-1144

Gao, J., He, D., Tan, X., Qin, T., Wang, L., & Liu, T.-Y. (2019). Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*. https://arxiv.org/abs/1907.12009

Hendrickson, A. E., & White, P. O. (1964). Promax: A quick method for rotation to oblique simple structure. *British Journal of Statistical Psychology*, *17*(1), 65–70. https://doi.org/10.1111/j.2044-8317.1964.tb00244.x

Hofmann, R. J. (1978). Complexity and simplicity as objective indices descriptive of factor solutions. *Multivariate Behavioral Research*, *13*(2), 247–250.

Hollis, G., & Westbury, C. (2016). The principals of meaning: Extracting semantic dimensions from co-occurrence models of semantics. *Psychonomic Bulletin & Review*, *23*(6), 1744–1756. https://doi.org/10.3758/s13423-016-1053-2

Holzinger, K. J., & Swineford, F. (1937). The bi-factor method. *Psychometrika*, *2*(1), 41–54. https://doi.org/10.1007/BF02287965

Huben, R., Cunningham, H., Riggs, L., Ewart, A., & Sharkey, L. (2024). Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations*. https://openreview.net/forum?id=F76bwRSLeK

Jaeger, H. (2014). *Controlling recurrent neural networks by conceptors*. arXiv. https://arxiv.org/abs/1403.3369

Jaeger, H. (2017). Using conceptors to manage neural long-term memories for temporal patterns. *Journal of Machine Learning Research*, *18*(13), 1–43.

Jennrich, R. I. (2001). A simple general procedure for orthogonal rotation. *Psychometrika*, *66*(2), 289–306. https://doi.org/10.1007/BF02294840

Jennrich, R. I., & Bentler, P. M. (2011). Exploratory bi-factor analysis. *Psychometrika*, *76*(4), 537–549. https://doi.org/10.1007/s11336-011-9218-4

Jennrich, R. I., & Sampson, P. F. (1966). Rotation for simple loadings. *Psychometrika*, *31*(3), 313–323. https://doi.org/10.1007/BF02289465

Kaiser, H. F. (1958). The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, *23*(3), 187–200. https://doi.org/10.1007/BF02289233

Karve, S., Ungar, L., & Sedoc, J. (2019). Conceptor debiasing of word representations evaluated on WEAT. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing* (pp. 40–48). Association for Computational Linguistics. https://doi.org/10.18653/v1/W19-3806

Kovaleva, O., Kulshreshtha, S., Rogers, A., & Rumshisky, A. (2021). BERT busters: Outlier dimensions that disrupt transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021* (pp. 3392–3405). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.findings-acl.300

Liu, T., Ungar, L., & Sedoc, J. (2019a). Unsupervised post-processing of word vectors via conceptor negation. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*(1), 6778–6785. https://doi.org/10.1609/aaai.v33i01.33016778

Liu, T., Ungar, L., & Sedoc, J. (2019b). Continual learning for sentence representations using conceptors. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Vol. 1)* (pp. 3274–3279). Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1331

Mansolf, M., & Reise, S. P. (2016). Exploratory bifactor analysis: The Schmid–Leiman orthogonalization and Jennrich–Bentler analytic rotations. *Multivariate Behavioral Research*, *51*(5), 698–717. https://doi.org/10.1080/00273171.2016.1215898

Moore, T. M. (2026). *Conceptors: An introduction from first principles*. Manuscript.

Mu, J., & Viswanath, P. (2018). All-but-the-top: Simple and effective postprocessing for word representations. In *International Conference on Learning Representations*. https://openreview.net/forum?id=HkuGJ3kCb

Murphy, B., Talukdar, P., & Mitchell, T. (2012). Learning effective and interpretable semantic models using non-negative sparse embedding. In *Proceedings of COLING 2012* (pp. 1933–1950). https://aclanthology.org/C12-1118/

Musil, T., & Mareček, D. (2024). Exploring interpretability of independent components of word embeddings with automated word intruder test. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)* (pp. 6922–6928). ELRA and ICCL. https://aclanthology.org/2024.lrec-main.605/

Park, S., Bak, J., & Oh, A. (2017). Rotated word vector representations and their interpretability. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 401–411). Association for Computational Linguistics. https://doi.org/10.18653/v1/D17-1041

Postmus, J., & Abreu, S. (2024). *Steering large language models using conceptors: Improving addition-based activation engineering*. arXiv. https://arxiv.org/abs/2410.16314

Puccetti, G., Rogers, A., Drozd, A., & Dell'Orletta, F. (2022). Outlier dimensions that disrupt transformers are driven by frequency. In *Findings of the Association for Computational Linguistics: EMNLP 2022* (pp. 1286–1304). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.findings-emnlp.93

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). *Language models are unsupervised multitask learners*. OpenAI.

