WEATHER FORECASTING

Learning skillful medium-range global weather forecasting

Remi Lam[1]*†, Alvaro Sanchez-Gonzalez[1]*†, Matthew Willson[1]*†, Peter Wirnsberger[1]†, Meire Fortunato[1]†, Ferran Alet[1]†, Suman Ravuri[1]†, Timo Ewalds[1], Zach Eaton-Rosen[1], Weihua Hu[1], Alexander Merose[2], Stephan Hoyer[2], George Holland[1], Oriol Vinyals[1], Jacklynn Stott[1], Alexander Pritzel[1], Shakir Mohamed[1]*, Peter Battaglia[1]*

Global medium-range weather forecasting is critical to decision-making across many social and economic domains. Traditional numerical weather prediction uses increased compute resources to improve forecast accuracy but does not directly use historical weather data to improve the underlying model. Here, we introduce GraphCast, a machine learning–based method trained directly from reanalysis data. It predicts hundreds of weather variables for the next 10 days at 0.25° resolution globally in under 1 minute. GraphCast significantly outperforms the most accurate operational deterministic systems on 90% of 1380 verification targets, and its forecasts support better severe event prediction, including tropical cyclone tracking, atmospheric rivers, and extreme temperatures. GraphCast is a key advance in accurate and efficient weather forecasting and helps realize the promise of machine learning for modeling complex dynamical systems.

It is 05:45 UTC (coordinated universal time) in mid-October 2022 in Bologna, Italy, at the recently opened high-performance computing facility of the European Centre for Medium-Range Weather Forecasts (ECMWF). For the past several hours, the Integrated Forecasting System (IFS) has been running sophisticated calculations to forecast Earth's weather over the next days and weeks, and its first predictions have just begun to be disseminated to users. This process repeats every 6 hours, every day, to supply the world with the most accurate weather forecasts available.

The IFS, and modern weather forecasting more generally, are triumphs of science and engineering. The dynamics of weather systems are among the most complex physical phenomena on Earth, and each day, countless decisions made by individuals, industries, and policy-makers depend on accurate weather forecasts, from deciding whether to wear a jacket to deciding whether to flee a dangerous storm. The dominant approach for weather forecasting today is numerical weather prediction (NWP), which involves solving the governing equations of weather using supercomputers. The success of NWP lies in the rigorous and ongoing research practices that provide increasingly detailed descriptions of weather phenomena and in how well NWP scales to greater accuracy with greater computational resources (*1*, *2*). As a result, the accuracy of weather forecasts has increased year after year, to the point where the path of a hurricane can be predicted many days ahead—a possibility that was unthinkable even a few decades ago.

But while traditional NWP scales well with compute, capitalizing on the vast amount of historical weather data to improve accuracy is not straightforward. Rather, NWP methods are improved by highly trained experts innovating better models, algorithms, and approximations, which can be a time-consuming and costly process.

Machine learning–based weather prediction (MLWP)—wherein forecast models are trained from historical data, including observations and analysis data—offers an alternative to traditional NWP. MLWP has the potential to improve forecast accuracy by capturing patterns in the data that are not easily represented in explicit equations. MLWP also offers opportunities for greater efficiency by exploiting modern deep learning hardware, rather than supercomputers, and striking more favorable speed–accuracy trade-offs. Recently, MLWP has helped improve on NWP-based forecasting in regimes where traditional NWP is relatively weak, for example, in subseasonal [illegible] wave prediction (*3*) and precipitation [illegible] casting from radar images (*4*–*7*), where accurate equations and robust numerical methods are not as available.

In medium-range weather forecasting—the prediction of atmospheric variables up to 10 days ahead—NWP-based systems such as the IFS are still most accurate. The top deterministic operational system in the world is ECMWF's high-resolution forecast (HRES), a configuration of IFS that produces global 10-day forecasts at 0.1° latitude and longitude resolution, in around an hour (*8*). However, over the past several years, MLWP methods for medium-range forecasting trained on reanalysis data have been steadily advancing, facilitated by benchmarks such as WeatherBench (*8*). Deep learning architectures based on convolutional neural networks (*9*–*11*) and Transformers (*12*) have shown promising results at latitude and longitude resolutions coarser than 1.0°, and recent works—which use graph neural networks (GNNs), Fourier neural operators, and Transformers (*13*–*16*)—have reported performance that begins to rival IFS's at 1.0° and 0.25° for a handful of variables and lead times up to 7 days.

GraphCast

Here, we introduce an MLWP approach for global medium-range weather forecasting called GraphCast, which produces an accurate 10-day forecast in under a minute on a single Google Cloud TPU (Tensor Processing Unit) v4 device and supports applications including predicting tropical cyclone tracks, atmospheric rivers, and extreme temperatures.

GraphCast takes as input the two most recent states of Earth's weather—the current time and 6 hours earlier—and predicts the next state of the weather 6 hours ahead. A single weather state is represented by a 0.25° latitude-longitude grid (721 by 1440), which corresponds to roughly 28 km by 28 km resolution at the equator (Fig. 1A), where each grid point represents a set of surface and atmospheric variables (listed in Table 1). Like traditional NWP systems, GraphCast

Table 1. Weather variables and levels modeled by GraphCast. The numbers in parentheses in the column headings are the number of entries in the column. Boldfaced variables and levels indicate those that were included in the scorecard evaluation. All atmospheric variables are represented at each of the pressure levels.

