Neural hierarchical models of ecological populations

Maxwell B. Joseph
Earth Lab
Cooperative Institute for Research in Environmental Sciences
University of Colorado Boulder
Boulder, CO 80303, USA
maxwell.b.joseph@colorado.edu

Abstract

Neural networks are increasingly being used in science to infer hidden dynamics of natural systems from noisy observations, a task typically handled by hierarchical models in ecology. This paper describes a class of hierarchical models parameterized by neural networks: neural hierarchical models. The derivation of such models analogizes the relationship between regression and neural networks. A case study is developed for a neural dynamic occupancy model of North American bird populations, trained on millions of detection/non-detection time series for hundreds of species, providing insights into colonization and extinction at a continental scale. Flexible models are increasingly needed that scale to large data and represent ecological processes. Neural hierarchical models satisfy this need, providing a bridge between deep learning and ecological modeling that combines the function representation power of neural networks with the inferential capacity of hierarchical models.

Introduction

Deep neural networks have proved useful in myriad tasks due their ability to represent complex functions over structured domains (LeCun *et al.* 2015). While ecologists are beginning to use such approaches, e.g., to identify plants and animals in images (Norouzzadeh *et al.* 2018; Fricker *et al.* 2019), there has been relatively little integration of deep neural networks with ecological models.

Ecological processes are difficult to observe. Inference often proceeds by modeling the relationship between imperfect data and latent quantities or processes of interest with hierarchical models (Wikle 2003). For example, occupancy models estimate the presence or absence of a species using imperfect detection data (MacKenzie *et al.* 2002), and "dynamic" occupancy models estimate extinction and colonization dynamics (MacKenzie *et al.* 2003). Population growth provides another example, motivating hierarchical models that link noisy observations to mechanistic models (De Valpine & Hastings 2002). In such models, it is often desirable to account for heterogeneity among sample units (e.g., differences among habitats and survey conditions), to better understand ecological dynamics.

Many hierarchical models in ecology account for heterogeneity among sample units using a linear combination of explanatory variables, despite there often being reasons to expect non-linearity (Lek *et al.* 1996; Austin 2002; Oksanen & Minchin 2002). A variety of solutions exist to account for non-linearity. For example, Gaussian processes have been used for species distribution models (Latimer *et al.* 2009; Golding & Purse 2016), in animal movement models (Johnson *et al.* 2008), and in point process models for distance sampling data (Johnson *et al.* 2010; Yuan *et al.* 2017). Generalized additive models also have been used to account for spatial autocorrelation (Miller *et al.* 2013; Webb *et al.* 2014), nonlinear responses to habitat characteristics (Knapp *et al.* 2003; Bled *et al.* 2013), and differential catchability in capture-recapture studies (Zwane & Van der Heijden 2004).

Machine learning provides additional tools for approximating nonlinear functions in hierarchical models. For example, Hutchinson *et al.* (2011) combined a site-occupancy model with an ensemble of decision trees to predict bird occurrence, combining a structured observation and process model from ecology with a flexible random forest model. Similar hybrid approaches could be developed for other classes of ecological models and/or machine learning methods. Neural networks seem particularly worthy of attention, given their success in other domains (LeCun *et al.* 2015).

Neural networks have been used in ecology, but to date have not been integrated into hierarchical models that explicitly distinguish between ecological processes and imperfect observations. For example, neural networks have modeled observed abundances of aquatic organisms, but without distinguishing observed from true abundance (Chon *et al.* 2001; Jeong *et al.* 2001, 2008; Malek *et al.* 2012). Neural networks also have been used to model stock-recruitment and apparent presence/absence data (Manel *et al.* 1999; Chen & Hare 2006; Özesmi *et al.* 2006; Harris 2015; Chen *et al.* 2016), but have not yet been extended to account for imperfect detection, despite increasing recognition of its importance (Guillera-Arroita 2017; Tobler *et al.* 2019). Notably, standard neural networks that classify or predict data are of limited use for ecological applications where imperfect data are used to learn about dynamics of hard to observe systems. As a solution, this paper describes neural hierarchical models that combine the function representation capacity of neural networks with hierarchical models that represent ecological processes.

Related work

Neural networks

Neural networks are function approximators. Linear regression is a special case, where an input vector $\mathbf{x}$ is mapped to a predicted value y:

$$y = \mathbf{w}^T\mathbf{x},$$

where $\mathbf{w}$ is a parameter vector (Fig. 1a). Predicted values are linear combinations of the inputs $\mathbf{x}$ due to the product $\mathbf{w}^T\mathbf{x}$, which restricts the complexity of the function mapping $\mathbf{x}$ to y.

Instead of modeling outputs as linear functions of $\mathbf{x}$, a model can be developed that is a linear function of a nonlinear transformation of $\mathbf{x}$. Nonlinear transformations can be specified via polynomial terms, splines, or another basis expansion (Hefley *et al.* 2017), but neural networks parameterize the transformation via a set of sequential "hidden layers" (Goodfellow *et al.* 2016).

In a neural network with one hidden layer, the first hidden layer maps the length D input $\mathbf{x}$ to a length $D^{(1)}$ vector of "activations" $\mathbf{a}^{(1)} = \mathbf{W}^{(1)}\mathbf{x}$, where $\mathbf{W}^{(1)}$ is a $D^{(1)} \times D$ parameter matrix. The activations are passed to a differentiable nonlinear activation function g to obtain the "hidden units" of the first layer $\mathbf{h}^{(1)} = g(\mathbf{a}^{(1)})$ (Fig. 1b).

The final layer of a neural network maps the a hidden layer to an output. For a neural network with one hidden layer, if the output variable $\mathbf{y}$ is a K dimensional vector, the output unit activations are given by $\mathbf{a}^{(2)} = \mathbf{W}^{(2)}\mathbf{h}^{(1)}$, where $\mathbf{W}^{(2)}$ is a $K \times D^{(1)}$ parameter matrix. The output activation can be written as a composition of functions that transform the inputs x:

$$\mathbf{a}^{(2)} = \mathbf{W}^{(2)}\Big(g(\mathbf{W}^{(1)}\mathbf{x})\Big).$$

Similar to link functions in generalized linear models, outputs can be transformed by an output activation function. For example, if $\mathbf{y}$ is unbounded, the identity function can act as an output activation so that $\mathbf{y} = \mathbf{a}^{(2)}$. Neural networks that predict probabilities typically use a sigmoid (inverse logit) activation function.

Neural networks usually are trained using stochastic gradient based optimization to minimize a loss function, e.g., the negative log likelihood of a Gaussian distribution for a regression task, a Bernoulli distribution for binary classification, or a Poisson distribution for a count model. Partial derivatives of the model parameters are computed with respect to the loss via backpropagation, and parameters are updated to reduce the loss. In practice, these partial derivatives are often computed via automatic

differentiation over a "mini-batch" of samples, which provides a noisy estimate of the gradient (Ruder 2016).

Neural networks are popular both because of their practical successes in a wide variety of applications, and because they possess some desirable theoretical properties. A neural network with suitable activation functions and a single hidden layer containing a finite number of neurons can approximate nearly any continuous function on a compact domain (Cybenko 1989; Hornik 1991). "Deep" neural networks with many sequential hidden layers also act as function approximators (Lu *et al.* 2017). The field of deep learning, which applies such networks, provides a variety of network architectures to account for temporal structure (Hochreiter & Schmidhuber 1997), spatial structure on regular grids (Long *et al.* 2015), or graphs (Niepert *et al.* 2016), sets of unordered irregular points (Li *et al.* 2018), and spatiotemporal data on grids or graphs (Xingjian *et al.* 2015; Jain *et al.* 2016).

The potential for deep learning has been recognized in Earth science (Reichstein *et al.* 2019), the natural sciences (Ching *et al.* 2018; Gazestani & Lewis 2019; Roscher *et al.* 2019), physical sciences (Carleo *et al.* 2019), chemical sciences (Butler *et al.* 2018), and ecology (Christin *et al.* 2018; Desjardins-Proulx *et al.* 2019). For example, models of lake temperature that combine neural networks with loss functions consistent with known physical mechanisms perform better than physical models and neural networks applied alone (Karpatne *et al.* 2017). Similarly, generative adversarial networks with loss functions that encourage mass-balance have expedited electromagnetic calorimeter data generation from the Large Hadron Collider (Paganini *et al.* 2018; Radovic *et al.* 2018). Convolutional neural networks also have been successfully deployed in population genetics to make inferences about introgression, recombination, selection, and population sizes (Flagel *et al.* 2018). There are various ways to combine science knowledge with deep learning. Ba *et al.* (2019) provide a useful taxonomy in the context of physics-based deep learning. In ecology, hierarchical models present an opportunity to build upon existing approaches to derive science-based deep learning methods.

Hierarchical models

Hierarchical models combine a data model, a process model, and a parameter model (Berliner 1996; Wikle 2003). Data models represent the probability distribution of observations conditioned on a process and some parameters, e.g., the probability of capturing a marked animal, given its true state (alive or dead). Process models represent states and their dynamics, conditioned on some parameters. State variables are often incompletely observed, e.g., whether an individual animal is alive or whether a site is occupied. Parameter models represent probability distributions for unknown parameters - priors in a Bayesian framework. In a non-Bayesian setting, parameters are treated as unknowns and estimated from the data, but parameter uncertainty is not represented using probability distributions (Cressie *et al.* 2009).

Neural hierarchical models

Neural hierarchical models are hierarchical models in which the observation, process, or parameter model is parameterized by a neural network. These models are hierarchical (sensu Berliner (1996)) if they distinguish between a modeled process and available data, e.g., between partial differential equations and noisy observations of their solutions (Raissi 2018). "Deep Markov models" - hidden Markov models parameterized by neural networks - provide an example, with successful applications in polyphonic music structure discovery, patient state reconstruction in medical data, and time series forecasting (Krishnan *et al.* 2017; Rangapuram *et al.* 2018). State-space neural networks that use recurrent architectures provide another example dating back two decades (Zamarreño & Vega 1998; Van Lint *et al.* 2002). This class of models inherits the flexibility and scalability of neural networks, along with the inferential power of hierarchical models, but applications in ecology and environmental science are only just beginning to emerge (Wikle 2019).

Construction of such models from existing hierarchical models is straightforward. For example, one can propose neural variants of occupancy models (MacKenzie *et al.* 2002), dynamic occupancy models (MacKenzie *et al.* 2003; Royle & Kéry 2007), N-mixture models (Royle 2004), mark-recapture models (Jolly 1965; Calvert *et al.* 2009), and other hidden Markov models (Patterson *et al.* 2009, 2017; Langrock *et al.* 2012). Output activation functions can be determined from inverse link functions, e.g., sigmoid (inverse logit) activations for probabilities, and loss functions

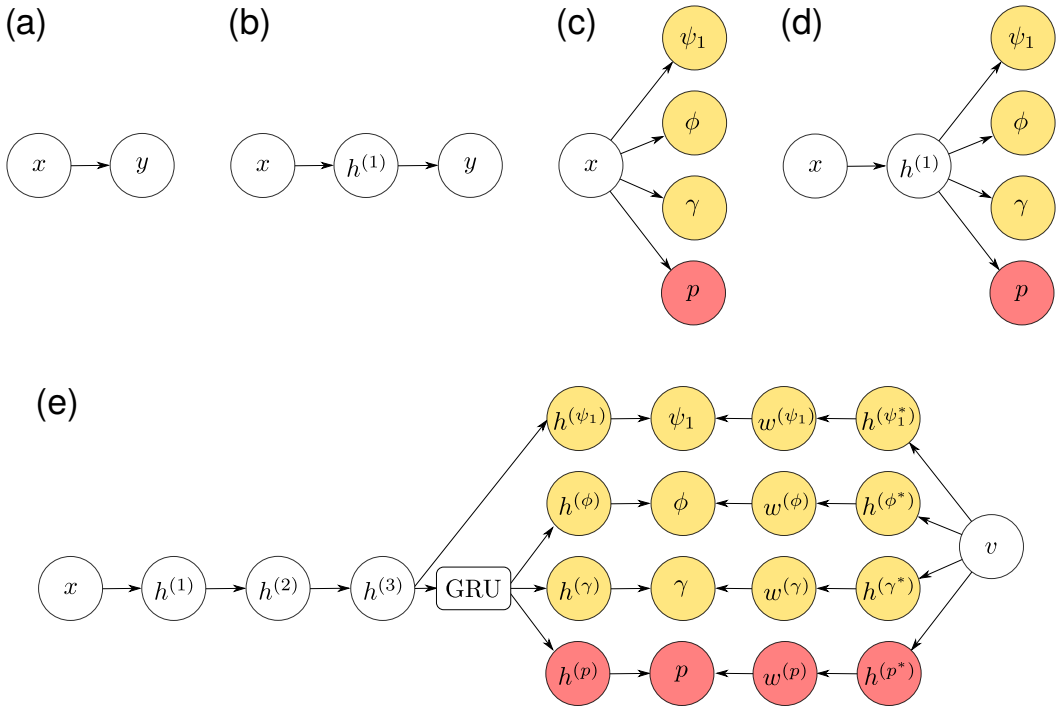

Figure 1: Computation graphs for (a) linear regression, (b) a neural network with one hidden layer, (c) a dynamic occupancy model, (d) a single species neural dynamic occupancy model, and (e) a deep multi-species neural dynamic occupancy model. Yellow and red indicate quantities specific to process and observation components respectively. Inputs are represented by x, and predicted values in panels (a) and (b) by y. Hidden layers are represented by h, with layer-specific superscripts. Outputs include initial occupancy (ψ_1), persistence (ϕ), colonization (γ), and detection (p) probabilities. Latent species embedding vectors are represented by v, and GRU indicates a gated recurrent unit.

can be constructed from the negative log likelihoods (see Appendix S1 in Supporting Information for example model specifications). Specialized neural network architectures that operate on structured data can be readily integrated into such models. For example, Appendix S2 in Supporting Information provides a simulated animal movement case study where aerial imagery is mapped to state transition probabilities of a hidden Markov model with a convolutional neural network. To provide an empirical use case, a neural dynamic occupancy model is developed for extinction and colonization dynamics of North American bird communities.

Case study

The North American Breeding Bird Survey (BBS) is a large-scale annual survey aimed at characterizing trends in roadside bird populations (Link & Sauer 1998; Sauer & Link 2011; Sauer *et al.* 2013; Pardieck *et al.* 2018). Thousands of routes are surveyed once a year during the breeding season. Surveys consist of volunteer observers that stop 50 times at points 800 meters apart on a transect, recording all birds detected within 400 meters for three minutes. Species can be present, but not detected, and may go locally extinct or colonize new routes from year to year, motivating the development of dynamic occupancy models which use imperfect detection data to estimate latent presence or absence states (MacKenzie *et al.* 2003; Royle & Kéry 2007).

This case study uses BBS data from 1997-2018 excluding unidentified or hybrid species, restricting the analysis to surveys meeting the official BBS criteria (Pardieck *et al.* 2018). The resulting data consists of 647 species sampled at 4,540 routes, for a total of 59,384 surveys (not every route is surveyed in each year), 2,937,380 observation history time series, and 38,421,448 detection/non-detection observations.

Process model

A multi-species dynamic occupancy model for spatially referenced routes $s = 1, ..., S$, surveyed in years $t = 1, ...T$, for species $j = 1, ..., J$ aims to estimate colonization and extinction dynamics through time. The true occupancy state $z_{t,s,j} = 1$ if species j is present, and $z_{t,s,j} = 0$ if species j is absent. The model represents $z_{t,s,j}$ as a Bernoulli distributed random variable, where $\Pr(z_{t,s,j} = 1) = \psi_{t,s,j}$. The probability of occurrence on the first timestep is $\psi_{t=1,s,j}$. Subsequent dynamics are determined by probabilities of persistence from time t to $t + 1$ denoted $\phi_{t,s,j}$, and probabilities of colonization from time t to $t + 1$ denoted $\gamma_{t,s,j}$, so that the probability of occurrence in timesteps $t = 2, ..., T$ is (MacKenzie *et al.* 2003):

$$\psi_{t,s,j} = z_{t-1,s,j}\phi_{t-1,s,j} + (1 - z_{t-1,s,j})\gamma_{t-1,s,j}.$$

Observation model

Let $y_{t,s,j}$ represent the number of stops where species j was detected in year t on route s. Conditional on a species being present, it is detected at each stop with probability $p_{t,s,j}$. Assume that there are no false-positive detections, so that if a species is absent, it cannot be detected (but see Royle & Link 2006). With $k = 50$ replicate stops on each transect, the observations can be modeled using a Binomial likelihood for one species-route-year combination:

$$[y_{t,s,j} \mid z_{t,s,j}, p_{t,s,j}] = \text{Binomial}(y_{t,s,j} \mid z_{t,s,j}p_{t,s,j}, k),$$

where square brackets denote the probability function (e.g., in this case, $[y_{t,s,j} \mid z_{t,s,j}, p_{t,s,j}] = \Pr(Y_{t,s,j} = y_{t,s,j} \mid z_{t,s,j}, p_{t,s,j})$ where $Y_{t,s,j}$ is a discrete random variable and $y_{t,s,j}$ is a particular value). The joint likelihood corresponds to the product of these terms for all years, routes, and species (Dorazio *et al.* 2010).

Parameter models

Heterogeneity in parameter values among routes, years, species, and surveys was modeled using three different approaches.

1. **Single-species baseline models** mapped input features to occupancy parameters with a linear combination on the logit scale (MacKenzie *et al.* 2003) (Fig. 1c).
2. **Single-species neural hierarchical models** mapped inputs to occupancy parameters using a neural network with one hidden layer (Fig. 1d).
3. **A multi-species deep neural hierarchical model** was developed to model occupancy dynamics of all species simultaneously (Fig. 1e).

Input features included Environmental Protection Agency (EPA) level one ecoregions, the first eight principal components of the standard 19 WorldClim Bioclimatic variables averaged across the years 1970-2000 (Fick & Hijmans 2017), BBS route spatial coordinates, distance from the coast, elevation, and road density within a 10 km buffer of each route (Meijer *et al.* 2018). The model of detection probabilities additionally included survey-level features including temperature, duration, wind, and air conditions.

The neural hierarchical models were motivated by joint species distribution models in which species load onto a shared set of latent factors (Thorson *et al.* 2015, 2016; Warton *et al.* 2015; Ovaskainen *et al.* 2016; Tikhonov *et al.* 2017), and by recent work on deep neural basis expansions (McDermott & Wikle 2019; Wikle 2019). Analogously, the neural networks combined inputs into a latent vector for each route (the hidden layers), which act as latent factors that are mapped to parameters (Fig. 1d). Because the single species neural hierarchical models were fit separately for each species, latent factors were species-specific. In contrast, latent factors were shared among species in the multi-species model.

