Using a Coevolutionary Postprocessor to Improve Skill for Both Forecasts of Surface Temperature and Nowcasts of Convection Occurrence

PAUL J. ROEBBER

Atmospheric Science Group, Department of Mathematical Sciences, University of Wisconsin–Milwaukee, Milwaukee, Wisconsin

JOHN CROCKETT

*Cooperative Institute for Research in the Atmosphere, Colorado State University, Fort Collins, Colorado, and
Meteorological Development Laboratory, Office of Science and Technology Integration, National
Weather Service, Silver Spring, Maryland*

(Manuscript received 11 March 2019, in final form 22 August 2019)

ABSTRACT

An evolutionary programming postprocessor, using coevolution in a predator–prey ecosystem model, is developed and applied both to 72-h, 2-m temperature forecasts for the conterminous United States and southern Canada and to 60-min nowcasts of convection occurrence for the United States east of 94°W. The new approach improves deterministic and probabilistic forecasts of surface temperature relative to bias-corrected numerical weather prediction forecasts and to an earlier version of evolutionary programming forecasts for these same data. The new method also improves deterministic performance for an artificial neural network trained and evaluated for these same data. Additionally, the new approach substantially improves these forecasts' reliability, as evidenced by reductions in the occurrence of excessive outliers in the rank histogram. The coevolutionary postprocessor also improves deterministic nowcasts of convection occurrence when compared to those produced by the National Weather Service's AutoNowCaster system and to those obtained using multiple logistic regression. Notably, the degree of improvement relative to traditional methods appears to be problem dependent, while the training and implementation of such a system requires additional effort. However, the coevolutionary system is shown to be robust to imbalances between the frequency of positive and null events in the training data, unlike many postprocessing methods; to be implementable and effective in an adaptive mode, removing the need for retraining as inputs (such as numerical weather prediction model data) change; and to provide a useful, alternative perspective on the likelihood of event occurrence when used in combination with other methods.

1. Introduction

We are motivated in this study to explore a new method for producing deterministic and probabilistic forecasts, which holds the promise of improving the ongoing problem of ensemble underdispersion. For that purpose, we use a temperature dataset developed by Roebber (2018). We also show that this technique can be skillfully applied to the classification problem for convective occurrence in the 1-h-ahead range.

The new method builds off the work of Roebber (2018), who developed an evolutionary programming (EP) postprocessor that used a static ecosystem model to produce deterministic and probabilistic 2-m temperature forecasts. These forecasts were more skillful than both

raw and bias-corrected forecasts obtained via multiple linear regression as well as forecasts from a numerical weather prediction (NWP) model ensemble. In that EP postprocessor (see also Roebber 2015a), forecast algorithms evolved based on a set of rules applied to a defined mathematical structure. The specific coefficients, operators, and variables of that structure changed over subsequent iterations (hereafter, generations), and the most optimal solutions were retained based on performance on a validation dataset. The ecosystem model held constant the number of forecast algorithms per generation by introducing new algorithms and removing existing algorithms at the same fixed rate, based on sorting using a root-mean-square error (RMSE) performance criterion.

We note that work in EP has been ongoing since the 1960s (e.g., Fogel 1999) and while the conceptual basis of this approach remains the same—using the principles of

Corresponding author: Paul J. Roebber, roebber@uwm.edu

evolution to produce successful algorithms, where success is measurable by a specified metric—the applications of this concept developed in Roebber (2018) (and references therein) are somewhat unique. In the present study, our interest is in developing the ecosystem model beyond the static approach used in Roebber (2018). Although the static ecosystem model was successful, an EP method that takes advantage of selective pressure in a dynamic, coevolving ecosystem might produce better *and* more diverse solutions, simultaneously furthering the twin goals of improving overall skill and reducing ensemble underdispersion. In this paper, we demonstrate this improved deterministic and probabilistic performance for temperature and also show skill for classifying convective occurrence.

There is a twofold rationale for expecting the new approach to be effective. First, coevolution (Ehrlich and Raven 1964), the process by which two or more species exert selective evolutionary pressure on each other, such as in predator–prey behavior, can lead to a dynamic in which the predator evolves more effective methods for capturing the prey, and the prey in response develops additional evasive abilities. Similarly, in a computational environment, numerically better solutions can be produced in this way, to the degree that the simulated species' evolution can be closely linked to the forecast algorithms' performance. Second, the predator–prey dynamic often leads to spatial clustering of individuals both in nature (e.g., Wilson 1975; Partridge 1982) and in computer simulations [e.g., Dewdney 1984; Olsen and Fraczkowski 2015; see also Couzin and Krause (2003) for an extensive review of both observations and simulations]. Roebber (2015a) used a prescribed method for creating such “evolutionary niches,” with the goal of introducing additional diversity in the ensemble solutions. The complex emergent behavior described here, however, can be produced from a small set of rules describing feeding, movement, and reproduction, and it can be accomplished while simultaneously optimizing solution performance.

To demonstrate these improvements and to facilitate direct comparison of the methods, the predator–prey dynamic is applied within EP to the same spatial temperature data studied by Roebber (2018). Also, to demonstrate its effectiveness for a particularly challenging forecast problem, the method is applied to convection occurrence using the AutoNowCaster data from Ba et al. (2017, hereafter BXCS).

The data used in this study are described in section 2. The new EP methodology is summarized in section 3 (with more extensive details provided in an appendix), while section 4 presents the results from its application

to the aforementioned forecast problems. A summary and discussion are presented in section 5.

2. Data

a. 72-h, 2-m temperature forecasts

These data are identical to those in Roebber (2018) but are summarized here for completeness. The data span the period 1 January 1985–14 May 2011 and lie on a $1^\circ \times 1^\circ$ latitude–longitude grid that spans 24° – 53° N and 125° – 66° W (i.e., the CONUS and southern Canada).

We define our reference temperature forecasts to be the 11-member Reforecast Version 2 (Hamill et al. 2013; hereafter RFv2) 72-h, 2-m temperature forecasts issued at 2100 UTC. For the EP methodology, we make use of the following data at each grid point:

- 1) the minimum of the reference 2-m temperature forecasts;
- 2) the 20th percentile of the reference 2-m temperature forecasts;
- 3) the median of the reference 2-m temperature forecasts;
- 4) the 80th percentile of the reference 2-m temperature forecasts;
- 5) the maximum of the reference 2-m temperature forecasts;
- 6) the RFv2 control member's corresponding 72-h forecast of 850-hPa temperature;
- 7) the RFv2 control member's corresponding 72-h forecast of cloud cover;
- 8) the RFv2 control member's corresponding 72-h forecast of precipitable water;
- 9) the RFv2 control member's corresponding 72-h forecast of 10-m wind speed;
- 10) the RFv2 control member's corresponding 72-h forecast of snow cover in excess of 1 in.;
- 11) an analog 72-h, 2-m temperature forecast (computed as in Roebber 2018, following the concept outlined in Delle Monache et al. 2011);
- 12) the climatological 2-m temperature at 2100 UTC (computed as in Roebber 2018); and
- 13) the cosine of the solar zenith angle at local noon (computed as in Roebber 2018).

All 72-h, 2-m temperature forecasts are verified using the corresponding RFv2 analysis at the valid time.