Surface variables (5)	Atmospheric variables (6)	Pressure levels (37)
2-m temperature (2*T*)	**Temperature** (*T*)	1, 2, 3, 5, 7, 10, 20, 30, **50**, 70,
10-m u wind component (10*U*)	**U component of wind** (*U*)	**100**, 125, **150**, 175, **200**, 225,
10-m v wind component (10*V*)	**V component of wind** (*V*)	**250**, **300**, 350, **400**, 450, **500**,
Mean sea level pressure (*MSL*)	**Geopotential** (*Z*)	550, **600**, 650, **700**, 750, 775,
Total precipitation (*TP*)	**Specific humidity** (*Q*)	800, 825, **850**, 875, 900, **925**,
	Vertical wind speed (*W*)	950, 975, and **1000** hPa

[1]Google DeepMind, London, UK. [2]Google Research, Mountain View, CA, USA.
*Corresponding author. Email: remilam@google.com (R.L.); alvarosg@google.com (A.S.-G.); matthjw@google.com (M.W.); shakir@google.com (S.M.); peterbattaglia@google.com (P.B.)
†These authors contributed equally to this work.

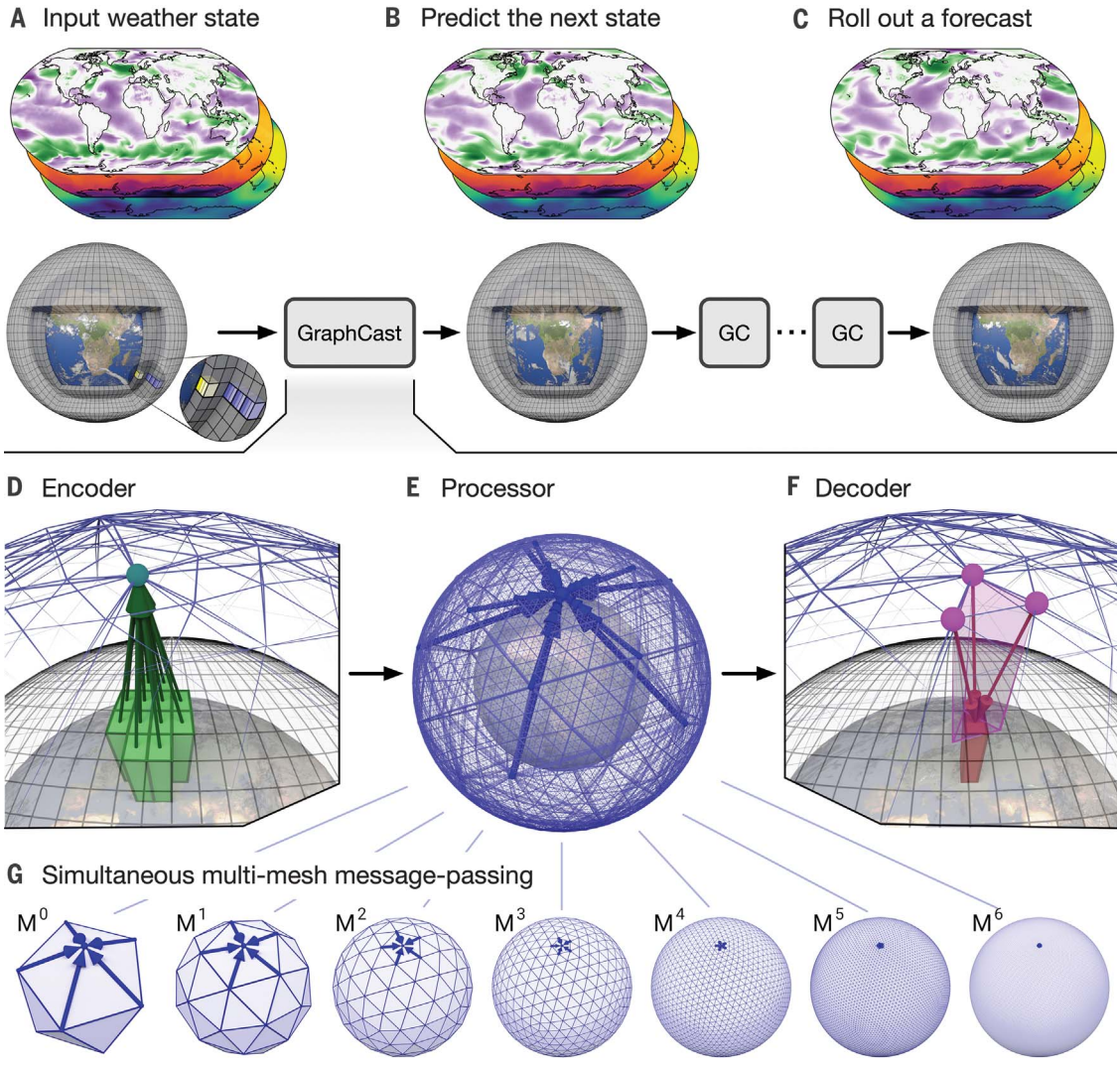

Fig. 1. Model schematic. (**A**) The input weather state(s) are defined on a 0.25° latitude-longitude grid comprising a total of 721 × 1440 = 1,038,240 points. Yellow layers in the close-up pop-out window represent the five surface variables, and blue layers represent the six atmospheric variables that are repeated at 37 pressure levels (5 + 6 × 37 = 227 variables per point in total), resulting in a state representation of 235,680,480 values. (**B**) GraphCast predicts the next state of the weather on the grid. (**C**) A forecast is made by iteratively applying GraphCast (GC) to each previous predicted state, to produce a sequence of states that represent the weather at successive lead times. (**D**) The encoder component of the GraphCast architecture maps local regions of the input (green boxes) into nodes of the multimesh graph representation (green, upward arrows that terminate in the green-blue node). (**E**) The processor component updates each multimesh node using learned message-passing (heavy blue arrows that terminate at a node). (**F**) The decoder component maps the processed multimesh features (purple nodes) back onto the grid representation (red, downward arrows that terminate at a red box). (**G**) The multimesh is derived from icosahedral meshes of increasing resolution, from the base mesh (M^0, 12 nodes) to the finest resolution (M^6, 40,962 nodes), which has uniform resolution across the globe. It contains the set of nodes from M^6 and all the edges from M^0 to M^6. The learned message-passing over the different meshes' edges happens simultaneously, so that each node is updated by all of its incoming edges. [The Earth texture in the figure is used under CC BY 4.0 from https://www.solarsystemscope.com/textures/]

is autoregressive: It can be "rolled out" by feeding its own predictions back in as input, to generate an arbitrarily long trajectory of weather states (Fig. 1, B and C).