The multi-species neural dynamic occupancy model additionally built upon previous work on deep multi-species embedding. Deep multi-species embedding uses vector-valued "entity embeddings" to represent each species (Chen *et al.* 2016; Guo & Berkhahn 2016). These entity embeddings are mappings from categorical data (e.g., a species identity) to continuous numeric vector representations.

Embeddings are used extensively in language models such as word2vec, which maps words to vector spaces (Mikolov *et al.* 2013).

Further, the multi-species model combined species embeddings with encoder-decoder components to estimate occupancy parameters. Encoder-decoder neural networks are used in sequence-to-sequence translation (Sutskever *et al.* 2014). They encode inputs (e.g., a sequence of words) into a latent vector space, then decode that vector representation to generate another sequence using a neural network. Similarly, the multi-species model encoded route-level features into vector representations. These route vectors were decoded by a recurrent neural network to generate a multivariate time series of latent vectors associated with colonization, persistence, and detection (Chung *et al.* 2014). Finally, the multi-species model combined these route-level latent vectors with species-level embeddings to compute colonization, persistence, and detection probabilities (Fig. 1e). For additional details, see Appendix S3 in Supporting Information.

Model comparisons

To compare the performance of the three modeling approaches, the data were partitioned into a training, validation, and test set at the EPA level two ecoregion level (Roberts *et al.* 2017). All routes within an ecoregion were assigned to the same partition. This resulted in 2154 training routes, 948 validation routes, and 1438 test routes. K-fold cross validation would also be possible, though it requires retraining of each model K times (Roberts *et al.* 2017). Because of the size of the BBS data and the computational resources required to train these models, a simpler train/validation/test split was used.

For each of the three modeling approaches, routes in the training set were used for parameter estimation. Single-species models were fit separately for each species (modeling approaches 1 and 2), and one multi-species model (approach 3) was fit using all of the training data. Then, using the trained models, the mean predictive log-density of the validation data was evaluated to identify the best performing model (Gelman *et al.* 2014). This step indicated which model fit best to the withheld validation data. Finally, the best performing model was retrained using the training and validation data, and its predictive performance was evaluated on the withheld test set (Russell & Norvig 2016).

Final model evaluation

The final model's performance was evaluated quantitatively and qualitatively. Quantitative predictive performance was evaluated in two ways. First, 95% prediction interval coverage was computed for test set counts. Prediction intervals were constructed using the quantiles of the binomial distribution, marginalizing over latent occupancy states. Second, the area under the receiver operator characteristic curve (AUC) was computed for binarized test set counts. The AUC analysis also marginalized over latent states to derive predicted probabilities, e.g., $\Pr(y_{t,s,j} = 0 \mid p_{t,s,j}, \psi_{t,s,j}) = 1 - \psi_{t,s,j} + \psi_{t,s,j}(1 - p_{t,s,j})^{50}$, where the parameters p and ψ are estimated by the model. These two approaches provide information on how well the final model could predict the number of stops with detections at each route, and whether any detections would occur on each route, respectively.

Qualitative analyses were based on predicted occupancy states from the final model. The most likely occupancy states were computed at all BBS routes for each year and species using the Viterbi algorithm (Viterbi 1967). The estimated occupancy states were then used to compute finite sample population growth rates for each species (Royle & Kéry 2007). Occupancy state estimates were also used to compute annual spatial centroids for each species, by taking the spatial centroids of occupied route coordinates. These are hereafter referred to as "BBS range centroids" to differentiate from the actual centroid of a species' entire range.

To evaluate whether the results were qualitatively consistent with previous findings about colonization and extinction gradients over species ranges, correspondence between BBS range centroids and both colonization and persistence probabilities were assessed using linear regression (Mehlman 1997; Doherty Jr *et al.* 2003; Royle & Kéry 2007). For this analysis, the response variable was colonization or persistence averaged over time, the predictor was distance from BBS range centroid. Survey routes were the sample units, and separate analyses were conducted for each species. To avoid bias associated with recently added BBS routes and variance associated with rare species, these analyses only used data from BBS routes surveyed in every year, species observed in every year, and species that occurred in 100 or more routes (Sauer *et al.* 2017).

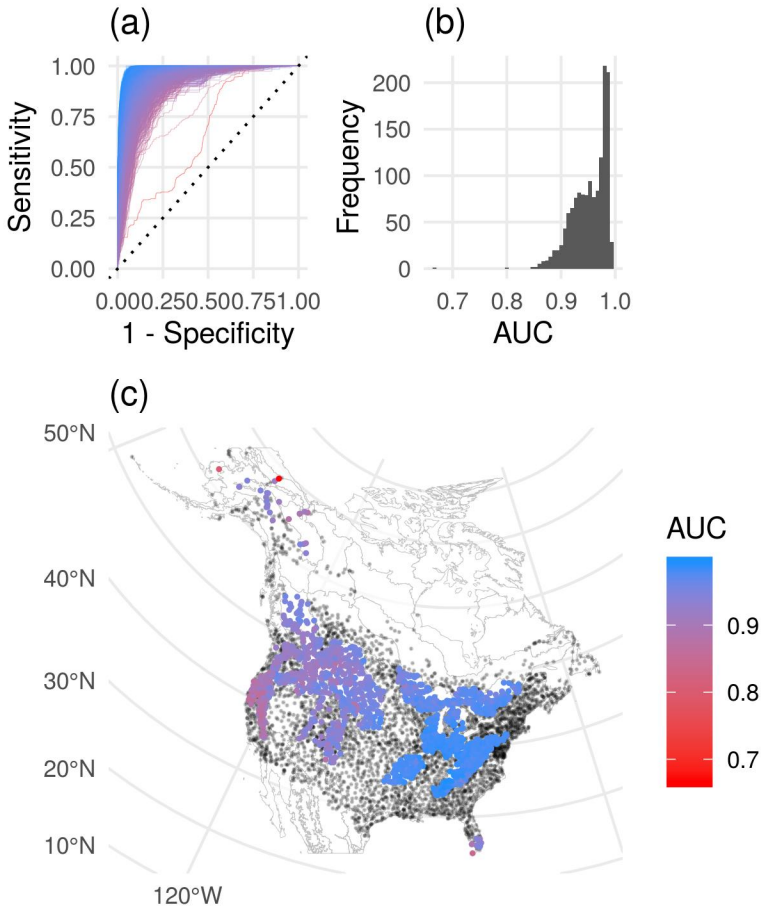

Figure 2: (a) Receiver operator characteristic curves of binarized detection/non-detection data for each survey route in the test set, colored by the area under the curve (AUC). Here, the x-axis is the complement of specificity: the ratio of the number of false positives (incorrectly predicted detections, with no observed detections) to the sum of false positives and true negatives (correctly predicted non-detections with observed non-detections). The y-axis is true positive rate: the ratio of the number of true positives (correctly predicted detections with observed detections) to the sum of true positives and false negatives (incorrectly predicted non-detections with observed detections). The overall distribution of AUC values is shown in (b), and (c) shows the locations of test set routes colored by AUC values, where black dots represent routes that were used to train the final model.

Finally, visualizations were developed to graphically interpret the final model. First, route-level features were visualized using t-distributed stochastic neighbor embedding, which maps high dimensional vectors to lower dimensional representations (Maaten & Hinton 2008). In this lower dimensional space, routes with similar embeddings are close together, and routes with dissimilar embeddings appear distant (Rauber *et al.* 2016). Second, species' loading vectors were compared in terms of cosine similarity, which measures the orientation of loading vectors in latent space. Species' occupancy should be positively related if loading vectors are oriented similarly. This expectation was checked graphically by comparing estimated occupancy time series of species that had the most similar and most different loading vectors.

Results

The multi-species neural hierarchical model performed best on the withheld validation routes. The difference in mean validation set negative log likelihood was 5.015 relative to the baseline model and 2.108 relative to the single-species neural hierarchical model. For the final model, 95% prediction interval coverage for observed counts at withheld test set routes was 93.6%, with a standard deviation of 1.5%, a minimum of 86.5%, and a maximum of 97.6%. The model also predicted whether species would be detected on test set routes fairly well, with a mean test set AUC of 0.953, an among-route standard deviation of 0.032, a minimum of 0.667, and a maximum of 0.993 (Fig. 2).

Qualitative results related to range shifts and population growth rates were plausible given previous work. The model identified the invasive Eurasian Collared-Dove (*Streptopelia decaocto*) as having the greatest range centroid displacement from 1997 to 2018 and the highest finite sample population growth rate (Fig. 3), consistent with its invasion of North America from Florida following its introduction in the 1980's (Bled *et al.* 2011). Increasing trends also have previously been reported for the species with the next three highest population growth rates: Bald Eagle (*Haliaeetus leucocephalus*), Wild Turkey (*Meleagris gallopavo*), and Osprey (*Pandion haliaetus*) (Sauer *et al.* 2013).

The majority (77%) of common species were less likely to persist at routes that were distant from their estimated BBS range centroids. Similarly, 98% of common species were less likely to colonize routes distant from their BBS range centroids. There were examples of species with positive and negative distance coefficients for persistence and colonization (Fig. 4a). Negative relationships were most apparent for common species that occupied a large fraction of BBS routes (Fig. 4b). Results for representative species are displayed in Fig. 4c-d.

Route vectors combined information from the categorical and continuous route-level features. Unsurprisingly (because ecoregion was an input feature), routes in the same ecoregions clustered together (Fig. 5a). Route embeddings also revealed relationships among ecoregions. For example, Marine West Coast Forest ecoregion routes were similar to Northwestern Forested Mountains routes, and most different from Northern Forests routes (Fig. 5a). Variation within clusters also related to continuous route-level features (Fig. 5b).

The model predicted non-linear dependence among species that is interpretable in terms of the cosine similarity among species-specific parameter vectors. For example, parameters for Mourning Dove (*Zenaida macroura*) were closest to those of Barn Swallow (*Hirundo rustica*), and most different from (Myrtle Warbler) Yellow-rumped Warbler (*Setophaga coronata coronata*) (Fig. 6a). On BBS routes where Mourning Doves are likely to occur, Barn Swallow are also likely to occur, and (Myrtle Warbler) Yellow-rumped Warbler are unlikely to occur. Species pairs of the most similar and most dissimilar loadings are also provided for Eurasian Collared-Dove and Bald Eagle (Fig. 6b-c).

Discussion

Neural hierarchical models provide a bridge between hierarchical models built from scientific knowledge, and neural networks that approximate functions over structured domains. This framework integrates research in science-based deep learning and ecological modeling, and can use existing hierarchical models as a starting point. The case study provides a proof of concept example of constructing a scalable and performant neural hierarchical model based on a multi-species dynamic occupancy model, providing insights about colonization and extinction dynamics of North American bird assemblages at a continental scale.

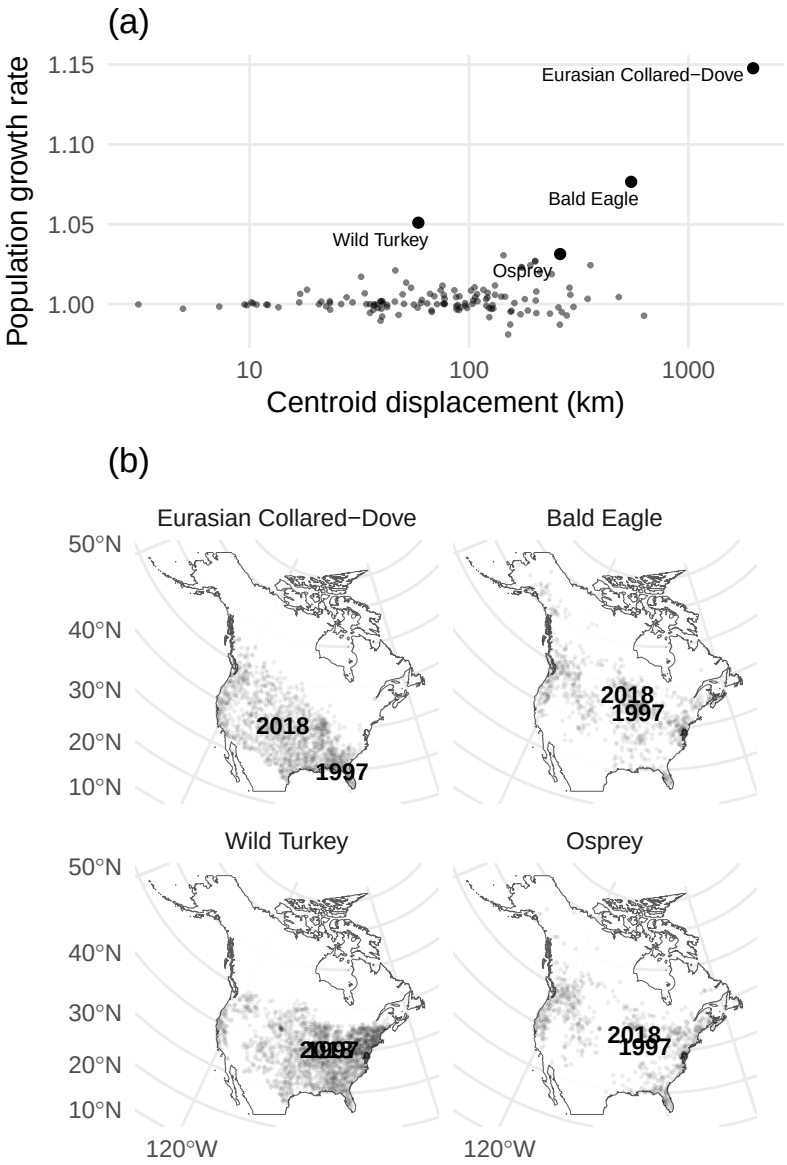

Figure 3: (a) Centroid displacement of bird species (x-axis) from 1997 to 2018 in kilometers vs. finite-sample population growth rate (y-axis), where a growth rate of 1 is stable, values less than one are decreasing, and values greater than one are increasing. Species with the highest population growth rates are highlighted. Panel (b) shows the locations of breeding bird survey range centroids in 1997 and 2018 for each highlighted species, along with grey points that represent survey routes where the species was estimated to be present in at least one year.

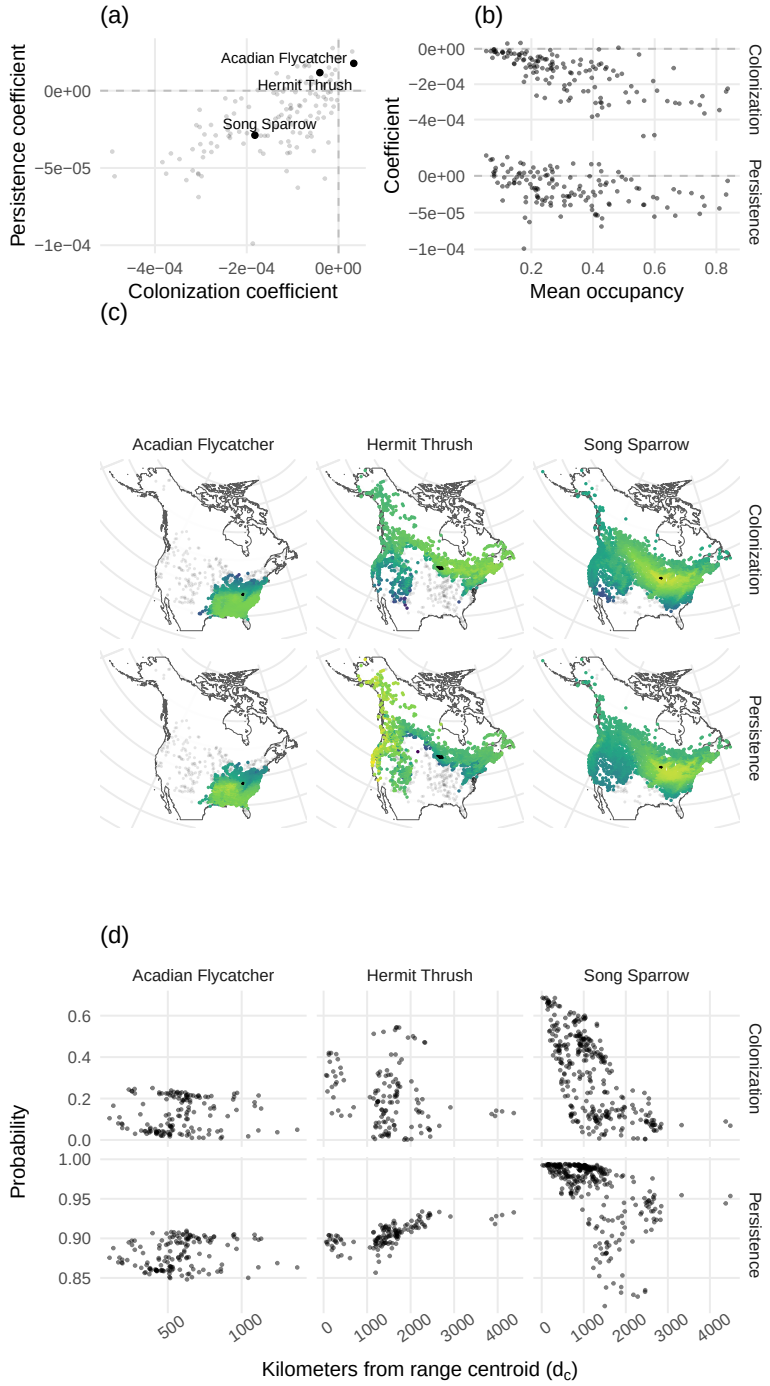

Figure 4: (a) Scatter plot relating species-specific distance decay coefficients for colonization and persistence, with focal species from each quadrant highlighted. (b) Mean finite-sample occupancy (x-axis) vs. coefficients relating distance from range centroid and the probability of colonization (y-axis, top row), and persistence (y-axis, bottom row). (c) Maps of colonization and persistence probabilities at each route where focal species were likely absent (in grey) and present (in color, normalized to increase the visibility of gradients). Centroids for each year are shown in solid black. (d) Colonization and persistence probabilities (y-axis) as a function of distance from range centroid, averaged among years.