The data are split in the same way as in Roebber (2018) with 4383 days (1 January 1985–31 December 1996) for training, 2922 days (1 January 1997–31 December 2004) for validation, and 2325 days (1 January 2005–14 May 2011) for testing. The training data are used to develop the EP solutions, while the validation data are used to determine which of these solutions to

FIG. 1. Logic map for the predator–prey ecosystem model. Diamond symbols represent ecosystem actions, while the colors represent increases (green) and decreases (red) to the algorithm population. See text and [appendix](#) for details.

retain. The test data are used to evaluate performance; all temperature results reported in this paper are based on the test data and are directly comparable to the results in [Roebber \(2018\)](#).

b. Convection occurrence nowcasts

The AutoNowCaster (ANC; [Mueller et al. 2003](#); [Lakshmanan et al. 2012](#)) generates 60-min nowcasts of convective likelihood (CL). As described in [BXCS](#), these data span the period 11 June 2012–30 September 2012 (from 1400 to 2359 UTC each day) and lie on a $0.02^\circ \times 0.02^\circ$ latitude–longitude grid that spans 31° – 45° N and 94° – 71° W (i.e., most of the United States east of the Mississippi River). The data are available every 5–6 min.

Based on [BXCS](#), we define an ANC convection occurrence nowcast to be $CL \geq 0.6$. In [BXCS](#), the predictors used by ANC to produce the 60-min nowcasts of CL were (i) the maximum convective available potential energy (CAPE) from 900 to 700 hPa, (ii) the mean convective inhibition from 975 to 900 hPa, (iii) the likelihood of a frontal zone, (iv) the average relative humidity from 875 to 625 hPa, (v) the vertical instability from 1000 to 700 hPa, (vi) the vertical velocity at 700 hPa, (vii) surface mass convergence, (viii) the lifted index, (ix) the rate of change of 10.7 micron GOES imagery, (x) areas classified as being free of clouds, and (xi) areas classified as having cumulus and/or cumulus congestus clouds. For the EP

methodology, we make use of the following data at each grid point:

- 1) the maximum CAPE from 900 to 700 hPa (ANC likelihood field);
- 2) the likelihood of a frontal zone (ANC likelihood field);
- 3) the average relative humidity from 875 to 625 hPa (ANC likelihood field);
- 4) surface mass convergence (ANC likelihood field);
- 5) ANC's 60-min nowcast of CL;
- 6) the sine of the local hour for the forecast initial time (i.e., 1 h prior to the valid time); and
- 7) the cosine of the solar zenith angle at local noon (computed as in [Roebber 2018](#)).

This more limited set of predictors is motivated by correlation analysis of ANC's predictors, physical reasoning regarding variables connected to convective processes, and the desire to limit the number of possible inputs for more efficient and effective exploration of the input space, given limits to the size of the training sample.

[BXCS](#) verified the CL nowcasts using data derived from WSR-88D reflectivity volumes. We do the same and, as in [BXCS](#), we employ spatial relaxation using a neighborhood size parameter N , yielding a $(2N + 1)$ by $(2N + 1)$ square neighborhood of points with the given forecast grid point at the center of the square. We set $N = 12$, corresponding to a spatial area of

FIG. 2. Evolution of predator and prey populations in the convection occurrence training. (bottom) Shown in red (blue) are the number of predators (prey) as a function of generation, along with the CSI for the validation data (dotted line). (top) Shown in solid red (blue) are the number of predator (prey) births, in blue dashed the number of prey deaths through predator capture, and in dotted red (blue) the number of predator (prey) deaths through starvation and aging.

approximately 50 km^2 . (See [BXCS](#) for a detailed description). However, unlike in [BXCS](#), we do not use overlapping 50 km^2 areas for verification (see their Fig. 3). Rather, we verify only nonoverlapping 50 km^2 areas.

Across all domain grid points and dates, the frequency of convective occurrence is approximately 14% for the sample period of 11 June–30 September 2012. Owing to variability in the frequency of observed convective occurrence during this period, however, we further split the data as follows. For any sequence of 4 consecutive days, the first two are initially set as training cases, the third as validation, and the fourth as testing, resulting in 50% training cases, 25% validation, and 25% testing. Experiments using logistic regression as a baseline comparison against EP (see below) showed that the former performed poorly owing to imbalance in the number of events compared to null cases, so a further stratification for training was performed except as noted below. Specifically, for training cases, we keep all observed events but randomly filter out 86% of the null events, thus approximately balancing the frequency of observed and null

events. No further changes are made to the validation and testing samples.

3. EP methodology

EP is a computational method in which the principles of evolution are used to devise solutions to a well-defined forecast problem. The conceptual series of steps required to produce these solutions are as follows:

- 1) Randomly initialize a population of forecast algorithms.
- 2) For both the training data and the validation data, evaluate each algorithm from that population based on a defined performance metric.
- 3) Remove the poorest performing algorithms, thus creating “ecosystem space” for new algorithms.
- 4) Based on the remaining algorithms’ performance, produce new algorithms, and introduce reproductive mutations to allow for potentially useful innovations.

Steps 2–4 are repeated through some number of iterations (hereafter, generations) until a stopping criterion is satisfied.

FIG. 3. Red (blue) symbols show the location of at least one predator (prey) algorithm in the 100×100 domain at (top left) generation 1, (top right) generation 50, (bottom left) generation 100, and (bottom right) generation 150 for the convection occurrence training.

As noted above, in this paper we introduce an ecosystem model with attendant predator–prey dynamics to further drive the evolutionary process. The details of this model are provided in the [appendix](#), while a conceptual overview is given here. The algorithms are evolved to perform their forecast task (here, 72-h temperature or 1-h-ahead convective occurrence) on an “ecosystem domain” that has no geographic relationship to the forecasts in question—the domain is provided simply to allow the predator–prey dynamic to unfold. The specific sequence of steps for this process are summarized in [Fig. 1](#): a large, random population of forecast algorithms are initialized, at a 3:1 ratio of prey to predators. These algorithms are evaluated on the training and validation datasets, using root-mean-square

error (RMSE) for temperature and critical success index (CSI)¹ for convection occurrence classification. A sorted, top-100 list is maintained, such that if any of these algorithms performance on the validation dataset is superior to an algorithm on that list, it replaces that poorer-performing algorithm. Next, the prey algorithms seek “food” by attempting to move to ecosystem grid points where all the variables used in its logic are available (variables are nonuniformly distributed across the ecosystem domain), while avoiding locations

¹ Critical success index is defined by three of the four elements of the 2×2 contingency table (i.e., correct null events are excluded), as follows: $CSI = [\text{hits}/(\text{hits} + \text{misses} + \text{false positives})]$.

FIG. 4. Boxplots of the improvement in (leftmost three plots) RMSE (RMS.Sk) and (rightmost two plots) RPS (RPS.Sk) relative to the bias-corrected RFSv2 ensemble for the test data for the ANN (RMS.SkANN), standard EP (RMS.SkEP), coevolutionary EP (RMS.SkCO), standard EP (RPS.SkEP), and coevolutionary EP (RPS.SkCO), for all temperature grid points.

inhabited by predator algorithms. Once this step is complete, the predator algorithms seek the prey by moving to ecosystem grid points where the most prey algorithms are located (and if it does so, then it consumes one prey algorithm and that algorithm is eliminated). All predator and prey movements are restricted to the “neighborhood domain” of a 3×3 box centered on the location of the algorithm in question, and become more effective (i.e., move with the above logic rather than randomly) as their evaluation metrics improve. In addition to predation, prey algorithms may be removed through aging and starvation; predator algorithms are also removed through aging and starvation. These removals provide ecosystem space for new algorithms to develop, and the top-100 list ensures that well performing algorithms that might otherwise have been lost are maintained in the record unless they are superseded. At this stage, all surviving algorithms that have fed reproduce a clone of itself, which then may also mutate (thus introducing innovation).

