REPRESENTATIONAL SIMILARITY ACROSS AI WEATHER MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

AI weather models produce skilful forecasts competitive with traditional Numerical Weather Prediction, but it remains unclear whether these independently developed models learn similar internal representations. We investigate representational similarity across six AI weather models, which comprise three variants of WeatherNext 1 Graph (GraphCast), AIFS-CRPS, Aurora and ArchesWeather, using five similarity metrics which include CKA and orthogonal Procrustes. Spatial and temporal variability in the atmosphere is often dominated by climatological latitude and seasonal variations that obscure the effects of synoptic weather systems. To account for this, we construct a null calibration of the metrics that preserves the large-scale structure, making the metrics more responsive to similarity in the representation of weather systems. We find non-zero calibrated similarity between all model pairs, strongest and most consistent within the GraphCast family, while rankings among the other models depend on the metric and spatial data used. Global CKA relationships remain stable throughout the year, whereas regional comparisons show greater seasonal variability, particularly in the midlatitudes. We also assessed representational similarity across processor depth in individual models. GraphCast and AIFS-CRPS representations evolve relatively smoothly, whereas Aurora and ArchesWeather change more strongly due to processor-layer boundaries. Together, these results suggest partially shared representations of atmospheric states, whose organisation and construction differ across models, and demonstrate the importance of calibrated, multi-metric analysis.

1 INTRODUCTION

Data-driven weather models have become increasingly competent in the past few years, to the extent that they can now match or outperform traditional numerical weather prediction (NWP) across a variety of forecast metrics (Lam et al., 2023), and are beginning to be incorporated into operational forecasting systems (Dueben et al., 2026). With these advances come many questions surrounding how they construct forecasts and whether their internal mechanisms are reliable.

Work on AI weather-model interpretability has begun, with many techniques for explaining pretrained models largely based on perturbation-, game-theory- and gradient-based attribution methods (Yang et al., 2024). Tempest et al. (2026) focus instead on visualising the latent representations and analysing the meaning of some channels, but cross-model and across-layer comparison remains difficult. Using intermediate decoding of these latents, Beylich et al. (2026) find that forecasts are constructed progressively through the processor, with successive blocks refining the prediction from larger to smaller spatial scales in a way that reflects model architecture. Craig et al. (2026) build on this to interpret latent-state evolution as an approximate gradient flow towards a minimum of a learned free-energy functional, and use Centered Kernel Alignment (CKA) to provide initial evidence that different AI weather-model families learn similar final-processor representations at selected grid points. Questions such as how the internal representation changes through a model's processor, and how much the representations of different models have in common are important building blocks for understanding how AI weather models operate.

Similarity studies have been used extensively in other fields of Machine Learning (ML) to understand learned representations. In general, similarity measures can be split into functional and rep-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

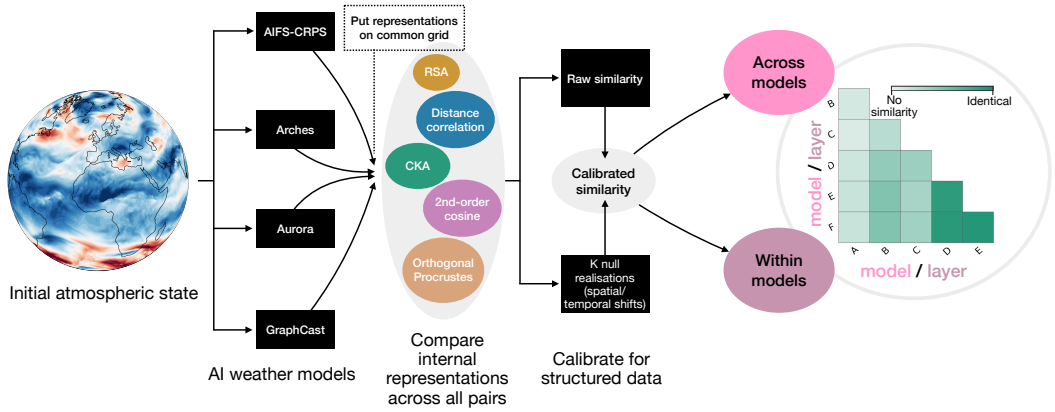


Figure 1: **Overview of the representational similarity analysis.** Internal representations are mapped to a common grid, compared using similarity measures, and calibrated using structure-preserving null transformations for comparisons both across, and within, AI weather models.

resentational measures: functional similarity concerns differences in model behaviour or outputs, whereas representational similarity compares intermediate activations within the models (Klabunde et al., 2025a). Previous studies have used representational similarity to understand differences between trained models and their learned features, as well as to study transfer learning and feature reuse (Kornblith et al., 2019; Neyshabur et al., 2021). Representational similarity therefore provides a natural framework through which to ask whether independently developed AI weather models have learned related internal descriptions of the atmosphere.

With hundreds of proposed similarity measures (Cloos et al., 2025), metric choice is seldom straightforward. Measures differ in their invariances, scale, and sensitivity to representation dimension, sample size and variance structure, and can disagree on both the magnitude and ranking of similarity (Klabunde et al., 2023; 2025a). No measure performs consistently best, as evaluated in the ReSi benchmark (Klabunde et al., 2025b); CKA, as an example of a popular metric, can inflate when model width is large relative to sample size (Gröger et al., 2026) and can be dominated by leading principal components (Ding et al., 2021). Because different metrics therefore emphasise different properties of a representation, Cloos et al. (2024) argue for multi-metric evaluation and dataset-specific interpretation. This is particularly relevant for AI weather models, where properties of the data can strongly influence the outcome of a metric. In particular, the variance of global meteorological data has large contributions associated with pole to equator contrasts, and with the seasonal cycle. A measure that is dominated by these components may provide little information about how weather systems are represented.

One way of making similarity scores easier to interpret is to calibrate them against a permutation-based null distribution, providing a baseline for the similarity expected without correspondence between the two representations (Gröger et al., 2026). For atmospheric data, however, unrestricted permutations destroy strong spatial and temporal dependencies like the climatological latitude and seasonal structures, and can therefore give an inappropriate null. We instead use structure-preserving null transformations that retain these dependencies while breaking correspondence between model representations on the space and time scales associated with weather systems.

In this paper we first evaluate several similarity measures and a structure-preserving calibration for AI weather representations from four different model families. Secondly, we use these measures to compare final-processor representations across models as well as representation evolution within individual processors (Figure 1). CKA is our primary metric, complemented by second-order cosine similarity, orthogonal Procrustes, representational similarity analysis (RSA), and distance correlation.

Our main contributions and findings are:

- Calibrating similarity scores is effective for observing similarity above that expected due to spatial and temporal dependencies of the atmospheric data. Metric verification is fur-

thermore essential, as magnitudes and in select cases, rankings of the scores, depend on the metric and spatial samples used.

- AI weather models partially share their learned representations. Similarity is found between all pairs tested, with the strongest and most consistent relationship occurring within a model family (GraphCast). Across-model global CKA is relatively stable throughout the year, although regional comparisons show greater temporal variability, particularly in the midlatitudes.
- Processor evolution is architecture-dependent. GraphCast and AIFS-CRPS change relatively smoothly, whereas Aurora and ArchesWeather change more strongly due to boundaries within their processor.

2 METHODS

2.1 AI WEATHER MODELS

We use data from four families: the Small, Standard and Operational variants of GraphCast/WeatherNext 1 Graph (Lam et al., 2023); Aurora v1.8.0 (Bodnar et al., 2025); ArchesWeather-M (Couairon et al., 2024a); and AIFS-CRPS N320 (aifs-ens-1.0) (Lang et al., 2026). Although AIFS-CRPS is trained for ensemble forecasting, we analyse representations from a single standard forecast run. Further architecture details are given in Appendix A.1.

Unless stated otherwise, we use final-processor representations. GraphCast and AIFS-CRPS provide one feature vector per native processor node whereas Aurora and ArchesWeather have an additional latent-level dimension, which we concatenate at each horizontal location. Processing the latent levels in a different way, by using only a single latent level instead, does not qualitatively change the results (Appendix A.11).

Times refer to the forecast time, meaning inter-model comparisons match the represented atmospheric state rather than forecast initialisation time. GraphCast, Aurora and AIFS-CRPS are initialised 6 h earlier whereas ArchesWeather is initialised 24 h earlier. Unless stated otherwise, single-timestep comparisons use the forecast time of 2020-06-01 18 UTC.

For inter-model comparisons, all representations are mapped to the $10{,}242$-node GraphCast-Small processor mesh. Standard and Operational GraphCast nodes are selected directly, while Aurora, ArchesWeather and AIFS-CRPS are interpolated from their native grids or mesh. The resulting representation for one forecast time is therefore $\mathbf{R} \in \mathbb{R}^{N \times d}$, where the number of samples, N is $10{,}242$ when all spatial nodes are used and d is the feature dimension after any latent levels have been concatenated. When multiple forecast times are combined, the spatial and temporal dimensions form the sample dimension, as described in Section 2.2.

For within-model processor comparisons, we additionally test sensitivity to changes in latent basis by repeating the analysis, where translators are available, after affine alignment of each intermediate representation to the final-layer representation space (Beylich et al., 2026); see Appendix A.2.

2.2 SIMILARITY METRICS

We compare representations using centred kernel alignment (CKA), representational similarity analysis (RSA), distance correlation, second-order cosine similarity, and aligned orthogonal Procrustes similarity. CKA is our primary measure and the others are chosen to include different metric families and transformation invariances. Scores range from 0 (no similarity) to 1 (perfect similarity). Definitions, invariances and implementation details are given in Appendix A.3, including a comparison of biased and unbiased CKA (Appendix A.3.7).

We use the notation of Klabunde et al. (2025b) and write the two representations to be compared as $\mathbf{R} \in \mathbb{R}^{N \times d}$ and $\mathbf{R'} \in \mathbb{R}^{N \times d'}$, where N is the common sample dimension. For one timestep, and if all mesh nodes are used, $N = n = 10{,}242$, where n is the selection of mesh nodes in this instance. If multiple timesteps are included, t, then $N = nt$. The feature dimensions d and d' may differ between models. Sensitivity to the number of spatial samples is examined in Appendix A.10.

2.2.1 CALIBRATING SIMILARITY METRICS

Raw similarity scores can contain substantial background similarity even in the absence of direct correspondence between two representations. We therefore follow Gröger et al. (2026) to null-calibrate the similarity scores, but adapt the null construction to preserve the potentially strong spatial and temporal dependencies present in atmospheric data.

For each comparison, we generate $K = 200$ structure-preserving null realisations, in which correspondence between the two representations is broken while their large-scale spatial and temporal structure is retained. We use the upper-tail threshold, τ, at $\alpha = 0.05$ as an effective zero point, representing the level of similarity expected from this preserved structure alone. Sensitivity to the choice of K is examined in Appendix A.4.1. The calibrated score is then

$$s_{\text{cal}} = \max\left(\frac{s_{\text{obs}} - \tau}{s_{\max} - \tau}, 0\right), \tag{1}$$

such that scores, s_{obs}, at or below the null threshold are mapped to zero and the maximum possible similarity, $s_{\max}$, remains one. The raw score therefore measures the total similarity detected by the metric, whereas the calibrated score measures the similarity remaining above that expected from the preserved spatial and temporal structure alone. A calibrated score of zero therefore does not imply zero raw similarity, but rather that the observed similarity does not exceed the chosen null threshold, τ.

We additionally calculate an empirical upper-tail p-value, which gives the fraction of null realisations producing a similarity at least as large as the observed raw score, with a standard $+1$ correction. Smaller p-values therefore indicate stronger evidence that the observed similarity exceeds that expected from the preserved structure alone. With $K = 200$, the minimum possible empirical p-value is $1/(K + 1) \simeq 0.005$.

Calibrated scores are displayed in the main paper, whilst the corresponding raw scores are provided in the appendix. Further details regarding the calibration are given in Appendix A.4, while the structure-preserving spatial and temporal transformations are described in Appendix A.4.2.

2.2.2 VERIFYING SIMILARITY METRICS

Following Ding et al. (2021), we test whether the similarity measures respond consistently to functionally meaningful changes in the representations. For this, we progressively perturb each model's final latent representation with Gaussian noise, decode the perturbed representation, and compare the reduction in representational similarity with the resulting change in the forecast. A useful similarity measure should decrease as the perturbation produces greater forecast degradation. Full details of the perturbation and forecast-error calculation are given in Appendix A.3.8.

3 VERIFICATION OF SCORES AND CALIBRATION TECHNIQUE

3.1 CONNECTION BETWEEN FUNCTIONAL AND INTERNAL REPRESENTATION SIMILARITY

First, we examine the relationship between representational and functional similarity within each of the weather models, as described in Section 2.2.2. Figure 2 shows that, across all five metrics, increasing latent-space noise reduces representational similarity and increases forecast error for the four model families, the expected behaviour for a functionally meaningful similarity measure.

The response nevertheless differs between metrics. At larger perturbations, several become less sensitive as forecast error continues to increase. This is clearest for second-order cosine similarity, which can remain relatively high even when forecasts differ substantially. Orthogonal Procrustes instead falls more rapidly, assigning lower similarity at smaller forecast errors. This agrees with previous work showing greater Procrustes sensitivity to lower-variance representation directions, whereas CKA emphasises high-variance components (Cloos et al., 2024).

