Geometry under anchoring: a whole-geometry measurement over the stored runs

A reanalysis of 131 stored checkpoints from three experiments. Past the first block, an anchored model does not arrange its colours the way an un-anchored one does. By the last block, the colour geometry of an anchored run correlates with that of a control at half the level two controls reach, in every experiment. Dropping the anchor axis brings the recipe at 50 epochs back to the control band. The heavier anchors, the longer-trained recipe, and the handover arms stay under it, and their seeds agree with each other more closely than controls do. So a light anchor seems to add an axis to a control's geometry; a heavier or longer one also moves the deep geometry to a different arrangement, one that seeds reproduce and one less like the RGB cube.

Observations

Each line below is a measurement on stored checkpoints, judged against the spread between un-anchored controls. None of them is a result; the closing section says what a preregistered follow-up would test.

- Whole geometry: at the embedding, anchored runs are about as close to a control as controls are to each other. The gap opens with depth. At the last block, the colour geometry of an anchored run correlates with that of a control at roughly half the control-against-control level. That holds for the D2.1 recipe, the six-op recipe at both lengths, the three heavier survey points, and all three handover arms.
- The anchor axis: with e_1 dropped from every run, the six-op recipe at 50 epochs comes back inside the control band at the last two blocks. The D2.1 recipe and the two mid-weight survey points come back most of the way, the 100-epoch recipe and the heaviest point about a third of the way, and the handover arms barely move. So what differs in those last cases is the geometry of the other 63 coordinates. At the embedding, dropping e_1 moves every anchored condition a little *under* the band, because it removes *red* from the anchored side only.
- Agreement among seeds: at depth, anchored runs of one condition agree with each other more closely than controls do. With e_1 dropped, the extra agreement stays for the heavier anchors, the 100-epoch recipe, and the handover arms. It goes away for the λ=0.1 recipes at 50 epochs, which are the conditions that dropping e_1 put back in the band. So where the geometry differs beyond the axis, seeds reproduce that difference.
- Dose: with the geometry as it is, the drop at the last block is a step at λ=0.1, with little further change up to λ=0.557. With e_1 dropped, the heavier anchors keep less of the control's geometry than the recipe does.
- The colour cube: at the last two blocks, anchored runs arrange the colours less like the RGB cube than controls do. The embedding and first block are alike.
- Procrustes agrees with RSA on every measurement, so a rotation and a rescaling would not remove the difference.

Scope

This is a reanalysis of stored checkpoints, planned in the backlog item: no training, no gates, no verdicts. Every anchored model we have has already been scored for the axis it was given (alignment, margin, containment) and for its task. But whether the *rest* of its colour geometry matches what an un-anchored model builds has only been measured through per-channel probe R^2 (the H2 of ex-2.1.12), which came back unresolved. Here we ask that of the geometry as a whole.

Why

The D2.1 post claims that anchoring guides one concept to a known place and leaves the rest of the representation to form as it would have. That claim rests on the task metric (an anchored model still answers as well as a control) and on probes for the other channels.

Neither one looks at the geometry itself. A model can score the same while arranging its colours differently, and a linear probe can find green wherever it is put. So the claim needs a statistic that compares the *shape* of the representation between an anchored run and an un-anchored one, with the ordinary variation in that shape between un-anchored seeds as the yardstick.

Representational similarity analysis (RSA) is such a statistic, and the one most of this report uses.[1] For each run it measures the distance in latent space between every pair of the 216 grid colours, which gives a matrix of distances. It then correlates the matrices of two runs. Two runs that place the colours the same way, up to a rotation, correlate at 1, whatever basis each of them chose; runs that arrange them differently score lower. The baseline is control against control: runs that differ only by seed set the correlation a faithful anchored run should reach. Procrustes, below, is the second shape statistic. It asks the same question through the coordinates rather than the distances, so agreement between the two is a check on each. The states lie on a unit hypersphere, so the Euclidean distance between two of them is the chord, which falls as the cosine rises (see Method).

Anchoring is meant to move something: *red* is asked to lie along the e_1 axis. So a second variant of every measurement drops e_1 from the states of every run before the distances are taken, anchored and control alike. That variant asks whether the geometry *other than the anchor axis* is what a control builds.