GraphCast is implemented as a neural network architecture, based on GNNs in an "encoder-processor-decoder" configuration (*13*, *17*), with a total of 36.7 million parameters (code, weights, and demos can be found at https://github.com/deepmind/graphcast). Previous GNN-based learned simulators (*18*–*20*) have been very effective at learning the complex dynamics of fluid and other systems modeled by partial differential equations, which supports their suitability for modeling weather dynamics.

The encoder (Fig. 1D) uses a single GNN layer to map variables (normalized to zero-mean unit variance) represented as node attributes on the input grid to learned node attributes on an internal "multimesh" representation. The multimesh (Fig. 1G) is a graph that is spatially homogeneous, with high spatial resolution over the globe. It is defined by refining a regular icosahedron (12 nodes, 20 faces, 30 edges) iteratively six times, where each refinement divides each triangle into four smaller ones (leading to four times more faces and edges), and reprojecting the nodes onto the sphere. The multimesh contains the 40,962 nodes from the highest-resolution mesh (which is roughly 1/25 the number of latitude-longitude grid points at 0.25°) and the union of all the edges created in the intermediate graphs, forming a flat hierarchy of edges with varying lengths. The processor (Fig. 1E) uses 16 unshared GNN layers to perform learned message-passing on the multimesh, enabling efficient local and long-range information propagation with few message-passing steps. The decoder (Fig. 1F) maps the final processor layer's learned features from the multimesh representation back to the latitude-longitude grid. It uses a single GNN layer and predicts the output as a residual update to the most recent input state (with output normalization to achieve unit variance on the target residual). See supplementary materials section 3 for further architectural details.

During model development, we used 39 years (1979–2017) of historical data from ECMWF's ERA5 (*21*) reanalysis archive. As a training objective, we averaged the mean squared error (MSE) between GraphCast's predicted states over N autoregressive steps and the corresponding ERA5 states, with the error weighted by

