Leveraging Sparse Observations to Predict Species Abundance Across Space and Time

SUPPLEMENTARY INFORMATION

Md Zahidul Islam
Cameron S. Fletcher
Ke Sun
Amir Dezfouli
Iadine Chades

1. COTS CONTROL ON THE GBR

Manual "search and kill" strategy is the most effective intervention to control Crown-of-Thorns Starfish (COTS) population [1, 2]. Currently, a surveillance and control strategy is deployed to monitor and control COTS at the Great Barrier Reef (GBR), Australia. The control strategy involves visiting the affected reef areas physically by boats and searching and culling the COTS [3, 4]. Each visit is called a *voyage* that lasts for fourteen days. In every fourth voyage, a quick *manta-tow survey* is performed around a reef to estimate COTS abundance. During a manta-tow survey, snorkel divers are towed at a relatively constant speed behind a boat to visually assess COTS activity and document their observations.

The GBR spans a 340,000 sq. kilometres area with thousands of reefs. In the COTS Control Program, each reef is divided into smaller ($\sim$10 hectare) control sites. Typically, only a small area of a reef can be culled per voyage and the manta-tow surveys provide estimates of the actual scenarios which are sufficient for the control team to prioritize the sites for culling. The control team must accurately rank sites by density and prioritize control activities accordingly before the COTS population goes beyond the ecologically sustainable level [2]. Manta-tow estimates are used as a proxy for COTS density, identifying areas of potential high density for prioritized control efforts. Each high density site is culled until the ecological threshold, measured as Catch-Per-Unit-Effort (CPUE) [5], is reached.

Manta-tow poses several challenges. For example, manta-tow relies on divers' ability to visually perceive COTS density. Accuracy of divers' estimation accuracy depends on COTS size, time of the visits after activities of the COTS population and accessibility to COTS locations on the reefs. The detectability of COTS in manta-tow is about 10% (the divers' detectability is 50% to 90%). The detectability decreases further when the diver is being manta-towed around a reef. To find hidden COTS, the divers often rely on the identification of white feeding scars on the coral to focus their search. These scars fade away with time and are barely noticeable after a week. In addition, on any given day some COTS or feeding scars are undetected due to the complex formation of the coral matrix [3].

2. COTS DATASET

The real-world COTS dataset we used in our experiments has two unique properties - highly sparse observations and zero inflation. The dataset is collected from a COTS control program implemented at the GBR [1, 6]. The dataset includes the number of COTS culled from a particular site on a control voyage. Decisions about which sites or locations on the GBR to target for control are made at the reef level. Control sites are defined around the circumference of a reef, typically 500m long by 200m wide, which are recorded in the dataset with an identifier and the latitude and longitude of the centroid. Once management begins at a reef, it continues until the COTS density at every site around the reef is reduced below the target density. If an entire site cannot be fully culled in a single dive, multiple dives are performed at the same site immediately. The dataset includes the actual number of COTS culled on a single day (if multiple dives are performed on the same date, the COTS numbers are summed). The data we used in our experiments is collected from November 5, 2018 through July 14, 2021.

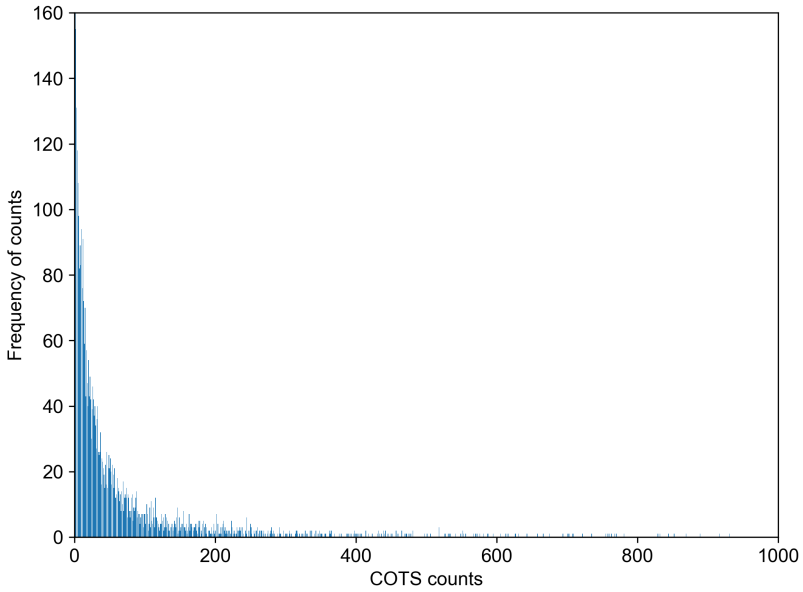

Fig. S1. Histogram of observed COTS counts. The observations are recorded from 5 Nov 2018 through 14 Jul 2021.

A. Missing observations and Zero-Inflation in the feature matrix

When we convert the COTS dataset into a feature matrix for training our model, each day corresponds to one row and each site corresponds to one column in the feature matrix. Only a few sites are culled on a particular day i.e., only a few cells in a row contain observations from that day. Similarly, a site is only culled if the number of COTS is estimated (by a manta-tow) to be above the ecological threshold. As a result, we only have few observations per column (or site) in the feature matrix. According to our representation, the frequency of the number of COTS recorded at a site is very sparse. Similar to [7], we use -1 to indicate missing observations emerge from the above two phenomena. To learn the spatial relationship, our model uses the locations of the sites.

We also observed a large number of zeros in the COTS observations. Figure S1 illustrates the distributions of COTS count in our dataset. The zeros are recorded in the dataset when a diver could not find any COTS to cull during a diving effort at a site. These false positives emerged from manta-tow estimations and resulted in a waste of effort driving down the efficiency of the COTS management process. We hope that our model will reduce the number of wasted efforts and thus increase the efficiency of the COTS management process.

Quantitatively, there are 4427 COTS observations recorded in the dataset from 1368 sites over 982 days. The converted feature matrix is 982×1368 dimensional corresponding to 982 days and 1368 data collection sites. In the feature matrix, 99.67% of the cells have -1 i.e., no observations are recorded for the corresponding days and sites and 11.48% have 0 i.e., divers did not find any COTS for the corresponding days and sites. Moreover, the maximum number of dives in a day is 23 i.e., we have a maximum of 23 observations in a day (the rest of the values in the row vectors are -1), the maximum number of dives per site is 29 (the rest of the values in the column vectors are -1) and there are no observations recorded for 166 days i.e., 166×1368 cells in the feature matrix have -1. Figure S2 illustrates the observation rates (top figure), maximum (middle figure) and minimum (bottom figure) number of days between two visits to the same site in the COTS dataset. The number of visits to a site (each visit corresponds to one day) is divided by the total observed days to compute the observation rate. In Figure S2, we see that very few sites have information to learn temporal patterns and some sites are visited more often (these are the sites where a COTS outburst occurred in the past), some sites are visited after a long period and some sites are visited only once (within the time steps recorded in the dataset).