The process then loops back to the evaluation stage for a new generation—these cycles are continued for a fixed but sufficiently large number of generations to reasonably ensure that the practicably attainable skill

has been captured (in practice, this number turned out to be 70 for temperature and 200 for convection occurrence; note that by maintaining the top-100 list, we do not need to apply specific convergence criteria as long as the number of generations is appropriately large).

Although a number of choices of parameter settings need to be made, in practice, the relevant consideration is simply that populations enter into a form of oscillatory coexistence rather than collapse, the behaviors for such systems that have long been studied in the field of ecosystem dynamics (Lotka 1925; Volterra 1931; Murray 2002; Sprott 2008). For the systems under consideration here, it turns out that obtaining this coexistence is relatively insensitive to a range of parameter choices, although some care was still needed in the case of the temperature algorithms for the entire grid (see the appendix).

In the case of the temperature forecasts, once the EP training has been completed, two further postprocessing steps are employed. First, we use weighted decay bias correction (Cui et al. 2012) as outlined in Roebber (2018). Next, we rank the 100 top-performing algorithms by RMSE on the validation data. We search for the top 5 of these that satisfy a root-mean-square

FIG. 5. The spatial distribution of RPS improvements relative to the bias-corrected RFSv2 ensemble for the test data for the coevolutionary EP at all temperature grid points.

difference (RMSD) criterion in which the RMSD between the candidate algorithm and all others in the top 5 list thus far must exceed a threshold relative to the average RMSD of all algorithms. In this study, we require that this threshold be 0.05 of the average RMSD, which is small but still sufficient to ensure some diversity between the selected algorithms. Bayesian Model Combination (Monteith et al. 2011; Roebber 2018 and references therein) is then applied to this set of 5 algorithms (using 4 raw weights requiring 4^5 combination evaluations) to produce final deterministic and probabilistic forecasts (under the assumption that each individual member of the weighted forecast is normally distributed) for temperature.

This postprocessing is not a requirement of the technique, but Roebber (2018) showed that bias correction and BMC provides further improvement in the context of temperature data. We perform these corrections here to maintain consistency with the prior results and to maximize skill. We note that postprocessing experiments with the convective nowcasting do not add skill and are not pursued in this paper.

To provide further comparative baselines to the EP method, we train (using the identical training and validation data and identical inputs) a multilayer perceptron (MLP) artificial neural network (ANN) for temperature, and logistic regression for convection occurrence. Based on validation data results, the best performing ANN was a single hidden layer

MLP with 8 hidden nodes, and is the basis for the temperature comparisons in section 4. We present results from two logistic regression equations: one where the training data were not filtered to balance the number of convection occurrence and null events, and one where this filtering was accomplished as noted above. Likewise, we trained EP for convection occurrence using these unbalanced and balanced datasets for direct comparison.

4. Results

The procedure outlined in section 3 results in classic, oscillatory predator–prey dynamics (Fig. 2; see section 3 and Hofbauer and Sigmund 1998), with lagged predator–prey population cycles forced by the availability of food and the consumption of prey. As anticipated, clustering of predator and prey algorithms emerges over time (Fig. 3), and the overall effect of the coevolution is to improve both deterministic (RMSE) and probabilistic [ranked probability score (RPS)] performance for temperature [coevolution RMSE of 2.95°F compared to 3.55°F for bias-corrected RFv2 and 3.24°F for the standard EP; see Fig. 4 for gridpoint improvements in RMSE relative to the RFSv2 for the ANN, the standard EP, and coevolutionary EP; also ranked probability score (RPS) improvements for the probabilistic forecasts; and Fig. 5 for the spatial distribution of RPS improvements relative to the RFSv2].

FIG. 6. Rank histograms for the bias-corrected (left) RFSv2 ensemble, (middle) standard EP, and (right) coevolutionary EP for the test data for all temperature grid points. Bins are based on the sorted list of forecasts, where bin 1 is for observations below the lowest forecast value, and so on, up to bin 6, which is for observations above the highest forecast value (note that for the 11-member RFv2, we have combined bins to yield the same number as for the 5-member EP ensembles).

It is notable that this deterministic performance is considerably better than we were able to obtain using either analog forecasts or the ANN (RMSE of 4.89°F and 3.37°F, respectively, after bias correction). As shown by the rank histogram (Fig. 6), the substantial improvement in reliability compared to the RFSv2 produced by the standard EP is increased further using the coevolutionary method described in this paper.

One way this superior performance can be achieved is through the conditional algorithm structure [see the appendix; e.g., as shown in Roebber (2015a) for snow on the ground]. In the present study, although 55% of the grid points do not have snow on the ground reported at any time over the training time period, snow cover is the most frequently invoked variable in the conditional portion of the full set of algorithms. A nonsystematic survey of the algorithms makes it clear that snow cover is often used as a correction factor for an apparent cold bias in the model forecasts – a bias that is confirmed in the training, validation and testing datasets, where the mean model forecast error (forecast minus observed) is 0.0°F when snow cover is not present and -1.5° , -1.6° , and -1.3° F when it is present. This represents just one way in which the conditional structure can positively affect performance—quite often these conditionals invoke a complex combination of interacting variables in the “result” portion of the algorithm, which can help to

further refine the calculation when the broad conditional is satisfied.

In section 3, we noted the large imbalance between null events and cases of convection occurrence in the sample. For the unbalanced training data, the coevolutionary EP method performs considerably better than both the ANC and the multiple logistic regression (MLoR), with much higher probability of detection than either, and only slightly increased false alarms relative to the ANC (note that the MLoR produces fewer false alarms owing to its relative inability to predict convection occurrence; Table 1).

Two examples from the test dataset illustrate the relative behaviors of ANC and the EP algorithm obtained from the unbalanced training, the first (1401 UTC

TABLE 1. Performance of the evolutionary program (EP), AutoNowCaster (ANC), and multiple logistic regression (MLoR) on the test data after training on the unbalanced and balanced data (note ANC is not trained here so there is only one entry). Shown are the probability of detection (POD), false alarm rate (FAR), and critical success index (CSI).

Method	POD	FAR	CSI
EP (not balanced)	0.682	0.658	0.295
ANC	0.261	0.639	0.178
MLoR (not balanced)	0.048	0.430	0.046
EP (balanced)	0.624	0.633	0.303
MLoR (balanced)	0.873	0.713	0.276

FIG. 7. Forecasts from the (top) ANC and (bottom) the best-performing coevolutionary EP algorithm for 1401:30 UTC 18 Sep 2012, a date from the test data with widespread convective activity. Red (blue) indicates locations where convection was (was not) occurring and filled symbols indicate that convection was forecast (i.e., filled red circles are hits, unfilled red circles are misses, and filled blue circles are false positives).

18 September 2012; Fig. 7) with widespread and the second (1932 UTC 26 September 2012; Fig. 8) with limited convection. In both instances, relative to ANC, the EP scores more hits with relatively fewer misses but also increased false positives, with few enough of the latter to result in overall increased skill [CSI for EP (ANC) is 0.299 (0.066) for 18 September; CSI for EP (ANC) is 0.192 (0.082) for 26 September].