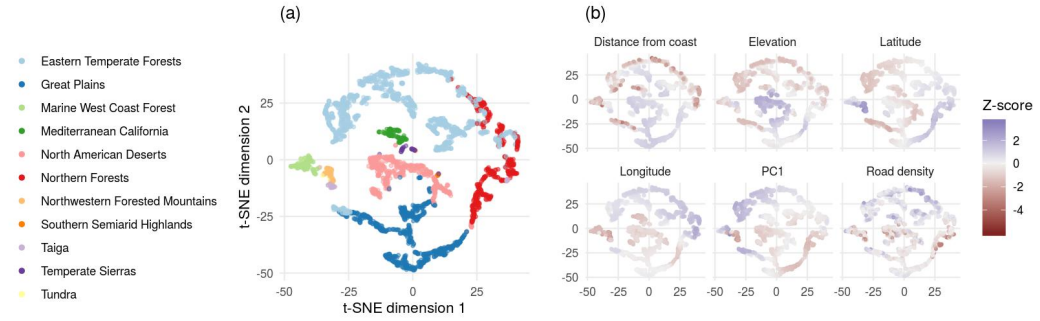

Figure 5: Clustering of North American Breeding Bird Survey routes by ecoregion and route-level features. All panels show a two dimensional plot of route vectors, computed using t-distributed stochastic neighbor embedding (t-SNE) on the latent vectors for initial occupancy, persistence, colonization, and detection probabilities. Each route is shown as a point. Color indicates (a) EPA level 1 ecoregion and (b) z-standardized continuous route-level features. PC1 refers to the first principal component of WorldClim climate data.

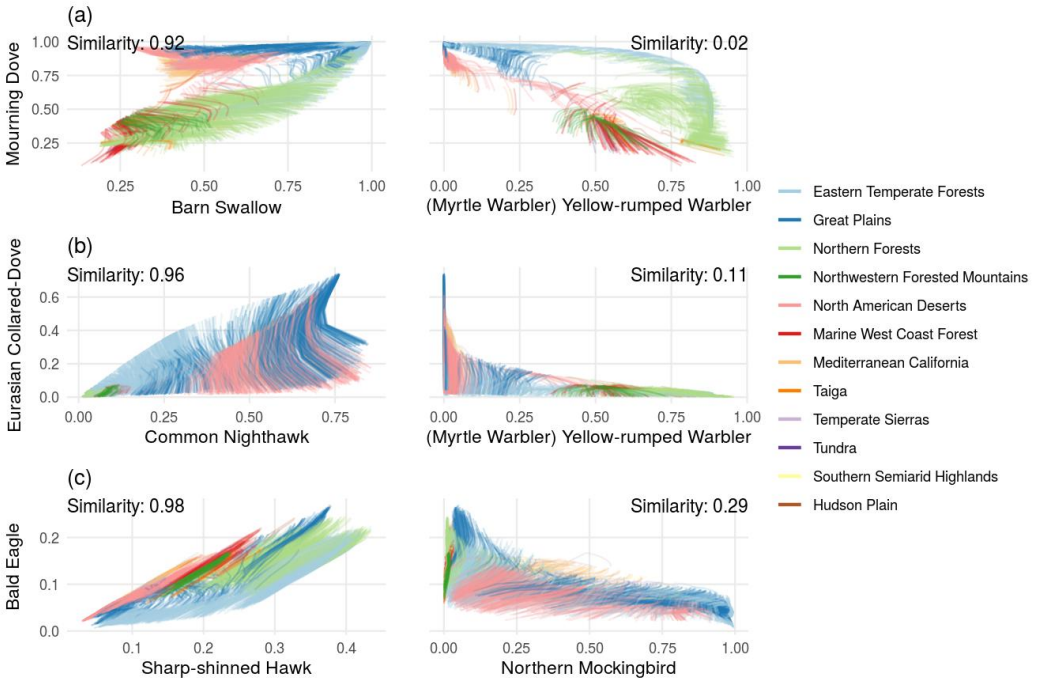

Figure 6: Relationships among species estimated occupancy probabilities through time, by location for species pairs that are nearest neighbors in parameter cosine distance (left column), and most cosine dissimilar (right column). Color indicates EPA level 1 ecoregions. Cosine similarity values for species pairs are printed in the upper corners of each panel. Each route is a line segment that connects estimates of occupancy probabilities for each year from 1997 to 2018. Species pairs with high cosine similarity tend to have positively related occupancy trajectories.

The breeding bird survey case study indicates that neural hierarchical models can outperform simpler models. Notably, the final model was performant both quantitatively and qualitatively, detecting population increases and range expansions that are consistent with prior work. The case study extends previous analyses of persistence probabilities at range edges (Royle & Kéry 2007), indicating that common species tend to have higher persistence and colonization probabilities at routes close to their BBS range centroid. Yet, interpreting this result is complicated by irregular range geometry which can lead to centroids that near the edge (or even outside) of a species range, and divergence between actual and estimated ranges due to incomplete sampling (Sagarin & Gaines 2002; Fortin *et al.* 2005; Dallas *et al.* 2017; Knouft 2018). Additional complexity is apparent in Fig. 4c, which indicates that gradients in colonization and persistence may not be isotropic (the same in every direction). With those caveats, the results are broadly consistent with a theoretical expectation that range boundaries can arise from gradients in local extinction and colonization rates (Holt & Keitt 2000).

The case study used a gated recurrent neural network architecture to handle temporal structure (Chung *et al.* 2014), and architectures designed for other data structures present additional opportunities for ecological applications. For example, neural hierarchical models could be used to couple a convolutional neural network observation model for camera trap images (Norouzzadeh *et al.* 2018; Tabak *et al.* 2019) with an ecological process model that describes animal density, movement, or community composition (Burton *et al.* 2015). Gridded data such as modeled climate data, remotely sensed Earth observations, and even 96 well microplates also might use convolutional neural network architectures (Rawat & Wang 2017). Appendix S2 provides a simulated example where gridded data (aerial imagery) are mapped to a state transition matrix of a hidden Markov model. In addition, many ecological datasets exhibit graph structure related to phylogenies, social networks, or network-like spatial structure. While it is possible to adapt convolutional neural networks to operate on distance matrices computed from graphs (Fioravanti *et al.* 2018), graph representation learning can also provide embeddings for nodes that encode network structure and node attributes (Hamilton *et al.* 2017; Cai *et al.* 2018).

Neural hierarchical models can scale to data that are too large to fit in computer memory. Indeed, memory limitations precluded comparison of a fully Bayesian multi-species dynamic occupancy model against the multi-species neural hierarchical model. The current state of the art multi-species occupancy models use approximately one tenth of the number of species, at about half the number of sites, and are static in the sense that colonization and extinction dynamics through time are not represented (Tobler *et al.* 2019). Species distribution models are increasingly scalable due to advances in approximate Gaussian process models (Tikhonov *et al.* 2019), but multi-species dynamic occupancy models have not previously been reported at this scale. This is particularly relevant for extensions of the BBS case study, given the volume of (imperfect) bird data accumulating through citizen science programs (Sullivan *et al.* 2009). The key strategy providing scalability in the case study is stochastic optimization that uses mini-batches - small subsets of a larger dataset - to generate noisy estimates of model performance and partial derivatives that can be used during training (Ruder 2016). With this approach, the entire dataset does not need to be loaded into memory at the same time. This could be useful as a way to scale models that integrate BBS, eBird, and other data (Ngiam *et al.* 2011; Pacifici *et al.* 2017; Zipkin *et al.* 2017; Lin *et al.* 2019).

Though neural hierarchical models can perform well in the large data regime, they might be overparameterized in a small data setting. For this reason, performance comparisons with simpler baseline models are useful. Appendix S2 provides a simulated example, where simple baselines work better with small datasets, and more complex neural hierarchical models work better with large datasets. Approaches familiar to ecologists such as k-fold cross validation and (to a lesser extent) information criteria such as the Akaike Information Criterion (AIC) have been used with neural networks, but such approaches are less common than cross-validation with training, validation, and test partitions of the data (Anderson & Burnham 2004; Jiang & Chen 2016; Ran & Hu 2017). With large datasets, considerable computational resources are required for training, so that retraining the same model many times might make k-fold cross validation infeasible. As a practical matter, even if a neural hierarchical model demonstrates superior predictive performance, a simpler model might be preferred if interpretability and/or explainability is a high priority, as it might be in a decision-making context (Rudin 2018).

Neural networks have a reputation for being "black-box" models. However, the interpretation and explanation of such models is an active area of research (Roscher *et al.* 2019). Interpretations map abstract concepts to domains that humans can make sense of, e.g., mapping neural network

parameters to species identity or spatial location (Montavon *et al.* 2018). Explanations are collections of features in such a domain that contributed to a prediction (Montavon *et al.* 2018), e.g., sensitivity of model output to perturbations of the input (Olden & Jackson 2002; Gevrey *et al.* 2003).

Looking ahead, in a forecasting or decision-making setting, it would be important to estimate both aleatoric uncertainty (arising from noise in an observation process) and epistemic uncertainty (arising from uncertainty in a model and its parameters) (Clark *et al.* 2001; Kendall & Gal 2017). Recent advances in accounting for uncertainty in neural network parameter estimates and architectures could be applied in future work. Approaches include "Bayes by backpropagation" (Blundell *et al.* 2015), normalizing flows for variational approximations (Kingma *et al.* 2016), adversarial training (Lakshminarayanan *et al.* 2017), methods that use dropout or its continuous relaxation (Gal *et al.* 2017), and ensemble approaches (McDermott & Wikle 2017). These methods can also help explain neural networks, e.g., to probabilistically estimate sensitivity to model inputs to random masking (Chang *et al.* 2017), or to decompose predictive uncertainty into component parts (Thiagarajan *et al.* 2019).

The potential for combining neural networks with mechanistic models was recognized more than thirty years ago (Psichogios & Ungar 1992; Meade Jr & Fernandez 1994; Lagaris *et al.* 1998). This potential is more easily realized today due to methodological spillover from deep learning into the natural sciences, but also increases in computing power, availability of ecological data, and the proliferation of educational content for quantitative ecology and machine learning. Further, modern deep learning frameworks provide abstractions that allow users to focus on model construction rather than the details of implementation, increasing accessibility in the same way that WinBUGS, JAGS, OpenBUGS, and Stan have done (Lunn *et al.* 2000; Plummer & others 2003; Spiegelhalter *et al.* 2005; Carpenter *et al.* 2017).

Although deep learning and ecological modeling may seem to be separate activities, neural hierarchical models bridge these disciplines. Given the increasing availability of massive ecological data, scalable and flexible science-based models are increasingly needed. Neural hierarchical models satisfy this need, and can provide a framework that links imperfect observational data to ecological processes and mechanisms by construction.

Acknowledgements

Thanks to David Zonana and Roland Knapp for discussions on the potential of neural hierarchical models, and to Susie Ellis for providing feedback on a draft of the manuscript. This manuscript was also greatly improved by three anonymous reviewers. Also thanks to Brandon Edwards for developing the bbsBayes R package, which was used to acquire and parse the North American Breeding Bird Survey data. This work was motivated by a workshop on machine learning in Earth science, hosted by Earth Lab and organized by the Federation of Earth Science Information Partners and the National Aeronatics and Space Administration Advanced Information Systems Technology program. This work was made possible by the CU Boulder Grand Challenge initiative and the Cooperative Institute for Research in the Environmental Sciences through their investment in Earth Lab. Thanks also to the thousands of U.S. and Canadian participants who annually perform and coordinate the North American Breeding Bird Survey.

Literature cited

Anderson, D. & Burnham, K. (2004). Model selection and multi-model inference. *Second. NY: Springer-Verlag*, 63.

Austin, M. (2002). Spatial prediction of species distribution: An interface between ecological theory and statistical modelling. *Ecological modelling*, 157, 101–118.

Ba, Y., Zhao, G. & Kadambi, A. (2019). Blending diverse physical priors with neural networks. *arXiv preprint arXiv:1910.00201*.

Berliner, L.M. (1996). Hierarchical bayesian time series models. In: *Maximum entropy and bayesian methods*. Springer, pp. 15–22.

Bingham, E., Chen, J.P., Jankowiak, M., Obermeyer, F., Pradhan, N. & Karaletsos, T. *et al.* (2018). Pyro: Deep Universal Probabilistic Programming. *arXiv preprint arXiv:1810.09538*.

Bled, F., Nichols, J.D. & Altwegg, R. (2013). Dynamic occupancy models for analyzing species' range dynamics across large geographic scales. *Ecology and evolution*, 3, 4896–4909.

Bled, F., Royle, J.A. & Cam, E. (2011). Hierarchical modeling of an invasive spread: The eurasian collared-dove streptopelia decaocto in the united states. *Ecological Applications*, 21, 290–302.

Blundell, C., Cornebise, J., Kavukcuoglu, K. & Wierstra, D. (2015). Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*.

Burton, A.C., Neilson, E., Moreira, D., Ladle, A., Steenweg, R. & Fisher, J.T. *et al.* (2015). Wildlife camera trapping: A review and recommendations for linking surveys to ecological processes. *Journal of Applied Ecology*, 52, 675–685.

Butler, K.T., Davies, D.W., Cartwright, H., Isayev, O. & Walsh, A. (2018). Machine learning for molecular and materials science. *Nature*, 559, 547.

Cai, H., Zheng, V.W. & Chang, K.C.-C. (2018). A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 30, 1616–1637.

Calvert, A.M., Bonner, S.J., Jonsen, I.D., Flemming, J.M., Walde, S.J. & Taylor, P.D. (2009). A hierarchical bayesian approach to multi-state mark–recapture: Simulations and applications. *Journal of Applied Ecology*, 46, 610–620.

Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M. & Tishby, N. *et al.* (2019). Machine learning and the physical sciences. *arXiv preprint arXiv:1903.10563*.

Carpenter, B., Gelman, A., Hoffman, M.D., Lee, D., Goodrich, B. & Betancourt, M. *et al.* (2017). Stan: A probabilistic programming language. *Journal of statistical software*, 76.

Chang, C.-H., Creager, E., Goldenberg, A. & Duvenaud, D. (2017). Interpreting neural network classifications with variational dropout saliency maps. In: *Proc. NIPS*. p. 6.

Chen, D.-G. & Hare, S.R. (2006). Neural network and fuzzy logic models for pacific halibut recruitment analysis. *ecological modelling*, 195, 11–19.

Chen, D., Xue, Y., Chen, S., Fink, D. & Gomes, C. (2016). Deep multi-species embedding. *arXiv preprint arXiv:1609.09353*.

Ching, T., Himmelstein, D.S., Beaulieu-Jones, B.K., Kalinin, A.A., Do, B.T. & Way, G.P. *et al.* (2018). Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15, 20170387.

Chon, T.-S., Kwak, I.-S., Park, Y.-S., Kim, T.-H. & Kim, Y. (2001). Patterning and short-term predictions of benthic macroinvertebrate community dynamics by using a recurrent artificial neural network. *Ecological Modelling*, 146, 181–193.

Christin, S., Hervet, E. & Lecomte, N. (2018). Applications for deep learning in ecology. *bioRxiv*, 334854.

Chung, J., Gulcehre, C., Cho, K. & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Clark, J.S., Carpenter, S.R., Barber, M., Collins, S., Dobson, A. & Foley, J.A. *et al.* (2001). Ecological forecasts: An emerging imperative. *science*, 293, 657–660.

Cressie, N., Calder, C.A., Clark, J.S., Hoef, J.M.V. & Wikle, C.K. (2009). Accounting for uncertainty in ecological analysis: The strengths and limitations of hierarchical statistical modeling. *Ecological Applications*, 19, 553–570.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2, 303–314.

Dallas, T., Decker, R.R. & Hastings, A. (2017). Species are not most abundant in the centre of their geographic range or climatic niche. *Ecology Letters*, 20, 1526–1533.

Dennis, E.B., Morgan, B.J. & Ridout, M.S. (2015). Computational aspects of n-mixture models. *Biometrics*, 71, 237–246.

Desjardins-Proulx, P., Poisot, T. & Gravel, D. (2019). Artificial intelligence for ecological and evolutionary synthesis. *bioRxiv*.

De Valpine, P. & Hastings, A. (2002). Fitting population models incorporating process noise and observation error. *Ecological Monographs*, 72, 57–76.

Doherty Jr, P.F., Boulinier, T. & Nichols, J.D. (2003). Local extinction and turnover rates at the edge and interior of species' ranges. In: *Annales zoologici fennici*. JSTOR, pp. 145–153.

Dorazio, R.M., Kery, M., Royle, J.A. & Plattner, M. (2010). Models for inference in dynamic metacommunity systems. *Ecology*, 91, 2466–2475.

Fick, S.E. & Hijmans, R.J. (2017). WorldClim 2: New 1-km spatial resolution climate surfaces for global land areas. *International journal of climatology*, 37, 4302–4315.

Fioravanti, D., Giarratano, Y., Maggio, V., Agostinelli, C., Chierici, M. & Jurman, G. *et al.* (2018). Phylogenetic convolutional neural networks in metagenomics. *BMC bioinformatics*, 19, 49.

Flagel, L., Brandvain, Y. & Schrider, D.R. (2018). The unreasonable effectiveness of convolutional neural networks in population genetic inference. *Molecular biology and evolution*, 36, 220–238.

Fortin, M.-J., Keitt, T.H., Maurer, B., Taper, M., Kaufman, D.M. & Blackburn, T. (2005). Species' geographic ranges and distributional limits: Pattern analysis and statistical issues. *Oikos*, 108, 7–17.

Fricker, G.A., Ventura, J.D., Wolf, J.A., North, M.P., Davis, F.W. & Franklin, J. (2019). A convolutional neural network classifier identifies tree species in mixed-conifer forest from hyperspectral imagery. *Remote Sensing*, 11, 2326.

Gal, Y., Hron, J. & Kendall, A. (2017). Concrete dropout. In: *Advances in neural information processing systems*. pp. 3581–3590.

Gazestani, V.H. & Lewis, N.E. (2019). From genotype to phenotype: Augmenting deep learning with networks and systems biology. *Current Opinion in Systems Biology*.

Gelman, A., Hwang, J. & Vehtari, A. (2014). Understanding predictive information criteria for bayesian models. *Statistics and computing*, 24, 997–1016.

Gevrey, M., Dimopoulos, I. & Lek, S. (2003). Review and comparison of methods to study the contribution of variables in artificial neural network models. *Ecological modelling*, 160, 249–264.

Golding, N. & Purse, B.V. (2016). Fast and flexible bayesian species distribution modelling using gaussian processes. *Methods in Ecology and Evolution*, 7, 598–608.

Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep learning*. MIT press.