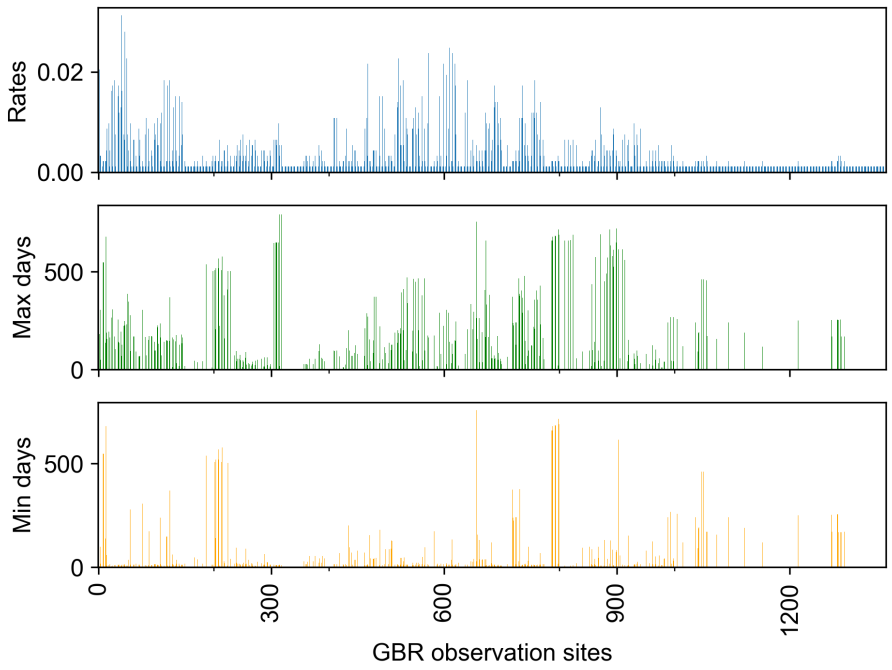

Fig. S2. Demonstration of missing observations on the GBR dataset. The top figure shows the observation rate of each site. The middle figure shows the maximum gap in days between two consecutive visits to the same site. The bottom figure shows the minimum gap in days between two consecutive visits to the same site.

Zero-inflated data are common in ecology where observations are recorded sparsely such as species abundance [8]. In our problem, we cannot fill up the missing observations with zeros and frame the problem as a zero-inflated count regression problem. Because sometimes the divers visit some sites and do not find any COTS, we call these situations **zero-observations**. If a site is not visited on a day we call it a **missing-observation**. The distinction between zero-observations and missing observations is important in the COTS control program as these factors are considered to decide whether to visit a site on a voyage or not. Other domains where zero-inflated data occurs include studies involving adverse side effects of vaccines or drugs and predictions of product sales online [9].

B. Spatial and temporal correlation

COTS outbreaks have an interesting spatiotemporal pattern. The GBR has experienced five outbreaks since the 1960s [10]. The outbreaks have first been detected around Lizard Island near Cairns, at intervals of fifteen to seventeen years [10]. During these outbreaks, the density of COTS on a reef increased from a background density of less than 1 ha^{-1} to up to 1000 ha^{-1} [10]. At these densities, the COTS can consume up to 96% of all the hard coral on a reef within a year, after which their population collapses [10]. Before this, though, they spawn, and larvae are carried downstream on prevailing ocean currents, leading to secondary outbreaks on downstream reefs typically detected two years later [10]. This generates a pattern of a latitudinal band of outbreaking reefs on the GBR that moves southwards from Lizard Island at a rate of around 100 km per year until it reaches the southern end of the GBR many years later.

We analyzed the current COTS cull data to find any spatial and temporal relationships with the number of COTS observed. We use Maximal Information Coefficient (MIC) [11] which is able to capture both linear and non-linear relationships between variables. To measure the spatial relationship, we use observed COTS at a site and the observations at its neighbouring sites between 1 to 10000 meters radius. Figure S3 shows the spatial relationship between the number of COTS at a site and its nearby sites. The first bar shows that there are 476 observations at the neighbours which are within 1000 meter radius and visited within the last 30 days and the mean MIC score between the number of COTS observed at the sites and the number of COTS observed

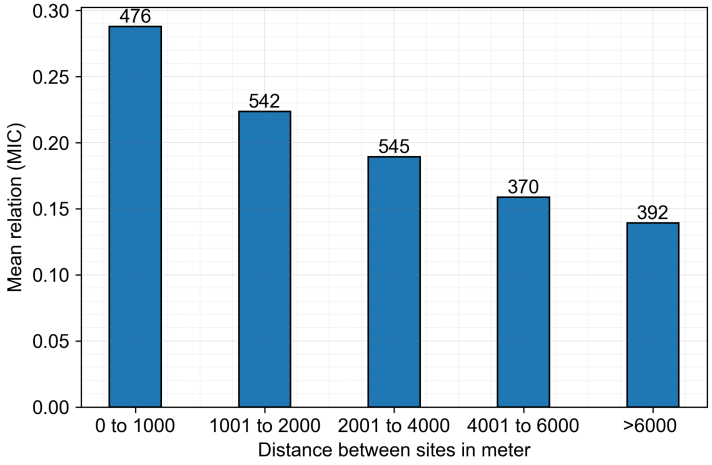

Fig. S3. Relationships of COTS observations between nearby sites.

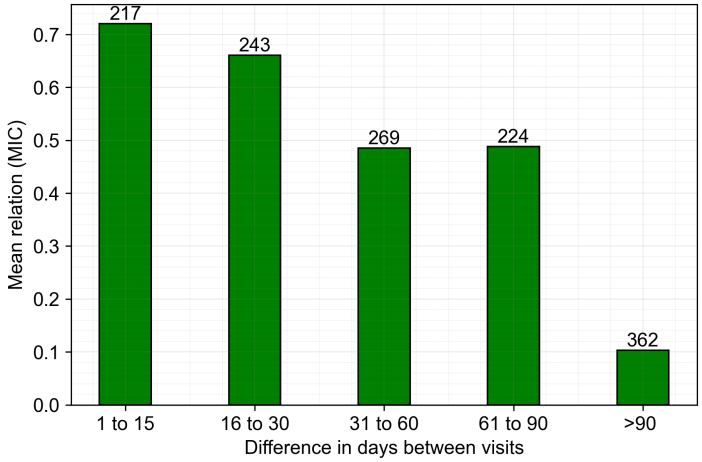

Fig. S4. Relationships of COTS observations between subsequent visits.

at their neighbours is 0.29. The correlation decreases with increasing distance indicating that the effect of numbers of COTS at neighbours decreases with distance. Similarly, the correlation between the number of COTS found at a site in two subsequent visits is shown in Figure S4. There are 217 (shown on top of the first bar) visits performed within 1 to 15 days at all the sites. We observed that as the gaps between consecutive visits increase, the temporal relation decreases.

3. THE STZIPN MODEL

In this section, we discuss the components of the proposed STZipN model in more detail. As discussed in the main text, the model has four components - *input encoder*, *temporal information encoder*, *spatial information encoder* and *output decoder*. The input $\mathbf{X}$ flows through each of the components and the component generates an enriched representation of the input. All the symbols used in the paper are given in Table S1 and a summary of the parameters of each component of STZipN model is shown in Table S2.

Table S1. Notations used in the paper.