An important consideration for operational use of guidance is that forecasters are able to understand the basis of those forecasts (i.e., their interpretability). Unfortunately, many machine-learning techniques are

FIG. 8. As in Fig. 7, but for 1932:40 UTC 26 Sep 2012, a date from the test data with limited convection.

sufficiently opaque to preclude such understanding. In contrast, the logic of EP algorithms obtained using the imposed structure here is readily interpretable. For example, the structure of the convection occurrence EP algorithm is such that there is a small baseline sum, which invokes the ANC (formed from an algorithm line that is always executed). The other four algorithm lines produce modifications to this baseline, which are functions primarily of the relative humidity, hour of the day, and frontal zone likelihood, such that when relative humidity is high, the classification of convection occurrence is substantially increased, particularly during midafternoon to early evening hours, or if the ANC likelihood is high, or if a frontal zone is involved. All of these factors are easily understandable by a forecaster, and the structure can further allow such an

expert to integrate their understanding of the uncertainty of these features into the calculation. In other words, if the relative humidity is high but there is some uncertainty in the timing of the frontal passage, it is possible to calculate how much this offset would change the classification. Tools for assessing such sensitivity (which for the temperature algorithm structure could be represented as the contribution of each input to the overall predicted deviation from the baseline RFv2 forecast) are readily constructed.

Interestingly, when training is conducted using a balanced dataset, the EP performance remains relatively unchanged, while the MLoR improves substantially, albeit with considerable overforecast bias (Table 1; Fig. 9). We note, as expected, that the best performance for both the EP and the MLoR occurs with a classification threshold of 0.5 (Fig. 9). Being able to access two or more (counting the ANC, as well as the EP and the MLoR) quasi-independent assessments of convection occurrence provides additional useful information for end-users. A simultaneous plot of the MLoR and EP assessments, stratified by convection occurrence, reveals this differential performance (Fig. 10). Although both the MLoR and EP suffer from false alarms, there is a tendency for them to make correct predictions of event occurrence when both are high, thus increasing confidence. Similarly when both are low, there is a tendency for them to correctly predict null events (i.e., they produce relatively few simultaneous misses). A high MLoR with a low EP, however, represents a cautionary signal since it is not clear whether this will result in an EP miss or a MLoR false alarm—but the distinct behavior of the two approaches would signal forecasters to pay special attention to such a case.

Thus, whether or not the increased complexity of the EP approach documented here is worthwhile for a given problem would appear to come down to several considerations. First, the EP method appears relatively robust to event frequency—this suggests that for somewhat rare events, it may be a better approach than more traditional methods. Second, the ease of applying the EP method in an adaptive mode [see Roebber (2015b) and (2018), and section 5 below] means that after initial training, it will be able to take advantage of future changes in inputs (as may occur with changes to NWP models or with additional observations), an important consideration in operational environments. Third, the relative performance improvements obtainable from the extra effort that EP implementation entails relative to standard techniques appear to be somewhat problem dependent, suggesting a focus on situations where additional

FIG. 9. Performance diagram (Roebber 2009) for the classification of convection occurrence from the best-performing evolutionary EP algorithm (red circles), as determined by CSI on the cross-validation data, the ANC with $CL = 0.6$ (blue circle with “A”), and the multiple logistic regression equation obtained from a rebalanced training sample (black squares). For both the EP and the logistic regression, results are shown for classification thresholds of 0.2, 0.5, and 0.8 (as labeled).

gains are valuable or otherwise important [e.g., as in the case of energy trading for temperature (Teisberg et al. 2005; Roebber 2010)]. Finally, the ability to use EP in combination with traditional measures as distinct assessments of a forecast in question might be extraordinarily valuable in real world situations.

5. Discussion and summary

Because postprocessors are, in the simplest of terms, a mapping of inputs to outputs, a valid mapping can be maintained only where the inputs have not been modified in some substantial way. A common example of a modification in operations is the occurrence of a major NWP model upgrade. Two remedies for postprocessors are available. The first is to freeze the inputs to the postprocessor (as happens when a NWP model has become outdated but is still run to provide those inputs) until such time as a new training dataset can be constructed using the upgraded NWP model. The second is to formulate the postprocessor in such a way as to allow it to adapt to the changes in the inputs. In other words, the mapping itself changes in response to those input changes.

EP is a natural way of accomplishing the second remedy and has been explored by Roebber (2015b, 2018). In each of these studies, a “mixed-mode” approach was

FIG. 10. Density plot stratified by observed convection occurrence [(left) no convection, (right) convection] for the coevolutionary EP algorithm (x axis) against the multiple logistic regression (y axis). The quadrants are labeled based on a 0.5 classification threshold.

employed in which the IF–THEN framework in existence for forecast time T is used to produce the next generation, which supplies the forecast for time $T + 1$. Additionally, the algorithm coefficients are optimized over a fixed number of trials using a sliding window of past observations. The best performer to that point, based on validation data, then provides the forecast for $T + 1$.

Here, we follow a similar, evolving procedure but simply use the current best performer “as is” from the top-performer list without adjusting the coefficients. Thus, while the “regular” training method which forms the basis of the text makes use of the entire training dataset in a block to produce the algorithms, here we use a portion of the training to produce an initial set of algorithms (a “population”), but then allow these populations to continue to evolve in response to blocks of data composed of a moving window of past observations up to but not including the current case. The adaptation occurs through the changes in the overall algorithm population as they are exposed to this moving window of cases, and the top-ranked algorithm in the top-100 list at a given time is used to provide the forecast.

Figure 11 provides a demonstration of the changes to EP performance following the introduction of an artificially improved ANC forecast input. We accomplish this improved input to the algorithms with a “new” ANC, formed by replacing one-fifth of the original ANC value with the actual observed value (which is always zero

or 1). The performance of the new ANC thus remains imperfect, but is better (average CSI over the period increases by 0.090).

As is evident from Fig. 11, the EP is able to incorporate the improved information within 200 generations, which in the context of the approximately 5-min cycle of convection occurrence nowcasts, represents less than one day. For a more typical forecast application, with a frequency of four forecast cycles per day, this implies incorporation of the improved information in less than two months. We note that adaptive techniques are also available, for example, for artificial neural networks, where online learning methods such as stochastic gradient descent can be used (e.g., [Toulis and Airolidi 2017](#)).

In this paper, we have developed an approach to producing evolutionary program forecasts that employs coevolution, a process that both in nature and in digital form can drive competition between species. In the forecast domain here, we have shown that this “arms race” leads to increasing skill in the predator and prey algorithms that make up the forecasts, and this skill is demonstrated for both deterministic and probabilistic forecasts of temperature and convection occurrence. In the former case, we have demonstrated that the coevolution improves the skill that was obtained on these same data using an earlier form of evolutionary programming, and further that it also improves forecast

FIG. 11. Simulation of adaptive coevolutionary EP response to improved input information (introduced at the generation indicated by the vertical line). Shown are the minimum (blue), mean (green), and maximum (red) CSI for the set of 100 best-performing EP algorithms for each generation prior to and after the introduction of the improved ANC input. Also shown is the observed frequency of convection occurrence at each generation (thick dotted line), and the distribution of ANC CSI values for the moving window used for all calculations (set to prior 100 forecast cycles).

reliability as evidenced by reductions in excessive outliers in the rank histogram. For convection occurrence, we have shown that the method improves nowcasts in the test dataset relative to ANC, a nowcasting tool currently employed by the NWS. Further, it can accomplish this without preprocessing the training data to balance events and nonevents, as is often necessary with traditional techniques (e.g., Batista et al. 2004 and references therein). Finally, the alternative view that the EP provides can be used by experts to further assess uncertainty when used in conjunction with a traditional technique such as multiple logistic regression, as was shown here for the convection occurrence problem.