Guillera-Arroita, G. (2017). Modelling of species distributions, range dynamics and communities under imperfect detection: Advances, challenges and opportunities. *Ecography*, 40, 281–295.

Guo, C. & Berkhahn, F. (2016). Entity embeddings of categorical variables. *arXiv preprint arXiv:1604.06737*.

Hamilton, W., Ying, Z. & Leskovec, J. (2017). Inductive representation learning on large graphs. In: *Advances in neural information processing systems*. pp. 1024–1034.

Harris, D.J. (2015). Generating realistic assemblages with a joint species distribution model. *Methods in Ecology and Evolution*, 6, 465–473.

Hefley, T.J., Broms, K.M., Brost, B.M., Buderman, F.E., Kay, S.L. & Scharf, H.R. *et al.* (2017). The basis function approach for modeling autocorrelation in ecological data. *Ecology*, 98, 632–646.

Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9, 1735–1780.

Holt, R.D. & Keitt, T.H. (2000). Alternative causes for range limits: A metapopulation perspective. *Ecology letters*, 3, 41–47.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4, 251–257.

Hutchinson, R.A., Liu, L.-P. & Dietterich, T.G. (2011). Incorporating boosted regression trees into ecological latent variable models. In: *Twenty-fifth aaai conference on artificial intelligence*.

Jain, A., Zamir, A.R., Savarese, S. & Saxena, A. (2016). Structural-rnn: Deep learning on spatio-temporal graphs. In: *Proceedings of the ieee conference on computer vision and pattern recognition*. pp. 5308–5317.

Jeong, K.-S., Joo, G.-J., Kim, H.-W., Ha, K. & Recknagel, F. (2001). Prediction and elucidation of phytoplankton dynamics in the nakdong river (korea) by means of a recurrent artificial neural network. *Ecological Modelling*, 146, 115–129.

Jeong, K.-S., Kim, D.-K., Jung, J.-M., Kim, M.-C. & Joo, G.-J. (2008). Non-linear autoregressive modelling by temporal recurrent neural networks for the prediction of freshwater phytoplankton dynamics. *ecological modelling*, 211, 292–300.

Jiang, P. & Chen, J. (2016). Displacement prediction of landslide based on generalized regression neural networks with k-fold cross-validation. *Neurocomputing*, 198, 40–47.

Johnson, D.S., Laake, J.L. & Ver Hoef, J.M. (2010). A model-based approach for making ecological inference from distance sampling data. *Biometrics*, 66, 310–318.

Johnson, D.S., Pelland, N.A. & Sterling, J.T. (2018). A continuous-time semi-markov model for animal movement in a dynamic environment. *BioRxiv*, 353516.

Johnson, D.S., Thomas, D.L., Ver Hoef, J.M. & Christ, A. (2008). A general framework for the analysis of animal resource selection from telemetry data. *Biometrics*, 64, 968–976.

Jolly, G.M. (1965). Explicit estimates from capture-recapture data with both death and immigration-stochastic model. *Biometrika*, 52, 225–247.

Karpatne, A., Watkins, W., Read, J. & Kumar, V. (2017). Physics-guided neural networks (pgnn): An application in lake temperature modeling. *arXiv preprint arXiv:1710.11431*.

Keller, M., Schimel, D.S., Hargrove, W.W. & Hoffman, F.M. (2008). A continental strategy for the national ecological observatory network. *Frontiers in Ecology and the Environment*, 6, 282–284.

Kendall, A. & Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? In: *Advances in neural information processing systems*. pp. 5574–5584.

King, R. (2012). A review of bayesian state-space modelling of capture–recapture–recovery data. *Interface Focus*, 2, 190–204.

Kingma, D.P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I. & Welling, M. (2016). Improved variational inference with inverse autoregressive flow. In: *Advances in neural information processing systems*. pp. 4743–4751.

Knapp, R.A., Matthews, K.R., Preisler, H.K. & Jellison, R. (2003). Developing probabilistic models to predict amphibian site occupancy in a patchy landscape. *Ecological Applications*, 13, 1069–1082.

Knouft, J.H. (2018). Appropriate application of information from biodiversity databases is critical when investigating species distributions and diversity: A comment on dallas et al.(). *Ecology letters*, 21, 1119–1120.

Krishnan, R.G., Shalit, U. & Sontag, D. (2017). Structured inference networks for nonlinear state space models. In: *Thirty-first aaai conference on artificial intelligence*.

Krizhevsky, A. (2014). One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*.

Lagaris, I.E., Likas, A. & Fotiadis, D.I. (1998). Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks*, 9, 987–1000.

Lakshminarayanan, B., Pritzel, A. & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In: *Advances in neural information processing systems*. pp. 6402–6413.

Langrock, R., King, R., Matthiopoulos, J., Thomas, L., Fortin, D. & Morales, J.M. (2012). Flexible and practical modeling of animal telemetry data: Hidden markov models and extensions. *Ecology*, 93, 2336–2342.

Latimer, A., Banerjee, S., Sang Jr, H., Mosher, E. & Silander Jr, J. (2009). Hierarchical models facilitate spatial analysis of large data sets: A case study on invasive plant species in the northeastern united states. *Ecology Letters*, 12, 144–154.

LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep learning. *nature*, 521, 436.

Lek, S., Delacoste, M., Baran, P., Dimopoulos, I., Lauga, J. & Aulagnier, S. (1996). Application of neural networks to modelling nonlinear relationships in ecology. *Ecological modelling*, 90, 39–52.

Li, Y., Bu, R., Sun, M., Wu, W., Di, X. & Chen, B. (2018). PointCNN: Convolution on x-transformed points. In: *Advances in neural information processing systems*. pp. 820–830.

Lin, T.-Y., Winner, K., Bernstein, G., Mittal, A., Dokter, A.M. & Horton, K.G. *et al.* (2019). MistNet: Measuring historical bird migration in the us using archived weather radar data and convolutional neural networks. *Methods in Ecology and Evolution*, 10, 1908–1922.

Link, W.A. & Sauer, J.R. (1998). Estimating population change from count data: Application to the north american breeding bird survey. *Ecological applications*, 8, 258–268.

Long, J., Shelhamer, E. & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In: *Proceedings of the ieee conference on computer vision and pattern recognition*. pp. 3431–3440.

Lu, Z., Pu, H., Wang, F., Hu, Z. & Wang, L. (2017). The expressive power of neural networks: A view from the width. In: *Advances in neural information processing systems*. pp. 6231–6239.

Lunn, D.J., Thomas, A., Best, N. & Spiegelhalter, D. (2000). WinBUGS-a bayesian modelling framework: Concepts, structure, and extensibility. *Statistics and computing*, 10, 325–337.

Maaten, L. van der & Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9, 2579–2605.

MacKenzie, D.I., Nichols, J.D., Hines, J.E., Knutson, M.G. & Franklin, A.B. (2003). Estimating site occupancy, colonization, and local extinction when a species is detected imperfectly. *Ecology*, 84, 2200–2207.

MacKenzie, D.I., Nichols, J.D., Lachman, G.B., Droege, S., Andrew Royle, J. & Langtimm, C.A. (2002). Estimating site occupancy rates when detection probabilities are less than one. *Ecology*, 83, 2248–2255.

Malek, S., Salleh, A., Milow, P., Baba, M.S. & Sharifah, S. (2012). Applying artificial neural network theory to exploring diatom abundance at tropical putrajaya lake, malaysia. *Journal of freshwater ecology*, 27, 211–227.

Manel, S., Dias, J.-M. & Ormerod, S.J. (1999). Comparing discriminant analysis, neural networks and logistic regression for predicting species distributions: A case study with a himalayan river bird. *Ecological modelling*, 120, 337–347.

McClintock, B.T. & Michelot, T. (2018a). MomentuHMM: R package for generalized hidden markov models of animal movement. *Methods in Ecology and Evolution*, 9, 1518–1530.

McClintock, B.T. & Michelot, T. (2018b). MomentuHMM: R package for generalized hidden Markov models of animal movement. *Methods in Ecology and Evolution*, 9, 1518–1530.

McDermott, P.L. & Wikle, C.K. (2017). An ensemble quadratic echo state network for non-linear spatio-temporal forecasting. *Stat*, 6, 315–330.

McDermott, P.L. & Wikle, C.K. (2019). Deep echo state networks with uncertainty quantification for spatio-temporal forecasting. *Environmetrics*, 30, e2553.

Meade Jr, A.J. & Fernandez, A.A. (1994). The numerical solution of linear ordinary differential equations by feedforward neural networks. *Mathematical and Computer Modelling*, 19, 1–25.

Mehlman, D.W. (1997). Change in avian abundance across the geographic range in response to environmental change. *Ecological Applications*, 7, 614–624.

Meijer, J.R., Huijbregts, M.A., Schotten, K.C. & Schipper, A.M. (2018). Global patterns of current and future road infrastructure. *Environmental Research Letters*, 13, 064006.

Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Miller, D.L., Burt, M.L., Rexstad, E.A. & Thomas, L. (2013). Spatial models for distance sampling data: Recent developments and future directions. *Methods in Ecology and Evolution*, 4, 1001–1010.

Montavon, G., Samek, W. & Müller, K.-R. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73, 1–15.

Nair, V. & Hinton, G.E. (2010). Rectified linear units improve restricted boltzmann machines. In: *Proceedings of the 27th international conference on machine learning (icml-10)*. pp. 807–814.

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H. & Ng, A.Y. (2011). Multimodal deep learning. In: *Proceedings of the 28th international conference on machine learning (icml-11)*. pp. 689–696.

Niepert, M., Ahmed, M. & Kutzkov, K. (2016). Learning convolutional neural networks for graphs. In: *International conference on machine learning*. pp. 2014–2023.

Norouzzadeh, M.S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M.S. & Packer, C. *et al.* (2018). Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proceedings of the National Academy of Sciences*, 115, E5716–E5725.

Oksanen, J. & Minchin, P.R. (2002). Continuum theory revisited: What shape are species responses along ecological gradients? *Ecological Modelling*, 157, 119–129.

Olden, J.D. & Jackson, D.A. (2002). Illuminating the "black box": A randomization approach for understanding variable contributions in artificial neural networks. *Ecological modelling*, 154, 135–150.

Ovaskainen, O., Roy, D.B., Fox, R. & Anderson, B.J. (2016). Uncovering hidden spatial structure in species communities with spatially explicit joint species distribution models. *Methods in Ecology and Evolution*, 7, 428–436.

Özesmi, S.L., Tan, C.O. & Özesmi, U. (2006). Methodological issues in building, training, and testing artificial neural networks in ecological applications. *Ecological Modelling*, 195, 83–93.

Pacifici, K., Reich, B.J., Miller, D.A., Gardner, B., Stauffer, G. & Singh, S. *et al.* (2017). Integrating multiple data sources in species distribution modeling: A framework for data fusion. *Ecology*, 98, 840–850.

Paganini, M., Oliveira, L. de & Nachman, B. (2018). Accelerating science with generative adversarial networks: An application to 3D particle showers in multilayer calorimeters. *Physical review letters*, 120, 042003.

Pardieck, L., K., Ziolkowski Jr., J., D., M., L. & Aponte *et al.* (2018). North american breeding bird survey dataset 1966 - 2018.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E. & DeVito, Z. *et al.* (2017). Automatic differentiation in PyTorch. In: *NIPS autodiff workshop*.

Patterson, T.A., Basson, M., Bravington, M.V. & Gunn, J.S. (2009). Classifying movement behaviour in relation to environmental conditions using hidden markov models. *Journal of Animal Ecology*, 78, 1113–1123.

Patterson, T.A., Parton, A., Langrock, R., Blackwell, P.G., Thomas, L. & King, R. (2017). Statistical modelling of individual animal movement: An overview of key methods and a discussion of practical challenges. *AStA Advances in Statistical Analysis*, 101, 399–438.

Plummer, M. & others. (2003). JAGS: A program for analysis of bayesian graphical models using gibbs sampling. In: *Proceedings of the 3rd international workshop on distributed statistical computing*. Vienna, Austria.

Psichogios, D.C. & Ungar, L.H. (1992). A hybrid neural network-first principles approach to process modeling. *AIChE Journal*, 38, 1499–1511.

Rabiner, L.R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77, 257–286.

Radovic, A., Williams, M., Rousseau, D., Kagan, M., Bonacorsi, D. & Himmel, A. *et al.* (2018). Machine learning at the energy and intensity frontiers of particle physics. *Nature*, 560, 41.

Raissi, M. (2018). Deep hidden physics models: Deep learning of nonlinear partial differential equations. *The Journal of Machine Learning Research*, 19, 932–955.

Ran, Z.-Y. & Hu, B.-G. (2017). Parameter identifiability in statistical machine learning: A review. *Neural Computation*, 29, 1151–1203.

Rangapuram, S.S., Seeger, M.W., Gasthaus, J., Stella, L., Wang, Y. & Januschowski, T. (2018). Deep state space models for time series forecasting. In: *Advances in neural information processing systems*. pp. 7785–7794.

Rauber, P.E., Fadel, S.G., Falcao, A.X. & Telea, A.C. (2016). Visualizing the hidden activity of artificial neural networks. *IEEE transactions on visualization and computer graphics*, 23, 101–110.

Rawat, W. & Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29, 2352–2449.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J. & Carvalhais, N. *et al.* (2019). Deep learning and process understanding for data-driven earth system science. *Nature*, 566, 195.

Roberts, D.R., Bahn, V., Ciuti, S., Boyce, M.S., Elith, J. & Guillera-Arroita, G. *et al.* (2017). Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*, 40, 913–929.

Roscher, R., Bohn, B., Duarte, M.F. & Garcke, J. (2019). Explainable machine learning for scientific insights and discoveries. *arXiv preprint arXiv:1905.08883*.

Royle, J.A. (2004). N-mixture models for estimating population size from spatially replicated counts. *Biometrics*, 60, 108–115.

Royle, J.A. & Kéry, M. (2007). A bayesian state-space formulation of dynamic occupancy models. *Ecology*, 88, 1813–1823.

Royle, J.A. & Link, W.A. (2006). Generalized site occupancy models allowing for false positive and false negative errors. *Ecology*, 87, 835–841.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Rudin, C. (2018). Please stop explaining black box models for high stakes decisions. *arXiv preprint arXiv:1811.10154*.

Russell, S.J. & Norvig, P. (2016). *Artificial intelligence: A modern approach*. Malaysia; Pearson Education Limited,

Sagarin, R.D. & Gaines, S.D. (2002). The 'abundant centre'distribution: To what extent is it a biogeographical rule? *Ecology letters*, 5, 137–147.

Sauer, J.R. & Link, W.A. (2011). Analysis of the north american breeding bird survey using hierarchical models. *The Auk*, 128, 87–98.

Sauer, J.R., Link, W.A., Fallon, J.E., Pardieck, K.L. & Ziolkowski Jr, D.J. (2013). The north american breeding bird survey 1966–2011: Summary analysis and species accounts. *North American Fauna*, 79, 1–32.

Sauer, J.R., Niven, D.K., Pardieck, K.L., Ziolkowski Jr, D.J. & Link, W.A. (2017). Expanding the north american breeding bird survey analysis to include additional species and regions. *Journal of Fish and Wildlife Management*, 8, 154–172.

Simonyan, K. & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Spiegelhalter, D., Thomas, A., Best, N. & Lunn, D. (2005). OpenBUGS version 2.10, user manual. *MRC Biostatistics Unit, Cambridge, United Kingdom*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15, 1929–1958.

Stan Development Team. (2018). RStan: The R interface to Stan.

Sullivan, B.L., Wood, C.L., Iliff, M.J., Bonney, R.E., Fink, D. & Kelling, S. (2009). EBird: A citizen-based bird observation network in the biological sciences. *Biological Conservation*, 142, 2282–2292.

Sutskever, I., Vinyals, O. & Le, Q.V. (2014). Sequence to sequence learning with neural networks. In: *Advances in neural information processing systems*. pp. 3104–3112.

Tabak, M.A., Norouzzadeh, M.S., Wolfson, D.W., Sweeney, S.J., VerCauteren, K.C. & Snow, N.P. *et al.* (2019). Machine learning to classify animal species in camera trap images: Applications in ecology. *Methods in Ecology and Evolution*, 10, 585–590.

Thiagarajan, J.J., Kim, I., Anirudh, R. & Bremer, P.-T. (2019). Understanding deep neural networks through input uncertainties. In: *ICASSP 2019-2019 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, pp. 2812–2816.

Thorson, J.T., Ianelli, J.N., Larsen, E.A., Ries, L., Scheuerell, M.D. & Szuwalski, C. *et al.* (2016). Joint dynamic species distribution models: A tool for community ordination and spatio-temporal monitoring. *Global Ecology and Biogeography*, 25, 1144–1158.

Thorson, J.T., Scheuerell, M.D., Shelton, A.O., See, K.E., Skaug, H.J. & Kristensen, K. (2015). Spatial factor analysis: A new tool for estimating joint species distributions and correlations in species range. *Methods in Ecology and Evolution*, 6, 627–637.

Tikhonov, G., Abrego, N., Dunson, D. & Ovaskainen, O. (2017). Using joint species distribution models for evaluating how species-to-species associations depend on the environmental context. *Methods in Ecology and Evolution*, 8, 443–452.

Tikhonov, G., Duan, L., Abrego, N., Newell, G., White, M. & Dunson, D. *et al.* (2019). Computationally efficient joint species distribution modeling of big spatial data. *Ecology*.

Tobler, M.W., Kéry, M., Hui, F.K., Guillera-Arroita, G., Knaus, P. & Sattler, T. (2019). Joint species distribution models with species correlations and imperfect detection. *Ecology*, e02754.

Van Lint, J., Hoogendoorn, S. & Zuylen, H.J. van. (2002). Freeway travel time prediction with state-space neural networks: Modeling state-space dynamics with recurrent neural networks. *Transportation Research Record*, 1811, 30–39.

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13, 260–269.

Warton, D.I., Blanchet, F.G., O'Hara, R.B., Ovaskainen, O., Taskinen, S. & Walker, S.C. *et al.* (2015). So many variables: Joint modeling in community ecology. *Trends in Ecology & Evolution*, 30, 766–779.

Webb, M.H., Wotherspoon, S., Stojanovic, D., Heinsohn, R., Cunningham, R. & Bell, P. *et al.* (2014). Location matters: Using spatially explicit occupancy models to predict the distribution of the highly mobile, endangered swift parrot. *Biological Conservation*, 176, 99–108.

Wikle, C.K. (2003). Hierarchical models in environmental science. *International Statistical Review*, 71, 181–199.