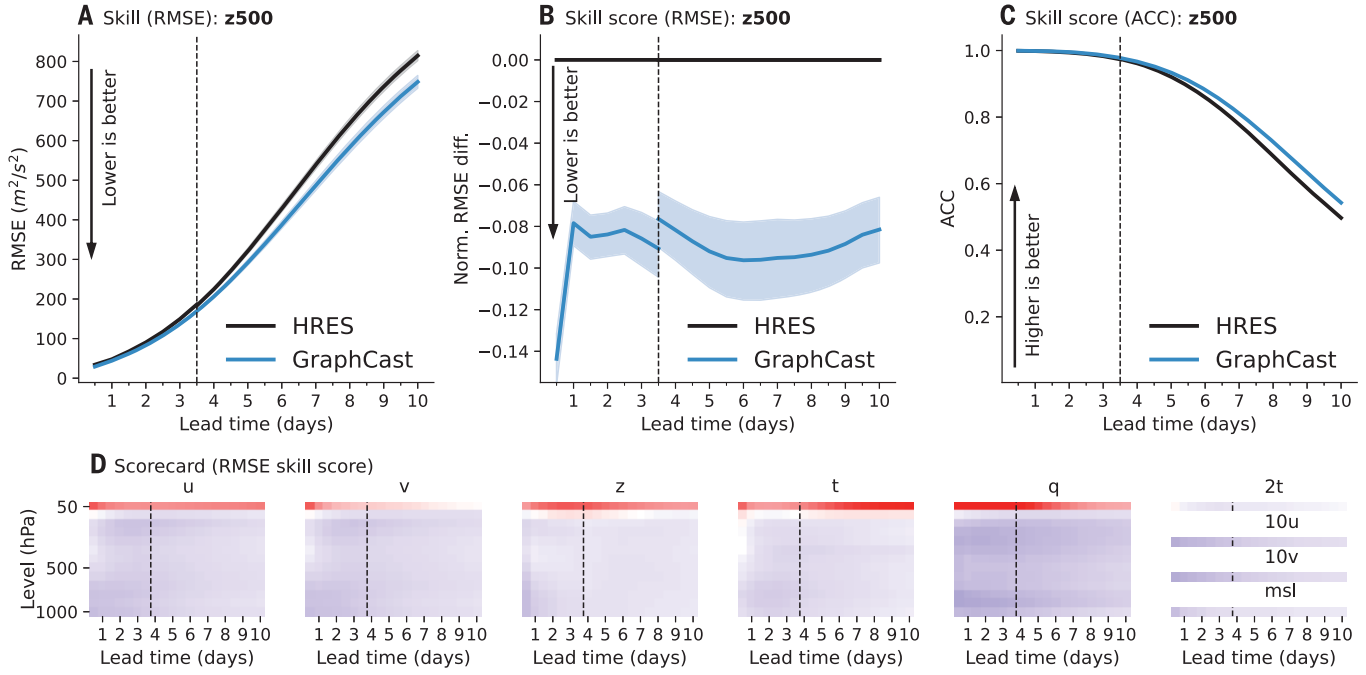

Fig. 2. Global skill and skill scores for GraphCast and HRES in 2018. (**A**) RMSE skill (*y* axis) for GraphCast (blue lines) and HRES (black lines), on *Z*500, as a function of lead time (*x* axis). Error bars represent 95% confidence intervals. The vertical dashed line represents 3.5 days, which is the last 12-hour increment of the HRES 06z/18z forecasts. The black line represents HRES, where lead times earlier and later than 3.5 days are from the 06z/18z and 00z/12z initializations, respectively. (**B**) RMSE skill score (*y* axis) for GraphCast versus HRES, on *Z*500, as a function of lead time (*x* axis). Error bars represent 95% confidence intervals for the skill score. We observe a discontinuity in GraphCast's curve because skill scores up to 3.5 days are computed between GraphCast (initialized at 06z/18z) and HRES's 06z/18z initialization, whereas skill scores after 3.5 days are computed with respect to HRES's 00z/12z initializations. (**C**) ACC skill (*y* axis) for GraphCast (blue lines) and HRES (black lines), on *Z*500, as a function of lead time (*x* axis). (**D**) Scorecard of RMSE skill scores for GraphCast, with respect to HRES. Each subplot corresponds to one variable: *U*, *V*, *Z*, *T*, *Q*, *2T*, *10U*, *10V*, and *MSL*. The rows of each heatmap correspond to the 13 pressure levels (for the atmospheric variables), from 50 hPa at the top to 1000 hPa at the bottom. The columns of each heatmap correspond to the 20 lead times at 12-hour intervals, from 12 hours on the left to 10 days on the right. Each cell's color represents the skill score, as shown in (B), where blue represents negative values (GraphCast has better skill) and red represents positive values (HRES has better skill).

vertical level (see supplementary materials eq. 19). The value of N was increased incrementally from 1 to 12 (i.e., from 6 hours to 3 days) over the course of training, and the gradient of the loss was computed by backpropagation through time (*22*). GraphCast was trained to minimize the training objective using gradient descent, which took roughly 4 weeks on 32 Cloud TPU v4 devices using batch parallelism. See supplementary materials section 4 for further training details. Consistent with real deployment scenarios, where future information is not available for model development, we evaluated GraphCast on the held-out data from the years 2018 onward (see supplementary materials section 5.1).

Verification methods

We verified GraphCast's forecast skill comprehensively by comparing its accuracy to that of HRES on a large number of variables, levels, and lead times. We quantified the respective skills of GraphCast, HRES, and ML baselines with two skill metrics: the root mean square error (RMSE) and the anomaly correlation coefficient (ACC).

Of the 227 variable and level combinations predicted by GraphCast at each grid point, we evaluated its skill versus HRES on 69 of them, corresponding to the 13 levels of WeatherBench (*8*) and variables (*23*) from the ECMWF Scorecard (*24*); see boldface variables and levels in Table 1 and supplementary materials section 1.2 for which HRES cycle was operational during the evaluation period. In addition to the aggregate performance reported in the main text, supplementary materials section 7 provides further detailed evaluations, including other variables, precipitation, regional performance, latitude and pressure level effects, spectral properties, blurring, biases, comparisons to other ML-based forecasts, and effects of model design choices.

In making these comparisons, skill was established on the basis of two key elements: (i) the selection of the ground truth for comparison and (ii) a careful accounting of the data assimilation windows used to infer this data from observations. We used ERA5 as the ground truth for evaluating GraphCast, because it was trained to take ERA5 data as input and predict ERA5 data as outputs. However, evaluating HRES forecasts against ERA5 would result in nonzero error on the initial forecast step. Instead, we constructed an "HRES forecast at step 0" (HRES-fc0) dataset to use as ground truth for HRES. HRES-fc0 contains the inputs to HRES forecasts at future initializations (see supplementary materials section 1.2), ensuring that each data point is grounded by recent observations and that the zeroth step of HRES forecasts will have zero error.

For a fair comparison, we had to ensure that the ERA5 initial conditions for GraphCast were derived from assimilation windows that look no further into the future than those used by HRES. HRES initializations (00z/06z/12z/18z, where 00z means 00:00 UTC in Zulu convention) always assimilate observations 3 hours into the future, whereas ERA5 initializations assimilate observations 9 hours into the future at 00z/12z and 3 hours into the future at 06z/18z. This constrained the choice of initialization times for GraphCast to 06z/18z in all our results. We used the same initializations for HRES when comparing performance up to 3.75 days. Beyond that, HRES archived forecasts are only available from 00z/12z initializations. The

transition from 06z/18z to 00z/12z initializations for HRES induced a small discontinuity in our plots, which is indicated by a vertically dashed line at the appropriate lead time. Supplementary materials section 5 contains further verification details, including details of the comparisons protocol between GraphCast and HRES (supplementary materials section 5.2) and the effect of initialization lookahead on both models' performance (supplementary materials section 5.2.2).

Forecast verification results

We find that GraphCast has greater weather forecasting skill than HRES when evaluated on 10-day forecasts at a horizontal resolution of 0.25° for latitude and longitude and at 13 vertical levels. Figure 2, A to C, shows how GraphCast (blue lines) outperforms HRES (black lines) on the *Z*500 (geopotential at 500 hPa) "headline" field in terms of RMSE skill, RMSE skill score [i.e., the normalized RMSE difference between model *A* and baseline *B* defined as $(RMSE_A - RMSE_B)/(RMSE_B)$], and ACC skill. Using *Z*500, which encodes the synoptic-scale pressure distribution, is common in the literature, as it has strong meteorological importance (*8*). The plots show that GraphCast has better skill scores across all lead times, with a skill score improvement of around 7 to 14%. Plots for additional headline variables are given in supplementary materials section 7.1.

Figure 2D summarizes the RMSE skill scores for all 1380 evaluated variables and pressure levels, across the 10-day forecasts, in a format analogous to the ECMWF Scorecard. The cell colors are proportional to the skill score, where blue indicates that GraphCast had better skill, and red indicates that HRES had better skill. GraphCast outperformed HRES on 90.3% of the 1380 targets and significantly ($P \leq 0.05$, nominal sample size $n \in \{729, 730\}$) outperformed HRES on 89.9% of targets. See supplementary materials section 5.4 for methodology and table S4 for *P* values, test statistics, and effective sample sizes.

The regions of the atmosphere in which HRES had better performance than GraphCast (top rows in red in the scorecards) were disproportionately localized in the stratosphere and had the lowest training loss weight (see supplementary materials section 7.2.2). When excluding the 50 hPa level, GraphCast significantly outperforms HRES on 96.9% of the remaining 1280 targets. When excluding levels 50 and 100 hPa, GraphCast significantly outperforms HRES on 99.7% of the remaining 1180 targets. When conducting per-region evaluations, we found that the previous results generally hold across the globe, as detailed in figs. S14 to S16.