All five measures therefore capture the expected broad relationship between representation and model behaviour, but the relationship is neither linear nor consistent across metrics, especially for strong perturbations. The same numerical score therefore need not imply the same functional simi-

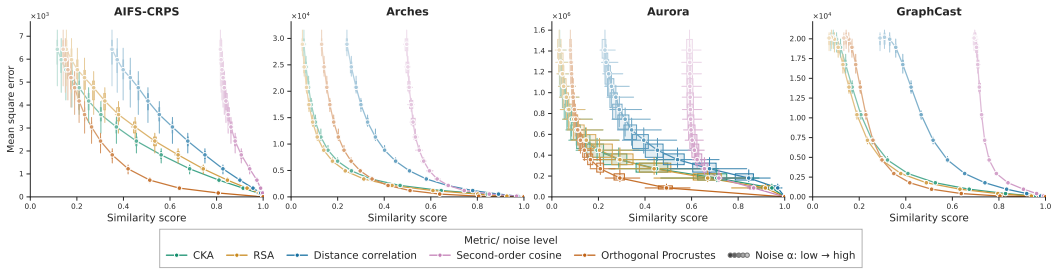


Figure 2: **Relationship between representational and functional similarity across AI weather models.** Similarity between the original and noise-perturbed latent representations is plotted against the resulting mean squared forecast error for AIFS-CRPS, ArchesWeather, Aurora and GraphCast Standard (left to right). Points show the median across 24 dates at each noise level, with boxes showing the interquartile range and whiskers the spread across dates.

larity, and systematically higher scores from measures such as second-order cosine similarity should be considered in inter-model comparisons.

3.2 Internal representations across timesteps

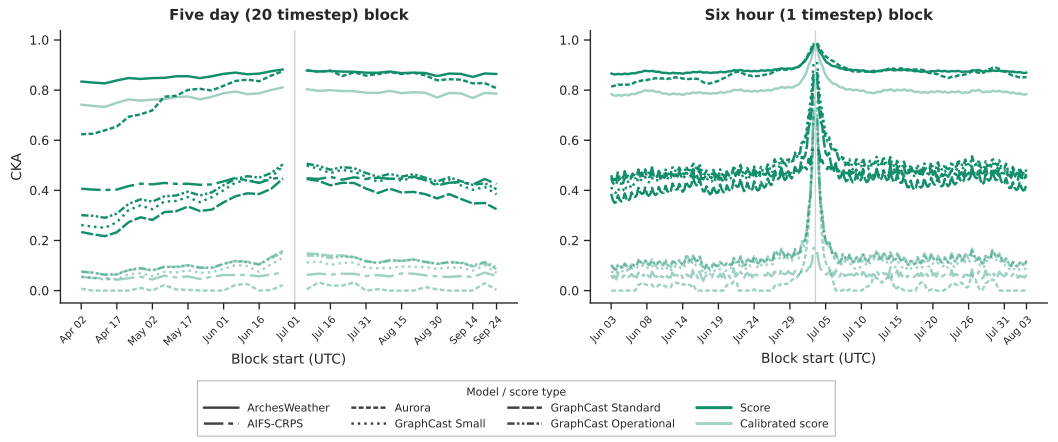


Figure 3: **Temporal similarity of model latent representations for different temporal block lengths.** Within each model, a fixed anchor is compared with non-overlapping blocks from the same time series using CKA. The left panel uses five-day blocks (anchor 1–5 July 2020), and the right uses individual six-hourly timesteps (anchor 3 July 2020 12 UTC).

To test the calibration, we compare each model's CKA at a fixed anchor time with representations at increasing temporal separation. Synoptic weather systems should remain related up to several (5-10) days, whereas similarity induced by persistent spatial structure may remain much longer. A useful calibration should therefore suppress this background without removing the expected near-anchor decay. We consider five-day blocks that should only resemble each other through the climatological background, and 6h (single timestep) blocks to test whether the local structures are retained after calibration. For both block lengths, temporal ordering is preserved and only longitude is rotated, so that latitudinal structure should also be removed by the calibration.

Figure 3 shows raw and calibrated CKA for five-day blocks (left) and individual six-hourly timesteps (right). The p-value takes its minimum possible value for nearly all comparisons, indicating that the similarity is statistically significant (Appendix A.4). The main exception is Aurora, for which it is above its minimum for 63% of five-day-block comparisons and 68% of single-timestep comparisons, reaching a maximum of 0.45. AIFS-CRPS has one comparison with a non-minimum p-value. This is not surprising considering how low the scores are for these cases, with Aurora being the only model

whose similarity reaches zero. This may be due to Aurora being the only model considered here that predicts the full atmospheric field directly, whereas the other models predict residual updates.

For five-day blocks, raw CKA has a large model-dependent background and, for four of the six models, higher similarity near the anchor. Far from the anchor, raw scores range from approximately 0.2 to 0.85. Calibration reduces most values drastically, to approximately 0-0.1, and removes the near-anchor peak, indicating that much of the raw similarity can be explained by the large-scale spatial structure retained in the representations. The removal of the peak is expected, since the atmosphere loses much of its synoptic-scale predictability over timescales of roughly 5-10 days. ArchesWeather is the clear exception, retaining calibrated CKA of approximately 0.75 even far from the anchor.

For individual timesteps, both raw and calibrated CKA show a much clearer peak near the anchor, decreasing with temporal separation and changing little after approximately five days. GraphCast and AIFS-CRPS also show a distinct cycle, with slightly greater similarity between timesteps at the same time of day. Calibration retains the local peak while removing the effects of similarities on longer time-scales.

Overall, calibration removes substantial model-dependent background similarity while preserving the expected short-timescale structure. ArchesWeather remains anomalous, which may reflect either a particularly persistent component of its latent representation or a limitation of longitude rotation as the null transformation for this model. Repeating the temporal comparison using an earlier set of anchor dates gives qualitatively similar behaviour (Appendix A.4.3).

4 Comparing representations across AI weather models

4.1 Comparing representations at a single forecast time

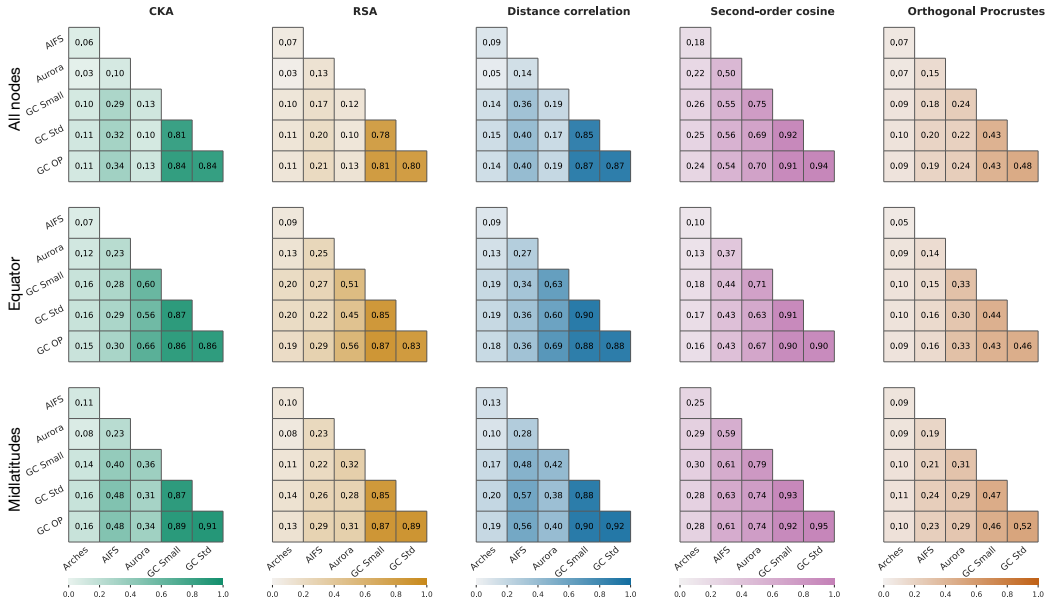


Figure 4: **Pairwise similarity of latent representations across AI weather models and spatial subsets.** Null-calibrated similarity between the final processor representations of the six models at 2020-06-01 18 UTC. Rows show all nodes (top), equatorial nodes between 10°S and 10°N (middle), and midlatitude nodes between 40° and 60° in both hemispheres (bottom). Columns show CKA, RSA, distance correlation, second-order cosine similarity and orthogonal Procrustes similarity (left to right).

Internal latent representations of the AI weather models are first compared at an individual timestep, 01-06-2020 18 UTC, using the final processor representations. Figure 4 shows the calibrated sim-

ilarity scores for the five similarity metrics between the six models, with the comparison repeated using three spatial subsets: all nodes, equatorial nodes, and midlatitude nodes. The corresponding raw scores are shown in Appendix A.5. All comparisons attain the minimum empirical p-value, $\simeq 0.005$, except for a small number of midlatitude comparisons involving ArchesWeather under CKA, RSA and distance correlation, for which $p = 0.01$.

Similarity is found between all model pairs in all similarity metrics for all three selections of nodes. Concentrating first on CKA using all nodes (top left), there is a wide range in the magnitude of similarity. As expected, the highest similarity is found within the GraphCast family, with calibrated CKA values of approximately 0.8. Among comparisons between different model families, AIFS-CRPS and the GraphCast models have relatively high similarity under global CKA, with calibrated values around 0.3. However, this ordering is not consistent when looking at different similarity metrics and spatial subsets.

The magnitude of the similarity also depends strongly on the measure used. For example, the similarity between Aurora and AIFS-CRPS ranges from 0.1 for CKA to 0.5 for second-order cosine similarity, which could be expected from Figure 2. The metrics broadly agree on which model pairs are more or less similar, but there are also differences in rank. For example, using all nodes, AIFS-CRPS is the model most similar to the GraphCast models under CKA, RSA and distance correlation, whereas Aurora is most similar under second-order cosine similarity and orthogonal Procrustes. The scale of the differences between model pairs also varies between metrics. These results demonstrate that, even after calibration, similarity values from different metrics should not be interpreted as directly equivalent.

To investigate the effect of the spatial samples used in the comparison, the calculations were repeated using specific latitude bands. The middle and bottom rows of Figure 4 show results for a 20° band around the equator and two 20° bands in the midlatitudes, respectively. In both cases, the calibrated similarity scores are generally larger than when all nodes are used. However, the increase is not uniform across models. For example, the CKA similarities involving AIFS-CRPS increase more strongly in the midlatitudes, whereas similarities involving Aurora increase more strongly at the equator.

The metrics therefore broadly identify common structure, but similarity magnitude and, in some cases, ranking, depend on both metric and spatial region. For the remainder of the paper we concentrate primarily on CKA while noting that results may be slightly different with other metrics. Repeating the analysis at 2020-01-24 06 UTC gives the same qualitative conclusions (Appendix A.6).

4.2 Comparing monthly internal representations

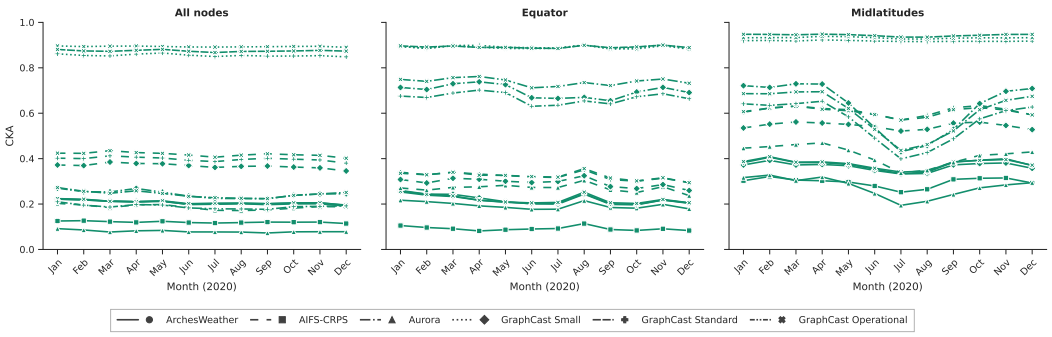


Figure 5: **Monthly pairwise calibrated CKA across spatial subsets.** Null-calibrated CKA between model representations throughout 2020, calculated using whole-month datasets for all nodes (left), equatorial nodes between 10°S and 10°N (middle), and midlatitude nodes between 40° and 60° in both hemispheres (right). Line and marker styles identify the two models in each pair. Note that all scores obtained the minimum p-value.

We next compute CKA for individual months throughout 2020, with representations including all timesteps from that month, for three spatial subsets. For these comparisons, the null calibration preserves both spatial and temporal structure, and CKA uses the minibatch implementation described

in Appendix A.3.6. Figure 5 shows that when all nodes are used (left), similarity between model pairs is fairly constant throughout the year. There is a small variability for the equatorial region (middle) however, and several midlatitude pairs (right) vary substantially. In particular, similarities involving Aurora decrease from approximately May before recovering by November. Separating the midlatitudes by hemisphere (Appendix A.7) shows that this variability arises in the Northern Hemisphere and is much clearer after calibration than in the raw CKA. The hemispheric datasets still contain approximately 1.4×10^5 samples, well above the latent dimensions (Appendix Table 1), making a small-sample explanation unlikely. These results indicate that the similarity is persistent rather than date-specific, but can vary seasonally within particular regions, suggesting differences in how specific regional atmospheric regimes are represented internally.