Wikle, C.K. (2019). Comparison of deep neural networks and deep hierarchical models for spatio-temporal data. *Journal of Agricultural, Biological and Environmental Statistics*, 1–29.

Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K. & Woo, W.-c. (2015). Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: *Advances in neural information processing systems*. pp. 802–810.

Xu, B., Wang, N., Chen, T. & Li, M. (2015). Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*.

Yuan, Y., Bachl, F.E., Lindgren, F., Borchers, D.L., Illian, J.B. & Buckland, S.T. *et al.* (2017). Point process models for spatio-temporal distance sampling data from a large-scale survey of blue whales. *The Annals of Applied Statistics*, 11, 2270–2297.

Zamarreño, J.M. & Vega, P. (1998). State space neural network. Properties and application. *Neural networks*, 11, 1099–1112.

Zipkin, E.F., Rossman, S., Yackulic, C.B., Wiens, J.D., Thorson, J.T. & Davis, R.J. *et al.* (2017). Integrating count and detection–nondetection data to model population dynamics. *Ecology*, 98, 1640–1650.

Zucchini, W., MacDonald, I.L. & Langrock, R. (2017). *Hidden markov models for time series: An introduction using r*. Chapman; Hall/CRC.

Zwane, E. & Van der Heijden, P. (2004). Semiparametric models for capture–recapture studies with covariates. *Computational Statistics & Data Analysis*, 47, 729–743.

Appendix S1

This appendix includes example specifications for neural occupancy, N-mixture, and hidden Markov models. The goal in providing these specifications is to give concrete examples of neural hierarchical models that are relatively simple, while drawing connections to existing models. PyTorch implementations in Jupyter notebooks are available at https://github.com/mbjoseph/neuralecology.

Generally, the construction of a hierarchical model requires:

1. A **process model** that represents some ecological dynamics or states that can depend on unknown parameters.
2. An **observation model** that relates available data to the process model, possibly dependent on unknown parameters.
3. A **parameter model** for unknown quantities (e.g., their dependence on some input, or relationship to each other). Traditionally, this is discussed in terms of prior distributions, though there may be rich structure encoded in a parameter model as well (Wikle 2003). For a fully Bayesian hierarchical model, all unknowns are represented by probability distributions. In an empirical Bayesian hierarchical model, point estimates are typically used for top-level parameters instead, so that these top-level parameters are treated as fixed, but unknown (Cressie *et al.* 2009). In the examples below, neural networks are applied at the parameter model stage to learn a mapping from inputs to parameters of process and observation models.

Given these components, parameter estimation can proceed in a few ways, depending on the inferential framework used. A loss function can be constructed to find the parameter values that maximize the probability of the data, possibly with some penalties for model complexity (in a maximum likelihood/penalized maximum likelihood framework), find the most probable parameter values, conditional on the data (in a maximum *a posteriori* framework), or compute a probability distribution for all unknowns, conditional on the data (in a Bayesian framework). For many models applied in deep learning, it is often the case that fully Bayesian inference is complicated by high-dimensional multimodal posterior geometry.

The examples here focus primarily on penalized maximum likelihood methods, though Bayesian approaches stand out as a key area for future work (see main text for relevant citations). Often, L_2 norm penalties (also referred to as "weight decay") are applied on the parameters of neural networks in the loss function to penalize model complexity in an effort to avoid overfitting. Additional strategies include early stopping, where an iterative stochastic optimization scheme is terminated before the training set loss stabilizes (or when the validation set loss begins to increase), which is particularly useful for complex models that can overfit quickly. Finally, unlike maximum likelihood estimation for simple models, because of the multimodality of the likelihood surface for neural network parameters, there are no guarantees that a global maximum exists, or that a particular optimum is not a local maximum. In practice, this is not a major limitation. For more discussion, see Goodfellow *et al.* (2016).

A single-species single-season neural occupancy model

A single-species single-season occupancy model estimates presence/absence states from imperfect detection/non-detection data (MacKenzie *et al.* 2002). Assume that n spatial locations are each surveyed k times, during a short time interval for which it is reasonable to assume that the presence or absence of a species is constant. Each spatial location has some continuous covariate value represented by x_i for site $i = 1, ..., n$, that relates to occupancy and detection probabilities.

Observation model

Observations at site i consist of k surveys, where each survey results in a detection or non-detection. Let y_i represent the number of surveys at site i for which a species is detected, and z_i represent the true presence/absence state (if the species is present: $z_i = 1$; if absent: $z_i = 0$). Assuming that the probability of detecting a species on a survey conditional on presence is p_i, and that each survey is conditionally independent, the observations can be modeled with a Binomial distribution:

$$y_i \sim \text{Binomial}(z_i p_i, k).$$

Process model

The true presence/absence occupancy state z can be treated as a Bernoulli random variable, with occupancy probability ψ_i:

$$z_i \sim \text{Bernoulli}(\psi_i).$$

Parameter model

A simple approach to account for the effect of the site-level covariate on occupancy and detection would be to include a slope and intercept parameter specific to each component, using the logit function to ensure that estimated probabilities are bounded between 0 and 1:

$$\psi_i = \text{logit}^{-1}\big(\alpha^{(\psi)} + \beta^{(\psi)} x_i\big),$$
$$p_i = \text{logit}^{-1}\big(\alpha^{(p)} + \beta^{(p)} x_i\big),$$

where α^j and β^j are intercept and slope parameters for component j.

In contrast, a neural hierarchical model might instead account for site-level variation by modeling occupancy and detection probabilities as outputs of a neural network:

$$\begin{bmatrix} \psi_i \\ p_i \end{bmatrix} = f(x_i),$$

where $f(x_i)$ is a neural network that takes a scalar as input (the site-level covariate x_i) and outputs a two dimensional vector containing occupancy and detection probabilities:

$$f(x_i) = \text{logit}^{-1}\big(\mathbf{W}^{(2)} g(\mathbf{W}^{(1)} x_i)\big),$$

where $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$ are parameter matrices for the first and second layer, g is a differentiable nonlinear activation function, e.g., the rectified linear unit activation function (Nair & Hinton 2010), and the inverse logit transform is applied element-wise to ensure that ψ_i and p_i lie between 0 and 1.

Loss function

The negative log observed data likelihood can be taken as the loss function (MacKenzie *et al.* 2002). Marginalizing over z and treating sites as conditionally independent yields:

$$L(\theta_f) = -\sum_{i=1}^{n} \log \Big(\psi_i \text{Binomial}(y_i \mid p_i, k) + I(y_i = 0)(1 - \psi_i)\Big),$$

where θ_f represents the parameters of the neural network f, and $I(y_i = 0)$ is an indicator function equal to one when $y_i = 0$ (zero otherwise).

The loss for any particular site can be computed efficiently using the log-sum-exp trick, which is particularly useful when the log of summands is available (i.e., $\log(\psi_i) + \log\binom{k}{y_i} + y_i \log(p_i) + \log(k - y_i) + \log(1 - p_i)$ and $\log(1 - \psi_i)$ are already computed), as is the case in both Stan and PyTorch, which provide the log probability from the Binomial distribution.

A single-species neural dynamic occupancy model

A single-species dynamic occupancy model can be used to estimate rates of colonization and extinction when detection is imperfect and sites are repeatedly sampled at multiple time points (MacKenzie *et al.* 2003). Assume that for T timesteps, n spatial locations are each surveyed k times, during a short time interval for which it is reasonable to assume that the presence or absence state is constant. Among timesteps, the true occupancy states of sites can change. Each spatial location has some continuous covariate value represented by x_i for site $i = 1, ..., n$, that relates to occupancy and detection probabilities.

Observation model

Observations at site i in timestep t consist of k surveys, where each survey results in a detection or non-detection. Let $y_{i,t}$ represent the number of surveys at site i in timestep t for which a species is detected, and $z_{i,t}$ represent the true presence/absence state (if the species is present: $z_{i,t} = 1$; if absent: $z_{i,t} = 0$). Assuming that the probability of detecting a species on a survey conditional on presence is p_i, the observations can be modeled using a Binomial distribution:

$$y_{i,t} \sim \text{Binomial}(z_{i,t}p_i, k).$$

Process model

Sites can transition from being unoccupied ($z_{i,t} = 0$) to occupied ($z_{i,t+1} = 1$) due to colonization, or from being occupied ($z_{i,t} = 1$) to unoccupied ($z_{i,t+1} = 0$) due to extinction. Let the initial occupancy state be treated as a random Bernoulli variable with probability of occupancy $\psi_{i,1}$:

$$z_{i,1} \sim \text{Bernoulli}(\psi_{i,1}).$$

Subsequent occupancy dynamics at site i for timesteps $t = 1, ..., T$ are related to the probability of colonization (γ_i) and the probability of persistence (ϕ_i), where the extinction probability is taken to be the complement of persistence ($1 - \phi_i$).

$$z_{i,t} \sim \text{Bernoulli}(z_{i,t-1}\phi_i + (1 - z_{i,t-1})\gamma_i).$$

Parameter model

In this example, heterogeneity among sites was accounted for using a single layer neural network that ingests the one dimensional covariate x_i for site i, passes it through a single hidden layer, and outputs a four dimensional vector of probabilities containing the probabilities of initial occupancy ($\psi_{i,1}$), persistence (ϕ_i), colonization (γ_i), and detection (p_i):

$$\begin{bmatrix} \psi_{i,1} \\ \phi_i \\ \gamma_i \\ p_i \end{bmatrix} = f(x_i),$$

where f is a neural network. Concretely, f was parameterized as follows:

$$f(x_i) = \text{logit}^{-1}\Big(\mathbf{W}^{(2)} g(\mathbf{W}^{(1)} x_i)\Big),$$

where $\mathbf{W}^{(1)}$ is a parameter matrix that generates activations from the inputs, g is the ReLU activation function, $\mathbf{W}^{(2)}$ is a parameter matrix that maps the hidden layer to the outputs, and logit^{-1} is the element-wise inverse logistic (sigmoid) function.

Loss function

The negative log observed data likelihood was used as the loss function, implemented in PyTorch following the description in MacKenzie *et al.* (2003) (equation 5-6), assuming that detection histories for site $i = 1, ..., n$ are conditionally independent and scaling the forward probabilities to avoid underflow as described in Rabiner (1989).

A neural N-mixture model

An N-mixture model can be used to estimate latent integer-valued abundance when unmarked populations are repeatedly surveyed and it is assumed that detection of individuals is imperfect (Royle 2004). Assume that J spatial locations are each surveyed K times, in a short time interval for which it is reasonable to assume that the number of individuals is constant within locations $j = 1, ..., J$. Each spatial location has some continuous covariate value represented by x_j, that relates to detection probabilities and expected abundance.

Observation model

Observations at site j in survey k yield counts of the number of unique individuals detected, denoted $y_{j,k}$ for all j and all k. Assuming that the detection of each individual is conditionally independent, and that each individual is detected with site-specific probability p_j, the observations can be modeled with a Binomial distribution where the number of trials is the true (latent) population abundance n_j:

$$y_{j,k} \sim \text{Binomial}(p_j, n_j).$$

Process model

The true population abundance n_j is treated as a Poisson random variable with expected value λ_j:

$$n_j \sim \text{Poisson}(\lambda_j).$$

Parameter model

Heterogeneity among sites was accounted for using a single layer neural network that ingests the one dimensional covariate x_i for site i, passes it through a single hidden layer, and outputs a two dimensional vector containing a detection probability p_i and the expected abundance λ_i:

$$\begin{bmatrix} \lambda_i \\ p_i \end{bmatrix} = f(x_i),$$

where f is a neural network with two dimensional output activations $\mathbf{h}(x_i)$ computed via:

$$\mathbf{h}(x_i) = \mathbf{W}^{(2)} g(\mathbf{W}^{(1)} x_i),$$

and final outputs computed using the log and logit link functions for expected abundance and detection probability:

$$f(x_i) = \begin{bmatrix} \exp(h_1(x_i)) \\ \text{logit}^{-1}(h_2(x_i)) \end{bmatrix}.$$

Here too $\mathbf{W}^{(1)}$ is a parameter matrix that generates activations from the inputs, g is the rectified linear unit activation function, and $\mathbf{W}^{(2)}$ is a parameter matrix that maps the hidden layer to the outputs. Additionally $h_1(x_i)$ is the first element of the output activation vector, and $h_2(x_i)$ the second element.

Loss function

The negative log likelihood was used as the loss function, enumerating over a large range of potential values of the true abundance (from $\min(y_{j\cdot})$ to $5 \times \max(y_{j\cdot})$, where $y_{j\cdot}$ is a vector of counts of length K) to approximate the underlying infinite mixture model implied by the Poisson model of abundance (Royle 2004). It is also worth noting that alternative specifications based on a multivariate Poisson model are possible (Dennis *et al.* 2015).

A neural hidden Markov model: capture-recapture-recovery

Consider a capture-recapture-recovery study aimed at estimating time-varying parameters (King 2012). Assume that individuals $i = 1, ..., N$ are initially captured, marked, and released on timestep $t = 0$.

The state of an individual i on time step t is denoted $z_{i,t}$. Individuals are either alive ($z_{i,t} = 0$), recently dead such that their bodies are discoverable (and marks identifiable) ($z_{i,t} = 1$), or long dead such that their bodies are not discoverable and/or marks are no longer identifiable ($z_{i,t} = 2$).

Observation model

Observations $y_{i,t}$ are made on time steps $t = 1, ..., T$, and if an individual is detected alive on timestep t, $y_{i,t} = 1$, and if an individual is detected as recently dead $y_{i,t} = 2$, otherwise if the individual is not detected $y_{i,t} = 0$.

Let $\mathbf{\Omega}_t$ denote a time-varying emission matrix containing state-dependent observation probabilities:

$$\mathbf{\Omega}_t = \begin{bmatrix} 1 - p_t & p_t & 0 \\ 1 - \lambda_t & 0 & \lambda_t \\ 1 & 0 & 0 \end{bmatrix},$$

where p_t is the probability of detecting an individual on timestep t, conditional on the individual being alive, and λ_t is the probability of recovering a recently dead individual on timestep t that has died since timestep $t - 1$. The rows of the emission matrix correspond to states (alive, recently dead, long dead), and the columns correspond to observations (not detected, detected alive, detected dead).

Process model

At time t, the transition probability matrix $\mathbf{\Gamma}_t$ contains the probability of transitioning from row j to column k:

$$\mathbf{\Gamma}_t = \begin{bmatrix} \phi_t & 1 - \phi_t & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix},$$

where survival probability is $\phi_t = P(z_{i,t+1} = 0 \mid z_{i,t} = 0)$, dead individuals stay dead, and the rows and columns of the transition matrix correspond to states (alive, recently dead, long dead).

Parameter Model

Heterogeneity among timesteps was accounted for using three two-layer neural networks (one for ϕ, one for λ, and one for p). Each network ingests a univariate time series and outputs a corresponding time series of parameter values (i.e., each network maps a sequence of inputs $x = (x_{t=1}, ..., x_{t=T})'$ to a sequence of parameter values, e.g., $p = (p_{t=1}, ..., p_{t=T})'$ for the detection probability network:

$$\begin{aligned} \phi &= \text{logit}^{-1} f^{(\phi)}(\mathbf{x}), \\ \lambda &= \text{logit}^{-1} f^{(\lambda)}(\mathbf{x}), \\ \mathbf{p} &= \text{logit}^{-1} f^{(p)}(\mathbf{x}), \end{aligned}$$

where $f^{(j)}$ is a neural network for parameter j.

Loss function

The negative log observed data likelihood was used as the loss function, and computed using the forward algorithm (Zucchini *et al.* 2017). Observation histories for individuals were assumed to be conditionally independent. As an aside, one can specify such models in terms of the complete data likelihood (i.e., in terms of the hidden states) using programming frameworks that implement automatic enumeration of discrete latent variables with finite support, such as Pyro (Bingham *et al.* 2018).

Appendix S2

This appendix describes an animal movement model parameterized by a convolutional neural network, including simulations to explore how the amount of training data affects performance relative to simpler baseline models. Convolutional neural networks (CNNs) are widely used in computer vision applications, acting as function approximators that take an image as input and output a vector of probabilities that the image is a picture of a cat, dog, car, etc. (Rawat & Wang 2017). In ecological contexts, CNNs have been used to identify plants and animals (Norouzzadeh *et al.* 2018; Fricker *et al.* 2019; Tabak *et al.* 2019). However, CNNs considered more broadly as function approximators have additional uses. For example, instead of mapping an image to a vector of class probabilities, a CNN might map an image to a state transition probability matrix in a hidden Markov model, such as those used for animal movement trajectories (Zucchini *et al.* 2017).

Generally, the movement of an individual animal depends on the spatiotemporal context around their location at any time, and on their behavioral state. In many animal movement models, covariates for the state transitions are included using a linear combination of values followed by a softmax transformation to ensure probabilities within rows of a state transition probability matrix sum to one. This allows inference about spatiotemporal covariates that explain movement such as distance to water, temperature, hour of day, wind, and ocean surface current (Patterson *et al.* 2017; Johnson *et al.* 2018; McClintock & Michelot 2018a). In most cases, only the values of the covariates at point locations get used. However, it is reasonable to expect that the spatiotemporal context around these point locations could also be relevant.

Instead of extracting values at points, consider using gridded raster data around point locations such as image "chips" centered around observed point locations containing contextual data within some spatiotemporal window. Input rasters might contain satellite and/or aerial imagery, continuous or categorical landscape features (e.g., a digital elevation model or land cover map), or meteorological data (e.g., gridded temperature or wind speed data).

Scenario description

Consider an animal that prefers to forage in and around tree canopies, but frequently moves between canopies over bare ground. To generate a semi-realistic simulation of this scenario, movement trajectories were simulated over vegetation canopy height models from the San Joaquin Experimental Reserve, one of the core National Ecological Observatory Network (NEON) sites (Keller *et al.* 2008). These canopy height models are estimated using lidar data on the NEON Airborne Observation Platform (AOP) - a plane that flies over NEON sites regularly and collects a variety of data, including high resolution orthorectified aerial imagery (Fig. S2.1).

In this simulation, the "correct" covariate data that affects state transitions – which is rarely known in real systems – is the height of the tree canopy. Assume that canopy height is unknown, but aerial imagery (or high resolution satellite imagery) is available. This aerial red-green-blue (RGB) imagery should to relate in some way to canopy height, but the mapping from an RGB image to canopy height is likely complex. This provides an opportunity to test whether a convolutional neural hierarchical model can learn such a mapping using information about animal movement.