We found that increasing the number of autoregressive steps in the MSE loss improves GraphCast performance at longer lead times

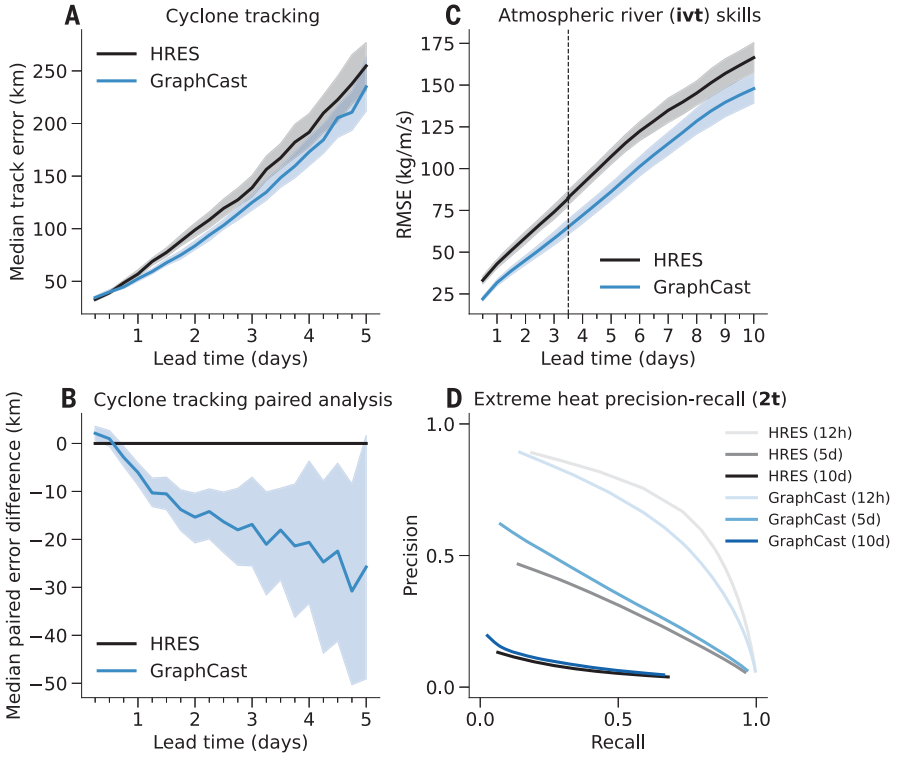

Fig. 3. Severe event prediction. (**A**) Cyclone tracking performances for GraphCast and HRES. The *x* axis represents lead times (in days), and the *y* axis represents median track error (in kilometers). Error bars represent bootstrapped 95% confidence intervals for the median. (**B**) Cyclone tracking paired error difference between GraphCast and HRES. The *x* axis represents lead times (in days), and the *y* axis represents median paired error difference (in kilometers). Error bars represent bootstrapped 95% confidence intervals for the median difference (see supplementary materials section 8.1). (**C**) Atmospheric river prediction (*IVT*) skills for GraphCast and HRES. The *x* axis represents lead times (in days), and the *y* axis represents RMSE. Error bars are 95% confidence intervals. (**D**) Extreme heat prediction precision-recall for GraphCast and HRES. The *x* axis represents recall, and the *y* axis represents precision. The curves represent different precision-recall trade-offs when sweeping over gain applied to forecast signals (see supplementary materials section 8.3).

(see supplementary materials section 7.3.2). It also encourages GraphCast to blur to a degree at longer lead times (see fig. S38), which means that its forecasts will lie somewhere between a traditional deterministic forecast and an ensemble mean. HRES's underlying physical equations, however, do not lead to blurred predictions. To assess whether GraphCast's relative advantage over HRES on RMSE skill is due to blurrier forecasts better optimizing RMSE, we artificially blurred HRES's forecasts with blurring filters. We fit filters for GraphCast and HRES by minimizing the RMSE between filtered predictions and the models' respective ground truths. We found that RMSE-optimized blurring applied to GraphCast has greater skill than analogous blurring applied to HRES on 88.0% of our 1380 verification targets, which is generally consistent with our above conclusions (see supplementary materials section 7.4). Still, blurrier forecasts may not be desirable for some applications, which we discuss further in the Conclusions section. We also compared GraphCast's performance to the top competing ML-based weather model, Pangu-Weather (*16*), and found that GraphCast outperformed it on 99.2% of the 252 targets presented (see supplementary materials section 6 for details).

Severe event forecasting results

Beyond evaluating GraphCast's forecast skill against HRES's on a wide range of variables and lead times, we also evaluated how its forecasts support predicting severe events, including tropical cyclones tracks, atmospheric rivers, and extreme temperature. These are key downstream applications for which GraphCast is not specifically trained, but which are very important for human activity.

Tropical cyclone tracks

Improving the accuracy of tropical cyclone tracking can help avoid injury and loss of life and can reduce economic harm (*25*). A cyclone's existence and trajectory are predicted by applying a tracking algorithm to forecasts of geopotential (*Z*), horizontal wind (10*U*/10*V*, *U*/*V*), and mean sea level pressure (*MSL*). We