The runs

Every run comes from a checkpoint a published experiment left in the store. Three experiments, four groups, each with its own un-anchored control seeds.

run	λ	seeds	what it is
`ex-2.1.10/lam0` (control)	0	3	un-anchored
`ex-2.1.10/either-t100`	0.1	9	the D2.1 recipe
`ex-2.2.3/control-short` (control)	0	5	un-anchored, 50 epochs
`ex-2.2.3/recipe-short`	0.1	20	the recipe, 50 epochs
`ex-2.2.3/t12`	0.361	5	survey proposal t12
`ex-2.2.3/t48`	0.431	5	survey proposal t48
`ex-2.2.3/t00`	0.557	5	survey proposal t00
`ex-2.2.3/control` (control)	0	5	un-anchored, 100 epochs
`ex-2.2.3/recipe`	0.1	20	the recipe, 100 epochs
`ex-2.2.9/control` (control)	0	5	un-anchored
`ex-2.2.9/handover`	0.1	20	untied readout, whole-line labeller
`ex-2.2.9/handover-slot`	0.1	20	untied readout, slot labeller
`ex-2.2.9/handover-tied`	0.1	9	tied readout, whole-line labeller

The runs. *Every seed of each condition is one checkpoint from the store. Conditions in one group share the control they are compared with, marked control in the table and in every legend. ex-2.2.3's* `recipe-short` *and* `recipe` *include the fifteen addendum seeds each; ex-2.2.9's* `handover-tied` *has nine.*

The measurement

For each run, the residual states at each site are collected into a `216 × 64` matrix, one row per grid colour. At *operand 1*, the state above the first token depends on that token alone, since attention is causal, so it is the context-free representation of the colour. At *operand 2*, the state depends on the whole prompt so far, and we average it over every first operand under the reference op (`mix`, or `+` on the D2.1 grammar). Both sites are taken at all five residual slices.

Each figure below shows, per condition and slice, a column of seed dots: the mean RSA of each run to the controls of its group.[2] For a control run, that mean is taken over the *other* controls. The grey strip behind each slice is the range of control-against-control pairs, so an anchored condition whose dots sit in the strip is as close to a control as controls are to each other.

The lower row of each figure drops e_1, the anchor axis, from the states of every run first. The checks below say how much of the variance of each run that coordinate accounts for, and whether it is where a probe finds *red*.

Whole geometry, per experiment

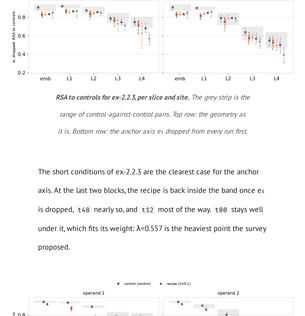
What we expected. If anchoring only moves *red* and leaves the rest alone, an anchored run should sit in the control band once e_1 is dropped, at every slice, and near it before.

What we saw. At the embedding, every anchored condition is within a few hundredths of the control band. The gap then opens block by block. At the last block, every anchored condition sits well under the band at both sites: the D2.1 recipe, the six-op recipe at either length, the three heavier survey points, and all three handover arms.

Dropping e_1 closes the gap for the six-op recipe at 50 epochs, which returns to the band at the last two blocks. It closes most of the gap for the D2.1 recipe and the two mid-weight survey points, about a third of it for the 100-epoch recipe and the heaviest point, and almost none on the handover grammar, where the three arms stay far under the band even with e_1 gone.

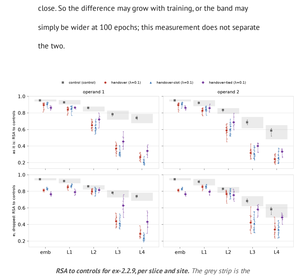
At the embedding, dropping e_1 goes the other way: every anchored condition falls a little under the band. We would expect that. For an anchored run e_1 holds *red*, and for a control it holds nothing in particular, so dropping the coordinate removes *red* from one side of the comparison only. A fairer variant would find the direction each control keeps *red* along, if it keeps one, and drop that too. We have not done that, and the omission works against the anchored side, so the deeper blocks come back in spite of it.

The band itself moves. Controls agree with each other closely at the embedding, and a little less at each block after it. The drop is steeper at operand 2, where the last-block band sits at about half its embedding level in every group. The same fall shows against the colour cube: the geometry of a control is most cube-like at the first block and drifts from the cube after that. So the part of the geometry that seeds share seems to be the part the input imposes, and each block replaces some of it with an arrangement of its own. At operand 2 the state is on its way to an answer, and there is more to replace. Why the seeds do not settle on one deep arrangement is open, and the follow-up below could take it up.