Reise, S. P., Moore, T. M., & Haviland, M. G. (2010). Bifactor models and rotations: Exploring the extent to which multidimensional data yield univocal scale scores. *Journal of Personality Assessment*, 92(6), 544–559. https://doi.org/10.1080/00223891.2010.496477

Rohe, K., & Zeng, M. (2023). Vintage factor analysis with Varimax performs statistical inference. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85(4), 1037–1060.

Rothe, S., Ebert, S., & Schütze, H. (2016). Ultradense word embeddings by orthogonal transformation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 767–777). Association for Computational Linguistics. https://doi.org/10.18653/v1/N16-1091

Schmid, J., & Leiman, J. M. (1957). The development of hierarchical factor solutions. *Psychometrika*, 22(1), 53–61. https://doi.org/10.1007/BF02289209

Schönemann, P. H. (1966). A generalized solution of the orthogonal Procrustes problem. *Psychometrika*, 31(1), 1–10. https://doi.org/10.1007/BF02289451

Smith, S. L., Turban, D. H. P., Hamblin, S., & Hammerla, N. Y. (2017). Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *International Conference on Learning Representations*. https://openreview.net/forum?id=r1Aab85gg

Stephenson, W. (1935). Correlating persons instead of tests. *Character and Personality*, 4(1), 17–24.

Subramanian, A., Pruthi, D., Jhamtani, H., Berg-Kirkpatrick, T., & Hovy, E. (2018). SPINE: SParse Interpretable Neural Embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence* (pp. 4921–4928). AAAI Press.

Thurstone, L. L. (1947). *Multiple-factor analysis*. University of Chicago Press.

Timkey, W., & van Schijndel, M. (2021). All bark and no bite: Rogue dimensions in transformer language models obscure representational quality. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (pp. 4527–4546). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.emnlp-main.372

Triantafyllopoulos, I., Cho, Y.-M., Tao, R., Miao, M. M., Rai, S., Ungar, L., Guntuku, S. C., Ryant, N., & Sedoc, J. (2026). *Conceptors for semantic steering*. arXiv. https://arxiv.org/abs/2605.04980

Westbury, C., Yang, M., & Anderson, K. (2025). The principal components of meaning, revisited. *Psychonomic Bulletin & Review*, *32*(1), 203–225. https://doi.org/10.3758/s13423-024-02551-y

Yamagiwa, H., Oyama, M., & Shimodaira, H. (2023). Discovering universal geometry in embeddings with ICA. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (pp. 4647–4675). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.283

Yifei, L. S., Ungar, L., & Sedoc, J. (2023). Conceptor-aided debiasing of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (pp. 10703–10727). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.661

Zhang, B., & Sennrich, R. (2019). Root mean square layer normalization. In *Advances in Neural Information Processing Systems 32*. https://arxiv.org/abs/1910.07467

A Reproducibility

Model and data. GPT-2 small (124M parameters; 12 layers; width 768; vocabulary 50,257), using OpenAI's released weights as exported in the ONNX Model Zoo (`gpt2-lm-head-10.onnx`, converted to a NumPy archive by `gpt2np.py`) and OpenAI's byte-pair tokenizer files (`encoder.json`, `vocab.bpe`), re-implemented in NumPy so that every residual-stream row can be read and edited and every weight transformed. The lookup table is the tensor `wte`; the same matrix is obtained from `GPT2Model.from_pretrained("gpt2").wte.weight`. The weights (635 MB) are not bundled; the archive's `README` gives the download locations and the conversion step. The NumPy tokenizer uses ASCII character classes where OpenAI's regular expression uses Unicode classes; all text processed here is ASCII, so the two agree on it, and the tokenizer was checked against OpenAI's on the sentences and prompts used. The 96 sentiment sentences (48 matched pairs) and 16 neutral sentences are those of Moore (2026) and are in `sentences.py`; their residual-stream rows are regenerated by `experiments_fast.py`. The corpus for Section 6 is the Gutenberg selection distributed with NLTK (six books, first 7,000 tokens of each, 41,984 tokens in all, processed in 128-token windows), with the corpus activations stored in half precision; all rotations and conceptor computations are in double precision, and the basis-change verification is in double precision throughout.