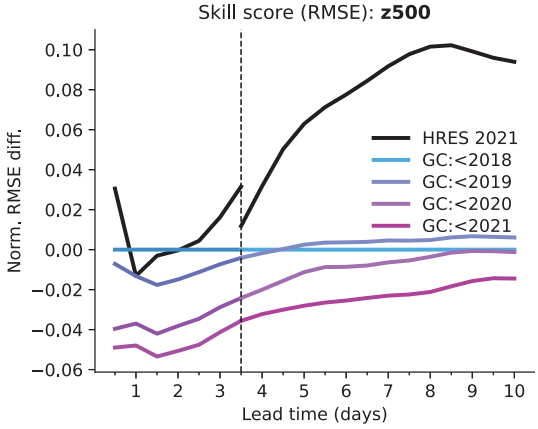

Fig. 4. Training GraphCast on more recent data. Each colored line represents GraphCast trained with data ending before a different year, from 2018 (blue) to 2021 (purple). The *y* axis represents RMSE skill scores on 2021 test data, for *Z*500, with respect to GraphCast trained up to before 2018, over lead times (*x* axis). The vertical dashed line represents 3.5 days, where the HRES 06z/18z forecasts end. The black line represents HRES, where lead times earlier and later than 3.5 days are from the 06z/18z and 00z/12z initializations, respectively.

implemented a tracking algorithm based on ECMWF's published protocols (*26*) and applied it to GraphCast's forecasts to produce cyclone track predictions (see supplementary materials section 8.1). As a baseline for comparison, we used the operational tracks obtained from HRES's 0.1° forecasts with ECMWF's own tracker, stored in the TIGGE (THORPEX Interactive Grand Global Ensemble) archive (*27*, *28*). The reason we used different trackers for GraphCast and HRES is because no single tracker should necessarily give best performance across different forecasts, therefore, we used the best tracker for each model. We measured errors for both models against the tracks from IBTrACS [International Best Track Archive for Climate Stewardship (*29*, *30*)], a separate reanalysis dataset of cyclone tracks aggregated rom various analysis and observational sources. To be consistent with established evaluation of tropical cyclone prediction (*26*), we evaluate all tracks when both GraphCast and HRES detect a cyclone, ensuring that both models are evaluated on the same events, and verify that each model's true-positive rates are similar.

Figure 3A shows that GraphCast has lower median track error than HRES over the period 2018–2021 (median was chosen to resist outliers). As per-track errors for HRES and GraphCast are correlated, we also measured the per-track paired error difference between the two models and found that GraphCast is significantly better than HRES for lead times of 18 hours to 4.75 days, as shown in Fig. 3B. The error bars show the bootstrapped 95% confidence intervals for the median (see supplementary materials section 8.1 for details).

Atmospheric rivers

Atmospheric rivers are narrow regions of the atmosphere that are responsible for most of the poleward water vapor transport across the mid-latitudes and generate 30 to 65% of annual precipitation on the US West Coast (*31*). Their strength can be characterized by the vertically integrated water vapor transport *IVT* (*32*, *33*), indicating whether an event will provide beneficial precipitation or be associated with catastrophic damage (*34*). *IVT* can be computed from the nonlinear combination of the horizontal wind speed (*U* and *V*) and specific humidity (*Q*), which GraphCast predicts. We evaluated GraphCast forecasts over coastal North America and the Eastern Pacific during cold months (October to April), when atmospheric rivers are most frequent. Despite not being specifically trained to characterize atmospheric rivers, Fig. 3C shows that GraphCast improves the prediction of *IVT* compared with HRES, from 25% at short lead time to 10% at longer horizons (see supplementary materials section 8.2 for details).

Extreme heat and cold

Extreme heat and cold are characterized by large anomalies with respect to typical climatology (*3*, *35*, *36*), which can be dangerous and disrupt human activities. We evaluated the skill of HRES and GraphCast in predicting events above the top 2% of historical values across location, time of day, and month of the year, for 2*T* at 12-hour, 5-day, and 10-day lead times, for land regions across the Northern and Southern hemispheres over their respective summer months. We plotted precision-recall curves (*37*) to reflect different possible trade-offs between reducing false positives (high precision) and reducing false negatives (high recall). For each forecast, we obtained the curve by varying a "gain" parameter that scales the 2*T* forecast's deviations with respect to the median climatology.

Figure 3D shows that GraphCast's precision-recall curves are above HRES's for 5- and 10-day lead times, suggesting that GraphCast's forecasts are generally superior to those of HRES at extreme classification over longer horizons. By contrast, HRES has better precision-recall at the 12-hour lead time, which is consistent with the 2*T* skill score of GraphCast over HRES being near zero, as shown in Fig. 2D. We generally find these results to be consistent across other variables relevant to extreme heat, such as *T*850 (temperature at 850 hPa) and *Z*500 (geopotential at 500 hPa) (*36*), other extreme thresholds (5, 2, and 0.5%), and extreme cold forecasting in winter. See supplementary materials section 8.3 for details.

Effect of training data recency

GraphCast can be retrained periodically with recent data, which in principle allows it to capture weather patterns that change over time—in response to, for example, the effects of climate change—and long climate oscillations. We trained four variants of GraphCast from scratch, with data that always began in 1979 but ended in 2017, 2018, 2019, and 2020, respectively (we label the variant ending in 2017 as "GraphCast: <2018," etc.). We compared their performances to HRES on 2021 test data.

Figure 4 shows the skill scores (normalized by GraphCast:<2018) of the four variants and HRES for *Z*500. We found that while GraphCast's performance when trained up to before 2018 is still competitive with HRES in 2021, training it up to before 2021 further improves its skill scores (see supplementary materials section 7.1.3). We speculate that this recency effect allows recent weather trends to be captured to improve accuracy. This shows that GraphCast's performance can be improved by retraining on more recent data.

Conclusions

GraphCast's forecast skill and efficiency compared with HRES shows that MLWP methods are now competitive with traditional weather forecasting methods. Additionally, GraphCast's performance on severe event forecasting, which it was not directly trained for, demonstrates its robustness and potential for downstream value. We believe this marks a turning point in weather forecasting, which helps open new avenues to strengthen the breadth of weather-dependent decision-making by individuals and industries by making cheap prediction more accurate and accessible as well as suitable for specific applications.