RSA to controls for ex-2.1.10, per slice and site. *The grey strip is the range of control-against-control pairs. Top row: the geometry as it is. Bottom row: the anchor axis e_1 dropped from every run first.*

ex-2.1.10 has three control seeds, so the band is only three pairs wide and should be taken loosely. The recipe falls under the band from the second block on. Dropping e_1 recovers most of the gap at the last two blocks, to within a few hundredths of the band.

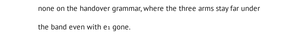


RSA to controls for ex-2.2.5, per slice and site. *The grey strip is the range of control-against-control pairs. Top row: the geometry as it is. Bottom row: the anchor axis e_1 dropped from every run first.*

The short conditions of ex-2.2.3 are the clearest case for the anchor axis. At the last two blocks, the recipe is back inside the band once e_1 is dropped, `t48` nearly so, and `t12` most of the way. `t00` stays well under it, which fits its weight: λ=0.557 is the heaviest point the survey proposed.

RSA to controls for ex-2.2.3, 100 epochs, per slice and site. *The grey strip is the range of control-against-control pairs. Top row: the geometry as it is. Bottom row: the anchor axis e_1 dropped from every run first.*

The same recipe trained for 100 epochs sits further under its band than the 50-epoch one, and dropping e_1 recovers less of the gap. As a fraction of the width of the band, though, the e_1-dropped row is still close. So the difference may grow with training, or the band may simply be wider at 100 epochs; this measurement does not separate the two.



RSA to controls for ex-2.2.9, per slice and site. *The grey strip is the range of control-against-control pairs. Top row: the geometry as it is. Bottom row: the anchor axis e_1 dropped from every run first.*

On the handover grammar, every arm is far under the band at the last two blocks, and dropping e_1 changes little. The tied readout (`handover-tied`) keeps the most, the slot labeller the least. The arms are compared at operand 1 under `mix`. The control shares the eleven-op grammar, stochastic rounding, and the untied readout with `handover` and `handover-slot`, so the grammar itself is not what makes the difference. The readout is where `handover-tied` differs, so it sitting closest to the control is not a like-for-like comparison.

run	as it is: to controls	to self	e_1 dropped: to controls	to self
ex-2.1.10/1a00 (control)	0.57–0.66	0.62	0.56–0.66	0.62
ex-2.1.10/either-t100	0.45	0.77	0.54	0.55
ex-2.2.3/control-short (control)	0.69–0.81	0.75	0.68–0.81	0.75
ex-2.2.3/recipe-short	0.55	0.88	0.71	0.74
ex-2.2.3/t12	0.54	0.87	0.65	0.78
ex-2.2.3/t48	0.59	0.87	0.68	0.82
ex-2.2.3/t00	0.52	0.87	0.56	0.85
ex-2.2.3/control (control)	0.69–0.82	0.76	0.69–0.82	0.76
ex-2.2.2/recipe	0.42	0.92	0.52	0.86
ex-2.2.9/control (control)	0.68–0.79	0.74	0.69–0.78	0.74
ex-2.2.9/handover	0.27	0.89	0.29	0.89
ex-2.2.9/handover-slot	0.21	0.93	0.24	0.92
ex-2.2.9/handover-tied	0.34	0.93	0.43	0.88

RSA at the last block, operand 1. *Per condition, for the geometry as it is and with e_1 dropped: to controls is the seed mean of each run's RSA to its group's controls, and to self the mean over every pair of runs within the condition. On a control's row, to controls is instead the control band, the range of those same pairs, and to self their mean.*

Agreement among seeds

What we expected. A run whose deep geometry is far from every control could get there two ways. Either anchoring adds seed-to-seed variation, in which case the run is far from every other run as well; or anchoring replaces one arrangement with another, in which case the run is close to the other runs of its own condition. Comparing runs within a condition tells the two apart.

What we saw. The second one, with the geometry as it is. Past the first block, anchored runs of one condition agree with each other more closely than controls do, in every group. The agreement among controls falls with depth, while the agreement within an anchored condition holds.