A number of questions related to evolutionary programming training remain to be explored. First, because the predator–prey dynamic can be unstable in certain parameter ranges, leading to collapse of one or both species, implementation of the method in any particular circumstance requires some attention to the details of the “ecosystem model” settings. If, for example, the prey population collapses, this would lead to the collapse of the predator population as well, with the consequence that the “best-performing” list of algorithms would after that time not be updated, perhaps a development of most consequence in the instance where an adaptive mode was employed. If only the predator population collapsed, the prey population and associated forecast algorithms would continue to be updated, but without the coevolutionary pressure, likely leading

to reduced performance compared to what might be obtainable in a more dynamic environment. An interesting theoretical question with practical implications is whether, in a forecast environment in which the inputs themselves are changing (such as through model improvements) and an adaptive system is being employed, these input changes would themselves help to reduce the possibility of population collapse by driving continuous changes in the algorithms. Regardless, in longer running simulations of the convection occurrence training with no adaptation, the populations did not collapse up to 1000 generations, when the simulation was stopped. Given the potential for stability in a multispecies environment, experiments with many more species, such as explored by Sprott (2008), could be undertaken. Such experiments might also extend beyond predator–prey to evaluate competitive and cooperative coevolution. Finally, it could be of interest to connect the 100×100 ecosystem model domain to the physical domain of the forecast quantity in question, in order to improve the efficiency of training.

Acknowledgments. This research was supported in part by the Cooperative Institute for Research in the Atmosphere (CIRA). We thank Dr. Stephan Smith, Dr. Mamoudou Ba, and Dr. Lingyan Xin of the National Weather Service’s Meteorological Development Laboratory for their advice and support during the conduct of this work.

APPENDIX

Evolutionary Program Ecosystem Model

EP is a computational method in which the principles of evolution are used to devise solutions to a well-defined forecast problem. The conceptual series of steps required to produce these solutions are as follows:

- 1) Randomly initialize a population of forecast algorithms.
- 2) For both the training data and the validation data, evaluate each algorithm from that population based on a defined performance metric.
- 3) Remove the poorest performing algorithms, thus creating “ecosystem space” for new algorithms.
- 4) Based on the remaining algorithms’ performance, produce new algorithms, and introduce reproductive mutations to allow for potentially useful innovations.

Steps 2–4 are repeated through some number of iterations (hereafter, generations) until a stopping criterion is satisfied.

In discussing the specifics of the process implemented here, we reference the above steps. Before this, however, we note the following four distinguishing characteristics between the two kinds of data used in this study.

- 1) For forecasting temperature, each forecast grid point has its own population of algorithms. For nowcasting convection occurrence, however, because of sample limitations, all forecast grid points share the same population of algorithms.
- 2) For temperature forecasts, there are 13 potential input variables; for convection occurrence nowcasts, there are 7.
- 3) For the temperature data, the performance metric used in step 2 above is RMSE improvement relative to the mean of the reference forecasts (i.e., $\text{RMSE}_{\text{EP}} - \text{RMSE}_{\text{XX}}$). For the convection occurrence data, the performance metric is critical success index (CSI) improvement relative to ANC (i.e., $\text{CSI}_{\text{EP}} - \text{CSI}_{\text{ANC}}$, using $\text{CL} \geq 0.6$).
- 4) With respect to temperature, the algorithms produce adjustments relative to the normalized mean of the reference forecasts.^{A1} With respect to convection occurrence, though, the algorithms produce the input to the logistic function for defining a probability of convection occurrence (see below).

^{A1} The algorithms produce a result in the range 0–1 that is then transformed back to the nonnormalized value based on the observed maximum and minimum of the training dataset for that grid location.

In deciding the details of the coevolutionary EP model, some discussion of the complexities of predator–prey ecosystems is warranted. Most notably, predator–prey systems can be unstable: a predator species may consume all the prey, or it may be unable to find enough prey and then starve. Alternatively, in a food-limited environment, overpopulation of a prey species could likewise lead to its starvation, followed quickly by the collapse of the predator population. The well-known Lotka–Volterra differential equation model (Lotka 1925; Volterra 1931) and its variants (e.g., Murray 2002) simulate the interactions between two species and have been employed for many years to study ecosystem dynamics. These equations can model predator–prey dynamics as well as species competition and cooperation through the assignment of the signs of model parameters. Sprott (2008) finds four equilibrium solutions: both species collapsing, one or other of the two species collapsing, and species coexistence. For the case of the predator–prey model, however, the coexisting solution, which consists of out-of-phase, oscillating populations, is not stable. In nature, given the availability of more than one prey species, so-called prey switching can occur (Murdoch 1969), where a predator preferentially consumes the more numerous prey, with a potentially stabilizing effect on the overall ecosystem (Oaten and Murdoch 1975; Shai and Ray 2010). Prey switching, however, does not guarantee stability. Consider the case where predation is not sufficient to reduce the numbers of an abundant prey species, and the predator population itself remains high, a circumstance that could simply lead to the elimination of a less abundant prey species. In the Lotka–Volterra framework, generalizing to N species does not help, as this leads to $2N$ equilibria, of which only one represents coexistence and is unlikely to be stable (Sprott 2008).

Spatial heterogeneity in the environment, which is a characteristic of agent-based models, may provide temporary refuge for prey, allowing for their numbers to increase when predation pressure declines under some circumstances (see section 4 above for an example.) Dewdney (1984) employed an agent-based model (specifically, a two species, discrete space and time ecosystem model) and found that while population collapses were ubiquitous, it was possible to find stable solutions in this environment. Using the aforementioned Lotka–Volterra variant generalized to N species and applied on a spatial domain, Sprott (2008) also found that under certain conditions, stable oscillatory solutions existed and that these temporal fluctuations correlated with spatial heterogeneity.

Finally, the possibility for coexistence may be increased when prey evolve in ways that improve their

FIG. A1. Evolutionary program architecture. Each of the 5 lines (L_1, L_2, L_3, L_4, L_5) in an algorithm follows the IF–THEN conditional logic as shown, where the yellow squares are input variables (normalized in the range $[0, 1]$ by the input data; $V_{2,n}$ can be a variable or unity), the light blue squares can be either $>$ or $\leq$, the dark blue squares are multiplication or addition, and the light green squares are coefficients in the range $[-1, 1]$. The result from each line is computed independently and then summed following one of the two formats indicated at the bottom (see text for details).

ability to survive in the face of predation. For this to be effective, however, the predator population must evolve in sync with the prey. Otherwise, the predators increasingly fail to capture prey and ultimately starve.

In designing the ecosystem model, we have attempted to incorporate many of the above characteristics. It is constructed such that all predators and prey are forecast algorithms that take a five-line form similar to prior work (e.g., Roebber 2018), where each algorithm line L_j is expressed as

$$\text{IF}(V_{1j} O_{Rj} V_{2j}) \text{ THEN } L_j \\ = (C_{1j} V_{3j}) O_{1j} (C_{2j} V_{4j}) O_{2j} (C_{3j} V_{5j}) \text{ ELSE } 0. \quad (\text{A1})$$

Here, $V_{1j}, V_{2j}, V_{3j}, V_{4j}$, and V_{5j} can be any of the 13 (7) input variables for temperature (convection occurrence). As a special case, V_{2j} may simply be set to unity. C_{1j}, C_{2j} , and C_{3j} are real-valued multiplicative constants in the interval $[-1, 1]$. O_{Rj} is one of two relational operators: $\leq$ or $>$. O_{1j} and O_{2j} can be either the addition or multiplication operator. (See Fig. A1.) This notation is such that V_{1j}, V_{2j} , etc. indicate the first input variable in line j , the second input variable in line j , and so on. Each input variable is normalized to $[0, 1]$ based on its minimum and maximum value in the training data, and the final forecast is reconstituted to the non-normalized value based on the minimum and maximum in the training verification data. (Note that the test data are nowhere used except in the final evaluation of performance.)