Execution order (all plain NumPy; two CPU cores; wall-clock times in parentheses). (1) `experiments_fast.py`: sentence states (minutes). (2) `factor_analyses.py`: R-mode and Q-mode analyses ($k = 30$) and the full varimax–promax rotation (varimax 42 min; promax under a minute); `consistent_transform.py` then derives the total promax transform consistent with the sign-flipped, reordered pattern. (3) `bifactor.py qmode 30` (bi-quartimin from eleven starts, varimax, and the target rotation; about 8 min) and `bifactor.py full` (bi-quartimin on the full table from the varimax solution; 400 iterations, 16 min). (4) `basis_change_demo.py`: the change-of-basis model (`GPT2InBasis`) and the

verification cases of Table 5 (about three minutes per case). (5) `collect_states.py` (38 min) and `layer_rotation.py` (25 min): Section 6. (6) `heldout_baselines.py`, `period_control.py` and `period_control2.py`: the baselines of Section 6 (the held-out comparators of Table 8, the sentence-final reference set, and the label-shuffled null; minutes). (7) `paper_computations.py` and `paper_computations2.py`: every number in the text and tables, written to `numbers.json`; `figs_paper2.py` and `figs_update.py`: every figure. `rotate_embeddings_any_model.py` applies the table rotations to any Hugging Face model at full or reduced rank; it does not perform the layer analyses. The varimax, promax, oblimin, bi-quartimin and target-rotation routines were checked against R's `stats::varimax`, `stats::promax` and `GPArotation` on small matrices.

Conventions. Rows are scaled to unit length before rotation (Kaiser normalization) and loadings are reported on that scale. Factors are sign-flipped so that each factor's largest loading is positive (the bifactor general factor so that its mean is positive) and ordered by their sum of squared loadings. Promax uses power 4 and the column scaling of R's `promax`. Bi-quartimin uses orthogonal gradient projection with the criterion of Jennrich and Bentler (2011), scaled by the number of rows; the target rotation uses Browne's (2001) partially specified target with a free first column and zeros at |varimax loading| < 0.2. Conceptors use $\lambda = \mathrm{tr}(R)/d$ unless stated, and a ridge of $10^{-8}I$ in the AND operation. Cohen's d uses sample variances. "Salient" means |loading| ≥ 0.3.

B The bi-quartimin group factors

The 29 group factors of the $k = 30$ Q-mode bi-quartimin solution of Section 3.4, with a reading from the top fifty tokens, the seven highest-loading tokens, and the number of tokens for which the factor is the primary group factor. Group loadings are small in absolute terms because the general factor holds 71% of the common variance; the readings rest on the ordering of tokens, which is unambiguous.

G	Reading	Highest-loading tokens (top loading)	primary for
1	frequent function words and punctuation	␣and, ␣in, ␣the, ␣a, ␣(, ␣to, ␣of (+0.69)	4,716
2	word-internal fragments	ane, ia, ous, iles, ale, oes, ade (+0.28)	5,313
3	government and administration	␣Government, ␣Federal, ␣Department, ␣governmental, ␣Administrative, ␣Executive, ␣Administration (+0.31)	2,222
4	three-digit numbers	␣277, ␣278, ␣279, ␣283, ␣267, ␣296, ␣257 (+0.47)	1,511
5	plural common nouns	␣teams, ␣cars, ␣boxes, ␣vehicles, ␣groups, ␣rooms, ␣locations (+0.36)	2,988
6	evaluative intensifiers	␣incredibly, ␣ridiculously, ␣extremely, ␣ridiculous, ␣incredible, ␣terrible, ␣insanely (+0.41)	1,945
7	sentence-initial words	Even, Most, There, Some, When, In, While (+0.37)	2,574
8	male first names	␣Scott, ␣John, ␣Michael, ␣Jason, ␣Brian, ␣Ryan, ␣Jeff (+0.37)	2,521