Simulating animal movement through tree canopies

Individual animal movement trajectories were simulated over a real canopy height model of the San Joaquin Experimental Reserve. The spatial region of interest is a 6 km by 5 km region, within which a 2018 NEON AOP mission was flown that generated a coregistered 1 m canopy height model (product code DP3.30015.001), and 10 cm high resolution orthorectified RGB camera reflectance data (product code DP3.30010.001). Due to some extreme outliers in the canopy height model, all values greater than 30m (~0.0025% of cells) were set to 30m. Then, the resulting raster was max-scaled (dividing by the maximum) to compress the range of values to the interval [0, 1].

An animal movement model with two states (foraging and in transit) was used to simulate trajectory data. Animals were more likely to forage where the canopy is high, and more likely to be in transit where canopy height is low.

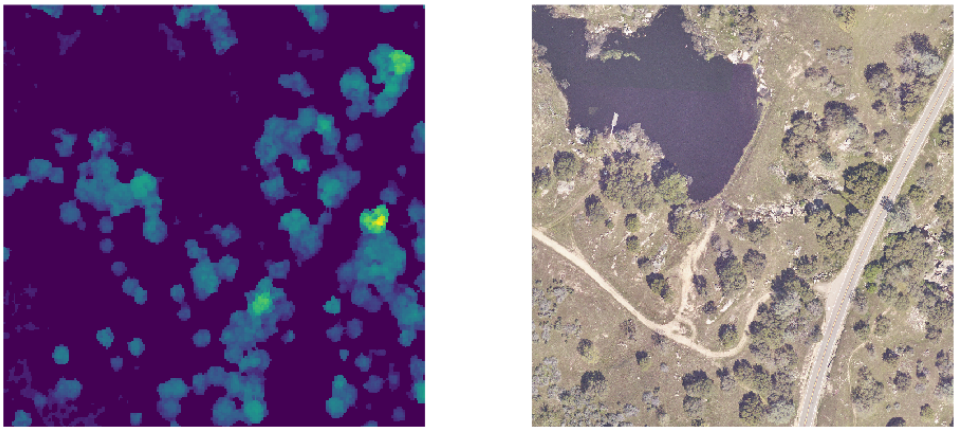

Figure S2.1: An example of a 1 m pixel resolution canopy height model (left) and corresponding high resolution (10 cm) orthorectified camera imagery over the San Joaquin Experimental Reserve. In the simulation, canopy height affects behavioral state transitions and subsequent animal movement, but the available data might consist only of aerial imagery. A 250 m by 250 m subset of the study area is displayed.

Behavioral states

Formally, consider a time series of length T containing the state of an animal at discrete times $t = 1, ..., T$, where $s_t = 1$ means the animal is "in transit" and $s_t = 2$ means the animal is "foraging" at time t. The state s_t is either 1 or 2 for any particular t. The probabilities of transitioning between states is time-varying, and is summarized in a matrix $\mathbf{\Gamma}^{(t)}$, which contains the transition probabilities $\gamma_{i,j}^{(t)}$ for states $i, j = 1, 2$ at time t. Each of these elements provides the probability of transitioning from one state to another, so that $\gamma_{i,j}^{(t)} = [s_{t+1} = j \mid s_t = i]$. For example $\gamma_{1,2}^{(t)}$ would provide the probability of transitioning from "in transit" in time t ($s_t = 1$) to "foraging" in time $t + 1$ ($s_{t+1} = 2$). At the first timestep, the state probabilities are contained in a row vector δ, where $\delta_i = [s_{t=1} = i]$, for states $i = 1$ and $i = 2$. In the simulation, the stationary state probabilities at randomly initialized starting locations were used as initial state probabilities.

To ensure that "foraging" was the more likely state in the canopy, and "in transit" was the most likely state over bare ground, the true state transition probabilities in the simulation were modeled as a logit-linear function of canopy height:

$$\text{logit}(\gamma_{1,2}^{(t)}) = -6 + 40x_t,$$
$$\text{logit}(\gamma_{2,1}^{(t)}) = 6 - 40x_t,$$

where x_t is the scaled canopy height at an animal's location in time t (Fig. S2.2A). This fully specifies the state transition matrix, because the rows must sum to one, implying for example that $\gamma_{1,1}^{(t)} = 1 - \gamma_{1,2}^{(t)}$.

State-dependent movement

The "foraging" and "in transit" behavioral states are associated with different movement patterns. Foraging is characterized by small step lengths with undirected turns. Movement trajectories for animals in transit are characterized by longer step lengths and more directed movements.

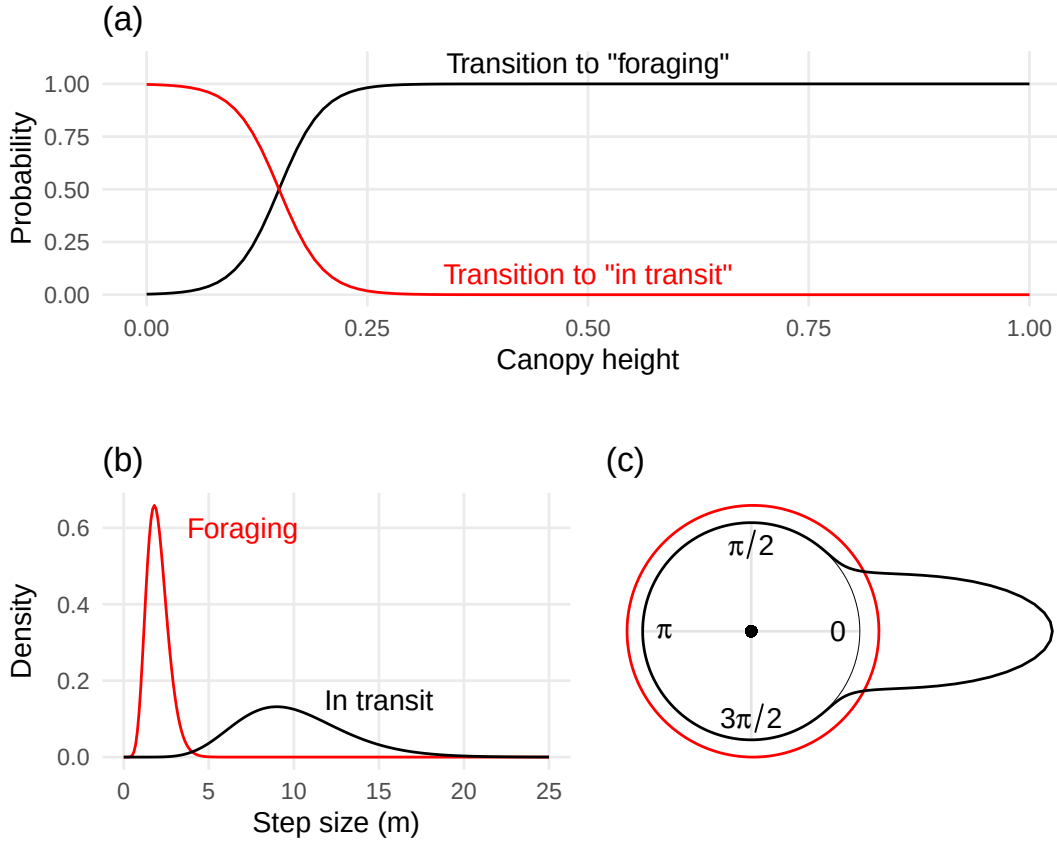

Figure S2.2: Panel A shows the true relationship between scaled canopy height (x-axis) and state transition probabilities in the simulation. Panels B and C show the densities of step sizes (B) and turn angles in radians (C) used in the simulation, colored by behavioral state.

Formally, if the vector $\mathbf{z}_t$ summarizes movement in the interval from time t to $t+1$, it is common to consider two quantities: the step size l_t and turning angle ϕ_t, so that $\mathbf{z}_t = (l_t, \phi_t)$ (Patterson *et al.* 2017). Thus, the movement model is a hidden Markov model with states $\{s_t\}_{t=1}^T$, transition probability matrices $\{\mathbf{\Gamma}^{(t)}\}_{t=1}^T$, and emissions $\{\mathbf{z}_t\}_{t=1}^T$.

In the simulation, step sizes were drawn from a gamma distribution with state-dependent parameters (Fig. S2.2B):

$$l_t \sim \begin{cases} \text{Gamma}(10, 1) & \text{if} \quad s_t = 1 \text{ ("in transit")} \\ \text{Gamma}(10, 5) & \text{if} \quad s_t = 2 \text{ ("foraging")} \end{cases},$$

for $t = 1, ..., T$. Turn angles were drawn from von Mises distributions with state-dependent parameters (Fig. S2.2C):

$$\phi_t \sim \begin{cases} \text{von Mises}(0, 20) & \text{if} \quad s_t = 1 \text{ ("in transit")} \\ \text{von Mises}(0, 0.1) & \text{if} \quad s_t = 2 \text{ ("foraging")} \end{cases},$$

for $t = 2, ..., T$. Initial movement directions (at $t = 1$) were randomly drawn from the uniform circular distribution, though strictly speaking these are not turn angles which require three points to compute.

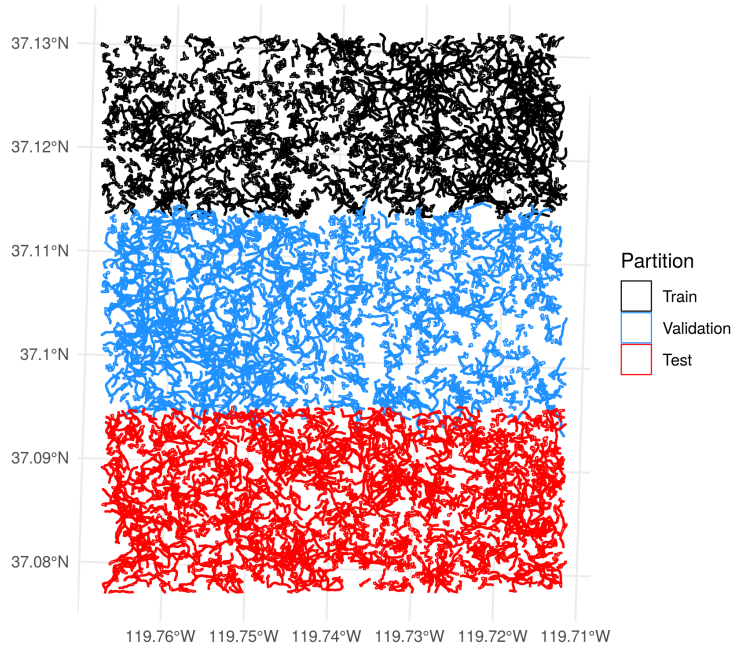

Figure S2.3: Map of the simulated trajectories in space, colored by dataset partition.

Trajectories simulated within the study area were partitioned into three sets based on the northing coordinate boundaries of the trajectory extents. Trajectories in the northern third of the study area were assigned to the training set, those in in the southern third were used as a withheld test set, and those in the middle third were used as a validation set (Fig. S2.3). For each partition, 1024 trajectories were simulated, to generate 3072 total trajectories across all partitions.

Model descriptions

Three models were developed, each of which maps a different set of inputs to transition probability matrices:

1. A **best case model** that takes canopy height as a state transition covariate. This represents the perfect scenario (unlikely in practice) where all relevant spatiotemporal information is provided to the model, and the generative model is correctly specified in every way. This best case model provides a useful upper bound on predictive performance.
2. A **point extraction model** that takes the RGB reflectance values from the aerial imagery extracted at point locations. This represents a more common scenario where covariate data indirectly related to the relevant spatiotemporal information (canopy height) are extracted at point locations. This model is likely to perform poorly, as the RGB reflectance at a point location may not contain much information about canopy height.
3. A **convolutional hidden Markov model** that takes image chips centered on point locations as input, and maps these image chips to transition matrices using a convolutional neural network. If RGB image chips contain more information about canopy height than simple RGB point extractions and sufficient training data are available, this model should perform better than the point extraction model.

Best case model

The best case model is the generative model for simulated trajectories. The relationship between canopy height and state transition probabilities is logit linear (as in the simulation), and intercept and slope parameters are estimated:

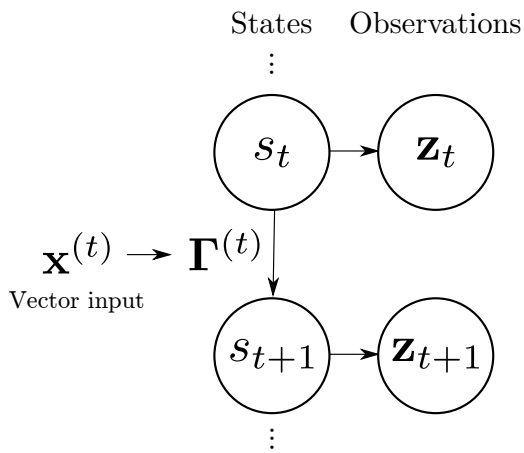

Figure S2.4: Graphical representation of a generic hidden Markov models for animal movement. The behavioral state s_t is associated with a state transition probability matrix $\mathbf{\Gamma}^{(t)}$, with observations $\mathbf{z}_t$ that represent the movement trajectory of the animal. Inputs contained in a vector $\mathbf{x}^{(t)}$ are mapped to the transition probability matrix by a function (usually a linear combination on a transformed scale).

$$\text{logit}(\gamma_{1,2}^{(t)}) = \alpha_{1,2} + \beta_{1,2}x_t,$$
$$\text{logit}(\gamma_{2,1}^{(t)}) = \alpha_{2,1} + \beta_{2,1}x_t.$$

Here as before $\gamma_{i,j}^{(t)}$ is the probability of transitioning from state $s_t = i$ to $s_{t+1} = j$. Intercept terms are represented by $\alpha_{i,j}$, and slopes by $\beta_{i,j}$, with x_t representing scaled canopy height. This model uses covariates for transition probabilities in the same way that many do: using a linear combination on a transformed scale (Fig. S2.4).

Point extraction model

The point extraction model includes parameters to map the RGB image reflectance values (r_t, g_t, and b_t) to the transition probabilities using a linear combination on the logit scale:

$$\text{logit}(\gamma_{1,2}^{(t)}) = \alpha_{1,2} + \beta_{1,2}^{r}r_t + \beta_{1,2}^{g}g_t + \beta_{1,2}^{b}b_t,$$

$$\text{logit}(\gamma_{2,1}^{(t)}) = \alpha_{2,1} + \beta_{2,1}^{r}r_t + \beta_{2,1}^{g}g_t + \beta_{2,1}^{b}b_t,$$

where $\beta_{i,j}^{k}$ is a coefficient for the $(i,j)^{th}$ transition probability and image band k. This model is also representative of the common approach taken to include covariates in such models - via a linear combination on a transformed scale (Fig. S2.4),

Convolutional hidden Markov model

The convolutional hidden Markov model is a neural hierarchical model that maps an image chip $\mathbf{X}^{(t)}$ centered on an animal's location at time t to a transition probability matrix $\mathbf{\Gamma}^{(t)}$. This is a departure from the previous two models. The input $\mathbf{X}^{(t)}$ is a multidimensional array instead of a real number (in the best case model) or a numeric vector (containing RGB reflectances in the point extraction model).

To generate image chip input arrays, square crops from the aerial RGB imagery were created centered on each simulated location. The spatial footprint of each chip was 128×128 pixels (≈ 13 m $\times$ 13 m). This created a $3 \times 128 \times 128$ array for each point location along each trajectory, where the three channels correspond to reflectance values in the red, green, and blue spectral bands (Fig. S2.5).

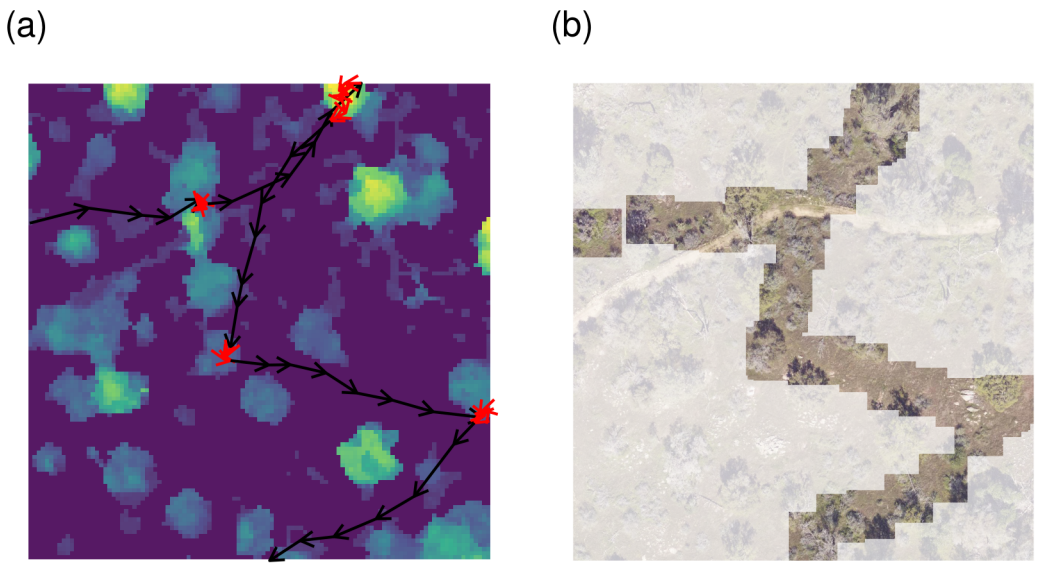

Figure S2.5: An example of a simulated movement trajectory. Panel (a) shows the true states and scaled canopy height, which determines the state transition probabilities. Red indicates that the animal is foraging, black indicates that it is in transit. The background color map shows the scaled canopy height. Panel (b) shows 128 by 128 image chips centered on point locations in the high resolution aerial imagery, which are used as inputs in the convolutional hidden Markov model of animal movement. Camera imagery outside of these image chips is shown in a lighter shade.

To illustrate the potential for including additional raster data in addition to imagery, an additional band was concatenated to each chip which contained zeros everywhere except for a 2×2 region in the center of the 128×128 grid, generating $4 \times 128 \times 128$ arrays. In real applications, this might represent additional raster data relevant to movement – inputs need not be images per se.