Experimentation with different models revealed that modeling convection occurrence was less susceptible to population collapses than modeling temperature. We speculate that this may relate to the spatial heterogeneity of the temperature dataset and its preferred predictors (see Roebber 2018). Regardless, in order to

maintain both stability and simplicity, our solution is to incorporate two prey species of the form F_a and F_b (and the possibility of prey switching) in the temperature ecosystem but only one prey species F_a in the convection occurrence ecosystem. These are composed of the algorithm lines L_j using

$$F_a = \sum_{j=1}^5 L_j, \quad (\text{A2})$$

$$F_b = L_1 + L_2 \times L_3 + L_4 \times L_5. \quad (\text{A3})$$

In both ecosystems, we employ one predator species, also of the form F_a . In addition to the possibility of improving the stability of the training of the temperature forecast algorithms across the full range of grid points, this additional complexity may allow further diversity in the resultant forecast solutions, as is seen in ecological systems and simulations (e.g., Chesson 2000; Shai and Ray 2010).

For step 1 above, predator and prey algorithms are initialized on a 100×100 “ecosystem grid,” and per note 1 above, for the temperature data, every forecast grid point has its own such ecosystem grid, while for the convection occurrence data, the forecast grid points share a single ecosystem grid. Although the ecosystem grid is rectangular, its dimensions wrap around, making its geometric form a torus. Training prey and predator algorithms follows the same general process; we note where they differ.

The number of predator and prey algorithms are initialized at fixed numbers and then allowed to evolve according to the described rules. Specifically, the number of predators (prey) for the temperature simulation is initialized as 5000 (1666) at every grid point (note that the two prey types are randomly populated with equal probability). For the convection forecasts, the number

FIG. A2. Example distribution of variables (which take the form of food for prey in the ecosystem) on the 100×100 ecosystem grid for the infinite food distribution mode. Ecosystem grid locations are typed using a *K*-means cluster analysis. Of 13 possible input variables for temperature, grid locations with green squares have a median of 13 variables; blue dashes have a median of 3 variables; purple and orange circles each have a median of 8 variables but with widely differing types.

of predators (prey) is also initialized as 5000 (1666), but here this is for the entire domain as there is only one ecosystem grid and all grid points are pooled. Variables operators and coefficients are all pulled with equal probability (e.g., 50% probability of multiplication or addition; 50% probability of conditional operator set to greater than or less than or equal to; random uniform coefficients between -1 and $+1$). Any variable can be placed in any position within the algorithm code structure where variables are allowed with equal probability and all variables are pulled with replacement.

A prey algorithm “feeds” at an ecosystem grid point if the variables there exactly match the specific requirements of that algorithm (i.e., every variable in an algorithm’s five lines must be at the location at which it feeds). There are two modes of food for prey algorithms: “inexhaustible” and “exhaustible.” The inexhaustible mode is used for temperature. In this mode, at the beginning of a training run, each ecosystem grid point is

randomly assigned a set of variables. A prey algorithm feeds when its needed variables are present and match its requirements, and those variables are never exhausted. Figure A2 provides an example of the distribution of the variables in one forecast grid point of the temperature forecast problem (i.e., inexhaustible food). Food-rich sites (colored green) can support any prey algorithm, whereas more sparse locations (blue dashes) only support limited numbers and very specific forms of prey. The exhaustible mode is used for convection occurrence. In this mode, if a prey algorithm feeds at a grid point, the amount of each needed variable there is decremented by one unit. The input variables “regrow” at the rate of one unit per generation to a maximum of 5 units, but they can be exhausted at a grid point if a swarm of prey is feeding. In exhaustible mode, feeding drives variable scarcity, so that all ecosystem grid points have all variables available when those variables have not been depleted through prey feeding.

Although both food modes were tested for each of the two forecast problems, we show results only for the inexhaustible mode for temperature and the exhaustible mode for convection occurrence. As stated previously, although multiple species decrease the odds of population collapse, such an outcome is not guaranteed, and experience showed that the temperature problem across the full forecast grid was sensitive to the mode of food while convection occurrence was not. In the case of convection occurrence, where only one prey species was used, the exhaustible food mode provides an additional means of introducing variety into the algorithm populations.

A predator algorithm feeds at an ecosystem grid point if there is a prey algorithm at that point. When it feeds, a predator algorithm consumes one and only one prey algorithm. Prey can stack up at a grid point, so if more than one prey algorithms are present, the predator feeds on the first in the stack list. These lists are shuffled through the course of the evolution, so this process does not impose any structure onto the decision. Although this does not guarantee any relationship between prey performance and which prey is chosen, such that there may be instances in which the best-performing algorithm is “trapped” and lost, those genes are likely preserved somewhere in the population and also in the top 100 list.

Prey may either move randomly or employ a food-seeking/predator-avoiding strategy. The latter causes a prey algorithm to seek the most suitable food location (based on the variable requirements of its algorithm) within the 3×3 ecosystem grid area centered on its current position *unless* there is a predator at that preferred location, in which case it will seek to avoid that site, regardless of the availability of food. The probability α of employing the strategy is defined by a relative performance metric, scaled according to the logistic $2P$ curve and limited to $[0.25, 1]$:

$$\alpha = \text{MAX} \left[0.25, \frac{1}{1 + e^{-\lambda \times (\text{metric}' - 0.0294)}} \right], \quad (\text{A4})$$

where metric' is the relative performance metric as defined in note (A3) above, and λ is a scaling parameter. For temperature, λ is set to 36.2275, whereas for convection occurrence, λ is set to 18.1138. (These values are arrived at empirically in order to provide reasonable performance curves.) As can be seen from Fig. A3, this probability rises rapidly with relative improvement, ensuring that better performing algorithms are more likely to survive to the next generation. For temperature, α increases from a baseline of 0.25 to above 0.90 with a 10% improvement in RMSE relative to the mean

FIG. A3. The probability, α , of a predator or prey algorithm employing strategy to seek prey (dashed line) or seek food/avoid predators (thick solid line), respectively, as a function of relative performance [for temperature (convection occurrence), fractional improvement in EP RMSE relative to the ensemble mean RMSE ($\text{CSI}_{\text{EP}} - \text{CSI}_{\text{ANC}}$ with $\text{CL} = 0.6$)]. The two thin solid lines are modifications to (A4) used in sensitivity tests. See text for details.

of the reference forecasts. For convection occurrence, α increases to nearly 0.70 with a 0.10 increase in CSI relative to ANC. Although α is an important ecosystem parameter and influences several additional species behaviors (see below), sensitivity tests with the convection occurrence model using both more and less rapid strategic improvements with relative performance (the latter being the two additional curves depicted in Fig. A3) resulted in only minor variations in the model outcomes. For example, none of these changes resulted in population collapse, and the best performing algorithms showed variations in CSI between runs no greater than 2%.

Similarly, predator algorithms may either move randomly or employ a food-seeking strategy. The predator strategy, also employed with probability α , is to seek within the 3×3 ecosystem grid area centered on its current position for that grid point where the maximum number of prey algorithms is. Thus, as with prey, better predator algorithms are more likely to survive to the next generation. This establishes an “arms race” between prey and predator algorithms, where prey seek to avoid predators and predators seek to find prey, with the success in doing so tied to forecast performance.

Feeding is critical for survival. If prey algorithms do not feed within a certain number of generations, then they die of hunger with probability β :

$$\beta = c(1 - \alpha), \quad (\text{A5})$$

where c is set to 0.125 for temperature and 0.250 for convection occurrence. For temperature (convection occurrence), prey algorithms die of hunger with probability β if they do not feed within 5 generations (1 generation).