5 Comparing internal representations across processor steps

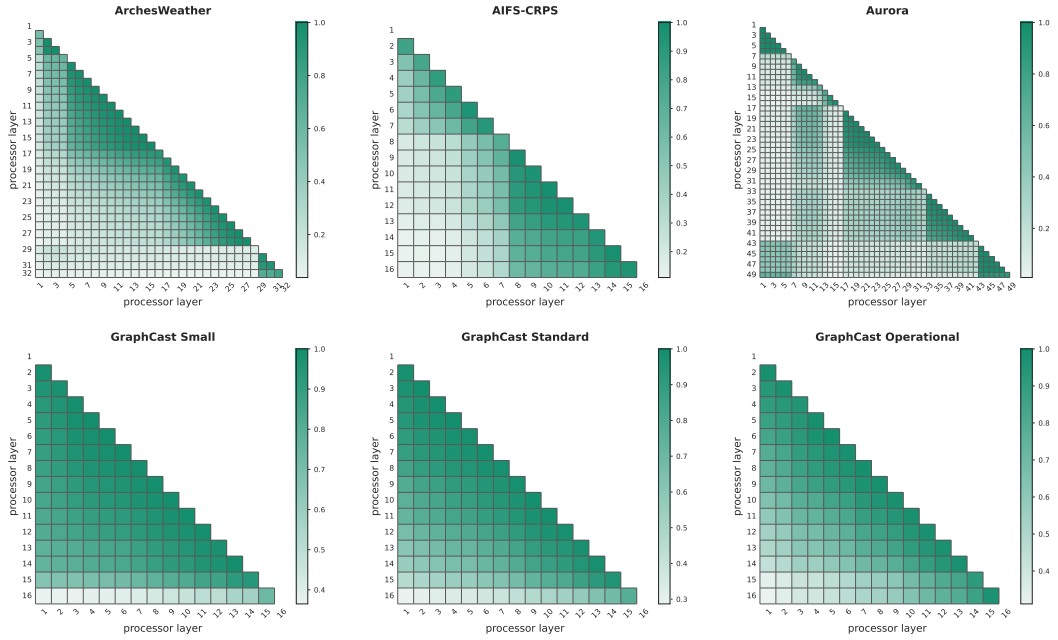


Figure 6: **Similarity across processor layers for a single timestep.** Within-model null-calibrated CKA between all pairs of processor-layer representations at 2020-06-01 18 UTC, calculated using all spatial nodes. Processor layers are ordered through each model, and colour bars are scaled separately from the minimum calibrated CKA for each model.

Having found measurable similarity between the final representations of the different models, we next investigate how the representations change within each model across its processor. A direct correspondence between processor steps of different models is difficult to define because the architectures contain different numbers and organisations of internal layers. Figure 6 shows the resulting null-calibrated CKA for the forecast time 2020-06-01 18 UTC using all mesh nodes. The raw scores are available in Appendix A.8, and all scores achieve the minimum p-value, except for 83 aurora values, which reach a maximum of just above 0.0225.

For AIFS-CRPS and the three GraphCast models, similarity is greatest for processor steps closest to each other, and generally decreases as the distance between steps increases, with the changes being relatively gradual. This is indicative of a progressive evolution of the latent representation through these processors. The three GraphCast variants show broadly similar behaviour, and retain comparatively high similarity throughout the processor, with calibrated CKA remaining above approximately 0.45.

ArchesWeather and Aurora evolve less gradually. In both, large similarity changes occur at boundaries between processor-layer groups, indicating strong changes in representational geometry between architectural stages, with further-away processor layers not necessarily being less similar than

those closer. In addition, Aurora approaches zero calibrated CKA for some representations, unlike the much higher GraphCast minima. This structure is notably similar to the covariance-similarity results of Beylich et al. (2026) for the same models except AIFS-CRPS.

After aligning each intermediate representation to the final-layer latent basis with a translator, similarity increases substantially, particularly for GraphCast and Aurora (Appendix A.2). However, broad model-dependent structure remains, meaning that latent-basis changes explain part, but not all, of the architecture-dependent evolution.

Overall, representations evolve smoothly through individual processors stages, with architectures comprising multiple stages exhibiting notable differences between them. Repeating the analysis at 2020-01-24 06 UTC gives the same architecture-dependent patterns (Appendix A.9).

6 Discussion

Significant similarity has been shown to exist between the final internal representations of different AI weather models in a variety of experiments, as well as within individual models processors. Across models, the GraphCast variants are consistently the most similar, as expected given that they belong to the same model family. Relationships between the different model families are less consistent, however, this global similarity does still indicate that all AI models tested do share similar structure globally. Computing similarity for separate regions (midlatitude, equatorial) shows that the similarity between a given pair of models depends on the region and is often stronger than the similarity computed globally.

Examining the representations after different processor layers in each model individually shows substantial similarity, but also features that are directly connected to the model architecture. The single 16-layer processor stage in GraphCast shows a smooth drift in representation, while abrupt changes are observed at the up/down-sampling stages in Aurora and ArchesWeather.

It has been additionally highlighted that great care is required when interpreting representational similarity, particularly for atmospheric data. In particular, it has been seen that the raw similarity contains a substantial contribution from the structure of the atmospheric data alone, and as such, can overstate representational correspondence if null calibration is not used to reduce much of this background similarity. This is model dependent however, as ArchesWeather retains much more similarity than the other models, suggesting an unknown, particularly persistent component of its latent representation. Differences between similarity metrics were also observed, demonstrating the importance of testing the metrics before any comparison of internal representations.

7 Limitations & future work

Although we study prominent AI weather models, the set is limited and new models have been released during this work. Furthermore, despite single-forecast time results having been repeated at a second date, the analysis still covers only a limited number of dates from 2020. Future work should test the robustness of these results across a broader range of dates and years. ArchesWeather's unusual calibration behaviour also motivates investigation of alternative structure-preserving null models. Finally, the presence of significant similarity between different AI weather models and at different processor layers in each model raises the question of whether the computations using the representations are also similar, given that they lead to similarly good weather forecasts. This is not addressed by the metrics shown in this work, but is an exciting topic for further research.

AI use statement

In this work, we used generative AI tools for code generation and revising the manuscript. In addition, we used AI tools to provide feedback on our research methodology and experiments. The research ideas and scientific questions were developed without generative AI assistance. Furthermore, no AI was used for supporting qualitative and thematic data analysis or interpreting results. All AI-assisted text and code has been reviewed line-by-line, and edited by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Reproducibility statement

Code used for the similarity calculations and experiments in Sections 3.2, 4 and 5 will be made available. The verification experiment from Section 3.1 uses the same similarity implementation, with the additional perturbation experiment described in Appendix A.3.8. Full experimental details are provided in Section 2 and the appendix: model architectures and latent representations are described in Appendix A.1, latent-basis alignment in Appendix A.2, similarity metric definitions and implementation details in Appendix A.3, and the null calibration procedure in Appendix A.4.

Latent representations can be extracted from the publicly available model implementations, either using forward hooks for PyTorch models or by modifying the relevant processor code to return intermediate activations. The GraphCast, Aurora, and ArchesWeather implementations are available through their public GitHub repositories (`google-deepmind/graphcast`, `microsoft/aurora`, and `gcouairon/ArchesWeather`, respectively). The AIFS-CRPS checkpoint (`ecmwf/aifs-ens-1.0`) is available from ECMWF on Hugging Face.

Acknowledgments

The authors thank the computing resources and support of the Physics department, as well as fruitful discussions with colleagues for advancing this topic.

References

Matthias Beylich, Kirsten I. Tempest, Tobias Selz, and George C. Craig. Interpretability of ai weather models via intermediate decoding. 2026. To be submitted.

Cristian Bodnar, Wessel P. Bruinsma, Ana Lucic, Megan Stanley, Anna Allen, Johannes Brandstetter, Patrick Garvan, Maik Riechert, Jonathan A. Weyn, Haiyu Dong, Jayesh K. Gupta, Kit Thambiratnam, Alexander T. Archibald, Chun-Chieh Wu, Elizabeth Heider, Max Welling, Richard E. Turner, and Paris Perdikaris. A foundation model for the earth system. *Nature*, 641 (8065):1180–1187, May 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09005-y. URL https://doi.org/10.1038/s41586-025-09005-y.

Nathan Cloos, Moufan Li, Markus Siegel, Scott L. Brincat, Earl K. Miller, Guangyu Robert Yang, and Christopher J. Cueva. Differentiable optimization of similarity scores between models and brains, 2024. URL https://arxiv.org/abs/2407.07059.

Nathan Cloos, Guangyu Robert Yang, and Christopher J. Cueva. A framework for standardizing similarity measures in a rapidly evolving field, 2025. URL https://arxiv.org/abs/2409.18333.

Guillaume Couairon, Christian Lessig, Anastase Charantonis, and Claire Monteleoni. Archesweather: An efficient ai weather forecasting model at 1.5deg $resolution, 2024a. URL.$

Guillaume Couairon, Christian Lessig, Anastase Charantonis, and Claire Monteleoni. ArchesWeather: An efficient AI weather forecasting model at 1.5{\deg} resolution, July 2024b. URL http://arxiv.org/abs/2405.14527. arXiv:2405.14527 [cs] version: 2 XXXXXXXXX.

George Craig, Tobias Selz, Matthias Beylich, and Kirsten I Tempest. The physics of ai weather models. *arXiv preprint arXiv:2605.23778*, 2026.

Frances Ding, Jean-Stanislas Denain, and Jacob Steinhardt. Grounding representation similarity with statistical testing, 2021. URL https://arxiv.org/abs/2108.01661.

Peter Dueben, Peter Bauer, Oliver Fuhrer, Nikolay Koldunov, and Jørn Kristiansen. Machine learning is revolutionizing weather forecasting – the next step is a change in how we work. *arXiv preprint arXiv:2606.25076*, 2026. 10.48550/arXiv.2606.25076.

Fabian Gröger, Shuo Wen, and Maria Brbić. Revisiting the platonic representation hypothesis: An aristotelian view, 2026. URL https://arxiv.org/abs/2602.14486.

Max Klabunde, Mehdi Ben Amor, Michael Granitzer, and Florian Lemmerich. Towards measuring representational similarity of large language models, 2023. URL https://arxiv.org/abs/2312.02730.

Max Klabunde, Tobias Schumacher, Markus Strohmaier, and Florian Lemmerich. Similarity of neural network models: A survey of functional and representational measures. *ACM Computing Surveys*, 57(9):1–52, May 2025a. ISSN 1557-7341. 10.1145/3728458. URL http://dx.doi.org/10.1145/3728458.

Max Klabunde, Tassilo Wald, Tobias Schumacher, Klaus Maier-Hein, Markus Strohmaier, and Florian Lemmerich. Resi: A comprehensive benchmark for representational similarity measures, 2025b. URL https://arxiv.org/abs/2408.00531.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited, 2019. URL https://arxiv.org/abs/1905.00414.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter Battaglia. Graphcast: Learning skillful medium-range global weather forecasting, 2023. URL https://arxiv.org/abs/2212.12794.

Simon Lang, Mihai Alexe, Mariana C. A. Clare, Christopher Roberts, Rilwan Adewoyin, Zied Ben Bouallègue, Matthew Chantry, Jesper Dramsch, Peter D. Dueben, Sara Hahner, Pedro Maciel, Ana Prieto-Nemesio, Cathal O'Brien, Florian Pinault, Jan Polster, Baudouin Raoult, Steffen Tietsche, and Martin Leutbecher. Aifs-crps: ensemble forecasting using a model trained with a loss function based on the continuous ranked probability score. *npj Artificial Intelligence*, 2(1):18, Feb 2026. ISSN 3005-1460. 10.1038/s44387-026-00073-7. URL https://doi.org/10.1038/s44387-026-00073-7.

Alex Murphy, Joel Zylberberg, and Alona Fyshe. Correcting biased centered kernel alignment measures in biological and artificial neural networks, 2024. URL https://arxiv.org/abs/2405.01012.

Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning?, 2021. URL https://arxiv.org/abs/2008.11687.

Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth, 2021. URL https://arxiv.org/abs/2010.15327.

Kirsten I. Tempest, Matthias Beylich, and George C. Craig. Mechanistic interpretability tool for ai weather models. In Maciej Paszynski, Amanda S. Barnard, and Yongjie Jessica Zhang (eds.), *Computational Science – ICCS 2026 Workshops*, pp. 115–127, Cham, 2026. Springer Nature Switzerland. ISBN 978-3-032-29915-4.

Ruyi Yang, Jingyu Hu, Zihao Li, Jianli Mu, Tingzhao Yu, Jiangjiang Xia, Xuhong Li, Aritra Dasgupta, and Haoyi Xiong. Interpretable machine learning for weather and climate prediction: A review. *Atmospheric Environment*, 338:120797, 2024. ISSN 1352-2310. https://doi.org/10.1016/j.atmosenv.2024.120797. URL https://www.sciencedirect.com/science/article/pii/S1352231024004722.