With e_1 dropped, that extra agreement stays for the heavier anchors, the 100-epoch recipe, and the handover arms. It goes away for the two λ=0.1 recipes at 50 epochs; in ex-2.1.10 it falls a little under the control level. Those are the same conditions that dropping e_1 put back in the band, so the two measurements tell one story.

The seeds of a light anchor share an axis and otherwise vary as controls do. The seeds of a heavier or longer anchor share an arrangement beyond the axis, and they reproduce it more closely than controls reproduce theirs.

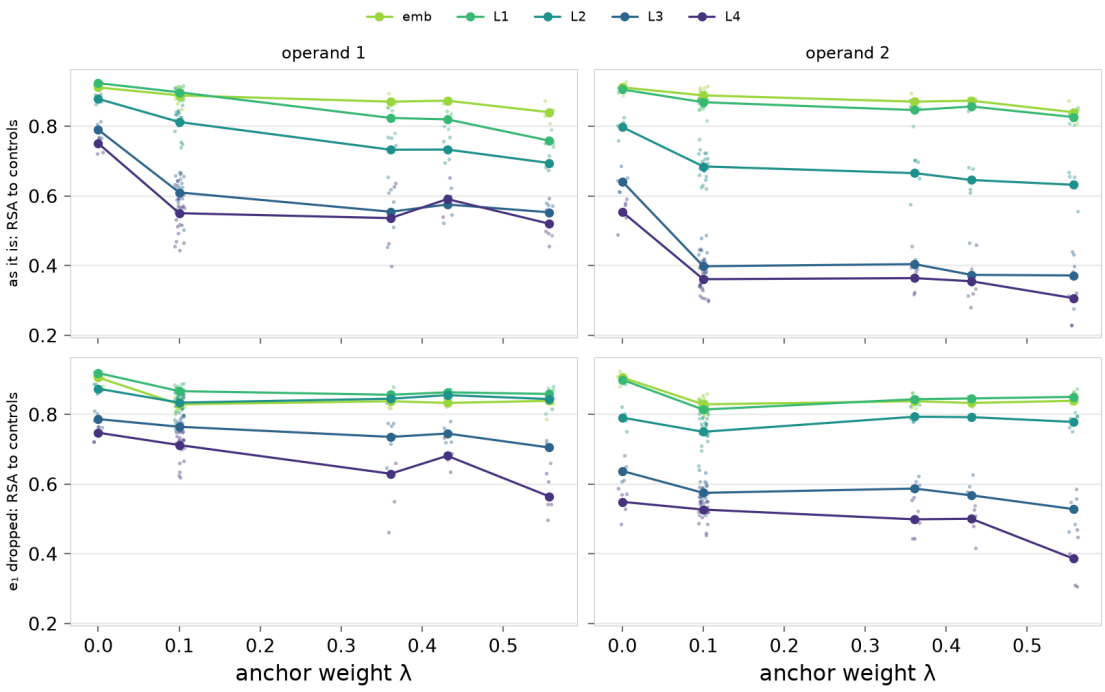
RSA within each condition, at operand 1. *Each small dot is one pair of runs of the same condition, the larger mark the mean over pairs, and the grey strip the range of control pairs (the same strip as in the figures above). Top row: the geometry as it is. Bottom row: e_1 dropped from every run.*

Dose

What we expected. If the difference scales with the anchor, the three heavier survey points should sit further from the controls than the recipe, in order of λ.

What we saw. With the geometry as it is, the drop at the last block is a step rather than a slope. The recipe at λ=0.1 is already most of the way down, and the heavier points scatter around it. At the earlier blocks the conditions do fall in order of λ.

With e_1 dropped, the last block is graded too: the recipe returns to the band, `t48` nearly reaches it, and `t12` and `t00` stay further off. So the anchor axis seems to account for a fixed part of the distance at any λ, while the rest grows with the weight.