Symbol	Dimension	Description
$\mathbb{N}_{>0}$		The set of positive integers greater than 0.
$\mathbb{N}_{+}$		The set of positive integers including 0.
$\mathbb{R}$		The set of real numbers.
m	$\mathbb{N}^1_{>0}$	The number of sites.
$\mathcal{S}$	$\mathbb{R}^{m\times 2}$	The site locations i.e., $\mathcal{S} = \{(\text{lon}_1, \text{lat}_1), \ldots, (\text{lon}_\text{m}, \text{lat}_\text{m})\}$.
t	$\mathbb{N}^1_{>0}$	The number of observed days.
d	$\mathbb{N}^1_{>0}$	The number of observed features.
$\mathbf{X}$	$\mathbb{R}^{t\times m\times d}$	A tensor of observations until time t.
$\mathbf{Y}$	$\mathbb{N}_{+}^{t\times m}$	A matrix of observations at sites.
$\mathbf{X}_i$	$\mathbb{R}^{m\times d}$	Observations at all m sites at a time step i.
$\mathbf{Y}_i$	$\mathbb{N}_{+}^{m}$	Number of observed pests at m sites on the next time step.
$\hat{\mathbf{Y}}$	$\mathbb{R}^{m}$	An estimation of the number of pests at time step $t+1$.
$\mathbf{Y}^i$	$\mathbb{N}_{+}^{1}$	Observations corresponding to site i
Y^i_j	$\mathbb{N}_{+}^{1}$	Observation corresponding to site i at time step j.
$\mathbf{Z}_\text{in}$	$\mathbb{R}^{m\times t\times \mathrm{H}}$	Output of the input encoder.
H	$\mathbb{N}^1_{>0}$	The number of hidden units.
$\mathbf{Z}_\text{temp}$	$\mathbb{R}^{m\times t\times \mathrm{H}}$	Output of the temporal information encoder.
$\mathbf{Z'}_\text{GCN}$	$\mathbb{R}^{m\times t\times \mathrm{H}}$	Output of the GCN submodule.
$\mathbf{Z}_\text{sp}$	$\mathbb{R}^{m\times \mathrm{H}}$	Output of the static feature encoder.
$\mathbf{Z}_\text{GCN}$	$\mathbb{R}^{m\times t\times \mathrm{H}}$	Output of the spatial information encoder.
D_{ij}	$\mathbb{R}^1$	The Haversine distance between sites i and j.
$\mathbf{D}$	$\mathbb{R}^{m\times m}$	The pairwise distance matrix using D_{ij}.
$A_{ij}(\tau,\epsilon)$	$\mathbb{R}^1$	Weights between the adjacent sites i and j.
$\mathbf{A}(\tau,\epsilon)$	$\mathbb{R}^{m\times m}$	Adjacency matrix representation of the graph constructed from $\mathcal{S}$.
τ	$\mathbb{N}^1_{>0}$	The kernel width for graph construction.
ϵ	$\mathbb{R}^1$	The threshold value to control sparsity in $\mathbf{A}$.
$\mathbf{A}'(\tau,\epsilon)$	$\mathbb{R}^{m\times m}$	$\mathbf{A}$ with self-loop included.
$\mathbf{Deg}(\tau,\epsilon)$	$\mathbb{N}_{+}^{m\times m}$	The node degree matrix corresponding to $\mathbf{A}$.
$\mathbf{1}_m$		m dimensional vector of 1s.
$\tilde{\mathbf{A}}(\tau,\epsilon)$	$\mathbb{R}^{m\times m}$	Normalized $\mathbf{A}'$.
$\boldsymbol{\pi}$	$\mathbb{R}^{m}$	Parameter of the Zero-Inflated Poisson (ZIP) distribution corresponding to the probability of zero.
$\boldsymbol{\lambda}$	$\mathbb{R}^{m}$	Parameter of the ZIP distribution corresponding to the expected Poisson count.
π^i	$\mathbb{R}^1$	The π parameter of the ZIP distribution corresponding to site i.
λ^i	$\mathbb{R}^1$	The λ parameter of the ZIP distribution corresponding to site i.
$\mathcal{L}$		Loss function of the STZipN model.

Table S2. Input, output and parameters of each module in STZipN model.

Module	Input	Parameters	Output
Input encoder	$\mathbf{X} \in \mathbb{R}^{t\times m\times d} \rightarrow \mathbf{X} \in \mathbb{R}^{m\times t\times d}$	$\mathbf{W}_{\text{in}}^{(1)} \in \mathbb{R}^{d\times \text{H}}, \mathbf{b}_{\text{in}}^{(1)} \in \mathbb{R}^{\text{H}}$, $\mathbf{W}_{\text{in}}^{(2)} \in \mathbb{R}^{\text{H}\times \text{H}}, \mathbf{b}_{\text{in}}^{(2)} \in \mathbb{R}^{\text{H}}$	$\mathbf{Z}_{\text{in}} \in \mathbb{R}^{m\times t\times \text{H}}$
Temporal information encoder	$\mathbf{Z}_{\text{in}}$	$\mathbf{W}_{\text{temp}} \in \mathbb{R}^{4\text{H}\times \text{H}}$ $\mathbf{b}_{\text{temp}} \in \mathbb{R}^{4\text{H}}$ (LSTM) or $\mathbf{W}_{\text{temp}} \in \mathbb{R}^{3\text{H}\times \text{H}}$, $\mathbf{b}_{\text{temp}} \in \mathbb{R}^{3\text{H}}$ (GRU)	$\mathbf{Z}_{\text{temp}} \in \mathbb{R}^{m\times t\times \text{H}}$
Spatial information encoder (GCN)	$\mathbf{Z}_{\text{temp}}, \mathcal{S} \in \mathbb{R}^{m\times 2}$	$\tau, \epsilon, \mathbf{W}_{\text{GCN}} \in \mathbb{R}^{H\times \text{H}}$ and $\mathbf{b}_{\text{GCN}} \in \mathbb{R}^{\text{H}}$	$\mathbf{Z}'_{\text{GCN}} \in \mathbb{R}^{m\times t\times \text{H}}$
Static feature encoder	$\mathcal{S} \in \mathbb{R}^{m\times 2}$	$\mathbf{W}_{\text{sp}}^{(1)} \in \mathbb{R}^{2\times \text{H}}, \mathbf{b}_{\text{sp}}^{(1)} \in \mathbb{R}^{\text{H}}$, $\mathbf{W}_{\text{sp}}^{(2)} \in \mathbb{R}^{\text{H}\times \text{H}}, \mathbf{b}_{\text{sp}}^{(2)} \in \mathbb{R}^{\text{H}}$	$\mathbf{Z}_{\text{sp}} \in \mathbb{R}^{m\times \text{H}}$
Spatial information encoder	$\mathbf{Z}'_{\text{GCN}}$, $\mathbf{Z}_{\text{sp}} \in \mathbb{R}^{m\times 1\times \text{H}}$		$\mathbf{Z}_{\text{GCN}} \in \mathbb{R}^{m\times t\times \text{H}}$
Output encoder	$\mathbf{Z}_{\text{GCN}}$	$\mathbf{W}_{\text{out}}^{(1)} \in \mathbb{R}^{H\times \text{H}}, \mathbf{b}_{\text{out}}^{(1)} \in \mathbb{R}^{\text{H}}, \ldots,$ $\mathbf{W}_{\text{out}}^{(nl)} \in \mathbb{R}^{\text{H}\times 2}, \mathbf{b}_{\text{out}}^{(nl)} \in \mathbb{R}^{2}$	$\boldsymbol{\pi}, \mathbf{log}\,\boldsymbol{\lambda}$

A. Input encoder

The input to our STZipN model is the feature matrix $\mathbf{X}$. Similar to [12], we project $\mathbf{X}$ into a latent embedded space in $\mathbb{R}^{\text{H}}$, where H is the dimension of the hidden state. Note that, we transpose the input tensor $\mathbf{X} \in \mathbb{R}^{t\times m\times d}$ to $\mathbf{X} \in \mathbb{R}^{m\times t\times d}$ as we intend to learn the feature representation per site. By projecting $\mathbf{X}$, the signals within it are enhanced, much like combining H variations of the input features. The input encoder module consists of a two layer feedforward neural network with H neurons in each linear layer and we use Exponential Linear Unit (ELU) [13] for nonlinearity. The output of the input encoder is given by:

$$\mathbf{Z}_{\text{in}} = \text{ELU}\left(\text{ELU}\left(\mathbf{X}\cdot\mathbf{W}_{\text{in}}^{(1)} + \mathbf{b}_{\text{in}}^{(1)}\right)\cdot\mathbf{W}_{\text{in}}^{(2)} + \mathbf{b}_{\text{in}}^{(2)}\right) \tag{S1}$$

where $\mathbf{W}_{\text{in}}^{(1)} \in \mathbb{R}^{d\times \text{H}}$ and $\mathbf{b}_{\text{in}}^{(1)} \in \mathbb{R}^{\text{H}}$ weights and biases corresponding to the input to the first hidden layer of the input encoder. Similarly, $\mathbf{W}_{\text{in}}^{(2)} \in \mathbb{R}^{H\times H}$ and $\mathbf{b}_{\text{in}}^{(2)} \in \mathbb{R}^{H}$ are associated with the first hidden layer to the output layer of the input encoder. The output $\mathbf{Z}_{\text{in}} \in \mathbb{R}^{m\times t\times \text{H}}$ of Eq S1 corresponds to the H dimensional embedding of the d dimensional input features.