Similarly, predator algorithms can also die from hunger, but unlike prey algorithms, predator algorithms store one “food unit” for every prey consumed. Only reproduction, which requires two food units, depletes this store. If a predator’s store is empty, then it also dies of hunger with probability β , where c is set to 0.200 for temperature and 0.005 for convection occurrence.

Prey algorithms can also die of age with probability γ :

$$\gamma = d(1 - \alpha), \quad (\text{A6})$$

where d is set to 0.1 for temperature and 1.0 for convection occurrence. For temperature (convection occurrence), prey algorithms die of age with probability γ after 6 generations (5 generations). Additionally, all prey algorithms in the convection occurrence ecosystem model die of age after 9 generations. Given the relative difficulty of the convective forecast problem, this modification more vigorously drives the solutions toward innovation. Sensitivity tests of these aging choices in the convection occurrence model, ranging from as little as 4/7 to as long as 10/19 generations (compared to the base of 5/9), again revealed no population collapses and only minor variation in final training outcomes.

Similarly, predator algorithms can also die of age with probability γ , where d is set to 0.3 for temperature and 1.0 for convection occurrence. For both temperature and convection occurrence, predator algorithms die of age with probability γ after 8 generations.

Per step 3 above, the removal of algorithms creates room for new algorithms. However, this can result in the undesirable loss of some well-performing algorithms. This problem is handled by maintaining a list of the 100 best performing algorithms—split between the top-50 prey algorithms and the top-50 predator algorithms—based on their performance in any generation on the validation data. At any time in the training process, if algorithm X performs better than an algorithm in the list, the worst performing member in the list is replaced with algorithm X . In Roebber (2015a), fatal disease created room for new algorithms, and a performance criterion provided selective pressure. Here, the ecosystem dynamics of predation, hunger, and aging creates space, and coevolution provides the selective pressure. As was shown in the main text, this methodology

outperforms that used in Roebber (2018) on both deterministic and probabilistic measures for the same training and test dataset.

In an ecosystem grid, there is an upper limit to the number of prey and predator algorithms that may be present. For temperature, these limits are 5000 prey algorithms and 5000 predator algorithms. For convection occurrence, these limits are doubled. Per step 4 above, when space for new algorithms exists (i.e., either of these upper limits has not been reached), new algorithms are produced through reproduction.

A prey algorithm reproduces if it feeds in the current generation. In this instance, a new prey algorithm that is a clone of its parent is placed randomly within the 3×3 ecosystem grid area centered on the parent’s current position. With probability $(1 - \alpha)$, one element of the clone may have a mutation, with a two-thirds chance that the mutation will be to one of the clone’s variables or operators and a one-third chance that it will be to one of the clone’s coefficients. Predator algorithms reproduce in the same way, although doing so also depletes their aforementioned store of food units.

Finally, using the defined performance metric in step 2, each prey algorithm compares its performance against the performance of all other prey algorithms within the 3×3 ecosystem grid area centered on that algorithm’s location, and it copies one of the five algorithm lines of the best performer in that area. In so doing, a form of learning takes place through horizontal gene transfer, which is distinct from the vertical gene transfer of the reproductive process. This has an effect similar to mutation in that it proves most effective at improving forecast performance during the earliest generations (e.g., Roebber 2015a). Unlike mutation, however, it does not introduce innovations but is simply another form of recombination. Thus, its overall contribution to forecast performance over many training generations is likely to be relatively minor, and if the desire is to limit the complexity of the overall training procedure, this variation could be left out of the ecosystem model. For the purposes of this study, however, we employ it in the form described above. As with prey, predators also “learn” from their most accomplished neighbors.

Table A1 summarizes the attributes used to model the coevolutionary ecosystem. See Olsen and Fraczkowski (2015) for the inspiration for some of these ideas.

Training begins by creating an initial population of 1667 predator algorithms and 5000 prey algorithms. In the temperature ecosystem, the prey algorithms have equal chances of taking either form F_a or F_b . All the algorithms’ variables, operators, and coefficients are chosen at random. In each generation, the performance metric of every algorithm is evaluated and used

TABLE A1. Attributes used to model prey and predator behaviors on the 100×100 ecosystem grid.

Attribute	Prey algorithms	Predator algorithms
Location	Initially randomly dispersed on 100×100 grid	Initially randomly dispersed on 100×100 grid
Structure	f (IF-THEN, VARS) $L_1 + L_2 + L_3 + L_4 + L_5$ (CO, T) or $L_1 + L_2 \times L_3 + L_4 \times L_5$ (T)	f (IF-THEN, VARS) $L_1 + L_2 + L_3 + L_4 + L_5$ (CO, T)
Food	T : variables (from 7–13 of list plus constant), randomly distributed on 100×100 grid. CO: all 7 variables of list plus constant, depleted and accumulated up to maximum of 5	Prey. Predator accumulates food reserves with each prey consumed and depletes this only through reproduction.
Feeding strategy	Seek food or avoid predator within 3×3 neighborhood with probability $\alpha = f(\text{RMSE for } T, \text{CSI for CO})$	Seek prey at location where maximum prey exist within the 3×3 neighborhood with probability α
Hunger	Prey feeds only if food at grid location contains all variables used by prey algorithm. After 5 (1) or more generations without feeding for T (CO), then prey may die with probability $\beta = f(1 - \alpha)$	If predator is hungry (has no accumulated food reserves) it may die with probability β
Aging	If prey has existed for at least 6 (5) generations for T (CO), then prey may die with probability $\gamma = f(1 - \alpha)$. CO: prey always dies after 9 generations.	If predator has existed for at least 8 generations then predator may die with probability γ . CO: predator always dies after 15 generations.
Reproduction	If prey has fed and the overall prey population on the grid is less than the prey carrying capacity (5000 T ; 10 000 CO), then prey produces one clone of itself within 3×3 neighborhood. The clone may have a mutation with probability $1 - \alpha$.	If predator has at least 2 accumulated food reserves and the overall predator population is less than the predator carrying capacity (5000 T ; 10 000 CO) then predator produces one clone of itself within the 3×3 neighborhood, depleting its food reserves by 2. The clone may have a mutation with probability $1 - \alpha$.
Learning	If there is another prey algorithm within the 3×3 neighborhood which has a higher performance level, then the worse-performing prey “learns” by copying at random one of the 5 lines from the better algorithm.	If there is another predator algorithm within the 3×3 neighborhood which has a higher performance level, then the worse-performing predator “learns” by copying at random one of the 5 lines from the better algorithm.

to update the list of best performing algorithms. They and any remaining algorithms may pass their genetic material to the next generation. Training is stopped after a fixed number of generations, set to a large enough number in order to ensure that the practicably attainable skill has been captured. In testing, monitoring changes in the overall and peak performance of the top-performer list helped establish to a reasonable approximation that 70 generations were sufficient for temperature, while 200 generations were sufficient for convection occurrence, and we use these numbers in this study.

In the case of the temperature forecasts, two further postprocessing steps are employed following the EP training. First, we use weighted decay bias correction (Cui et al. 2012) as outlined in Roebber (2018). Next, we rank the 100 top-performing algorithms by RMSE on the validation data. We search for the top 5 of these that satisfy a root-mean-square difference (RMSD) criterion in which the RMSD between the candidate algorithm and all others in the top 5 list thus far must exceed a threshold relative to the average RMSD of all algorithms. In this study, we require that this threshold be 0.05 of the average RMSD, which is small but still sufficient to ensure some diversity between the selected algorithms. Bayesian Model Combination (Monteith

et al. 2011; Roebber 2018 and references therein) is then applied to this set of 5 algorithms (using 4 raw weights requiring 4^5 combination evaluations) to produce final deterministic and probabilistic forecasts (under the assumption that each individual member of the weighted forecast is normally distributed) for temperature.