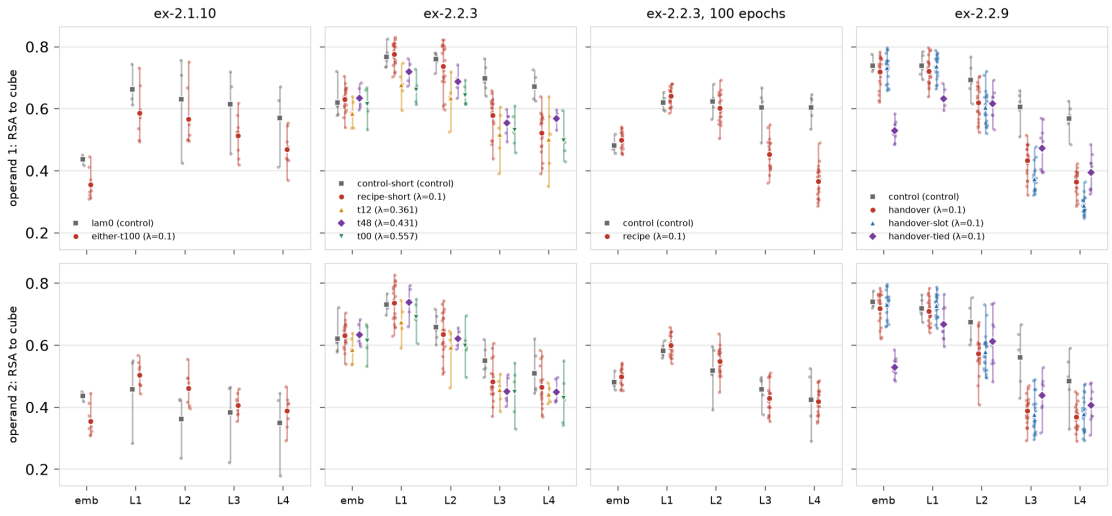
RSA to controls against anchor weight, on ex-2.2.3's short conditions.

Each line is one residual slice (the embedding lightest, the last block darkest), joining the seed means; the dots behind them are seeds. λ=0 is the control, compared with the other controls. Top row: the geometry as it is. Bottom row: e_1 dropped from every run.

Against the colour cube

A control does not arrange the colours as the RGB cube does either, but it comes closer than an anchored run does. This measurement correlates each run's colour distances with the straight-line distances between the same colours in RGB.

At the embedding and the first block, the conditions are alike, with the embedding of the tied readout as the one exception. From the second block on, the anchored conditions fall further. At the last block on the handover grammar, a control correlates with the cube at about a half and the handover arms at about a third. The heavier ex-2.2.3 anchors and `handover-tied` sit between.



RSA between each run's colour geometry and the RGB cube itself, per slice and site. *The cube's distances are the Euclidean distances between the 216 grid colours in RGB. Each dot is one seed, the larger mark the seed mean. A run that arranges the colours as the cube does scores 1.*

Checks on the anchor axis

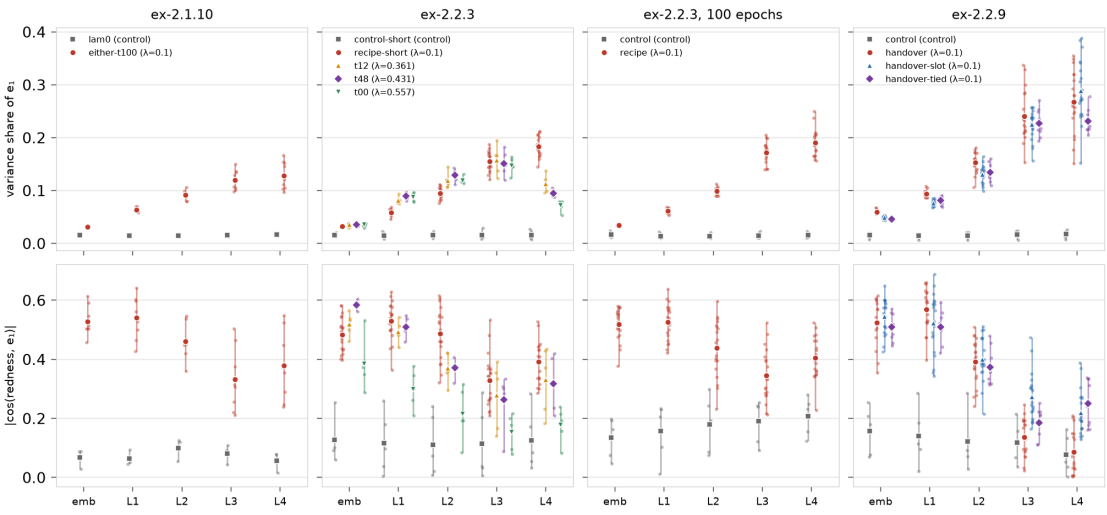
Two measurements of e_1 itself, so that the `e1 dropped` rows above can be interpreted.