A Appendix

A.1 AI weather models

Each of the AI weather models used in this study can broadly be described as using an encoder-processor-decoder architecture, although the form of the latent representation and the operations performed within the processor differ between models. The encoder maps one or more recent atmospheric states from the input weather grid into a structured latent representation. The processor then updates this representation through a sequence of neural network layers, before the decoder

Table 1: Overview of AI weather model architectures and latent representations used in this study. Layers denotes the number of internal layers inside the processor. In the case of the processor containing several stages, these stage boundaries are ignored and the individual processor blocks are counted. Comparison dimension refers to the dimension of the final processor layer latent, as this can vary in the case of up- and downsampling, as is used in Aurora and ArchesWeather.

Model	Version(s)	Processor	Layers	Per-level dim.	Latent levels	Comparison dim.
GraphCast / WeatherNext 1 Graph	Small, Standard & Operational	Message-passing GNN	16	512	–	512
Aurora	aurora-0.25-finetuned	3D Swin Transformer U-Net	48	1024	4	4096
ArchesWeather	Archesweather-M	3D Swin Transformer U-Net	32	384	8	3072
AIFS-CRPS	aifs-ens-1.0	Sliding-window Transformer	16	1024	–	1024

maps the final representation back to meteorological variables on the output grid. For GraphCast, the latent representation is defined on a graph and is updated through message passing, whereas the other models primarily use transformer-based processing on structured latent grids.

Table 1 provides an overview of the architectures and the latent representations used in this study. Here, processor layers refers to the internal layers included in our processor-step comparison, while per-level latent dimension is the length of the feature vector at one spatial location and one latent level. For Aurora and ArchesWeather, which contain multiple latent levels, these levels are concatenated before the similarity calculations. The feature dimension used in those calculations is therefore the per-level latent dimension multiplied by the number of latent levels.

A.1.1 GRAPHCAST

GraphCast, renamed to WeatherNext 1 Graph, (Lam et al., 2023), is a graph neural network (GNN) with an encoder-processor-decoder architecture. The encoder maps meteorological variables from a regular latitude-longitude grid onto an internal multimesh representation. The multimesh is constructed from a hierarchy of increasingly refined icosahedral meshes, allowing both local and longer-range spatial interactions to be represented within the graph.

The processor consists of 16 message-passing GNN layers with a latent feature dimension of 512. At each processor layer, edge features are first updated using the features of the adjacent mesh nodes. Information from the incoming edges is then aggregated at each node and used to update the node representation, with residual connections between processor layers. The final processor representation therefore contains one 512-dimensional latent feature vector at each mesh node.

We use all three publicly available pre-trained variants. GraphCast Standard is the high-resolution model used in Lam et al. (2023), operating at 0.25° resolution with 37 atmospheric pressure levels and trained on ERA5 from 1979–2017. GraphCast Small operates at 1° resolution with 13 pressure levels and uses a smaller mesh; it was trained on ERA5 from 1979–2015. GraphCast Operational operates at 0.25° resolution with 13 pressure levels and was pre-trained on ERA5 from 1979–2017 before being fine-tuned on HRES data from 2016–2021.

The GraphCast-small mesh contains $10{,}242$ nodes and is used as the common spatial basis for all inter-model comparisons in this study. The Standard and Operational meshes contain the GraphCast-small locations as a subset, allowing their representations to be selected directly at the corresponding nodes without interpolation.

A.1.2 AURORA

Aurora is a transformer-based model with a 3D Swin U-Net encoder-processor-decoder architecture (Bodnar et al., 2025). The model employs an attention-based 3D Perceiver encoder and decoder. The encoder tokenises and compresses the input from a regular latitude-longitude grid into a 3D latent representation of patches, with two dimensions corresponding to latitude and longitude, and the third corresponding to vertical levels.

The processor consists of 48 processor blocks, each applying a Swin-transformer self-attention operation followed by an MLP, organised into three encoding and three decoding stages separated by up- and downsampling operations. The stages contain 6, 10, 8, 8, 10, and 6 processor blocks respectively. Between stages, the up- and downsampling operations are implemented as linear projections that merge or split patches: downsampling blocks halve the horizontal resolution and double the channel dimension, while upsampling blocks do the reverse. Latent states are of shape (V, H, W, C), where V denotes the vertical dimension, H, W the horizontal resolution, and C the channel dimension. The resulting shapes (V, H, W, C) are $(4, 180, 360, 512)$, $(4, 90, 180, 1024)$, and $(4, 45, 90, 2048)$ for encoding stages 1–3 (and decoding stages 3–1, respectively; see Figure 2 in (Beylich et al., 2026)). The output of encoding stage 1 is concatenated along the channel dimension to the output of decoding stage 3 via a skip connection, resulting in a representation with $(V, H, W, C) = (4, 180, 360, 1024)$, which is passed to the decoder.

We use the fine-tuned version of Aurora, which operates on a 0.25° input and output grid with 13 pressure levels. It was pre-trained on ERA5 from 1979–2020, and subsequently fine-tuned on IFS HRES-0.25 T0 data from 2016–2021.

A.1.3 ARCHESWEATHER

ArchesWeather is a transformer-based model with a 3D Swin U-Net encoder-processor-decoder architecture (Couairon et al., 2024b). The encoder and decoder are implemented as convolutional layers. The encoder maps meteorological variables from a regular latitude-longitude grid onto an internal 3D latent representation of patches, with two dimensions corresponding to latitude and longitude, with the third corresponding to vertical levels.

The processor consists of 32 blocks organised into four layers, with layers 1 and 4 containing 4 processor blocks each and layers 2 and 3 containing 12 processor blocks each. Each block applies a Swin-transformer self-attention operation followed by an MLP. After layer 1, the representation is downsampled, and it is upsampled again after layer 3, with down- and upsampling implemented as described above for Aurora. ArchesWeather incorporates a single skip connection, which concatenates the output of layer 1 with the input of layer 4 along the channel dimension. Latent states are of shape (V, H, W, C), where V denotes the vertical dimension, H, W the horizontal resolution, and C the channel dimension. The resulting shapes (V, H, W, C) are $(8, 60, 120, 192)$ for layer 1, $(8, 30, 60, 384)$ for layers 2 and 3, and $(8, 60, 120, 384)$ for layer 4.

Three model sizes are available, differing in depth: small (16 layers, $\sim$49M parameters), medium (32 layers, $\sim$89M parameters), and large (64 layers, $\sim$164M parameters). All operate at 1.5° horizontal resolution with 13 pressure levels. Throughout this study we use the medium version (ArchesWeather-M). All model versions are trained on ERA5 from 1979–2018 (2019 for validation, 2020 for evaluation).

A.1.4 AIFS-CRPS

AIFS-CRPS (Lang et al., 2026) is a probabilistic machine-learned ensemble forecast model with an encoder-processor-decoder architecture. The encoder and decoder are transformer-based GNNs, while the processor is a sliding-window transformer. The processor consists of 16 layers with an embedding dimension of 1024 and 8 attention heads, operating on an O96 octahedral reduced Gaussian processor grid for the N320 model configuration.

AIFS-CRPS is trained probabilistically using an almost fair Continuous Ranked Probability Score (afCRPS) loss function. Model uncertainty is introduced stochastically through an independent Gaussian noise tensor, which is embedded using a two-layer multilayer perceptron (MLP), followed by layer normalisation. Although the model can generate multiple exchangeable ensemble members

from the same atmospheric state, we analyse the internal activations from a single member generated using the default inference procedure.

A.2 Effect of latent-basis alignment on processor-layer similarity

A.2.1 Translator method

Comparisons between processor layers may reflect both changes in the information represented by the model and changes in the latent basis in which that information is expressed. Beylich et al. (2026) showed that the latent basis of AI weather models can change substantially with processor depth. To compensate for this, they introduced learned affine translators to align intermediate latent states with the representation space of the last processor stage. In addition, due to CKA being invariant to orthogonal transformations but not to general affine changes of basis, CKA will not account for this change of basis and it is therefore of interest to measure the similarity again, with translated versions of the latents.

For an intermediate latent representation $\tilde{X}_i$, the translated representation is

$$\tilde{X}_i^F = W_i \tilde{X}_i + b_i, \tag{2}$$

where W_i and b_i are the learned affine transformation, mapping processor layer i into the representation space of the final processor layer. The transformation is independent of spatial location, such that the same weights are applied at every spatial point. See Beylich et al. (2026) for further details as to how the translator is applied, and how adjustments are made when up- and down-sampling occurs.

To account for this representational drift, we repeat the similarity analysis after applying these translators, translating each intermediate representation onto the final processor-layer representation space before CKA is calculated. The same structure-preserving null calibration described in Section 2.2.1 is then applied to the translated representations, such that both the observed CKA and the corresponding null distribution are calculated after translation. Results are shown for ArchesWeather, Aurora and the three GraphCast variants. AIFS-CRPS is omitted due to no translator currently available.

A.2.2 Translator results

Translation substantially increases the similarity between processor-layer representations for all models considered (Figure 7 shows the calibrated scores and Figure 8 the raw scores). The p-value takes its minimum possible value in all cases. The effect of translation is notably similar to the covariance-similarity results of Beylich et al. (2026). In particular, similarity increases substantially for GraphCast and Aurora, and more moderately for ArchesWeather, while similar block structure remains visible, for example in Aurora. Some differences remain however, particularly for calibrated ArchesWeather. Overall, this corroborates the conclusion of Beylich et al. (2026), that a considerable fraction of the apparent change through processor depth is associated with changes in latent basis. Nevertheless, the translated representations are not identical, and model-dependent structure remains.

A.3 Similarity metric definitions, invariances and implementation

Let $\mathbf{R} \in \mathbb{R}^{N \times d}$ and $\mathbf{R}' \in \mathbb{R}^{N \times d'}$ denote two representations evaluated over the same N samples. We use five representational similarity measures with their invariance properties, following Klabunde et al. (2025a), shown in Table 2. These invariances determine which transformations of a representation leave the corresponding similarity unchanged, and therefore form part of the definition of similarity imposed by each metric.

The implementations used for each of the metrics, apart from CKA, are from the Git repository `nacloos/similarity-repository`, and listed in Table 3, whereby resi relates to the implementations developed for Klabunde et al. (2025b).

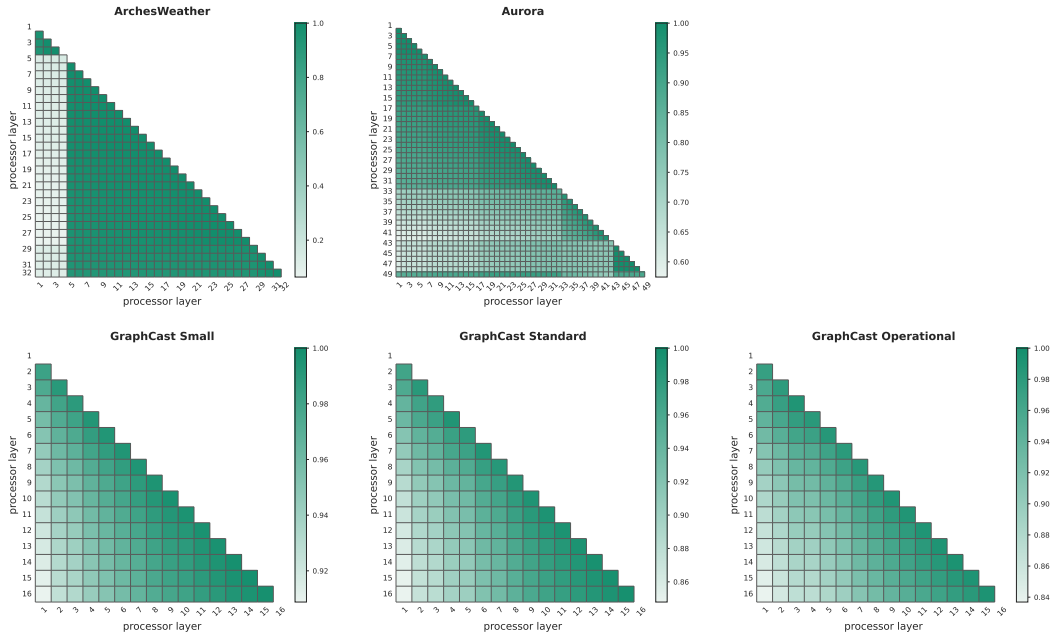


Figure 7: **Calibrated CKA similarity across translated processor-layer representations for a single timestep.** Within-model similarity of translated latent representations between all pairs of processor layers for ArchesWeather, Aurora, GraphCast Small, GraphCast Standard and GraphCast Operational, calculated using all nodes at the forecast time of 2020-06-01 18 UTC. Colour bars are scaled separately for each model from the minimum calibrated CKA value in that panel to 1.

Table 2: **Transformation invariances of the similarity metrics used in this study.** A check mark indicates that the similarity measure is invariant to the corresponding transformation.