B. Temporal information encoder

The purpose of our temporal information encoder is to learn the temporal dependence in the data. The Recurrent Neural Network (RNN) is widely used for processing temporal sequence data. The LSTM [14] and GRU [15] are two popular variants of RNN. We have experimented with both LSTM and GRU to encode the temporal dependence in our data. Here, we refer to the temporal module (using LSTM or GRU units) of our model as TE. The TE takes the outputs of the input encoder $\mathbf{Z}_{\text{in}}$ and generates a new encoding as follows:

$$\mathbf{Z}_{\text{temp}} = \text{TE}\left(\mathbf{Z}_{\text{in}}\cdot\mathbf{W}_{\text{temp}} + \mathbf{b}_{\text{temp}}\right) \tag{S2}$$

where, $\mathbf{W}_{\text{temp}} \in \mathbb{R}^{l\cdot \text{H}\times \text{H}}$ and $\mathbf{b}_{\text{temp}} \in \mathbb{R}^{l\cdot \text{H}}$ are the learnable parameters of TE. For LSTM, the parameters $\mathbf{W}_{\text{temp}} \in \mathbb{R}^{4\cdot \text{H}\times \text{H}}$ and $\mathbf{b}_{\text{temp}} \in \mathbb{R}^{4\cdot \text{H}}$ correspond to the input, forget and output gates, and cell state. Similarly, for GRU, $\mathbf{W}_{\text{temp}} \in \mathbb{R}^{3\cdot \text{H}\times \text{H}}$ and $\mathbf{b}_{\text{temp}} \in \mathbb{R}^{3\cdot \text{H}}$ correspond to the reset, update and candidate gates. The output $\mathbf{Z}_{\text{temp}} \in \mathbb{R}^{m\times t\times \text{H}}$ corresponds to temporal encoding of $\mathbf{Z}_{\text{in}}$.

The inputs to our TE are the feature representations from the input encoder and the hidden state from the previous time step. For LSTM network, the hidden states include the cell states and hidden states and for the GRU network, there are no cell states. Thus, TE obtains the representations of pests at each timestamp t by taking the hidden states at timestamp $t-1$ and the current observations as inputs. To obtain the representation for timestep $t+1$, the hidden state from t is fed back to the recurrent network. TE eventually retains the trends of input observations until $t-1$ to generate representations for timestamp t.

C. Spatial feature encoding

We apply a GCN [16] layer to the outputs of the temporal encoder $\mathbf{Z}_{\text{temp}}$ to capture the spatial and temporal dependencies together. We have a static feature encoder to encode the site locations $\mathcal{S}$(see Fig. 5) in our model. The GCN output $\mathbf{Z'}_{\text{GCN}}$ and the static feature encoder output are concatenated to provide both dynamic (observations varying over time) and static (site locations) signals. We apply the nonlinear ELU [13] transformation to the combined linear outputs from the GCN and static encoder submodules.

Given an adjacency matrix of a graph and feature representation of the input data, the GCN layer convolve features from neighbouring nodes to capture spatial dependencies. We do not assume an explicit graph structure in the observed dataset (which is the case for most real-world pest abundance datasets). Inspired by [17], we construct the graph structure from the site locations $\mathcal{S}$. Each site becomes a node in the graph and the edges are determined based on the distance between sites. Using the Haversine formula [18], the distance between sites i and j is computed as Eq. (S3):

$$D_{ij} := 2\arcsin\sqrt{\begin{array}{l}\sin^2\left(\dfrac{\text{lat}_i-\text{lat}_j}{2}\right)+\\ \cos(\text{lat}_i)\cos(\text{lat}_j)\sin^2\left(\dfrac{\text{lon}_i-\text{lon}_j}{2}\right)\end{array}} \tag{S3}$$

All pairwise site distances are represented by a matrix $\mathbf{D} := (D_{ij})_{m\times m}$. Our graph adjacency matrix is $\mathbf{A} := (A_{ij})_{m\times m}$, where we compute the weight $A_{ij}(\tau,\epsilon)$ corresponding to sites i and j using distance D_{ij}, kernel width τ and cut-off threshold ϵ:

$$A_{ij}(\tau,\epsilon) := \exp\left(-\frac{D_{ij}^2}{\tau^2}\right)\cdot\left[\!\!\left[\exp\left(-\frac{D_{ij}^2}{\tau^2}\right)>\epsilon\right]\!\!\right], \tag{S4}$$

where $[\![expression]\!]$ is 1 if the *expression* is true, 0 otherwise. In STZipN, τ is a hyperparameter depending on how fast the pests move. We determine the value of τ through hyperparameter search. Note that our graph representations such as $\mathbf{A}(\tau,\epsilon)$ and the corresponding degree matrix $\mathbf{Deg}(\tau,\epsilon)$ depend on τ and ϵ. For simplicity, we denote $\mathbf{A} := \mathbf{A}(\tau,\epsilon)$ and $\mathbf{Deg} := \mathbf{Deg}(\tau,\epsilon)$.

We normalize $\mathbf{A}$ as $\tilde{\mathbf{A}} := \mathbf{Deg}^{-\frac{1}{2}}\mathbf{A'}\mathbf{Deg}^{-\frac{1}{2}}$ for numerical stability. $\mathbf{A'} := \mathbf{A}+\mathbf{I}_m$ is the adjacency matrix with self-loop, $\mathbf{Deg} = \text{diag}\{\mathbf{A'}\cdot\mathbf{1}_\text{m}\}$ where $\mathbf{1}_m$ denotes a m dimensional vector of 1 and the diag(.) function constructs a diagonal matrix from the result of the operation $\mathbf{A'}\cdot\mathbf{1}_m$. Similar operations are used in [16]. In our model, the GCN submodule aggregates the temporal encoding of the m sites $\mathbf{Z}_{\text{temp}}$ using $\tilde{\mathbf{A}}$. For simplicity and faster training, we use a one-layer GCN [16] to obtain the spatial dependence. The static feature encoder embeds the site locations $\mathcal{S}$ using a two-layer network as shown in Fig. 1 (in the main paper). Finally, a non-linear transformation (using ELU [13]) of the aggregated embeddings from the GCN and static encoder submodules generate the spatiotemporal encoding of the input as shown in Eq S5:

$$\mathbf{Z}_{\text{GCN}} := \text{ELU}\left(\begin{array}{l}\underbrace{\tilde{\mathbf{A}}\cdot\mathbf{Z}_{\text{temp}}\cdot\mathbf{W}_{\text{GCN}}}_{\mathbf{Z}_{\text{GCN}'}}+\\ \underbrace{\text{ELU}(\mathcal{S}\cdot\mathbf{W}_{\text{sp}}^{(1)}+\mathbf{1}_\text{H}\cdot\mathbf{b}_{\text{sp}}^{(1)\mathsf{T}})\cdot\mathbf{W}_{\text{sp}}^{(2)}}_{\mathbf{Z}_{\text{sp}}}\\ +\mathbf{1}_\text{H}\cdot\mathbf{b}\end{array}\right), \tag{S5}$$

where $\mathbf{W}_{\text{GCN}}\in\mathbb{R}^{\text{H}\times\text{H}}$ is a learnable network parameter corresponding to the GCN submodule.

The static encoder submodule is a two-layer perceptron network i.e., $\mathbf{Z}_{sp} = \text{MLP}(\mathcal{S})$. We include spatial features (latitude and longitude) in addition to the GCN to maximize the use of

available information in a data-limited setting. While they partly duplicate information in the graph structure, they serve complementary roles: the graph structure enables the model to learn from connected sites based on their relational proximity, while the spatial features help learn from the absolute physical locations of sites irrespective of their graph-based relationships with other sites. This dual representation provides additional pathways to extract spatial patterns, which we found beneficial given our limited training data. The output of the static encoder is a matrix $\mathbf{Z}_{\text{sp}} \in \mathbb{R}^{m \times \text{H}}$ of H dimensional encoding of the static features. The shape of $\mathbf{Z}_{\text{GCN}'}$ is $m \times t \times \text{H}$. The parameters of the static enocoder network are $\mathbf{W}_{\text{sp}}^{(1)} \in \mathbb{R}^{2 \times \text{H}}$, $\mathbf{W}_{\text{sp}}^{(2)} \in \mathbb{R}^{\text{H} \times \text{H}}$, $\mathbf{b}_{\text{sp}}^{(1)} \in \mathbb{R}^{\text{H}}$ and $\mathbf{b}_{\text{sp}}^{(2)} \in \mathbb{R}^{\text{H}}$. In Eq. (S5), $\mathbf{b}$ represents bias terms corresponding to the GCN layer and the second layer of the static encoder i.e., $\mathbf{b}_{\text{sp}}^{(2)}$.