The first is how much of a run's variance sits along e_1. In an anchored run it is twice the control level at the embedding, and it grows with depth, reaching ten to eighteen times the control level at the last block. For a control the share is one part in 64, the same as any other coordinate. So the axis is where anchoring put its variance, as intended.

At the last block on the handover grammar, dropping e_1 removes up to a quarter of an anchored run's total variance, against under two percent for a control.

The second is how closely e_1 lines up with the direction a ridge probe reads *red* from. For anchored runs the cosine between them is about a half at the embedding and the first block, and it falls with depth; for controls it is about a tenth. So the probe finds red mostly along e_1 early and less so late, which is how ex-2.1.12 saw it.

Together these say what the `e1 dropped` rows take away: a coordinate that is a large part of the deep variance of an anchored run and only part of where its *red* lives. The coordinate is largest on the handover grammar, and even there the last-block gap remains once it is gone.

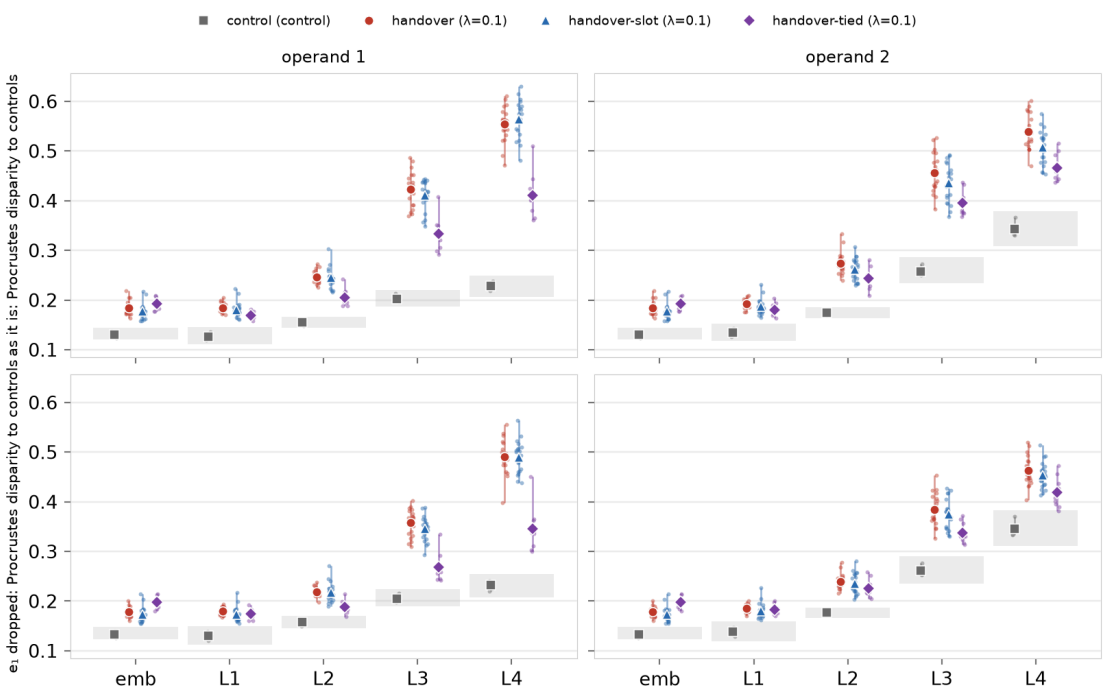


The anchor axis at operand 1. *Top: the share of the states' total variance along e_1. Bottom: the absolute cosine between e_1 and the direction a ridge fit (to the grading target `sim_to_red`, power 1.5) finds red along. Each dot is one seed, the larger mark the seed mean.*

A second statistic: Procrustes

RSA works from distances, so it cannot see a rotation. Procrustes disparity takes a different route to the same question. It finds the rotation, reflection, and scale that best map the states of one run onto those of another, then reports what is left over. So it also compares shape, but through the coordinates rather than the distances.

It agrees with RSA on every measurement. Anchored runs sit above the control band from the second block on, which here means further from the controls. Dropping e_1 changes little on the handover grammar, and the tied readout keeps the most. The figure shows the handover grammar; the other groups are in the store.