Metric	Permutation	Orthogonal	Isotropic scaling	Translation
Second-order cosine	✓	✓	✓	
Aligned orthogonal Procrustes	✓	✓	✓	✓
RSA	✓		✓	✓
Distance correlation	✓	✓		✓
CKA	✓	✓	✓	✓

Several of the measures use a representational similarity matrix (RSM), $\mathbf{S} \in \mathbb{R}^{N \times N}$, which describes the pairwise similarity between samples within a representation:

$$S_{ij} = s(R_i, R_j), \tag{3}$$

where s is the similarity function associated with the metric. In the following, the similarity measures used in this paper are described.

A.3.1 Second-order cosine similarity

This neighbourhood-based measure compares the local neighbourhood geometry around each sample, i, across the two representations.

For each sample i, cosine similarity is first used to identify its k nearest neighbours in each representation. The union of these neighbourhoods is formed as $\{j_1, \ldots, j_{K(i)}\} := \mathcal{N}^k_{\mathbf{R}}(i) \cup \mathcal{N}^k_{\mathbf{R}'}(i)$. The cosine similarity between the two resulting similarity vectors for each representation is then calculated for every sample:

$$\left(v^k_{\text{2nd-Cos}}\left(\mathbf{R}, \mathbf{R}'\right)\right)_i := \text{cos-sim}\left(\left(S_{i,j_1}, \ldots, S_{i,j_{K(i)}}\right), \left(S'_{i,j_1}, \ldots, S'_{i,j_{K(i)}}\right)\right).$$

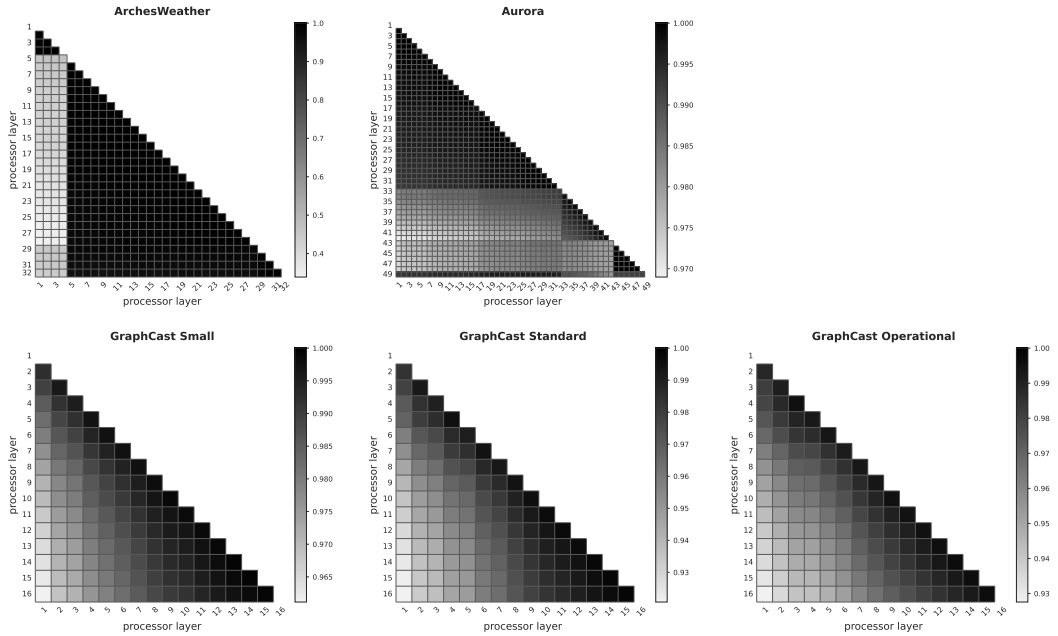


Figure 8: **Raw CKA similarity across translated processor-layer representations for a single timestep.** Within-model similarity of translated latent representations between all pairs of processor layers for ArchesWeather, Aurora, GraphCast Small, GraphCast Standard and GraphCast Operational, calculated using all nodes at the forecast time of 2020-06-01 18 UTC. Colour bars are scaled separately for each model from the minimum CKA value in that panel to 1.

Table 3: Metrics implementation

Metric	similarity-repository label
Second order cosine similarity	measure/resi/second_order_cosine_similarity
Orthogonal Procrustes	measure/resi/orthogonal_procrustes_centered_and_normalized
RSA	measure/resi/rsa-rdm=correlation-compare=spearman
Distance correlation	measure/resi/distance_correlation

These are then averaged over all instances:

$$m_{\text{2nd-Cos}}(\mathbf{R}, \mathbf{R}') = \frac{1}{N} \sum_{i=1}^{N} v_{\text{2nd-Cos}}(\mathbf{R}, \mathbf{R}')_i.$$

We use $k = 10$.

A.3.2 Aligned orthogonal Procrustes similarity

This alignment-based metric returns the distance between representations, once the best possible alignment has been found between them using orthogonal transformations. It is defined by:

$$m_{\text{OrthProc}}(\mathbf{R}, \mathbf{R}') = \min_{\mathbf{Q} \in O(D)} \|\mathbf{R}\mathbf{Q} - \mathbf{R}'\|_F = \left(\|\mathbf{R}\|_F^2 + \|\mathbf{R}'\|_F^2 - 2\|\mathbf{R}^\mathsf{T}\mathbf{R}'\|_*\right)^{\frac{1}{2}},$$

where $\mathbf{Q}$ is the matrix that provides the best orthogonal transformation. As a preprocessing step, we centre the columns of each representation and normalise it to unit Frobenius norm. Furthermore, an aligned version is used in the paper, which rescales the distance to be between 0 and 1, where 0 indicates no similarity and 1 perfect similarity:

$$\text{aligned score} = 1 - \frac{\text{score}}{\sqrt{2}}.$$

A.3.3 REPRESENTATIONAL SIMILARITY ANALYSIS (RSA)

This RSM-based measure correlates the pairwise similarity between inputs of two representations. An inner similarity function, s_{in}, is first used for calculating each RSM. The lower triangles of $\mathbf{S}$ and $\mathbf{S}'$ are then vectorised and compared with an outer similarity function, s_{out}:

$$m_{\text{RSA}}(\mathbf{R}, \mathbf{R}') = s_{\text{out}}(\mathbf{v}(\mathbf{S}), \mathbf{v}(\mathbf{S'}))\,.$$

In this paper, s_{in} is cosine similarity, and s_{out} is the Spearman rank-order correlation coefficient.

It returns a statistic with range between -1 and 1, whereby 1 is perfect correlation, 0 indicates totally independent representations, and -1 a perfect negative monotonic relationship. The null-calibration applied rescales this to be between 0 and 1.

A.3.4 DISTANCE CORRELATION

Another RSM-based measure, the distance correlation compares how pairwise distances between inputs vary between representations. Here, Euclidean distance is used to compute $\mathbf{S}$ and $\mathbf{S'}$, which are then double centred. The squared sample distance covariance is then calculated as:

$$\mathrm{dCov}^2(\mathbf{S}, \mathbf{S}') = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} S_{i,j}\, S'_{i,j}.$$

This is then normalised to obtain the distance correlation:

$$m_{\text{DistCorr}}(\mathbf{R}, \mathbf{R}') = \frac{\sqrt{\mathrm{dCov}^2(\mathbf{S}, \mathbf{S}')}}{\sqrt{\mathrm{dCov}^2(\mathbf{S}, \mathbf{S})\ \mathrm{dCov}^2(\mathbf{S}', \mathbf{S}')}}.$$

.

A.3.5 CENTRED KERNEL ALIGNMENT

The final RSM-based metric employed in this paper, CKA, compares the centred Gram matrices of the two representations. Kernel functions are first used for calculating the RSM of each representation, which are then compared using the Hilbert-Schmidt Independence Criterion (HSIC), and normalised. Due to little advantage being seen from using a non-linear kernel(Kornblith et al., 2019), a linear kernel is applied in this study. The score is as follows:

$$m_{\text{CKA}}(\mathbf{R}, \mathbf{R}') = \frac{\mathrm{HSIC}(\mathbf{S}, \mathbf{S}')}{\sqrt{\mathrm{HSIC}(\mathbf{S}, \mathbf{S})\,\mathrm{HSIC}(\mathbf{S}', \mathbf{S}')}},$$

whereby the biased HSIC can be empirically estimated (Kornblith et al., 2019) by

$$\mathrm{HSIC}(\mathbf{S}, \mathbf{S'}) = \frac{1}{(N-1)^2}\,\mathrm{tr}(\mathbf{SHS'H}),$$

where $\mathbf{H}$ is the centering matrix, $H_N = I_N - \frac{1}{N}\mathbf{1}\mathbf{1}^{\mathrm{T}}$. Note that in line with recommendations of Klabunde et al. (2025b), the representations $\mathbf{R}$ are in addition preprocessed by centering their columns beforehand.

An unbiased HSIC estimator also exists (Murphy et al., 2024). We compare the resulting unbiased CKA with the biased estimator in Appendix A.3.7, finding negligible differences for the sample sizes considered here.

A.3.6 CKA IMPLEMENTATION

CKA was calculated using our own implementation to allow for a *fast* and *minibatch* version, allowing for representations with more instances. We based this on the work of Kornblith et al. (2019) and Nguyen et al. (2021), respectively.

The *fast* version was used for all CKA scores, whereby

$$\frac{1}{(n-1)^2}\,\mathrm{tr}\left(\mathbf{R}\mathbf{R}^{\mathrm{T}}\mathbf{R}'\mathbf{R}'^{\mathrm{T}}\right) = \left\|\mathrm{cov}\left(\mathbf{R}^{\mathrm{T}}, \mathbf{R}'^{\mathrm{T}}\right)\right\|_{\mathrm{F}}^2$$

can be used to replace the trace operation of the HSIC. This works assuming a linear kernel.

For comparisons containing large numbers of samples, particularly the monthly calculations, we use a *minibatch* estimator following Nguyen et al. (2021). We take one timestep as a minibatch, containing all spatial samples used in the comparison at that timestep. The HSIC terms are then averaged across the b minibatches before normalisation:

$$\mathrm{CKA}_{\mathrm{minibatch}} = \frac{\frac{1}{b}\sum_{i=1}^{b}\mathrm{HSIC}(\mathbf{S}, \mathbf{S'})}{\sqrt{\frac{1}{b}\sum_{i=1}^{b}\mathrm{HSIC}(\mathbf{S}, \mathbf{S})}\sqrt{\frac{1}{b}\sum_{i=1}^{b}\mathrm{HSIC}(\mathbf{S'}, \mathbf{S'})}}.$$

A.3.7 UNBIASED CKA VERSUS BIASED CKA

To assess whether the use of unbiased CKA provides a meaningful advantage in our setting, we repeat the all-node comparison from Figure 4 using unbiased CKA with a linear kernel. Figure 9 shows that the resulting similarity scores are effectively unchanged from those obtained using biased CKA. Given the large number of samples relative to the latent dimensionality in these experiments, the difference between the two estimators is therefore negligible for our purposes. We consequently use biased CKA throughout our analysis. The broader sensitivity of the similarity measures to sample size is examined in Appendix A.10.



Figure 9: **Pairwise similarity calculated using unbiased CKA.** Comparison to biased CKA scores for all nodes in Figure 4. Calibrated scores are shown in the lower triangle and the corresponding raw score in the upper triangle.

A.3.8 SIMILARITY-METRIC VERIFICATION

As suggested by Ding et al. (2021), we test whether the results from the similarity measures are also functionally meaningful. For each AI weather model, we perturb the final internal representa-

tion used to produce the forecast by adding Gaussian noise. For a representation $\mathbf{R}$, the perturbed representation is

$$\widetilde{\mathbf{R}} = \mathbf{R} + \alpha\sigma_{\mathbf{c}}\epsilon, \tag{4}$$

where ϵ is independent Gaussian noise generated for every channel and spatial node, $\sigma_{\mathbf{c}}$ is the standard deviation of latent channel c calculated over the sample dimensions, and α controls the perturbation amplitude. Scaling the perturbation by $\sigma_{\mathbf{c}}$ accounts for differences in the natural magnitude of the latent channels.

We use 15 values of α between 0 and 4, with $\alpha = 0$ corresponding to the unperturbed representation. The experiment is repeated at 24 forecast times throughout 2020, using 06 UTC on the 14th, and 18 UTC on the 29th of each month. The perturbed representation is passed through the decoder to produce a perturbed forecast $\widehat{Y}_\alpha$.

We quantify the resulting change in model behaviour using the mean-squared difference from the unperturbed forecast, $\widehat{Y}_0$:

$$\mathrm{MSE}_\alpha = \mathrm{MSE}\left(\widehat{Y}_\alpha, \widehat{Y}_0\right). \tag{5}$$

The spatial MSE is calculated separately for each forecast variable and, where applicable, each pressure level, before taking the unweighted mean across these components.

For each perturbation amplitude, we additionally calculate the similarity between the original and perturbed representations, $s(\mathbf{R}, \widetilde{\mathbf{R}})$. A similarity measure that captures functionally meaningful differences should assign progressively lower similarity as the perturbation increasingly changes the decoded forecast.

A.4 Null calibration

We follow Gröger et al. (2026) to null-calibrate the similarity scores, such that any score above zero is indicative of some similarity.