D. Output decoder

The spatial information encoder takes the temporal encoding $\mathbf{Z}_{\text{temp}}$ as input and produces a spatiotemporal embedding $\mathbf{Z}_{\text{GCN}}$.

The final module in our STZipN model is an output network with two output units corresponding to the two parameters λ and π of our intended Zero-Inflated Poisson model. The input to the output encoder in the output of the spatial information encoder i.e., $\mathbf{Z}_{\text{GCN}} \in \mathbb{R}^{m \times t \times \text{H}}$. We consider $\log \lambda$ and $\texttt{logit}\pi$ as outputs to simplify our implementation. The outputs of the output encoder module are given by:

$$[\log \boldsymbol{\lambda}; \texttt{logit}\boldsymbol{\pi}] = \text{ELU}\left(\text{ELU}\left(\mathbf{Z}_{\text{GCN}} \cdot \mathbf{W}_{\text{out}}^{(1)} + \mathbf{b}_{\text{out}}^{(1)}\right) \cdot \mathbf{W}_{\text{out}}^{(2)} + \mathbf{b}_{\text{out}}^{(2)}\right) \cdots \mathbf{W}_{\text{out}}^{(nl)} + \mathbf{b}_{\text{out}}^{(nl)} \tag{S6}$$

where the number of layers of the feedforward output network nl is a hyperparameter. The parameters $\mathbf{W}_{\text{out}}^{(l)} \in \mathbb{R}^{\text{H} \times \text{H}}$ and $\mathbf{b}_{\text{out}}^{(l)} \in \mathbb{R}^{\text{H}}$ correspond to layers $l = \{1, 2, \ldots, nl-1\}$ and parameters $\mathbf{W}_{\text{out}}^{(nl)} \in \mathbb{R}^{\text{H} \times 2}$ and $\mathbf{b}_{\text{out}}^{(nl)} \in \mathbb{R}^{2}$ correspond to the nl^{th} layer.

E. Computation of loss function

Note that there are three types of values in our inputs $\mathbf{X}$ - zero, positive integers and missing values. Hence, we compute loss $\mathcal{L}$ for the targets $\mathbf{Y}$ where $\mathbf{Y} \geq 0$. We represent missing values by -1 and do not consider them in $\mathcal{L}$. Considering this factors, there are two components in our loss function - one for the zero observations 0 i.e., $\mathbf{Y} = 0$ and the other is for nonzero observations i.e., $\mathbf{Y} > 0$. Let $\mathcal{L}_{\mathbf{Y}=0}^{i}$ and $\mathcal{L}_{\mathbf{Y}>0}^{i}$ be the log likelihoods of $\mathbf{Y} = 0$ and $\mathbf{Y} > 0$ at site i respectively. We can define the loss $\mathcal{L}$ as follows:

$$\mathcal{L} = \sum_{i=1}^{m} \mathcal{L}_{\mathbf{Y}=0}^{i} + \mathcal{L}_{\mathbf{Y}>0}^{i} \tag{S7}$$

$$\mathcal{L}_{\mathbf{Y}=0}^{i} = \sum_{j:\text{Y}_j^i=0} \log\left(\pi^i + (1-\pi^i)e^{-\lambda^i}\right) \tag{S8}$$

$$\mathcal{L}_{\mathbf{Y}>0}^{i} = \sum_{k:\text{Y}_k^i>0} \log\left((1-\pi^i)\frac{{\lambda^i}^{\text{Y}_k^i} e^{-\lambda^i}}{\text{Y}_k^i!}\right) \tag{S9}$$

We now briefly discuss how we compute $\mathcal{L}_{\mathbf{y}=0}^{i}$ and $\mathcal{L}_{\mathbf{y}>0}^{i}$ from our network outputs. We ignore sper/subscriptions i and j in the following discussion to simplify our notations. From Equations S8 and S9, we need to compute:

$$\mathcal{L}_{\mathbf{Y}=0} = -\log\left(\pi + (1-\pi)e^{-\lambda}\right) \tag{S10}$$

$$\begin{aligned}\mathcal{L}_{\mathbf{Y}>0} &= -\log\left((1-\pi)\frac{\lambda^{\text{Y}} e^{-\lambda}}{\text{Y}!}\right) \\ &= -\log(1-\pi) - \text{Y}\log\breve{} + \breve{} + \log \text{Y}!\end{aligned} \tag{S11}$$

According to our model architecture, the outputs of STZipN are 1ogitß and $\log \lambda$. Using the

softplus function, we have:

$$\begin{aligned}\texttt{softplus}(\texttt{logit}ß) &= \log(1+e^{\texttt{logit}ß}) \\ &= \log\left(1+e^{\log\left(\frac{\pi}{1-\pi}\right)}\right) \\ &= -\log(1-\pi) \end{aligned} \tag{S12}$$

From the model output $\texttt{logit}\pi$, we compute p_0:

$$\begin{aligned} p_0 &= \texttt{logit}\pi - \texttt{softplus}(\texttt{logit}\pi) \\ &= \log\left(\frac{\pi}{1-\pi}\right) - \text{softplus}((\texttt{logit}\pi) \\ &= \log\pi - \log(1-\pi) + \log(1-\pi) \\ &= \log\pi \end{aligned} \tag{S13}$$

From our network outputs $\texttt{logit}\pi$ and $\log\lambda$ we compute p_1:

$$\begin{aligned} p_1 &= e^{\log\lambda} + \texttt{softplus}(\texttt{logit}\pi) \\ &= e^{\log\lambda} - \log(1-\pi) \\ &= \lambda - \log(1-\pi) \end{aligned} \tag{S14}$$

We know from the Log-Sum-Exp function (LSE) [19]:

$$\text{LSE}(x) = \log\left(\sum_{i=1} e^{x_i}\right) \tag{S15}$$

Thus, using the LSE function, we have:

$$\begin{aligned} \text{LSE}(p_0, -p_1) &= \log\left(\sum_{x\in\{p_0,-p_1\}} e^x\right) \\ &= \log(e^{\log\pi} + e^{-(\lambda-\log(1-\pi))}) \\ &= \log\left(\pi + e^{-\lambda}\times e^{\log(1-\pi)}\right) \\ &= \log(\pi + (1-\pi)e^{-\lambda}) = \mathcal{L}_{Y=0} \end{aligned} \tag{S16}$$

Similarly, we get: $\texttt{softplus}(\texttt{logit}\pi) - \text{Y}\times\log\text{˘} + \text{e}^{\log\text{˘}} + \log\text{Y!} = -\log(1-ß) - \text{Y}\log\text{˘} + \text{˘} + \log\text{Y!} = \mathcal{L}_{\text{Y}>0}$. From the above, we see that the loss $\mathcal{L}$ can be computed from the model outputs $\texttt{logit}\pi$ and $\log\lambda$, the `softplus` and the LSE functions.

4. EVALUATION

A. Hyperparameter search

The hyperparameter search space for the deep learning models (Trivial, Temporal, Spatial and STZipN) was defined based on insights gained from a series of 350 prior optimization trials. We used the Optuna optimization framework [20] for hyperparameter search. Key hyperparameters included the number of hidden layers (*nl*), the number of ELU units per layer (H), learning rate, the backward time window size, and graph convolutional kernel width τ, where applicable. Notably, the search ranges were carefully constrained to reflect empirically informed priors to improve efficiency of the optimization process and ensuring relevant regions of the hyperparameter space were prioritized by Optuna. In the hyperparameter search space, we vary τ from 800 to 1300, *nl* from 1 to 6, H from 16 to 256 and the learning rate from 0.0001 to 0.01. We use a multistep decaying learning rate policy decreasing by 0.2 every 50 epochs.