The convolutional hidden Markov model for animal movement mapped these $4 \times 128 \times 128$ image chips to 2×2 state transition probability matrices $\{\mathbf{\Gamma}^{(t)}\}_{t=1}^{T}$ (Fig. S2.6). The architecture of the convolutional neural network is a simplified version of the AlexNet model that plays an important role in the history of deep learning in computer vision (Krizhevsky 2014), though more modern architectures might perform better. Briefly, the input image is passed through a series of 2-dimensional convolutions, followed by nonlinear activation functions, followed by 2d max-pooling layers, creating a $64 \times 2 \times 2$ lower spatial resolution array with many "channels". This three dimensional array is flattened to a one dimensional array, creating a vector of length $64 \times 2 \times 2$, which is then passed to a series of fully connected hidden layers with nonlinear activations and dropout regularization (Srivastava *et al.* 2014) to create a vector of length 4. This vector is reshaped to a 2×2 matrix, then a softmax transformation is applied row-wise to ensure that the row probabilities sum to one (as they should in a state transition probability matrix). The resulting 2×2 matrix is the transition probability matrix $\mathbf{\Gamma}^{(t)}$, generated from the input $\mathbf{X}^{(t)}$.

The precise structure of the convolutional neural network that maps the input raster to the state transition probability matrix was as follows (in PyTorch-like psuedocode):

```
Sequential(
  Conv2d(4, 16, kernel_size=9, stride=3),
  LeakyReLU(),
```

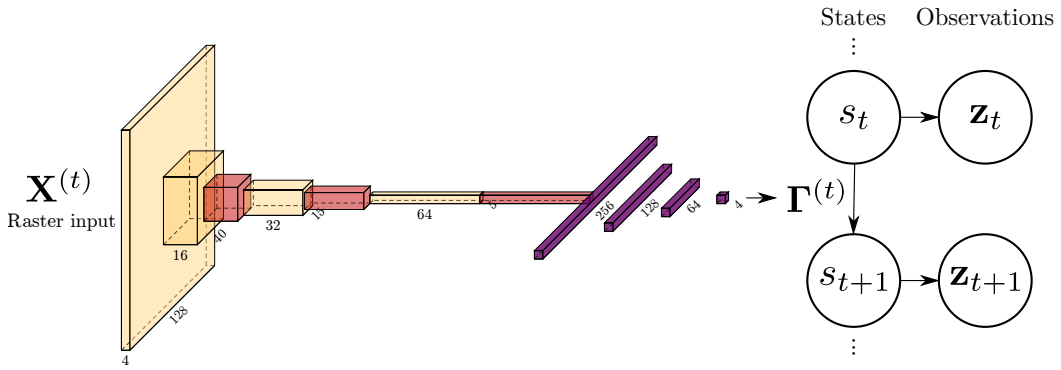

Figure S2.6: A convolutional neural network that maps a raster (in this case a $4 \times 128 \times 128$ grid) to a state transition probability matrix of a hidden Markov model. Yellow boxes indicate input arrays and outputs from convolutional layers, with labeled dimensions. Red boxes represent two dimensional maximum pooling layers. Purple boxes represent fully connected hidden layers. The final vector of length 4 is reshaped to a 2 by 2 matrix, and then a softmax transform is applied row-wise to ensure that the rows sum to one for the transition probability matrix at timestep t, denoted $\mathbf{\Gamma}^{(t)}$.

```
  MaxPool2d(kernel_size=3, stride=2),
  Conv2d(16, 32, kernel_size=5),
  LeakyReLU(),
  MaxPool2d(kernel_size=3, stride=2),
  Conv2d(32, 64, kernel_size=3),
  LeakyReLU(),
  MaxPool2d(kernel_size=3, stride=2)
  view(-1), # flatten into a vector
  Dropout(),
  Linear(64 * 2 * 2, 128),
  LeakyReLU(),
  Dropout(),
  Linear(128, 64),
  LeakyReLU(),
  Dropout(),
  Linear(64, 4),
  view(2, 2),
  softmax(dim=-1)
)
```

In addition to using dropout to regularize the convolutional model, an L_2 penalty of 10^{-5} was also applied. To further regularize the model, image augmentation (random horizontal and vertical flips) was also applied while training, though this might not be desirable in cases where directional orientation of the input image is important (Simonyan & Zisserman 2014). Image augmentation generally includes strategies to perturb data while training computer vision models in an attempt to generate a robust model (i.e., one that is insensitive to translation, orientation, hue, contrast, etc.).

In contrast to more common applications of convolutional neural networks where individual images are labeled (e.g., with bounding boxes and species identities in camera trap imagery), the observed movement trajectories (step sizes and turning angles) can be thought of as analogous to implicit "labels" for this convolutional hidden Markov model.

Model comparisons

The predictive performance of the three models (best case, point extraction, and convolutional) was compared across a gradient of training set sizes. This gradient included datasets that consisted of 16, 32, 64, 128, 256, 512, and 1024 simulated trajectories, each of which consisted of 50 timesteps.

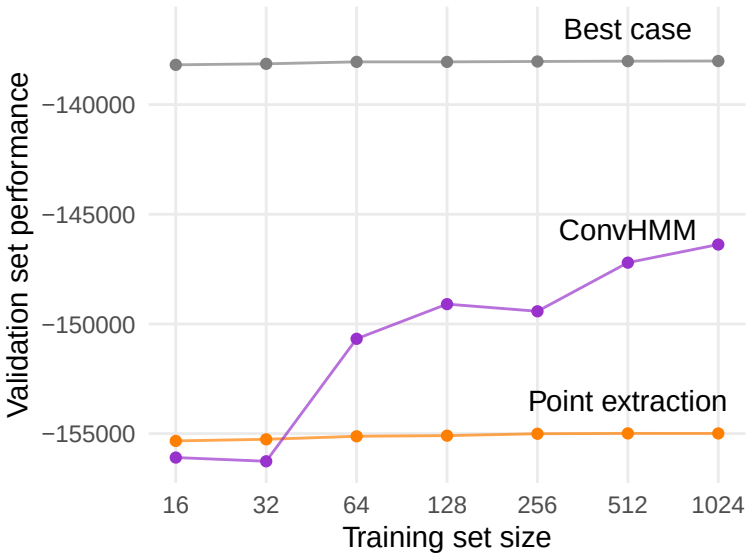

Figure S2.7: Model performance on withheld validation data. The x-axis is the number of movement trajectories in the training data. The y-axis shows the performance (predictive log-likelihood) on withheld validation data. Each point is the result of a simulation, and colored lines connect results for each model. ConvHMM is the convolutional hidden Markov model of animal movement.

The smaller datasets were subsets of the larger datasets. For each model/training set combination, predictive performance was evaluated using the out of sample log-likelihood, computed for the 1024 withheld validation trajectories using the forward algorithm (Patterson *et al.* 2017). After using the validation data to compare models, the preferred model was retrained using the training and validation data, and final predictive performance was evaluated on the still withheld test set.

Results

As the training set increased in size to more than 64 trajectories, the convolutional movement model's performance exceeded the point extraction baseline (Fig. S2.7). When the training set was relatively small, the convolutional movement model performed worse than the point extraction baseline. Validation set performance continued to increase as training data were added for all modles. The rate of increase was greatest for the convolutional model, however, and validation set performance did not appear to saturate even for the largest training data set (Fig. S2.7). As expected, the best case model, which is correctly specified and has access to the "correct" input data (canopy height) always had the highest performance. But, restricting attention to models that only have access to the RGB imagery, the convolutional movement model was preferred.

Final predictive checks on the withheld test set indicated that the convolutional model was able to estimate the true state transition probabilities fairly well. Transition probabilities from "in transit" ($s_t = 1$) to "foraging" ($s_t = 2$) were not captured as well as transitions from "foraging" ($s_t = 2$) to "in transit" ($s_t = 1$) (Fig. S2.8). In particular, the estimated distribution of transition probabilities was bimodal, but not as sharply peaked as the distribution of true transition probabilities.

Last, to provide a qualitative sense of what the final model had learned, test set image chips with the highest predicted transition probabilities are visualized in Fig. S2.9. Consistent with the underlying generative model, image chips centered on tree canopies were associated with the highest probabilities of transitioning into the "foraging" state, and image chips centered on bare ground (often with intermittent rocks) were associated with the highest probabilities of transitioning into the "in transit" state.

Taken together, these results indicate that simpler models might perform better when limited training data are available, but that neural hierchical models might provide predictive performance improvements for large datasets.

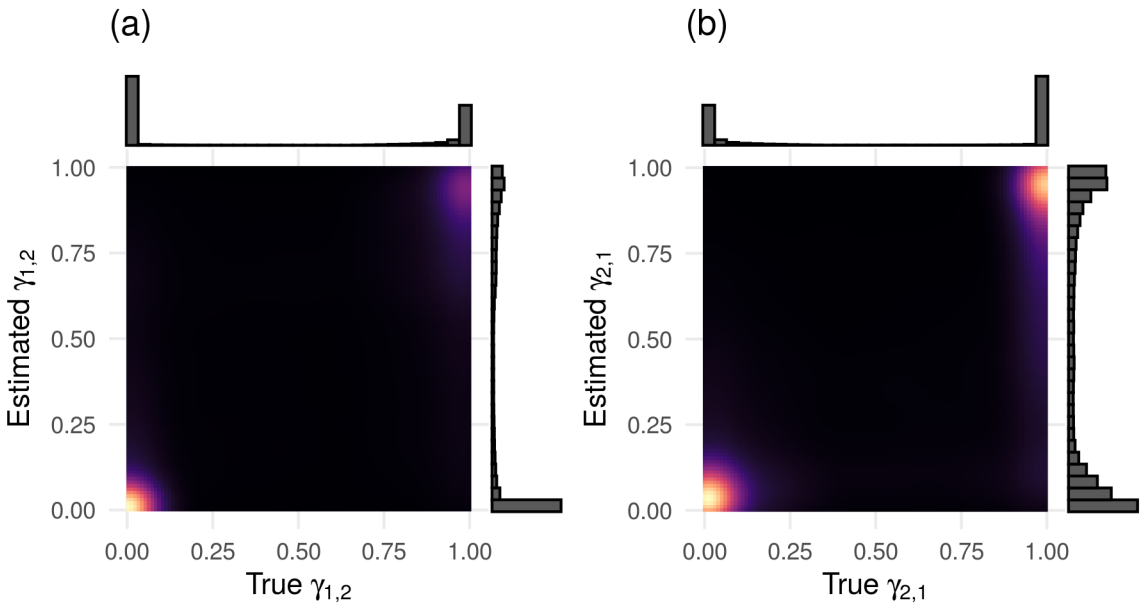

Figure S2.8: Joint densities of true (x-axis) and estimated (y-axis) state transition probabilities in the final convolutional movement model for the withheld test set. Panel (a) shows the distribution for $\gamma_{1,2}$ (transitions to "foraging"), with marginal histograms for the true and estimated probabilities. Panel (b) shows the same for $\gamma_{2,1}$ (transitions to "in transit"). Cell color represents the density of observations, with brighter colors indicating higher densities.

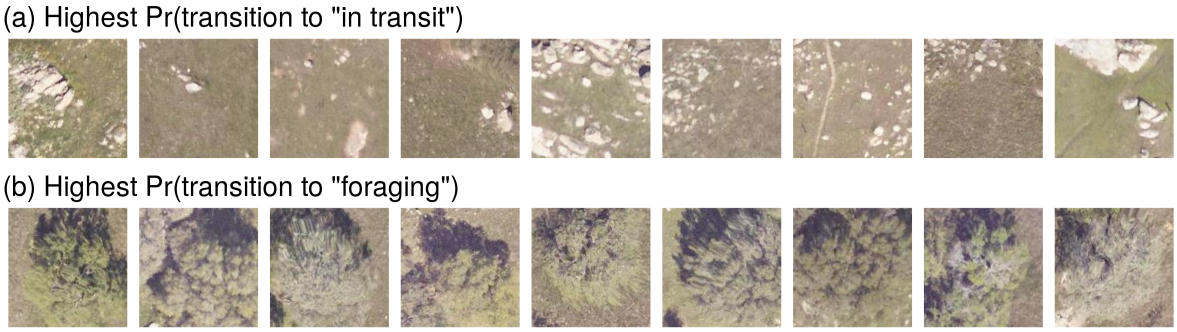

Figure S2.9: Top nine test set image chips with the highest probabilities of transitioning (a) from a "foraging" state to "in transit", and (b) from "in transit" to "foraging".

Implementation notes

The best case and point extraction models were both fit using the `momentuHMM` R package (McClintock & Michelot 2018b). The convolutional movement model was implemented with PyTorch (Paszke *et al.* 2017). All code required to reproduce the analysis is available on GitHub at https://www.github.com/mbjoseph/neuralecology.

Appendix S3

This appendix includes details on the structure and implementation of the baseline and neural dynamic occupancy models for breeding bird survey data.

Baseline model structure

Process model

The baseline model was a hierarchical Bayesian dynamic occupancy model, fit separately to each species. Let $s = 1, ..., S$ index survey routes, $t = 1, ..., T$ index years, and $j = 1, ..., J$ index bird species. Dropping subscripts for species j, for any particular species' model, presence/absence states are characterized by the following dynamics:

$$z_{t=1,s} \sim \text{Bernoulli}(\psi_{t=1,s}),$$

$$z_{t,s} \sim \text{Bernoulli}(\phi_s z_{t-1,s} + \gamma_s(1 - z_{t-1,s})), \quad \text{for } t = 2, ..., T,$$

where $\psi_{t,s}$ is the probability of occurrence at route s in year t, ϕ_s is the probability of persistence conditional on presence in the previous year, and γ_s is the probability of colonization conditional on absence in the previous year.

Observation model

Detection/non-detection data arise via a binomial observation process:

$$y_{t,s} \sim \text{Binomial}(p_{t,s} z_{t,s}, 50),$$

for all s and t, where the binomial sample size 50 arises from having 50 stops along each route.

Parameter model

Heterogeneity in occupancy dynamics was introduced in the model via additive adjustments for EPA level one ecoregions and route-level characteristics (centered and scaled to have mean zero and unit variance):

$$\text{logit}(\psi_{t=1,s}) = \mathbf{X}_s \beta^{(\psi_1)} + \alpha^{(\psi_1)}_{r[s]},$$

where $\mathbf{X}_s$ is row s from the design matrix $\mathbf{X}$ containing the route-level features, $\beta^{(\psi_1)}$ is a parameter vector associated with initial occupancy, and EPA level one adjustments are denoted $\alpha^{(\psi_1)}_{r[s]}$, where $r[s]$ represents the region r containing route s. Similarly, persistence and colonization probabilities are modeled as:

$$\text{logit}(\phi_s) = \mathbf{X}_s \beta^{(\phi)} + \alpha^{(\phi)}_{r[s]},$$

$$\text{logit}(\gamma_s) = \mathbf{X}_s \beta^{(\gamma)} + \alpha^{(\gamma)}_{r[s]}.$$

Heterogeneity in detection probability was included as:

$$\text{logit}(p_{t,s}) = \mathbf{X}^{(p)}_{(s,t)} \beta^{(p)} + \alpha^{(p)}_{r[s]},$$

where $\mathbf{X}^{(p)}_s$ is a augmented version of $\mathbf{X}_s$ that adds survey-level features related to detection probability (survey duration, along with start and end sky, temperature, and wind conditions, which vary by route and year), $\beta^{(p)}$ is a coefficient vector, and $\alpha^{(p)}_{r[s]}$ is an ecoregion adjustment.

Prior distributions

Prior distributions were constructed to facilitate borrowing of information across the four parameter dimensions of initial occupancy, persistence, colonization, and detection. Ecoregion adjustments were assigned multivariate normal priors:

$$\begin{bmatrix} \alpha_r^{(\psi_1)} \\ \alpha_r^{(\phi)} \\ \alpha_r^{(\gamma)} \\ \alpha_r^{(p)} \end{bmatrix} \sim \text{MultivariateNormal}\big(0, \boldsymbol{\Sigma}\big),$$

for regions $r = 1, ..., R$. The covariance matrix $\boldsymbol{\Sigma}$ was constructed for each level as $\boldsymbol{\Sigma} = \text{diag}(\sigma)\boldsymbol{\Omega}\text{diag}(\sigma)$, where σ is a vector of length 4, $\text{diag}(\sigma)$ is a 4×4 diagonal matrix with entries equal to σ, and $\boldsymbol{\Omega}$ is a 4×4 correlation matrix.

Prior distributions were specified as follows:

$\sigma \sim \text{Gamma}(1.5, 10)$,
$\boldsymbol{\Omega} \sim \text{LKJ-Correlation}(10)$,
$\beta^{(\psi_1)} \sim \text{Normal}(0, 1)$,
$\beta^{(\phi)} \sim \text{Normal}(0, 1)$,
$\beta^{(\gamma)} \sim \text{Normal}(0, 1)$,
$\beta^{(p)} \sim \text{Normal}(0, 1)$.

Posterior distribution

The posterior distribution is proportional to:

$$\prod_t \prod_s \text{Binomial}(y_{t,s} \mid z_{t,s} p_{t,s}, 50) \times$$
$$\prod_s \text{Bernoulli}(z_{t=1,s} \mid \psi_{t=1,s}) \prod_{t=2}^{T} \text{Bernoulli}(z_{t,s} \mid \phi_s z_{t-1,s} + \gamma_s(1 - z_{t-1,s})) \times$$
$$\prod_r [\alpha_r \mid \sigma, \boldsymbol{\Omega}] \times$$
$$[\beta^{(\psi_1)}][\beta^{(\phi)}][\beta^{(\gamma)}][\beta^{(p)}][\sigma][\boldsymbol{\Omega}].$$

Single species neural hierarchical model structure

The single species neural hierarchical model is a dynamic occupancy model parameterized by a neural network.

Process model

Presence/absence states are characterized by the following dynamics:

$$z_{t=1,s} \sim \text{Bernoulli}(\psi_{t=1,s}),$$

$$z_{t,s} \sim \text{Bernoulli}(\phi_{t-1,s} z_{t-1,s} + \gamma_{t-1,s}(1 - z_{t-1,s})), \quad \text{for } t = 2, ..., T,$$

where $\psi_{t,s}$ is the probability of occurrence at route s in year t, $\phi_{t-1,s}$ is the probability of persistence conditional on presence in the previous year: $\Pr(z_{t,s} = 1 | z_{t-1,s} = 1) = \phi_{t-1,s}$, and $\gamma_{t-1,s}$ is the probability of colonization conditional on absence in the previous year: $\Pr(z_{t,s} = 1 | z_{t-1,s} = 0) = \gamma_{t-1,s}$.

Observation model

Detection data are modeled using a binomial distribution:

$$y_{t,s} \sim \text{Binomial}(z_{t,s}p_{t,s}, 50),$$

for all s, and t.