For two representations $\mathbf{R}$ and $\mathbf{R}'$, the observed, raw, similarity is first calculated as

$$s_{\mathrm{obs}} = s(\mathbf{R}, \mathbf{R}'). \tag{6}$$

We then generate K null scores by applying a transformation π_k to one representation while leaving the other fixed:

$$s^{(k)} = s\left(\mathbf{R}, \pi_k(\mathbf{R}')\right), \qquad k = 1, \ldots, K. \tag{7}$$

The transformation is constructed to break the correspondence between samples in $\mathbf{R}$ and $\mathbf{R}'$, while retaining the spatial and temporal structure within each representation. The resulting null distribution therefore estimates the similarity expected in the absence of sample-wise correspondence.

The observed score and null scores form the combined multiset:

$$\left\{s_{\mathrm{obs}}, s^{(1)}, \ldots, s^{(K)}\right\}. \tag{8}$$

Allowing ties, the upper-tail critical value at significance level α is defined as

$$\tau = s_{(\lceil (1-\alpha)(K+1) \rceil)}, \tag{9}$$

where the subscript denotes the corresponding ordered value of the $(K+1)$-element multiset. The threshold τ can therefore be interpreted as an effective zero point for the similarity measure.

The corresponding empirical upper-tail p-value is

$$p = \frac{1 + \#\left\{k \in \{1, \ldots, K\} : s^{(k)} \geq s_{\mathrm{obs}}\right\}}{K+1}. \tag{10}$$

Throughout this study, we use $\alpha = 0.05$ and $K = 200$, giving a minimum attainable empirical p-value of 0.00497512. The sensitivity of the calibrated scores to the number of null realisations is examined in Appendix A.4.1.

Using τ, we define the null-calibrated similarity score as

$$s_{\mathrm{cal}} = \max\left(\frac{s_{\mathrm{obs}} - \tau}{s_{\max} - \tau}, 0\right), \tag{11}$$

where $s_{\max}$ is the maximum possible value of the similarity measure.

A.4.1 SENSITIVITY TO THE NUMBER OF NULL REALISATIONS

To determine how many null realisations are required for sufficiently stable calibration, we repeat the comparison from Figure 4 using different values of K for the case of all nodes being included. Figures 10 and 11 show the resulting calibrated similarity scores and calibration thresholds, respectively, for $K \in \{10, 20, 60, 100, 150, 200\}$. The largest changes generally occur at small K, with both the calibrated scores and calibration thresholds beginning to plateau at approximately $K = 60$.

We use $K = 200$ throughout the main analysis. This lies well within the region where the calibrated scores have largely stabilised, while also giving a minimum attainable empirical p-value of $1/(K + 1) \simeq 0.005$. This provides a balance between the computational cost of generating additional null realisations and obtaining a sufficiently stable estimate of the calibration threshold.

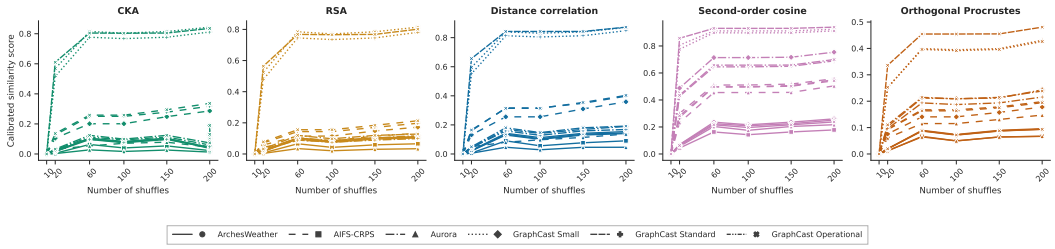


Figure 10: **Calibrated similarity scores as a function of the number of shuffles, K.** The comparison uses all spatial nodes at the same forecast time as Figure 4. Line style denotes the first model in each pair and marker style the second.

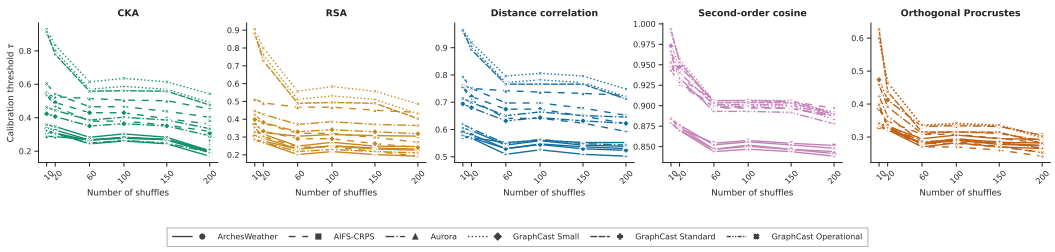


Figure 11: **Calibration threshold τ as a function of the number of shuffles, K.** The comparison uses all spatial nodes at the same forecast time as Figure 4. Line style denotes the first model in each pair and marker style the second.

A.4.2 STRUCTURE-PRESERVING NULL TRANSFORMATIONS

The samples used to calculate the similarity metrics contain strong spatial and temporal dependencies, meaning that positive similarity can arise simply because corresponding samples occupy related spatial or temporal positions. We therefore avoid independently permuting individual samples, since the assumption of exchangeability under the null is not appropriate here. Instead, we construct null representations that break the correspondence between the two representations while preserving the internal spatial and temporal structure of the transformed representation. In each null realisation, **R** is left unchanged and a structure-preserving transformation is applied only to **R'**, with the resulting transformed representation then compared against **R**. We do this in two steps, first considering the spatial structure and then, where applicable, the temporal structure.

To preserve spatial structure within null realisation k, a random longitude offset, $\Delta\lambda_k$, with random sign and magnitude between $2°$ and $180°$ is selected. The complete representation **R'** is rotated by this offset about the longitude axis. Since the GraphCast-small mesh is not a regular latitude-longitude grid, each rotated location is mapped to the nearest node on the original mesh. The same spatial transformation is applied at every timestep of **R'**. This changes which geographical locations correspond between **R** and **R'**, while retaining the spatial coherence within **R'**.

Where the comparison contains a sufficiently long time window (more than five days), an additional circular temporal shift is applied within the individual month time frames. A single temporal offset selected is used for each **R'**, which has a random direction and is between 5 and 29 days, subject to the shift being no greater than half of the available time window.

For comparisons containing only one timestep, the null representation is generated solely by the longitude rotation. Similarly, the 20-timestep (5-day) temporal blocks used in the temporal comparison are rotated spatially but are not shifted in time. Spatial transformations are always constructed using the full GraphCast-small grid before any spatial subset is selected.

A.4.3 Sensitivity to the temporal anchor date

To test whether the calibration behaviour shown in Figure 3 is dependent on the particular anchor period, we repeat the analysis using earlier dates in 2020. The experimental setup is otherwise unchanged. For the five-day comparison, the anchor block spans 2020-04-04 to 2020-04-08 and the single-timestep comparison uses 2020-04-05 06 UTC as the anchor.

Figure 12 shows qualitatively similar behaviour to Figure 3. In particular, the raw CKA retains a substantial model-dependent background, whereas calibration removes much of this background. The individual-timestep comparison again shows enhanced similarity close to the anchor and a reduction with increasing temporal separation. The persistence of comparatively high calibrated similarity for ArchesWeather is also reproduced. These results indicate that the main conclusions of Section 3.2 are not specific to the particular anchor used.

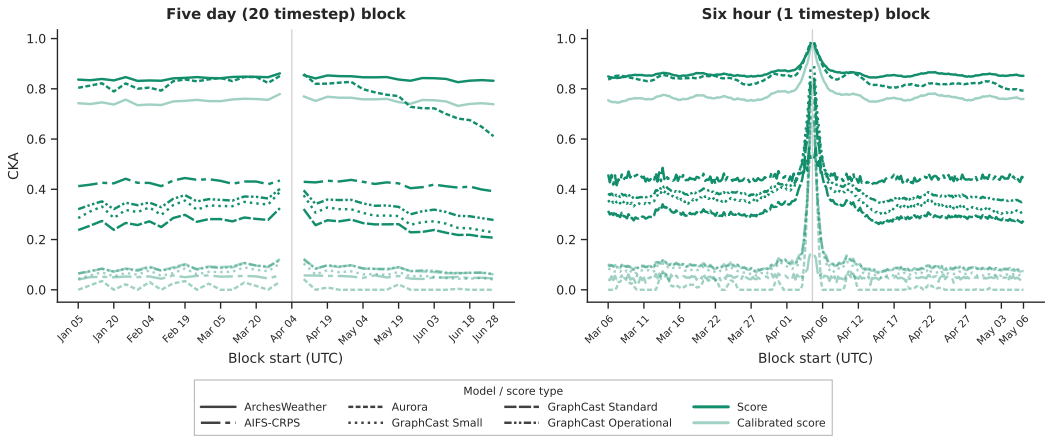


Figure 12: **Temporal similarity of model latent representations using alternative anchor dates.** The temporal comparison from Figure 3 is repeated using an earlier period in 2020. Within each model, a fixed anchor is compared with non-overlapping blocks from the same time series using CKA. The left panel uses five-day blocks with an anchor spanning 4–8 April 2020, while the right panel uses individual six-hourly timesteps with an anchor at 2020-04-05 06 UTC.

A.5 One timestep raw and calibrated similarity scores

Figure 13 shows the raw and null-calibrated similarity scores corresponding to Figure 4. Results are shown for all nodes, the equatorial band, and the midlatitudes. Across these comparisons, the empirical p-value takes its minimum possible value, $\simeq 0.005$, in all but a small number of cases. The exceptions occur in the midlatitudes for comparisons involving ArchesWeather under CKA, RSA and distance correlation, for which $p = 0.01$.

A.6 Single-timestep similarity at a second forecast time

To assess whether the relationships found in the single-timestep comparison in Section 4.1 depend on the particular forecast time considered, we repeat the analysis for a second time, 2020-01-24

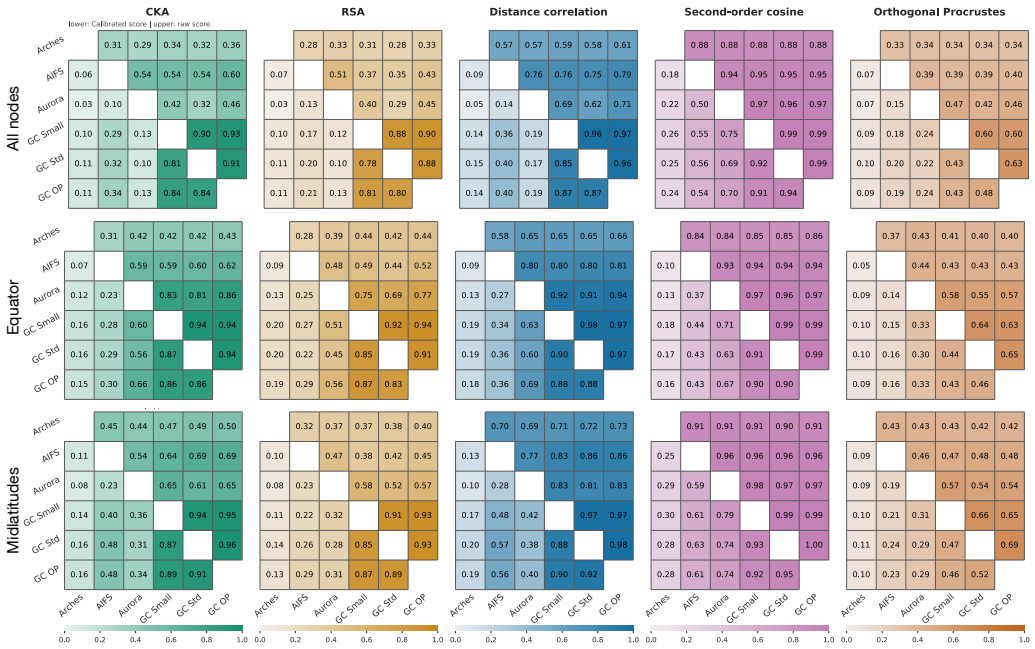


Figure 13: **Pairwise raw and null-calibrated similarity across spatial subsets.** Pairwise representation similarity between ArchesWeather, AIFS-CRPS, Aurora, GraphCast Small (GC Small), GraphCast Standard (GC Std) and GraphCast Operational (GC OP) at 2020-06-01 18 UTC. Rows show all nodes (top), equatorial nodes between $10°\mathrm{S}$ and $10°\mathrm{N}$ (middle), and midlatitude nodes between $40°$ and $60°$ in both hemispheres (bottom). For each matrix, the lower triangle shows the null-calibrated similarity score and the upper triangle the corresponding raw score.

06 UTC. Figure 14 shows the raw and null-calibrated similarity scores for the same five similarity metrics and three spatial subsets used in Figure 4.

The qualitative behaviour is consistent with that found for 2020-06-01 18 UTC. Similarity remains after calibration for all model pairs, metrics and spatial subsets, with the strongest and most consistent similarities occurring within the GraphCast family. The dependence on similarity metric and spatial subset is also retained. In particular, CKA similarities involving AIFS-CRPS and GraphCast are enhanced most clearly in the midlatitudes, while similarities involving Aurora and GraphCast increase strongly for the equatorial subset. The main conclusions of the single-timestep comparison are therefore not specific to the forecast time used in the main-text figure.