Procrustes disparity to controls for ex-2.2.9, per slice and site. *The grey strip is the range of control-against-control pairs. Top row: the geometry as it is. Bottom row: the anchor axis e_1 dropped from every run first.*

What we make of it

The D2.1 post claims that anchoring leaves the rest of the representation to form as it would have. For a light anchor and a short run, that is what we see here once the axis is set aside: the ex-2.2.3 recipe at 50 epochs is a control geometry plus e_1, and the D2.1 recipe is close to that.

It is not what we see for a heavier anchor, a longer run, or the handover grammar. There the colours are arranged differently beyond the axis, and the difference grows with depth, with training, and with the anchor weight. Seeds reproduce that arrangement more closely than un-anchored seeds reproduce theirs, and it is less like the RGB cube.

Three things this analysis does not say. It does not say the task is affected: every one of these conditions matched its control on held-out exact match in its own report. It does not say the other channels are lost: ex-2.1.12 found green and blue as decodable as before. And it does not say where in training the two geometries part company, since we only measured final checkpoints.

What it says is that the *arrangement* of the colours changes, which is what a whole-geometry statistic measures and a per-channel probe does not.

Perhaps the anchor term reshapes the deep geometry around the axis it is given, so the rest of the space organises relative to *red* rather than as an un-anchored model would have it. The falling RSA against the cube and the rising agreement within a condition both fit that.

Or maybe the anchored geometry is the un-anchored one, stretched along e_1 and sheared, in a way that dropping a single coordinate cannot undo. The recipe at 50 epochs fits the stretching part, since dropping e_1 puts it back in the band; whether the handover arms fit the shearing part is open. Telling the two readings apart is a preregistered question, and the statistics here are cheap enough to run at every checkpoint of a training run.

To do. A preregistered experiment with one hypothesis: with e_1 dropped, the last-block RSA of an anchored run to the controls is inside the control band. The stored runs already say it holds for the six-op recipe at 50 epochs and misses for the handover grammar. So the experiment should measure the handover grammar through training, at every saved checkpoint, and add an arm at a lower λ.

Two exploratory measurements to carry with it: comparing each run against a *redness-only* set of distances, which would say whether the anchored arrangement organises by red; and the same figures at the answer position, where the side-effects of an intervention would land.

Method

Prompts. Every ordered pair of the 216 grid colours runs as the three-token prompt `a op b`, with `op` the grammar's `mix` where it has one and `+` on the D2.1 grammar. The residual stream is taken at every slice above positions 0 and 2. The state above position 0 is the same for every `b` (checked to 1e-5 in every run); the state above position 2 is averaged over `a`.

Dissimilarity. From a `216 × 64` state matrix we take the pairwise Euclidean distances and keep the upper triangle. The states lie on the nGPT hypersphere, so this is the chord distance, which falls as the cosine rises. A cosine dissimilarity would order every pair the same way, since one is a monotone function of the other. Pearson correlation does respond to that change of scale, so the numbers would shift a little, but no ordering in this report would change. RSA is then the Pearson correlation between the upper triangles of two runs.

The anchor axis. The `e₁ dropped` variant deletes coordinate 0 of every run's states before the distances are taken, and the states are not renormalised. The redness direction in the checks comes from a ridge fit (`l2 = 1e-2`, on centred states) from the states of a run to the grading target `sim_to_red(GRID_RGB, power=1.5)`, normalised to a unit vector.

Procrustes. SciPy's `procrustes`. It centres both matrices and scales them to unit Frobenius norm, finds the best orthogonal map from one to the other, and reports the sum of squared residuals as the disparity. Two geometries that differ only by a rotation, a reflection, or a scale score 0.

Where the data is. The run table is under the `metrics` ref of `reports/m2/geometry-rsa`, and every pairwise matrix under the `arrays` ref. The raw states of each run are under `states/{experiment}/{label}`, so a follow-up can read them without a forward pass.

---

1. Those matrices of distances are *representational dissimilarity matrices*. Correlating two of them asks whether both runs find the same colours near each other and far apart, without requiring them to use the same coordinates. The measure does not change if you rotate the state space, so it is blind to *where* the anchor put red. ↩

2. In every figure of this kind, each small dot is one seed's mean over the controls, the larger mark is the seed mean, and the bar is the seed range. RSA is the correlation of colour-distance matrices between one run and each of the group's control runs, so higher is more alike. Procrustes disparity is the residual after the best rotation and scale between the two, so lower is more alike. ↩