Parameter model

Heterogeneity in occupancy and observation parameters was included by modeling initial occupancy, persistence, colonization, and detection probabilities as outputs of a neural network (Fig. S3.1). Categorical inputs to the network included EPA level one ecoregions, which were mapped to 8 dimensional numeric vector embeddings, $\mathbf{h}_r^{(L_1)}$ for regions $r = 1, ..., R$ (Guo & Berkhahn 2016). Categorical entity embeddings are equivalent to using one layer of a neural network with 8 dimensional output, where the inputs are one-hot encoded ecoregion codes, but in practice using an embedding layer is more computationally efficient because it capitalizes on the sparsity of a one-hot encoding.

For example, let R represent the number of level 1 ecoregions containing BBS routes ($R = 14$). Let $\mathbf{x}_{r[s]}^{(L_1)}$ represent a one-hot encoded vector of length R, where all entries are zero except for the index of the level 1 ecoregion containing route s (denoted $r[s]$), which is set equal to one. Then, let $\mathbf{W}^{(L_1)}$ represent a parameter matrix of size $D^{(L_1)} \times R$, where $D^{(L_1)}$ is the dimensionality of the embedding ($D^{(L_1)} = 8$). Note that one-hot encodings are often used linear models, where categorical covariates are coded as dummy variables, and this is a special case where the embeddings are one dimensional. Then, the embedding $\mathbf{h}_{r[s]}^{(L_1)}$ for the level 1 ecoregion r containing route s is therefore a vector with $D^{(L_1)}$ elements, given by:

$$\mathbf{h}_{r[s]}^{(L_1)} = \mathbf{W}^{(L_1)}\mathbf{x}_{r[s]}^{(L_1)},$$

which essentially corresponds to extracting column r from the embedding matrix $\mathbf{W}^{(L_1)}$.

Embeddings for route s are concatenated with the numeric route-level features $\mathbf{x}_s$ (a vector containing climate principal components, road density, distance from coast, latitude, and longitude) to obtain a vector for route s that is the zeroth hidden layer of the neural network $\mathbf{h}_s^{(0)}$:

$$\mathbf{h}_s^{(0)} = \begin{bmatrix} \mathbf{h}_{r[s]}^{(L_1)} \\ \mathbf{x}_s \end{bmatrix}.$$

This vector $\mathbf{h}_s^{(0)}$ is mapped to next hidden layer with $D^{(1)} = 4$ hidden units via a fully-connected layer, followed by leaky rectified linear unit activation functions (Xu *et al.* 2015). The leaky rectified linear unit activation function is $g(x) = \max(0, x) - 0.01 \times \min(0, x)$. Thus, the next hidden layer is obtained in a forward pass via:

$$\mathbf{h}_s = g(\mathbf{W}^{(1)}\mathbf{h}_s^{(0)}),$$

where $\mathbf{W}^{(1)}$ is a $D^{(1)} \times D^{(0)}$ parameter matrix. For the single species models, $D^{(0)} = 21$, because there is one embedding of size 8, and an additional 13 elements in the route-level covariate vector $\mathbf{x}_s$ (eight principal component axes, elevation, road density, distance from coast, latitude, and longitude).

The hidden layer is then mapped to initial occupancy, persistence, colonization, and detection probabilities, via fully connected layers:

$$\begin{aligned} \text{logit}(\psi_{s,t=1}) &= \mathbf{W}^{(\psi_1)}\mathbf{h}_s, \\ \text{logit}(\gamma_s) &= \mathbf{W}^{(\gamma)}\mathbf{h}_s, \\ \text{logit}(\phi_s) &= \mathbf{W}^{(\phi)}\mathbf{h}_s, \end{aligned}$$

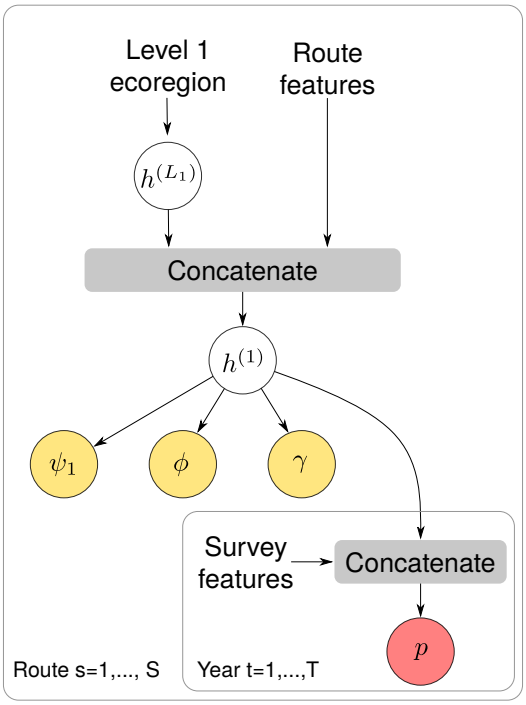

Figure S3.1: Extended computational diagram for the single species neural dynamic occupancy model. Outer grey boxes indicate the different levels of the model (route and year) that index quantities inside the boxes. Yellow nodes indicate occupancy parameters, and red nodes indicate detection parameters. Hidden layers are represented by h, with layer-specific superscripts. Outputs include initial occupancy (ψ_1), persistence (ϕ), colonization (γ), and detection probabilities (p).

$$\text{logit}(p_{t,s}) = \mathbf{W}^{(p)}(\mathbf{h}_s, \mathbf{x}_{t,s}^{(p)})' \quad \text{for } t = 1, ..., T.$$

Here $\mathbf{W}^{(\psi_1)}$, $\mathbf{W}^{(\gamma)}$, and $\mathbf{W}^{(\phi)}$ are 4×1 parameter matrices, and $\mathbf{W}^{(p)}$ is a $(4 + n_p) \times 1$ parameter matrix that maps the concatenation of detection-related hidden unit vector $\mathbf{h}_s^{(p)}$ with n_p survey specific features contained in the vector $\mathbf{x}_{t,s}^{(p)}$ to the detection probability $p_{t,s}$.

Multi-species neural hierarchical model structure

Process model

The process model for the multi-species model extends the single-species models to the multiple species case. Presence/absence states $z_{t,s,j}$ for all s, t, and j are equal to 0 if species j is absent from route s in time t, and $z_{t,s,j} = 1$ if the species is present. Occupancy dynamics are modeled as a function of initial occupancy, persistence, and colonization probabilities (Royle & Kéry 2007):

$$z_{t=1,s,j} \sim \text{Bernoulli}(\psi_{t=1,s,j}),$$

$$z_{t,s,j} \sim \text{Bernoulli}(\phi_{t-1,s,j} z_{t-1,s,j} + \gamma_{t-1,s,j}(1 - z_{t-1,s,j})), \quad \text{for } t = 2, ..., T,$$

where $\psi_{t,s,j}$ is the probability of occurrence of species j at route s in year t, $\phi_{t-1,s,j}$ is the probability of persistence conditional on presence in the previous year, and $\gamma_{t-1,s,,j}$ is the probability of colonization conditional on absence in the previous year.

Observation model

The observation model similarly is a multi-species extension of the single species observation models. The observations $y_{t,s,j}$ are integer-valued counts in the set $0, 1, 2, ..., 50$ that indicate the number of stops for which species j was detected in year t on route s. Detection/non-detection data are modeled as binomial random variables:

$$y_{t,s,j} \sim \text{Binomial}(z_{t,s,j} p_{t,s,j}, 50),$$

for all s, t, and j.

Parameter model

Route representations

Heterogeneity in occupancy dynamics was introduced by allowing initial colonization, persistence, colonization, and detection probabilities to relate to latent spatiotemporal feature vectors (Fig. S3.2). These latent spatiotemporal feature vectors represent a nonlinear combination of route-level inputs. For the multi-species model, all feature vectors are 64 dimensional. Hidden layers are 32 dimensional unless stated otherwise.

Categorical inputs to the network included EPA level one ecoregion, mapped to categorical entity embeddings (Guo & Berkhahn 2016). As with the single-species models:

$$\mathbf{h}_{r[s]}^{(L_1)} = \mathbf{W}^{(L_1)} \mathbf{x}_{r[s]}^{(L_1)},$$

where $\mathbf{h}_{r[s]}^{(L_1)}$ is the embedding for a level one ecoregion r containing route s, $\mathbf{W}^{(L_1)}$ is a parameter matrix, and $\mathbf{x}_{r[s]}^{(L_1)}$ is a one-hot encoding for the level one ecoregion r containing route s. Concatenating the ecoregion embedding with route-level features $\mathbf{x}_s$ for route s provides the zeroth hidden layer of the network.

$$\mathbf{h}_s^{(0)} = \begin{bmatrix} \mathbf{h}_{r[s]}^{(L_1)} \\ \mathbf{x}_s \end{bmatrix}.$$

This zeroth hidden layer $\mathbf{h}_s^{(0)}$ that combines an ecoregion embedding and route-level features is then passed to a sequence of hidden layers:

$$\begin{aligned} \mathbf{h}_s^{(1)} &= g(\mathbf{W}^{(1)} \mathbf{h}_s^{(0)}), \\ \mathbf{h}_s^{(2)} &= g(\mathbf{W}^{(2)} \mathbf{h}_s^{(1)}), \\ \mathbf{h}_s^{(3)} &= g(\mathbf{W}^{(3)} \mathbf{h}_s^{(2)}), \end{aligned}$$

where g is the leaky ReLU activation function, so that $\mathbf{h}_s^{(3)}$ is a vector valued nonlinear combination of route-level features.

This hidden layer is then mapped to parameter-specific hidden layers. The initial occupancy probability hidden layer uses a fully connected layer:

$$\mathbf{h}_s^{(\psi_1)} = g(\mathbf{W}^{(\psi_1)} \mathbf{h}_s^{(3)}).$$

Colonization, persistence, and detection probability hidden layers were allowed to vary in space and time and modeled using a recurrent neural network. Temporal variation was modeled by treating $\mathbf{h}_s^{(3)}$ as a route-level encoding, that is decoded by a two-layer gated recurrent unit – a particular type of sequence model often used in time series analysis and text modeling, similar to a long short term memory model (Chung *et al.* 2014). The gated recurrent unit takes the vector $\mathbf{h}_s^{(3)}$ as an input, and outputs a multivariate sequence of shape $T \times (32 \times 3)$, which is reshaped into a $T \times 32 \times 3$

dimensional array that has dimensions for years, hidden features, and model components (persistence, colonization, and detection).

$$\begin{bmatrix} \mathbf{h}_{1:T}^{(\phi)} \\ \mathbf{h}_{1:T}^{(\gamma)} \\ \mathbf{h}_{1:T}^{(p)} \end{bmatrix}_s = \text{GRU}(\mathbf{h}_s^{(3)}),$$

where the subscript $1:T$ indicates that values are generated for timesteps $t = 1, ..., T$. These latent spatiotemporal route features are then combined with species-specific parameters generated via deep multi-species embedding (Chen *et al.* 2016).

Hierarchical deep multi-species embedding

Species-specific parameters were modeled as outputs of a neural network that ingests species-level traits (in this case, species identity, genus, family, and order). Taxonomic embeddings enable information to be shared among species within genera, families, and orders. A vector valued embedding $\mathbf{v}_j$ for species j is generated by concatenating embeddings at each taxonomic level:

$$\begin{aligned} \mathbf{h}_j &= \mathbf{W}^{(\text{Species})}\mathbf{x}_j, \\ \mathbf{h}_j^{(g)} &= \mathbf{W}^{(\text{Genus})}\mathbf{x}_j^{(g)}, \\ \mathbf{h}_j^{(f)} &= \mathbf{W}^{(\text{Family})}\mathbf{x}_j^{(f)}, \\ \mathbf{h}_j^{(o)} &= \mathbf{W}^{(\text{Order})}\mathbf{x}_j^{(o)}, \end{aligned}$$

$$\mathbf{v}_j = \begin{bmatrix} \mathbf{h}_j \\ \mathbf{h}_j^{(g)} \\ \mathbf{h}_j^{(f)} \\ \mathbf{h}_j^{(o)} \end{bmatrix}.$$

Here $\mathbf{h}_j$ is the vector valued embedding for species j, $\mathbf{W}^{(\text{Species})}$ is a parameter matrix, $\mathbf{x}_j$ is a one hot encoded vector, $\mathbf{h}_j^{(g)}$ is the vector valued embedding for the genus containing species j, etc. The vector $\mathbf{v}_j$ is mapped to parameter-specific hidden units via fully connected layers:

$$\begin{aligned} \mathbf{h}_j^{(\psi_1^*)} &= g(\mathbf{W}^{(\psi_1^*)}\mathbf{v}_j), \\ \mathbf{h}_j^{(\phi^*)} &= g(\mathbf{W}^{(\phi^*)}\mathbf{v}_j), \\ \mathbf{h}_j^{(\gamma^*)} &= g(\mathbf{W}^{(\gamma^*)}\mathbf{v}_j), \\ \mathbf{h}_j^{(p^*)} &= g(\mathbf{W}^{(p^*)}\mathbf{v}_j). \end{aligned}$$

These hidden layers are then mapped to species-specific parameters via fully connected layers with identity activation functions:

$$\begin{aligned} \mathbf{w}_j^{(\psi_1)} &= \mathbf{W}^{(w_{\psi_1})}\mathbf{h}_j^{(\psi_1^*)}, \\ \mathbf{w}_j^{(\phi)} &= \mathbf{W}^{(w_\phi)}\mathbf{h}_j^{(\phi^*)}, \\ \mathbf{w}_j^{(\gamma)} &= \mathbf{W}^{(w_\gamma)}\mathbf{h}_j^{(\gamma^*)}, \\ \mathbf{w}_j^{(p)} &= \mathbf{W}^{(w_p)}\mathbf{h}_j^{(p^*)}. \end{aligned}$$

By modeling parameters as outputs of a neural network with shared features, relationships among occupancy and detection parameters can be learned. By including features (embeddings) at multiple taxonomic levels, information among taxonomically similar species can be shared.

Combining route features and species-specific parameters

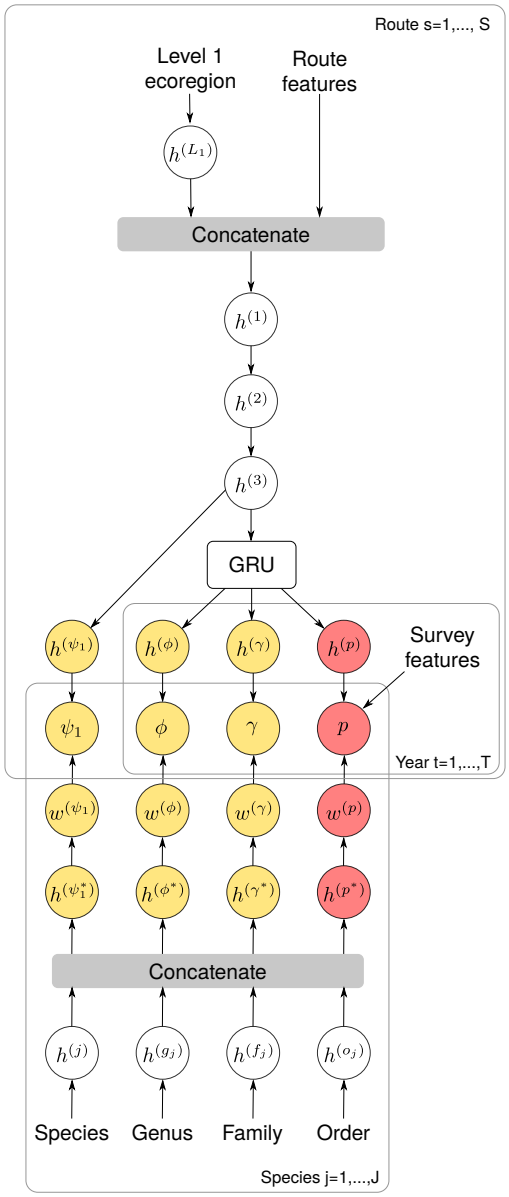

Figure S3.2: Extended computational diagram for the multi-species neural hierarchical dynamic occupancy model. Outer grey boxes indicate the different levels of the model that index quantities inside the boxes. Yellow nodes indicate occupancy parameters, and red nodes indicate detection parameters. Outputs include initial occupancy (ψ_1), persistence (ϕ), colonization (γ), and detection probabilities (p). The box labeled GRU is a gated recurrent unit that decodes temporal sequences of hidden layers from encoded route vectors.

A dot product combines latent spatiotemporal route features with species-specific parameters:

$$\begin{aligned}
\text{logit}(\psi_{t=1,s,j}) &= \mathbf{h}_s^{(\psi_1^*)} \cdot \mathbf{w}_j^{(\psi_1)}, \\
\text{logit}(\phi_{t,s,j}) &= \mathbf{h}_{t,s}^{(\phi^*)} \cdot \mathbf{w}_j^{(\phi)}, \\
\text{logit}(\gamma_{t,s,j}) &= \mathbf{h}_{t,s}^{(\gamma^*)} \cdot \mathbf{w}_j^{(\gamma)}, \\
\text{logit}(p_{t,s,j}) &= \mathbf{h}_{t,s}^{(p^*)} \cdot \mathbf{w}_j^{(p)},
\end{aligned}$$

where $\mathbf{x} \cdot \mathbf{y}$ is the dot product of $\mathbf{x}$ and $\mathbf{y}$: $\mathbf{x} \cdot \mathbf{y} = \sum_i x_i y_i$.

Implementation

Maximum *a posteriori* estimates of baseline model parameters were obtained with Stan (Carpenter *et al.* 2017; Stan Development Team 2018). Penalized maximum likelihood estimates of the neural hierarchical model were obtained with PyTorch (Paszke *et al.* 2017), with L2 regularization penalties on the network parameters, which is equivalent to maximum a posteriori estimation for the neural network with Gaussian priors for the parameters (Blundell *et al.* 2015). The discrete latent occupancy states were marginalized using the forward algorithm, so that the optimization objective used the observed data likelihood (rather than the complete data likelihood). This optimization proceeded in one step, in contrast to some previous approaches that combine deep neural networks with spatiotemporal models using two-stage least squares parameter estimation (McDermott & Wikle 2019). To avoid underflow, forward probabilities were computed on the log scale for the baseline model in Stan, and using forward probability scaling in PyTorch for the neural hierarchical models (Rabiner 1989). All code required to reproduce the analysis is available on GitHub at https://www.github.com/mbjoseph/neuralecology.