A.7 Monthly similarity scores for the Northern and Southern Hemisphere

Figures 15 and 16 show the calibrated and raw monthly CKA scores, respectively, with the midlatitudes separated by hemisphere. Nodes are selected between $40°$ and $60°$ in each hemisphere, and every comparison attains the minimum empirical p-value.

The seasonal variability seen for the combined midlatitudes in Figure 5 is concentrated in the Northern Hemisphere. In particular, several similarities involving Aurora decrease through the middle of the year before recovering towards November, whereas the corresponding Southern Hemisphere relationships remain comparatively stable. This variability is also less apparent in the raw scores, indicating that calibration makes the seasonal changes clearer by removing background similarity.

A.8 Raw similarity across processor steps

Figure 17 shows the raw CKA scores corresponding to the calibrated scores in Figure 6. The raw scores are generally higher, reflecting the background similarity removed by the null calibration, although the same broad architecture-dependent structure remains.

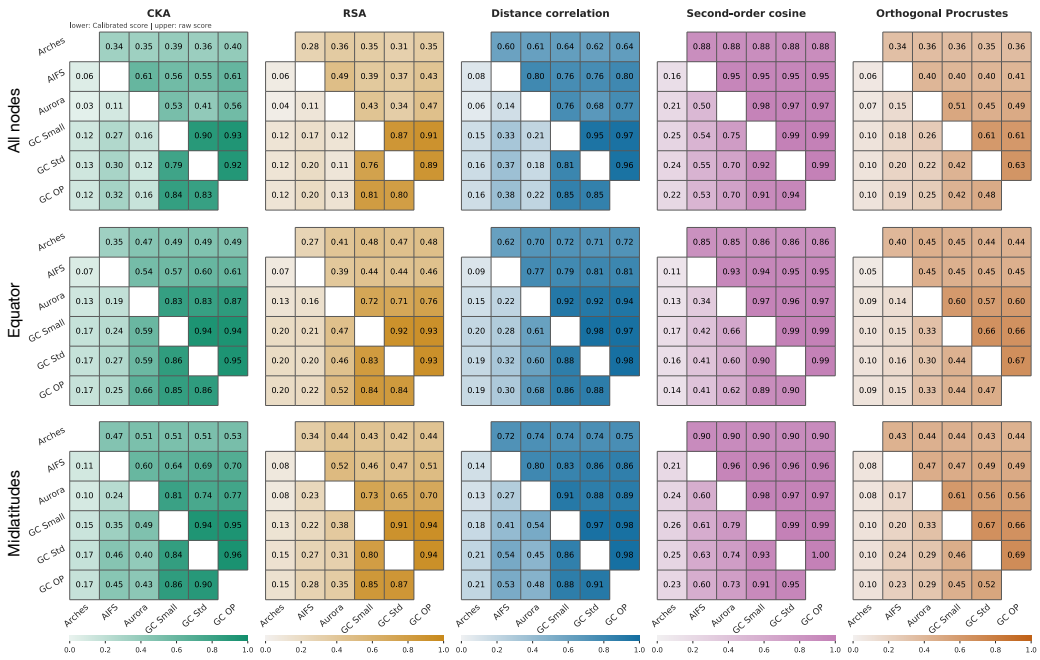


Figure 14: **Pairwise similarity of model latent representations at a second forecast time.** Pairwise representation similarity between ArchesWeather, AIFS-CRPS, Aurora, GraphCast Small (GC Small), GraphCast Standard (GC Std) and GraphCast Operational (GC OP) at the forecast time of 2020-01-24 06 UTC. Rows show all nodes (top), equatorial nodes between $10°$S and $10°$N (middle), and midlatitude nodes between $40°$ and $60°$ in both hemispheres (bottom). For each metric, the lower triangle shows the null-calibrated similarity score and the upper triangle shows the corresponding raw score.

A.9 Processor-layer similarity at a second forecast time

To assess whether the processor-layer relationships described in Section 5 depend on the particular forecast time considered, we repeat the analysis for a second forecast time, 2020-01-24 06 UTC. Figure 18 shows the calibrated CKA scores in the native latent bases, with the corresponding raw scores shown in Figure 19. Figures 20 and 21 show the corresponding calibrated and raw scores after translating the intermediate representations to the final-layer latent basis.

The qualitative behaviour is consistent with that found for 2020-06-01 18 UTC in both the native and translated latent bases. In particular, the broad model-dependent patterns in processor-layer similarity and the effect of latent-basis translation are retained. The main conclusions from the processor-layer analysis are therefore not specific to the forecast time used in the main-text figure.

A.10 Number of samples

To determine whether the number of spatial samples used in the single-timestep experiments is sufficient, and to assess how the similarity scores are affected when fewer samples are available, we repeat the comparison from Figure 4 using progressively smaller subsets of the original $10{,}242$ GraphCast-small mesh nodes. The selected nodes are approximately evenly distributed over the mesh, such that reducing the number of samples does not preferentially retain a particular spatial region. Figure 22 shows the calibrated similarity scores using $K = 200$ null realisations, while Figure 23 shows the corresponding raw scores.

The sensitivity to sample size differs between the similarity measures. For the raw scores, CKA and RSA are comparatively stable once a thousand nodes are included, being already close to their full-grid values. Distance correlation, second-order cosine similarity and orthogonal Procrustes however

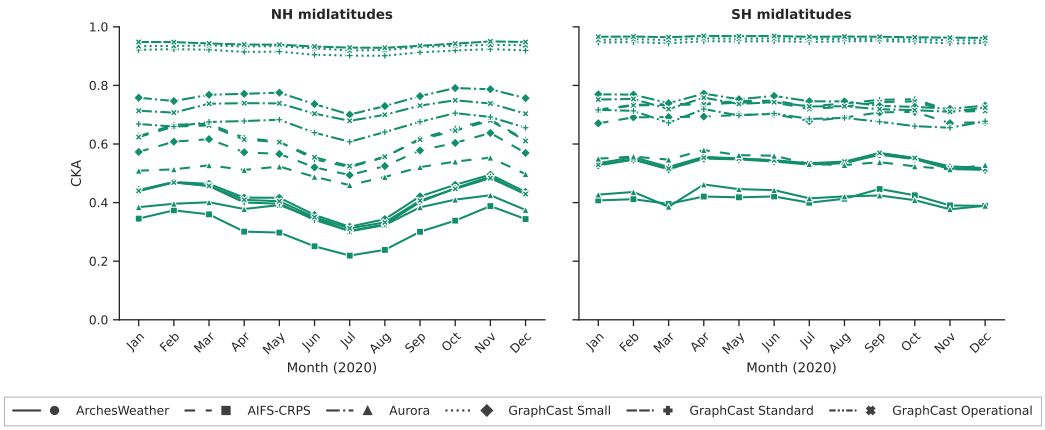


Figure 15: **Monthly pairwise null-calibrated CKA across Northern and Southern Hemisphere midlatitudes.** Pairwise similarity of the latent representations of ArchesWeather, AIFS-CRPS, Aurora, GraphCast Small, GraphCast Standard and GraphCast Operational over 2020, measured using null-calibrated CKA. The two panels show results calculated using Northern Hemisphere midlatitude nodes between $40°\mathrm{N}$ and $60°\mathrm{N}$ (left) and Southern Hemisphere midlatitude nodes between $40°\mathrm{S}$ and $60°\mathrm{S}$ (right). Each curve represents one model pair; line style identifies one model in the pair and marker style identifies the other.

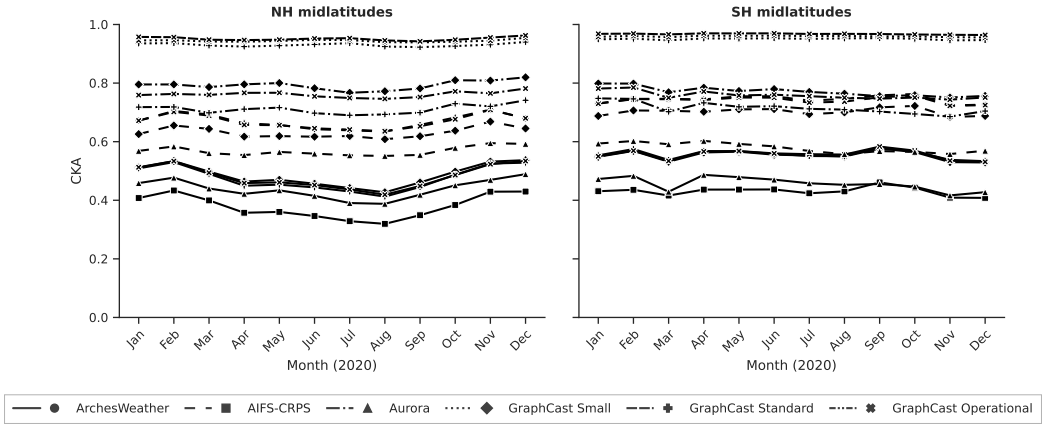


Figure 16: **Monthly pairwise raw CKA across Northern and Southern Hemisphere midlatitudes.** Pairwise similarity of the latent representations of ArchesWeather, AIFS-CRPS, Aurora, GraphCast Small, GraphCast Standard and GraphCast Operational over 2020, measured using raw CKA. The two panels show results calculated using Northern Hemisphere midlatitude nodes between $40°\mathrm{N}$ and $60°\mathrm{N}$ (left) and Southern Hemisphere midlatitude nodes between $40°\mathrm{S}$ and $60°\mathrm{S}$ (right). Each curve represents one model pair; line style identifies one model in the pair and marker style identifies the other.

show a stronger dependence on sample size, with particularly large changes for some comparisons involving ArchesWeather. Null calibration is seen to remove a substantial part of this sample-size dependence for all measures, being most apparent for those measures with greater dependency on sample size in the raw scores. These results indicate that the number of nodes used in this paper are sufficient.

A.11 Latent levels

As described in Section 2.1, Aurora and ArchesWeather contain an additional latent-level dimension that is not present in GraphCast or AIFS-CRPS. In the main analysis, we concatenate the feature

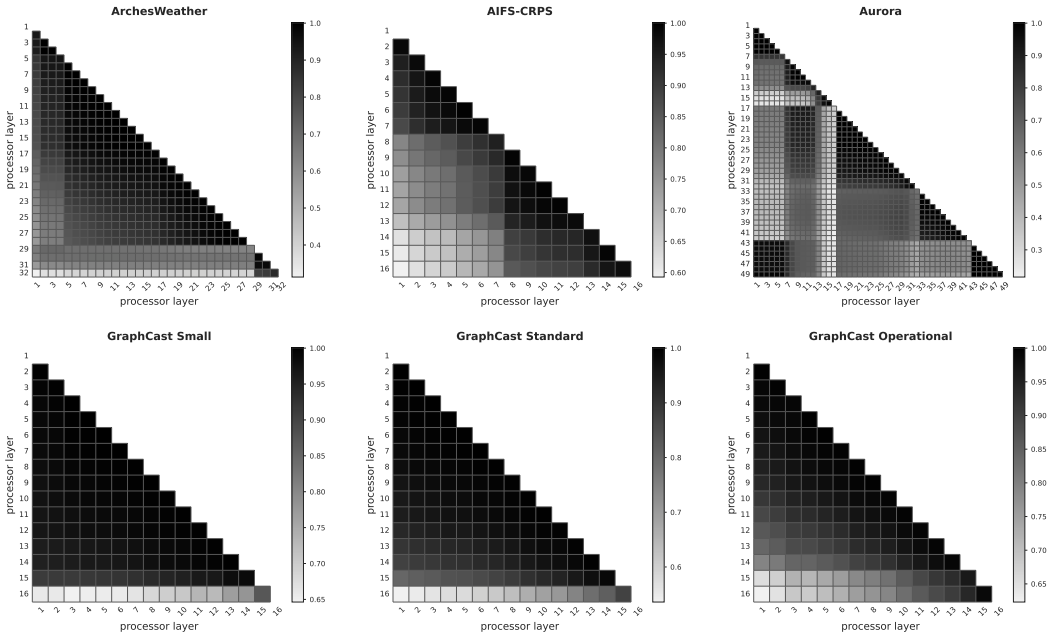


Figure 17: **Raw CKA similarity across processor layers for a single timestep.** Within-model similarity of latent representations between all pairs of processor layers for ArchesWeather, AIFS-CRPS, Aurora, GraphCast Small, GraphCast Standard and GraphCast Operational, calculated using all nodes at the forecast time of 2020-06-01 18 UTC. Colour bars are scaled separately for each model from the minimum CKA value in that panel to 1.

vectors across these latent levels at each horizontal location before calculating similarity. To test whether this choice has a substantial effect on the inferred relationships between models, we repeat the CKA comparison with all nodes from Figure 4, using individual latent levels rather than the concatenated representation.

Figure 24 shows the resulting similarities when individual levels are selected for Aurora and ArchesWeather. The absolute similarity values involving these models vary to some extent with the level used. However, the broad relationships between models remain similar to those obtained using the concatenated representations.