B. Expert guided regression (ExpertR)

The objective of this study is to predict COTS densities at a daily and site-specific level. Practical experience from management operations indicates that recent observations from a site are the most reliable predictors of current density, as ecological processes rarely change substantially within short time frames. Empirical results further suggest that a simple "last-seen" baseline often performs competitively with more complex algorithmic models. However, as observations

become temporally distant, or when recent data are unavailable for a site, it becomes necessary to rely increasingly on correlated environmental and spatial factors.

Previous studies and field experience identify four principal correlates influencing high COTS densities at a site on a given day [1–3, 6, 21, 22]:

1. Whether the site lies within the current outbreak front (approximately 100 km scale);
2. Whether the site belongs to an outbreaking reef (approximately 10 km scale);
3. Whether the site exhibited high COTS densities during the most recent survey (within 56 days);
4. Whether neighbouring sites exhibit high densities (approximately 1 km scale).

These factors can be interpreted hierarchically. If criterion (1) is not satisfied, the site is highly likely to have low or zero COTS density, except in rare instances marking the onset of a new large-scale outbreak. If criterion (1) is met but criterion (2) is not, the likelihood of low or zero density remains high, except when an outbreak is emerging but not yet evident in observational data. When both (1) and (2) are met but (3) is not, a low density is again likely, with probability weighted by the time elapsed since the last observation. COTS individuals can relocate within a reef; thus, densities may increase at previously low-density sites after prolonged intervals. When all three criteria (1)–(3) are satisfied, the site most likely exhibits high density, approximately proportional to the previously recorded cull numbers. If recent observations (3) are unavailable, yet (1) and (2) are satisfied, the site may still present moderate densities.

Operationalising these rules presents challenges, as some concepts are intuitive rather than formally defined. For example, whether a site is “at an outbreaking reef" is better determined by reef membership than by simple spatial proximity, since COTS can move considerable distances within reefs but not between them. Similarly, the “outbreak front" lacks a formal definition; managers typically estimate it by fitting a Gaussian function to outbreak densities along latitude and designating the region within approximately $\pm$100 km of the Gaussian peak as the outbreak front.

Based on these empirical insights, we formulated an expert-guided regression baseline, hereafter referred to as ExpertR:

- All sites are initially assumed to have zero COTS density.
- Alternatively, densities can be set equal to the last recorded cull value for each site—although conceptually simplistic, this baseline performs competitively in practice. For sites lacking prior observations, the nearest available observation is used.
- Predictions for a given site are refined by incorporating weighted correlates, including:
  - Year,
  - Latitude,
  - Mean COTS density across all reefs within approximately $\pm$100 km latitude,
  - Mean COTS density at the current reef,
  - Mean COTS density at neighbouring sites, and
  - Potential additional covariates such as reef size (larger reefs can sustain larger outbreaks), shelf position (inshore reefs rarely experience outbreaks), and available coral habitat area.

In management applications, false positives at sites with zero COTS density are of minor concern, as divers can rapidly verify and adjust upon arrival. In contrast, false negatives pose substantial risk, as they may result in failure to control high-density populations that cause immediate coral damage and spawn larvae driving downstream outbreaks. Consequently, the most valuable predictive models are those capable of identifying high-density sites even in the absence of recent or strongly correlated observations.

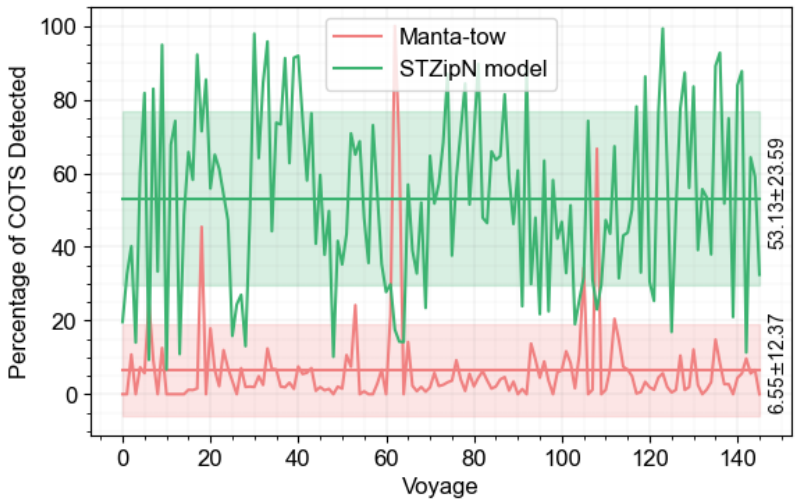

Fig. S5. Comparison of detectability of manta-tow and the detectability of our STZipN model. The x axis represents voyages and y axis presents the detectability in percentage. The straight lines presents mean detectability over 151 voyages and the light shading represents the standard error of the mean detectability.

C. Comparison of detectabilities of manta-tow and our model

We measure the detectability as the percentage of COTS culled from sites recommended by our model or manta-tow. For example, let there be x COTS culled in a voyage from all the sites and manta-tow and our model estimate y and z COTS respectively. The detectability of manta-tow is $\frac{y}{x} * 100\%$ and the detectability of our model is $\frac{z}{x} * 100\%$. We assume three site are culled in a voyage. We do not consider the voyages where no COTS are culled, the voyages where manta-tow or our model overestimates the number of COTS. Figure S5 shows the results of our detectability comparison.

D. Comparison of recommendations by STZipN and manta-tow

The COTS control team uses a ranked site priority list to deploy culling activities. Currently, the manta-tow estimates are used to rank the sites according to the estimated COTS densities at the sites. We compare the ranks produced by manta-tow estimates and STZipN estimates against the number of COTS culled from the sites in each voyage. Hence, we assume the cull numbers provide the ground truth ranks. The result of our rank comparison using the Spearman rank correlation is shown in Figure S6.

Assuming the null hypothesis is true, the p-value of Spearman rank correlation indicates the probability of getting a test-statistic at least as extreme as the observed correlation. A p-value close to 1 suggests no correlation other than due to random chance and a p-value close to 0 suggests the observed correlation is unlikely to be due to chance. According to previous research presented in [23], p-value is only accurate for very large samples (> 500 samples). In our cull data, at most 52 sites are culled per voyage on average. To obtain a reliable estimate of the p-value, we follow the permutation test approach [23]. The results p-value comparison is shown in Figure S7. We observe that p-values of the rank correlation coefficient of cull and manta-tow ranks are very high and average p-values is 0.34 indicating that there is no significant relationship between manta-tow ranks and actual ranks observed in the cull data. On the other hand, the p-values of the ranks produced by our model with cull is lower than manta-tow ranks. The average p-value is 0.06 which indicates statistically significant correlation between cull and our model's estimates.

E. Power consumption by STZipN model

The training of the STZipN model with its relatively simple architecture,including one or two-layer networks in the modules (except for the output decoder), we are able to train the model for 200 epochs using the entire COTS dataset on a MacBook with 10th generation Intel Iris Plus Graphics equivalent to CuDA compatible Nvidia GT 1030 or Nvidia GeForce MX150 GPU within 4 hours. The MacBook GPU operates at a frequency of 350 MHz which can be boosted up to 1000

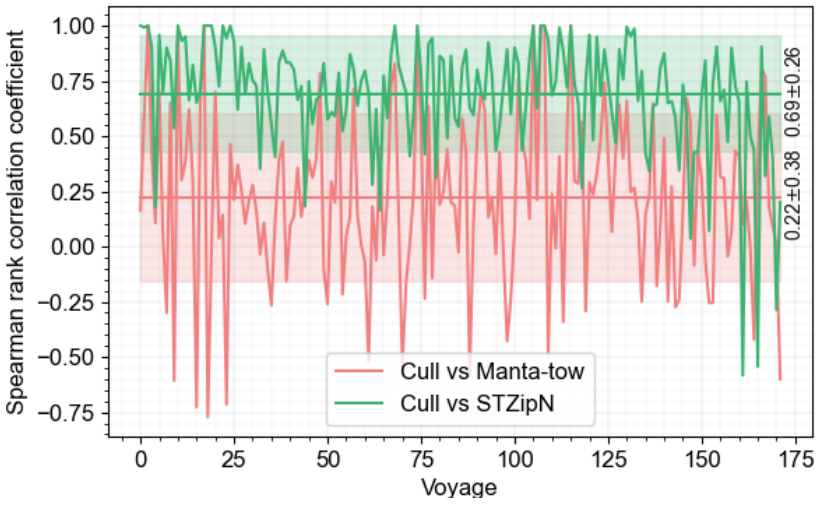

Fig. S6. Comparison of rank correlation between our model and cull data and manta-tow and the cull data. The x axis represents voyages and the y axis presents the Spearman rank correlation coefficient between the site ranks. The straight lines present the mean rank correlation coefficient and the light shading represents the standard error of the mean.