With 36.7 million parameters, GraphCast is a relatively small model by modern ML standards, chosen to keep the memory footprint tractable. And while HRES is released on 0.1° resolution, 137 levels, and up to 1-hour time steps, GraphCast operates on 0.25° latitude and longitude resolution, 37 vertical levels, and 6-hour time steps, because of the ERA5 training data's native 0.25° resolution and engineering challenges in fitting higher-resolution data on hardware. Generally, GraphCast should be viewed as a family of models, with the current version being the largest we can practically fit under current engineering constraints, but which have the potential to scale much further in the future with greater compute resources and higher-resolution data.

One key limitation of our approach is in how uncertainty is handled. We focused on deterministic forecasts and compared against HRES, but the other pillar of ECMWF's IFS, the ensemble forecasting system ENS, is especially important for quantifying the probability of extreme events and as the skill of the forecast decreases at longer lead times. The nonlinearity of weather dynamics means that there is increasing uncertainty at longer lead times, which is not well captured by a single deterministic forecast. ENS addresses this by generating multiple, stochastic forecasts, which approximate a predictive distribution over future weather; however, generating multiple forecasts is expensive. By contrast, GraphCast's MSE training objective encourages it to spatially blur its predictions in the presence of uncertainty, which may not be desirable for some applications where knowing tail, or joint, probabilities of events is important. Building probabilistic forecasts that model uncertainty more explicitly, along the lines of ensemble forecasts, is a crucial next step.

It is important to emphasize that data-driven MLWP relies critically on large quantities of data and their quality, which, in the case of models trained on reanalysis, depend on the fidelity of NWPs. Therefore, rich high-quality data sources such as ECMWF's MARS (Meteorological Archival and Retrieval System) archive (*38*) are invaluable. Our approach should not be regarded as a replacement for traditional weather forecasting methods, which have been developed for decades, rigorously tested in many real-world contexts, and offer many features we have not yet explored. Rather, our work should be interpreted as evidence that MLWP is able to meet the challenges of real-world forecasting problems and has the potential to complement and improve the current best methods.

Beyond weather forecasting, GraphCast can open new directions for other important geospatiotemporal forecasting problems, including climate and ecology, energy, agriculture, and human and biological activity, as well as other complex dynamical systems. We believe that learned simulators, trained on rich, real-world data, will be crucial in advancing the role of machine learning in the physical sciences.

REFERENCES AND NOTES

1. S. G. Benjamin *et al*., *Meteorol. Monogr.* **59**, 13.1–13.67 (2019).
2. P. Bauer, A. Thorpe, G. Brunet, *Nature* **525**, 47–55 (2015).
3. I. Lopez-Gomez, A. McGovern, S. Agrawal, J. Hickey, *Artif. Intell. Earth Syst.* **2**, e220035 (2022).
4. X. Shi *et al*., *Adv. Neural Inf. Process. Syst.* **30**, 5617–5627 (2017).
5. C. K. Sønderby *et al*., arXiv:2003.12140 [cs.LG] (2020).
6. S. Ravuri *et al*., *Nature* **597**, 672–677 (2021).
7. L. Espeholt *et al*., *Nat. Commun.* **13**, 5145 (2022).
8. S. Rasp *et al*., *J. Adv. Model. Earth Syst.* **12**, e2020MS002203 (2020).
9. J. A. Weyn, D. R. Durran, R. Caruana, *J. Adv. Model. Earth Syst.* **11**, 2680–2693 (2019).
10. J. A. Weyn, D. R. Durran, R. Caruana, *J. Adv. Model. Earth Syst.* **12**, e2020MS002109 (2020).
11. S. Rasp, N. Thuerey, *J. Adv. Model. Earth Syst.* **13**, e2020MS002405 (2021).
12. T. Nguyen, J. Brandstetter, A. Kapoor, J. K. Gupta, A. Grover, arXiv:2301.10343 [cs.LG] (2023).
13. R. Keisler, Forecasting global weather with graph neural networks. arXiv:2202.07575 [physics.ao-ph] (2022).
14. J. Pathak *et al*., arXiv:2202.11214 [physics.ao-ph] (2022).
15. T. Kurth *et al*., arXiv:2208.05419 [physics.ao-ph] (2022).
16. K. Bi *et al*., arXiv:2211.02556 [physics.ao-ph] (2022).
17. P. W. Battaglia *et al*., arXiv:1806.01261 [cs.LG] (2018).
18. A. Sanchez-Gonzalez *et al*., *Proc. Mach. Learn. Res.* **119**, 8459–8468 (2020).
19. T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, P. Battaglia, "Learning mesh-based simulation with graph networks," International Conference on Learning Representations (ICLR 2021), 3 to 7 May 2021.
20. F. Alet *et al*., *Proc. Mach. Learn. Res.* **97**, 212–222 (2019).
21. H. Hersbach *et al*., *Q. J. R. Meteorol. Soc.* **146**, 1999–2049 (2020).
22. P. Werbos, *Proc. IEEE* **78**, 1550–1560 (1990).
23. Because precipitation in ERA5 has known biases (*39*), no development decision for GraphCast was made to improve performance on precipitation, and GraphCast simply uses precipitation as an auxiliary input and output. Note that precipitation is sparse and non-Gaussian and would have possibly required different modeling decisions than the other variables. Additionally, precipitation is not available in the HRES analysis products in a form amenable to our evaluation protocol (see next paragraphs). Thus, any claim about precipitation prediction is left out of the scope of this work, and we show precipitation evaluation using a different protocol in supplementary materials section 7.1.4 for completeness only.
24. T. Haiden *et al*., "Evaluation of ECMWF forecasts, including the 2018 upgrade," ECMWF Technical Memorandum No. 831 (European Centre for Medium-Range Weather Forecasts, 2018); https://doi.org/10.21957/ldw15ckqi.
25. A. B. Martinez, *Econometrics* **8**, 18 (2020).
26. L. Magnusson *et al*., "Tropical cyclone activities at ECMWF," ECMWF Technical Memorandum No. 888 (European Centre for Medium-Range Weather Forecasts, 2021); https://doi.org/10.21957/zzxzzygwv.
27. P. Bougeault *et al*., *Bull. Am. Meteorol. Soc.* **91**, 1059–1072 (2010).
28. R. Swinbank *et al*., *Bull. Am. Meteorol. Soc.* **97**, 49–67 (2016).
29. K. R. Knapp, M. C. Kruk, D. H. Levinson, H. J. Diamond, C. J. Neumann, *Bull. Am. Meteorol. Soc.* **91**, 363–376 (2010).
30. K. R. Knapp, H. J. Diamond, J. P. Kossin, M. C. Kruk, C. J. Schreck III, International Best Track Archive for Climate Stewardship (IBTrACS) Project, version 4, NOAA National Centers for Environmental Information (2018); https://doi.org/10.25921/82ty-9e16.
31. W. Chapman, A. Subramanian, L. Delle Monache, S. Xie, F. Ralph, *Geophys. Res. Lett.* **46**, 10627–10635 (2019).
32. P. J. Neiman, F. M. Ralph, G. A. Wick, J. D. Lundquist, M. D. Dettinger, *J. Hydrometeorol.* **9**, 22–47 (2008).
33. B. J. Moore, P. J. Neiman, F. M. Ralph, F. E. Barthold, *Mon. Weather Rev.* **140**, 358–378 (2012).
34. T. W. Corringham, F. M. Ralph, A. Gershunov, D. R. Cayan, C. A. Talbot, *Sci. Adv.* **5**, eaax4631 (2019).
35. L. Magnusson, T. Haiden, D. Richardson, "Verification of extreme weather events: Discrete predictands," ECMWF Technical Memorandum No. 731 (European Centre for Medium-Range Weather Forecasts, 2014); https://doi.org/10.21957/1iql31n2c.
36. L. Magnusson, ECMWF confluence wiki: 202208 - Heatwave - UK (2022); https://confluence.ecmwf.int/display/FCST/202208+-+Heatwave+-+UK.
37. T. Saito, M. Rehmsmeier, *PLOS ONE* **10**, e0118432 (2015).
38. C. Maass, E. Cuartero, MARS user documentation (2022); https://confluence.ecmwf.int/display/UDOC/MARS+user+documentation.
39. D. A. Lavers, A. Simmons, F. Vamborg, M. J. Rodwell, *Q. J. R. Meteorol. Soc.* **148**, 3152–3165 (2022).
40. D. Fan *et al*., *Sci. Robot.* **4**, eaay5063 (2019).
41. T. Ewalds, Source code for GraphCast, google-deepmind/graphcast: Version 0.1, Zenodo (2023); https://doi.org/10.5281/zenodo.10058758.