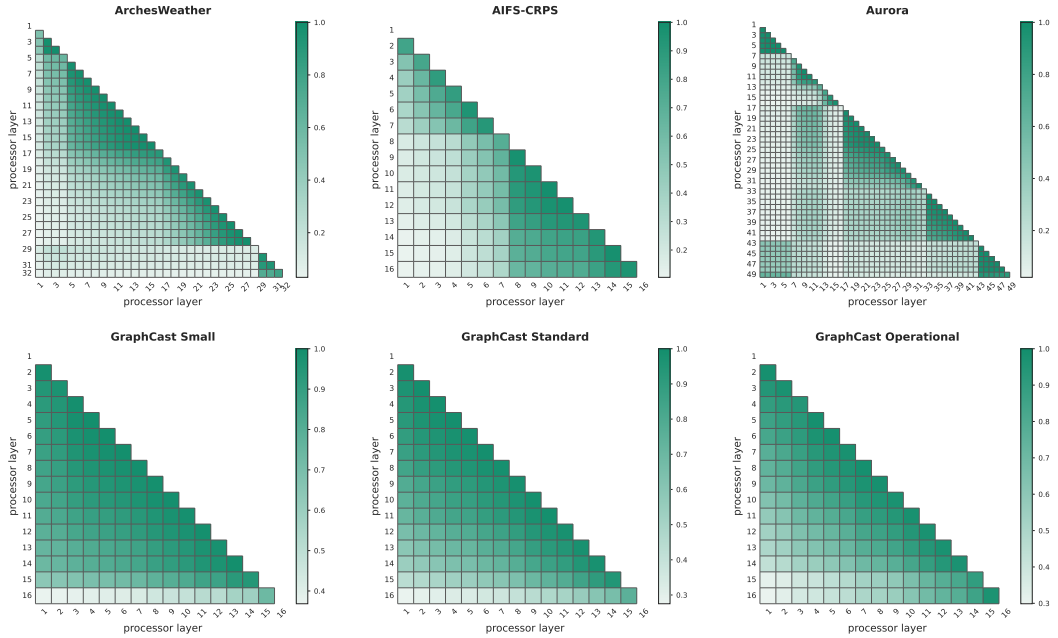


Figure 18: **Calibrated CKA similarity across processor layers at a second forecast time.** Within-model null-calibrated CKA between all pairs of processor-layer representations at 2020-01-24 06 UTC. Colour bars are scaled separately from the minimum calibrated CKA for each model.

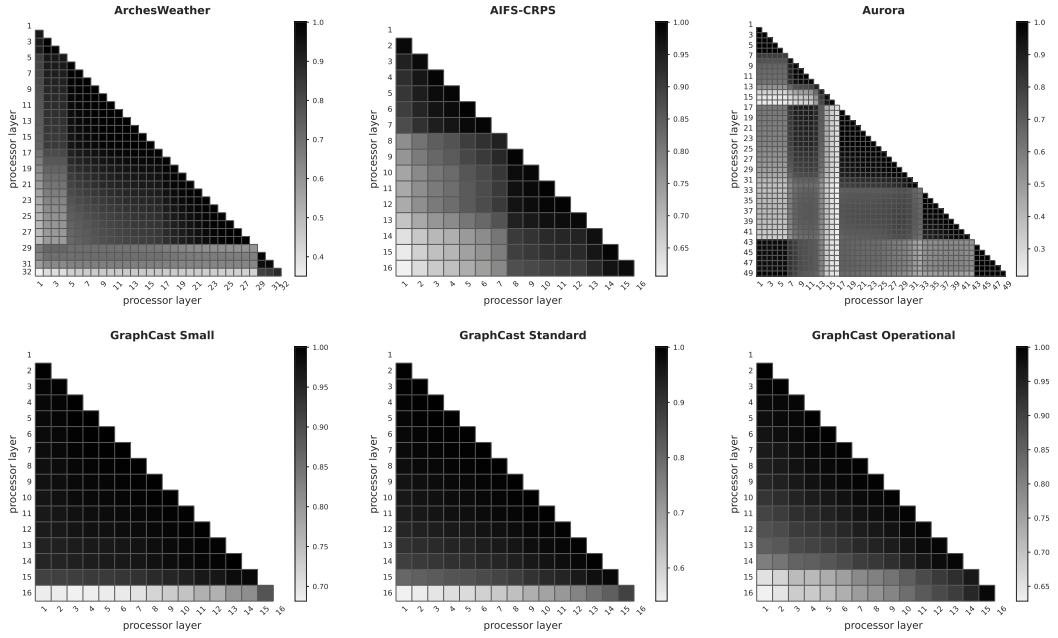


Figure 19: **Raw CKA similarity across processor layers at a second forecast time.** Within-model raw CKA between all pairs of processor-layer representations at 2020-01-24 06 UTC. Colour bars are scaled separately from the minimum CKA for each model.

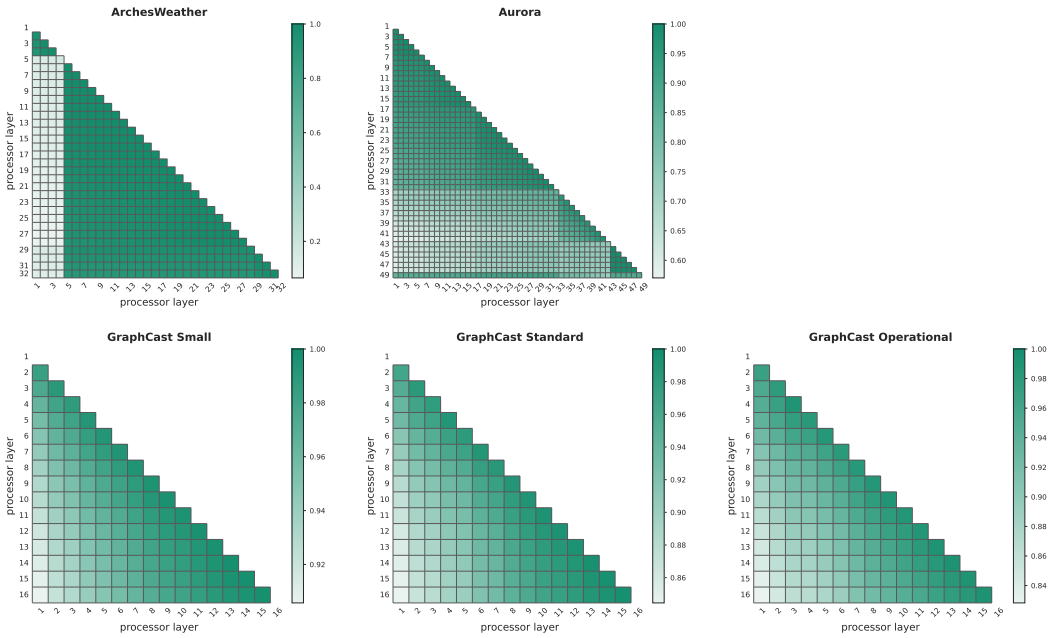


Figure 20: **Calibrated CKA similarity across translated processor-layer representations at a second forecast time.** Within-model null-calibrated CKA between all pairs of translated processor-layer representations at 2020-01-24 06 UTC. Colour bars are scaled separately from the minimum calibrated CKA for each model.

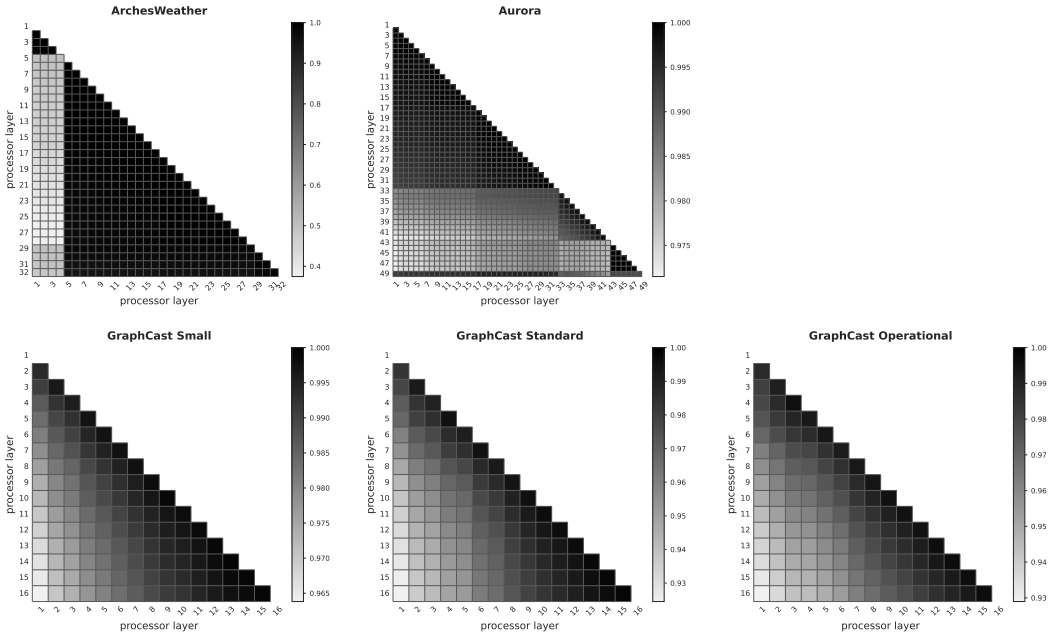


Figure 21: **Raw CKA similarity across translated processor-layer representations at a second forecast time.** Within-model raw CKA between all pairs of translated processor-layer representations at 2020-01-24 06 UTC, calculated using all spatial nodes. Colour bars are scaled separately from the minimum CKA for each model.

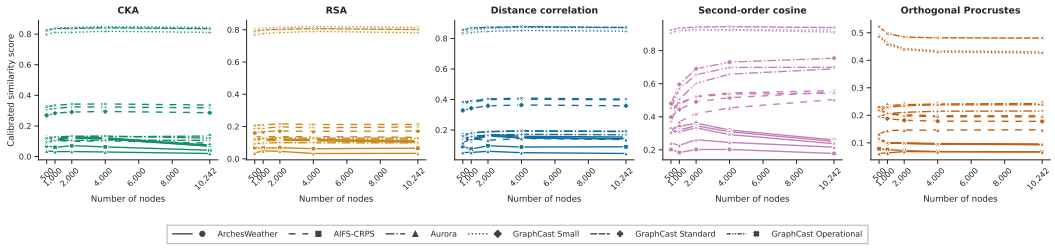


Figure 22: **Calibrated pairwise similarity as a function of spatial samples.** Line style denotes the first model in each pair and marker style the second.

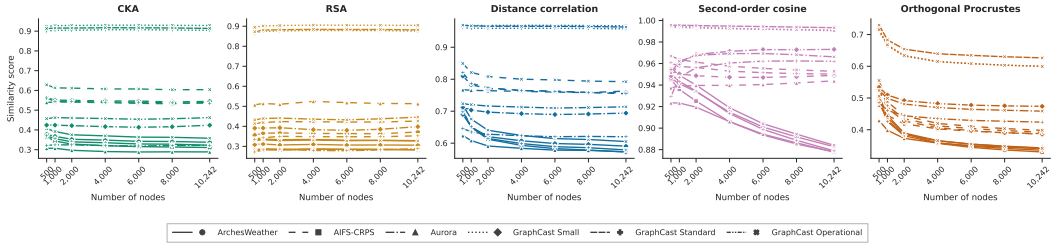


Figure 23: **Raw pairwise similarity as a function of spatial samples.** Line style denotes the first model in each pair and marker style the second.

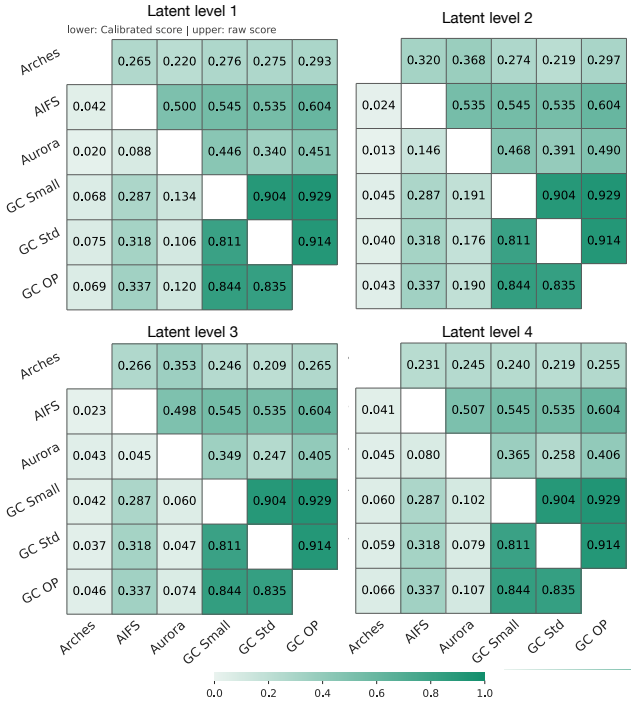


Figure 24: **Pairwise CKA using four different latent levels for Aurora and ArchesWeather.** The lower and upper triangles of each matrix show the calibrated and raw similarity scores respectively.