G	Reading	Highest-loading tokens (top loading)	primary for
9	capitalized game and weapon nouns	▁Attack, ▁Laser, ▁Power, ▁Weapon, ▁Magic, ▁Armor, ▁Speed (+0.30)	2,406
10	violence and crime	▁violence, ▁murder, ▁destruction, ▁killing, ▁terror, ▁abuse, ▁terrorist (+0.35)	1,217
11	past-tense verbs	▁launched, ▁pushed, ▁returned, ▁brought, ▁dropped, ▁attacked, ▁pulled (+0.38)	2,080
12	Asian and Middle-Eastern nationalities	▁Turkish, ▁Azerbaijan, ▁Indonesian, ▁Kazakhstan, ▁Malaysia, ▁Istanbul, ▁Malaysian (+0.40)	1,719
13	chemistry and biomedicine	▁chemical, ▁vitamin, ▁chemicals, ▁bacteria, ▁bacterial, ▁protein, ▁pharmaceutical (+0.38)	1,339
14	gender and sexuality	▁gay, ▁gender, ▁LGBT, ▁sexual, ▁feminist, ▁transgender, ▁socially (+0.35)	1,040
15	entertainment events	▁championship, ▁announcement, ▁superstar, ▁anniversary, ▁premiere, ▁celebration, ▁celebrity (+0.26)	1,415
16	finance	▁profit, ▁savings, ▁profits, ▁cash, ▁Profit, ▁financial, ▁payment (+0.30)	1,196
17	ALL-CAPS words	▁ALL, ▁YOU, ▁ANY, ▁SH, ▁EVERY, ▁NOT, ▁NO (+0.33)	2,414
18	computing	▁Linux, ▁blockchain, ▁desktop, ▁computer, ▁binary, ▁digital, ▁software (+0.31)	1,267
19	the divine and eternal	▁divine, ▁immortality, ▁Gods, ▁eternal, ▁Eternal, ▁gods, ▁resurrection (+0.32)	1,097
20	-ing forms	▁creating, ▁bringing, ▁putting, ▁making, ▁removing, ▁providing, ▁taking (+0.43)	1,467
21	geography and the urban	▁southwest, ▁coastal, ▁northwest, ▁rural, ▁neighborhood, ▁Northwest, ▁Lake (+0.32)	1,038
22	quantitative decline (deficit, regression)	▁reduction, ▁regression, ▁deficit, ▁slump, ▁decrease, ▁progression, ▁stagnation (+0.29)	945
23	research and science	▁research, ▁analysis, ▁science, ▁researcher, ▁scientific, ▁scientist, ▁researchers (+0.33)	902
24	soldiers, warriors, weapons	▁soldier, ▁warrior, ▁knight, ▁fighter, ▁captain, ▁sword, ▁assassin (+0.38)	1,001
25	programming identifiers	▁identifier, ▁object, ▁exclude, ▁adapter, ▁assign, ▁interface, ▁imply (+0.26)	993
26	innovation and enterprise	▁energ, ▁innov, ▁entrepreneurs, ▁accelerate, ▁entrepreneur, ▁innovate, ▁innovation (+0.30)	871
27	political rhetoric	▁rhetoric, ▁ideology, ▁stance, ▁rhetorical, ▁Bannon, ▁ideological, ▁rebuke (+0.35)	767
28	anxiety and distress	▁anxiety, ▁sleep, ▁anguish, ▁sigh, ▁awake, ▁anxious, ▁despair (+0.30)	696
29	years	▁1929, ▁1939, ▁1933, ▁1945, ▁1960, ▁1930, ▁1959 (+0.34)	597

Roughly half the factors describe form (1, 2, 4, 5, 7, 11, 17, 20, 29) and half topic; tokens typically load on one of each, the cross-loading pattern of Section 3.2 at coarser resolution.

C Derivations

C.1 Folding LayerNorm's affine parameters. With $n = (h - \mu\mathbf{1})/\sigma$, a LayerNorm followed by a read is $\mathrm{LN}(h)W + b = (\gamma \odot n + \beta)W + b = n\,\mathrm{diag}(\gamma)W + (\beta W + b)$. In new coordinates $n' = nM$, so $n\,\mathrm{diag}(\gamma)W = n'M^{-1}\mathrm{diag}(\gamma)W$; the new read is $M^{-1}\mathrm{diag}(\gamma)W$, the new bias is $b + \beta W$, and the new LayerNorm returns n' with unit gain and zero bias. For the tied unembedding the same fold gives $U' = M^{-1}\mathrm{diag}(\gamma_f)E^{\mathrm{T}}$ and the logit offset $\beta_f E^{\mathrm{T}}$.