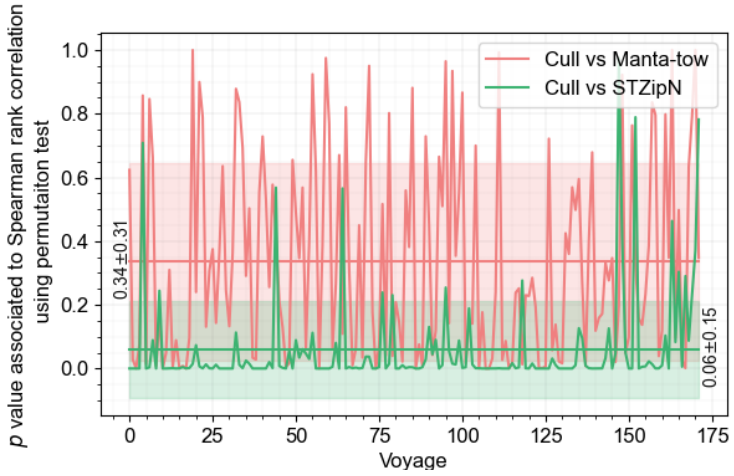

Fig. S7. Comparison of the p-values associated to the Spearman rank correlation comparison. The x axis represents voyages and y axis presents the p-values. The solid straight lines presents mean p-values over all the voyages and the light shading represents the standard error of the mean.

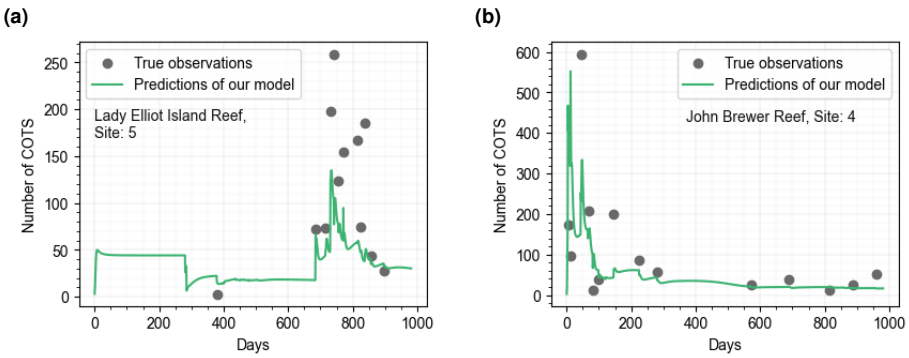

Fig. S8. Predictions of our STZipN model with very few observations in the training set. The x axis represents the observation days and the y axis represents the number of COTS observed on the corresponding day.

MHz and its power consumption ranges from 12 to 25 Watts[1].

5. PREDICTIONS AT LOW OBSERVATION SITES

Our proposed model can make meaningful predictions with a low amount of historical data at a site. Figure S8 shows two examples (more examples are shown in the Result section of the manuscript) of sites with few historical data and our model's predictions at those sites. More specifically, Figure S8a shows that our model correctly predicts increasing COTS at site 5 at the Lady Elliot Island reef on day 685 and beyond with only a single observation before day 685. In Figure S8b our model predicts a COTS outbreak initially and correctly predicts fewer COTS in later days - not predicting the initial high COTS patterns. The Lady Elliot Island reef has fewer past data points than the John Brewer reef. A manual cross-checking of the data shows that for site 5 at Lady Elliot Island reef, 14 observations were collected at neighbouring sites before day 685 (Figure S8a).

Table S3 shows some properties of the graph we constructed from the data and used to learn features from nearby sites which aids in making predictions with low data observations. The mean node degrees shows that each site learns features from 18 to 19 nearby sites. Interestingly, we found that 22 sites (or nodes) have no neighbours according to our learned kernel width which suggests culling voyages rarely took place at sites far from most COTS-infested sites. The in-field COTS management teams could consider targeted data collection around sites with few neighbouring data points in the dataset to provide better future COTS density predictions in those locations

A. Non-zero overestimation despite zero-inflated Poisson distribution model

COTS exhibit clumped distributions at multiple spatial scales (e.g. within a latitudinal band some individual reefs experience starfish outbreaks while others don't, and within an individual reef experiencing an outbreak, COTS cluster in certain locations or sites and not others). This behaviour leads to spatial distributions that are zero inflated at multiple scales. In addition, COTS are hard to detect. All current COTS estimation methods – whether algorithmic or real-world monitoring – can overestimate. However, overestimation is less harmful to control program success than underestimation. Overestimation decreases the efficiency of the overall control process by deploying effort to low priority sites – however, once control at a site begins, any prior overestimation becomes apparent, and resources can be redirected. Underestimation of COTS, in contrast, may lead to significant coral damage and increase the chance of an uncontrolled outbreak.

6. SENSITIVITY ANALYSIS

In this section, we discuss how we generated the gradient-plot. Let, $\mathbf{X}$ be the input of our model and $\mathbf{O}$ be the learned output with respect to the input $\mathbf{X}$, STZipN model is a non-linear mapping of

[1] https://www.notebookcheck.net/Intel-Iris-Plus-Graphics-G7-Ice-Lake-64-EU-Laptop-GPU.422866.0.html

Table S3. Key statistics on the graph we constructed using a learned kernel width from the site locations and cull data.

Property	Value
Number of nodes	1368
Number of edges	12934
Network density	0.0138
Number of connected components	133
Diameter of the largest component	6
Mean path length of the largest component	2.39
Mean diameter	1.53
Median diameter	1
Nodes with no connections	22
Max node degree	76
Mean node degree	18.91
Median node degree	14
Mean clustering coefficient	0.83

Table S4. An example feature matrix.

X			
	s_1	s_2	s_3
t_1			
t_2			
t_3			
t_4			
t_5			

O			
	s_1	s_2	s_3
t_2			
t_3			
t_4			
t_5			
t_6			

the input to the output which can be represented as $\mathbf{O} = \text{STZipN}(\mathbf{X})$. By computing the gradients of $\mathbf{O}$ with respect to $\mathbf{X}$, we gain insights into how small changes in the COTS observation affect the estimated parameters of the distribution, providing us with sensitivity information about the model's output.