ACKNOWLEDGMENTS

In alphabetical order, we thank K. Allen, C. Blundell, M. Botvinick, Z. B. Bouallegue, M. Brenner, R. Carver, M. Chantry, M. Deisenroth, P. Deuben, M. Garnelo, R. Keisler, D. Kochkov, C. Mattern, P. Mirowski, P. Norgaard, I. Price, C. Qin, S. Racanière, S. Rasp, Y. Rubanova, K. Shah, J. Smith, D. Worrall, and countless others at Alphabet and ECMWF for advice and feedback on our work. We also thank ECMWF for providing invaluable datasets to the research community. The style of the opening paragraph was inspired by (*40*). **Funding:** All research in this study was funded by Google DeepMind and Alphabet. There was no external funding. **Author contributions:** Conceptualization: R.L., A.S.-G., M.W., S.M., and P.B. Data curation: R.L., A.S.-G., M.W., A.M., and P.B. Formal analysis: R.L., A.S.-G., M.W., P.W., M.F., F.A., S.R., T.E., Z.E.-R., W.H., A.P., S.M., and P.B. Investigation: R.L., A.S. -G., M.W., P.W., M.F., F.A., S.R., T.E., Z.E.-R., W.H., A.P., S.M., and P.B. Methodology: R.L., A.S.-G., M.W., P.W., M.F., F.A., S.R., T.E., Z.E.-R., W.H., A.P., S.M., and P.B. Project administration: R.L., A.S.-G., M.W., G.H., O.V., J.S., S.M., and P.B. Software: R.L., A.S.-G., M.W., P.W., M.F., F.A., S.R., T.E., Z.E.-R., W.H., A.P., and P.B. Supervision: R.L., S.M., and P.B. Validation: R.L., A.S.-G., M.W., P.W., M.F., F.A., S.R., T.E., Z.E.-R., W.H., A.P., S.M., and P.B. Visualization: R.L., A.S.-G., M.W., F.A., and P.B. Writing – original draft: R.L., A.S.-G., M.W., P.W., M.F., F.A., S.R., T.E., Z.E.-R., S.H., A.P., S.M., and P.B. Writing – review & editing: R.L., A.S.-G., M.W., P.W., M.F., F.A., S.R., T.E., Z.E.-R., S.H., A.P., S.M., and P.B. **Competing interests:** This work was done in the course of employment at Google DeepMind, with no other competing financial interests. A.S.-G., R.L., P.B., M.W., P.W., M.F., and A.P. have filed a provisional patent application relating to machine learning for learned medium-range global weather forecasting (US Provisional App. no. US63/435,163). **Data and materials availability:** GraphCast's code and trained weights are publicly available on GitHub at https://github.com/deepmind/graphcast (*41*). This work used publicly available data from the ECMWF. We used the ECMWF archive (expired real-time) products for ERA5, HRES, and TIGGE products, whose use is governed by the Creative Commons Attribution 4.0 International (CC BY 4.0). We used IBTrACS Version 4 from https://www.ncei.noaa.gov/products/international-best-track-archive and (*29*, *30*), as required.

SUPPLEMENTARY MATERIALS

science.org/doi/10.1126/science.adi2336
Materials and Methods
Supplementary Text
Figs. S1 to S53
Tables S1 to S4
References (*42*–*75*)

Submitted 18 April 2023; accepted 1 November 2023
Published online 14 November 2023
10.1126/science.adi2336