C.2 The normalized vector in new coordinates. $\mu = h\mathbf{1}^{\mathrm{T}}/d = h'M^{-1}\mathbf{1}^{\mathrm{T}}/d = h'c/d$ with $c = M^{-1}\mathbf{1}^{\mathrm{T}}$. The centred vector $h - \mu\mathbf{1}$ has new coordinates $(h - \mu\mathbf{1})M = h' - \mu\mathbf{1}'$. Its squared Euclidean length is $(h' - \mu\mathbf{1}')M^{-1}M^{-\mathrm{T}}(h' - \mu\mathbf{1}')^{\mathrm{T}} = (h' - \mu\mathbf{1}')\Phi(h' - \mu\mathbf{1}')^{\mathrm{T}}$, so σ^2 in equation (5) equals σ^2 in equation (2), stabilizer included. If M is orthogonal with $M^{\mathrm{T}}\mathbf{1}^{\mathrm{T}} = \mathbf{1}^{\mathrm{T}}$ then $c = \mathbf{1}^{\mathrm{T}}$, $\mathbf{1}' = \mathbf{1}$ and $\Phi = I$, so (5) is (2).

C.3 Pattern in, structure out. With $\Lambda = E$ and $T = M$: the input table EM is the pattern matrix ΛT; the transposed unembedding without the final gain, $EM^{-\mathrm{T}} = \Lambda T^{-\mathrm{T}} = \Lambda T(T^{\mathrm{T}}T)^{-1} = \Lambda_{\mathrm{p}}\Phi$, is the structure matrix. A logit is then (pattern coordinates of the state) · (structure coordinates of the token), the usual pairing of a vector with the dual basis. With the final gain folded in, the same holds for $\Lambda = E\,\mathrm{diag}(\gamma_f)$.

C.4 The conceptor fitting formula in oblique coordinates. Let $R' = M^{\mathrm{T}}RM$ be the covariance matrix of the centred examples in the new coordinates, so $R = M^{-\mathrm{T}}R'M^{-1}$. Then $R + \lambda I = M^{-\mathrm{T}}(R' + \lambda M^{\mathrm{T}}M)M^{-1}$, and $(R + \lambda I)^{-1} = M(R' + \lambda M^{\mathrm{T}}M)^{-1}M^{\mathrm{T}}$. Substituting into $C' = M^{-1}R(R + \lambda I)^{-1}M$ gives $C' = M^{-1}M^{-\mathrm{T}}R'(R' + \lambda M^{\mathrm{T}}M)^{-1}M^{\mathrm{T}}M = \Phi R'(R' + \lambda\Phi^{-1})^{-1}\Phi^{-1}$, which is equation (7). The default rule transports as $\mathrm{tr}(R) = \mathrm{tr}(M^{-\mathrm{T}}R'M^{-1}) = \mathrm{tr}(R'M^{-1}M^{-\mathrm{T}}) = \mathrm{tr}(R'\Phi)$. Self-adjointness: $C'\Phi = M^{-1}CM(M^{\mathrm{T}}M)^{-1} = M^{-1}CM^{-\mathrm{T}}$, which is symmetric because C is.

C.5 Reading a gain along an axis. For a unit row vector a in the trained coordinates, aCa^{T} is the component the filter keeps along a from an input along a; it lies in $[0, 1)$ and equals the gain when a is an eigenvector, and it does not describe the components the filter may move into other directions. For an orthonormal set of axes A ($d \times k$), the share of the conceptor's trace inside their span is $\mathrm{tr}(A^{\mathrm{T}}CA)/\mathrm{tr}(C)$. Both are what Section 6 reports.

C.6 The bi-quartimin criterion. For a $p \times k$ loading matrix Λ with group columns Λ_2 = columns 2 to k, $B(\Lambda) = \Sigma_i\, \Sigma_{r \neq s}\, \lambda_{ir}{}^2\lambda_{is}{}^2$ over the group columns, with gradient $4(0, \Lambda_2 \odot (\Lambda_2{}^{\odot 2}\mathrm{N}))$, N the off-diagonal ones matrix. It is zero if and only if each row of Λ_2 has at most one nonzero entry (Jennrich & Bentler, 2011, Theorem 1), and it does not depend on the first column, which the orthogonal rotation is free to fill with whatever the group columns leave.