Let, X_j^i be the input corresponding to site i and day j and O_l^k be the estimated output corresponding to site k and day l and $l > j$. We compute the gradient $\partial \mathrm{O}_l^k / \partial \mathrm{X}_j^i$ to estimate the effect of observation at site i on time step j on the number of COTS observed at site k on time step l. We compute the spatial and temporal differences between O_l^k and X_j^i as D_{ik} and $T_{jl} = |l - j|$ respectively. To measure the effects of observations within a particular distance and time step, we sum the gradients within the required range and represent them as $f(\Delta s, \Delta t)$. Let $\Delta_s = (\rho_s, \rho_e)$ is a spatial range from ρ_s to ρ_e. The choice of spatial interval as 400 meters seems reasonable given that the sites are approximately 200 meters apart and selecting a spatial interval larger than the distance between adjacent sites should be appropriate to capture the influence of neighbouring sites on the model's output. Let $\Delta_t = (\eta_s, \eta_e)$ is a temporal range from η_s to η_e. Setting the temporal interval to 1 day is a reasonable choice as we have daily observations data and by selecting a temporal interval of 1 day, our gradient analysis should capture changes of the model's

output with respect to daily changes in the COTS population. Then,

$$f(\Delta s, \Delta t) := \sum_{\substack{i,k:\rho_s \le D_{ik} \le \rho_e \\ j,l:\eta_s \le T_{jl} \le \eta_e}} \frac{\partial O_l^k}{\partial X_j^i} \tag{S17}$$

In the given context, if we have $\rho_s = 400$ (lower distance threshold), $\rho_e = 800$ (upper distance threshold), $\eta_s = 14$ (lower time threshold), and $\eta_e = 15$ (upper time threshold), the function $f(\Delta s, \Delta t)$ represents the sensitivity of outputs to the sites that satisfy the two criteria: (1) the distance between the sites is more than 400 meters but less than 800 meters and (2) the sites were visited within 14 to 15 days.

By defining $f(\Delta s, \Delta t)$ in this way, we are specifying a subset of sites that fall within a specific distance range and were visited within a specific time range. The sensitivity of the outputs refers to how the model's predictions or estimations respond to changes in these selected sites.

We use the PyTorch autograd library that provides automatic differentiation capabilities. Automatic differentiation allows us to compute gradients of scalar-valued functions (in our case the STZipN model) with respect to input variables efficiently and accurately. Let us consider an example of our input **X** and output **O** given in Table S4. The example includes observations from three sites - s_1, s_2 and s_3 and six time steps t_1 to t_6. From the autograd library, we get gradients of each value of **O** with respect to all the values in **X**. For the shaded cell in **O**, corresponding to t_6 and s_2, we get the following gradients.

$$\begin{pmatrix} \frac{\partial O_{t_6}^{s_2}}{\partial X_{t_1}^{s_1}} & \frac{\partial O_{t_6}^{s_2}}{\partial X_{t_1}^{s_2}} & \frac{\partial O_{t_6}^{s_2}}{\partial X_{t_1}^{s_3}} \\ \frac{\partial O_{t_6}^{s_2}}{\partial X_{t_2}^{s_1}} & \frac{\partial O_{t_6}^{s_2}}{\partial X_{t_2}^{s_2}} & \frac{\partial O_{t_6}^{s_2}}{\partial X_{t_2}^{s_3}} \\ \vdots & \vdots & \vdots \\ \frac{\partial O_{t_6}^{s_2}}{\partial X_{t_5}^{s_1}} & \frac{\partial O_{t_6}^{s_2}}{\partial X_{t_5}^{s_2}} & \frac{\partial O_{t_6}^{s_2}}{\partial X_{t_5}^{s_3}} \end{pmatrix}$$

From such gradient matrices, we can summarise the gradients as follows:

δt	δs	grad value
$\lvert t_1 - t_6 \rvert$	$\lvert s_1 - s_2 \rvert$	$\frac{\partial O_{t_6}^{s_2}}{\partial X_{t_1}^{s_1}}$
$\lvert t_1 - t_6 \rvert$	$\lvert s_2 - s_3 \rvert$	$\frac{\partial O_{t_6}^{s_2}}{\partial X_{t_1}^{s_3}}$
$\vdots$	$\vdots$	$\vdots$
$\lvert t_5 - t_6 \rvert$	$\lvert s_2 - s_3 \rvert$	$\frac{\partial O_{t_6}^{s_2}}{\partial X_{t_5}^{s_3}}$

We compute the spatial differences δs between sites i and j as D_{ij} according Haversine formula [18].We use $f(\Delta s, \Delta t)$ to represent the effects of observations within a certain distance range and temporal range. For a spatial range from ρ_s to ρ_e and a temporal range from η_s to η_e, $f(\Delta s, \Delta t)$ is computed by Eq. S17.

REFERENCES

1. D. A. Westcott, C. S. Fletcher, F. J. Kroon, *et al.*, "Relative efficacy of three approaches to mitigate crown-of-thorns starfish outbreaks on australia's great barrier reef," Sci. Reports **10** (2020).
2. R. C. Babcock, E. E. Plaganyi, S. A. Condie, *et al.*, "Suppressing the next crown-of-thorns outbreak on the great barrier reef," Coral Reefs **39**, 1233–1244 (2020).
3. C. S. Fletcher and D. A. Westcott, "Strategies for surveillance and control: Using crown-of-thorns starfish management program data to optimally distribute management resources between surveillance and control," Reef Rainfor. Res. Centre Limited, Cairns **1** (2016).
4. D. A. Westcott, C. S. Fletcher, D. Gladish, and R. Babcock, "Monitoring and surveillance for the expanded crown-of-thorns starfish management program," Reef Rainfor. Res. Centre Limited, Cairns **1** (2021).

5. J. Gulland, "Catch per unit effort as a measure of abundance," Rapp. p.-v. Réun. Cons. Int. Explor. Mer **155**, 8–14 (1964).
6. D. W. Gladish, C. S. Fletcher, S. Condie, and D. A. Westcott, "Environmental factors influencing the distribution of crown of thorns starfish on the great barrier reef," Reef Rainfor. Res. Centre Limited, Cairns **1** (2020).
7. D. Pagendam, S. Janardhanan, J. Dabrowski, and D. MacKinlay, "A log-additive neural model for spatio-temporal prediction of groundwater levels," Spatial Stat. **55**, 100740 (2023).
8. S. Kong, J. Bai, J. H. Lee, *et al.*, "Deep hurdle networks for zero-inflated multi-target regression: Application to multiple species abundance estimation," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence,* (2021), IJCAI'20.
9. C. X. Feng, "A comparison of zero-inflated and hurdle models for modeling zero-inflated count data," J. statistical distributions applications **8**, 1–19 (2021).
10. M. S. Pratchett, C. F. Caballes, J. A. Rivera-Posada, and H. P. Sweatman, "Limits to understanding and managing outbreaks of crown-of-thorns starfish (acanthaster spp)," Oceanogr. Mar. Biol. An Annu. Rev. **52**, 133–200 (2014).
11. D. N. Reshef, Y. A. Reshef, H. K. Finucane, *et al.*, "Detecting novel associations in large data sets," Science **334**, 1518–1524 (2011).
12. A. Tran, A. Mathews, C. S. Ong, and L. Xie, "Radflow: A recurrent, aggregated, and decomposable model for networks of time series," in *Proceedings of the Web Conference 2021,* (Association for Computing Machinery, 2021), WWW '21, p. 730–742.
13. T. Clevert, Djork-Arnéand Unterthiner and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)." in *International Conference on Learning Representation,* (2016).
14. S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation **9**, 1735–1780 (1997).
15. K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," Syntax, Semant. Struct. Stat. Transl. **1**, 103 (2014).
16. T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations,* (2017).
17. B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence,* (2018), pp. 3634–3640.
18. C. C. Robusto, "The cosine-haversine formula," The Am. Math. Mon. **64**, 38–40 (1957).
19. C. Shen and H. Li, "On the dual formulation of boosting algorithms," IEEE Transactions on Pattern Analysis Mach. Intell. **32**, 2216–2231 (2010).
20. T. Akiba, S. Sano, T. Yanase, *et al.*, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining,* (Association for Computing Machinery, 2019), KDD '19, p. 2623–2631.
21. C. S. Fletcher, M. C. Bonin, and D. A. Westcott, "An ecologically-based operational strategy for cots control:integrated decision making from the site to the regional scale," Reef Rainfor. Res. Centre Limited, Cairns (2020).
22. S. A. Condie, K. R. N. Anthony, R. C. Babcock, *et al.*, "Large-scale interventions may delay decline of the great barrier reef," Royal Soc. Open Sci. **8**, 201296 (2021).
23. B. Phipson and G. K. Smyth, "Permutation p-values should never be zero: calculating exact p-values when permutations are randomly drawn," Stat. applications genetics molecular biology **9** (2010).