This postprocessing is not a requirement of the technique, but Roebber (2018) showed that bias correction and BMC provides further improvement in the context of temperature data. We perform these corrections here to maintain consistency with the prior results and to maximize skill. We note that postprocessing experiments with the convective nowcasting do not add skill and are not pursued in this paper (see below).

For convection occurrence, the five lines of an EP algorithm defines an equation X , which is placed in the range 0–1 using the logistic function:

$$p(\text{CO}) = \frac{1}{1 + e^{-X}}. \quad (\text{A7})$$

This value is then used for classification, using a threshold value of 0.5. EP training and validation performance is then evaluated using the CSI.

REFERENCES

- Ba, M., L. Xin, J. Crockett, and S. B. Smith, 2017: Evaluation of NCAR's AutoNowCaster for operational application within the National Weather Service. *Wea. Forecasting*, **32**, 1477–1490, <https://doi.org/10.1175/WAF-D-16-0173.1>.
- Batista, G. E. A. P. A., R. C. Prati, and M. C. Monard, 2004: A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor.*, **6**, 20–29, <https://doi.org/10.1145/1007730.1007735>.
- Chesson, P., 2000: Mechanisms of maintenance of species diversity. *Annu. Rev. Ecol. Syst.*, **31**, 343–366, <https://doi.org/10.1146/annurev.ecolsys.31.1.343>.
- Couzin, I., and J. Krause, 2003: Self-organization and collective behavior in vertebrates. *Adv. Stud. Behav.*, **32**, 1–75, [https://doi.org/10.1016/S0065-3454\(03\)01001-5](https://doi.org/10.1016/S0065-3454(03)01001-5).
- Cui, B., Z. Toth, Y. Zhu, and D. Hou, 2012: Bias correction for global ensemble forecast. *Wea. Forecasting*, **27**, 396–410, <https://doi.org/10.1175/WAF-D-11-00011.1>.
- Delle Monache, L., T. Nipen, Y. Liu, G. Roux, and R. Stull, 2011: Kalman filter and analog schemes to postprocess numerical weather predictions. *Mon. Wea. Rev.*, **139**, 3554–3570, <https://doi.org/10.1175/2011MWR3653.1>.
- Dewdney, A. K., 1984: Computer recreations. *Sci. Amer.*, **252**, 14–24, <https://doi.org/10.1038/scientificamerican0584-14>.
- Ehrlich, P. R., and P. H. Raven, 1964: Butterflies and plants: A study in coevolution. *Evolution*, **18**, 586–608, <https://doi.org/10.1111/j.1558-5646.1964.tb01674.x>.
- Fogel, L. J., 1999: *Intelligence through Simulated Evolution: Forty Years of Evolutionary Programming*. John Wiley, 162 pp.
- Hamill, T. M., G. T. Bates, J. S. Whitaker, D. R. Murray, M. Fiorino, T. J. Galarneau Jr., Y. Zhu, and W. Lapenta, 2013: NOAA's second-generation global medium-range ensemble reforecast dataset. *Bull. Amer. Meteor. Soc.*, **94**, 1553–1565, <https://doi.org/10.1175/BAMS-D-12-00014.1>.
- Hofbauer, J., and K. Sigmund, 1998: *Evolutionary Games and Population Dynamics*. Cambridge University Press, 323 pp.
- Lakshmanan, V., J. Crockett, K. Sperow, M. Ba, and L. Xin, 2012: Tuning AutoNowcaster automatically. *Wea. Forecasting*, **27**, 1568–1579, <https://doi.org/10.1175/WAF-D-11-00141.1>.
- Lotka, A. J., 1925: *Elements of Physical Biology*. Williams and Wilkins, 495 pp.
- Monteith, K., J. Carroll, K. Seppi, and T. Martinez, 2011: Turning Bayesian Model Averaging into Bayesian Model Combination. *Proc. Int. Joint Conf. on Neural Networks (IJCNN'11)*, San Jose, CA, IEEE, 2657–2663.
- Mueller, C., T. Saxen, R. Roberts, J. Wilson, T. Betancourt, S. Dettling, N. Oien, and H. Yee, 2003: NCAR Auto-Nowcast system. *Wea. Forecasting*, **18**, 545–561, [https://doi.org/10.1175/1520-0434\(2003\)018<0545:NAS>2.0.CO;2](https://doi.org/10.1175/1520-0434(2003)018<0545:NAS>2.0.CO;2).
- Murdoch, W. W., 1969: Switching in generalist predators: experiments on prey specificity and stability of prey populations. *Ecol. Monogr.*, **39**, 335–354, <https://doi.org/10.2307/1942352>.
- Murray, J. D., 2002: *Mathematical Biology. I. An Introduction*. Springer-Verlag, 553 pp.
- Oaten, A., and W. W. Murdoch, 1975: Switching, functional response, and stability in predator-prey systems. *Amer. Nat.*, **109**, 299–318, <https://doi.org/10.1086/282999>.
- Olsen, M. M., and R. Fraczkowski, 2015: Co-evolution in predator prey through reinforcement learning. *J. Comput. Sci.*, **9**, 118–124, <https://doi.org/10.1016/j.jocs.2015.04.011>.
- Partridge, B. L., 1982: The structure and function of fish schools. *Sci. Amer.*, **246**, 114–123, <https://doi.org/10.1038/scientificamerican0682-114>.
- Roebber, P. J., 2009: Visualizing multiple measures of forecast quality. *Wea. Forecasting*, **24**, 601–608, <https://doi.org/10.1175/2008WAF2222159.1>.
- , 2010: Seeking consensus: A new approach. *Mon. Wea. Rev.*, **138**, 4402–4415, <https://doi.org/10.1175/2010MWR3508.1>.
- , 2015a: Evolving ensembles. *Mon. Wea. Rev.*, **143**, 471–490, <https://doi.org/10.1175/MWR-D-14-00058.1>.
- , 2015b: Adaptive evolutionary programming. *Mon. Wea. Rev.*, **143**, 1497–1505, <https://doi.org/10.1175/MWR-D-14-00095.1>.
- , 2018: Using evolutionary programming to add deterministic and probabilistic skill to spatial model forecasts. *Mon. Wea. Rev.*, **146**, 2525–2540, <https://doi.org/10.1175/MWR-D-17-0272.1>.
- Shai, J., and T. S. Ray, 2010: Maintenance of species diversity by predation in the Tierra system. *Proc. 12th Int. Conf. on the Synthesis and Simulation of Living Systems*, Odense, Denmark, ALIFE, 533–540.
- Sprott, J. C., 2008: Predator-prey dynamics for rabbits, trees, and romance. *Unifying Themes in Complex Systems IV (Part II)*, A. A. Minai and Y. Bar-Yam, Eds., Springer, 231–238, https://doi.org/10.1007/978-3-540-73849-7_26.
- Teisberg, T. J., R. F. Weiher, and A. Khotanzad, 2005: The economic value of temperature forecasts in electricity generation. *Bull. Amer. Meteor. Soc.*, **86**, 1765–1771, <https://doi.org/10.1175/BAMS-86-12-1765>.
- Toulis, P., and E. M. Airolidi, 2017: Asymptotic and finite-sample properties of estimators based on stochastic gradients. *Ann. Stat.*, **45**, 1694–1727, <https://doi.org/10.1214/16-AOS1506>.
- Volterra, V., 1931: Variations and fluctuations of the number of individuals in animal species living together. *Animal Ecology*, R. N. Chapman, Ed., McGraw-Hill, 9–21.
- Wilson, E. O., 1975: *Sociobiology*. Harvard University Press, 720 pp.