**Overview of Model Training**

To build a model that uses environmental and temporal context to predict fish occupation of discrete depth bins, we applied probabilistic deep learning (Durr, 2020) and framed the problem as a classification of fish into those bins. As such, the model was trained over a dataset where each sample in that dataset corresponds to one fish at a specific location and time and that fish's "classification" is defined as the depth bin it occupied. Features for each sample are derived from the environmental and/or temporal context associated with the depth bins at that position. Model training then aimed to reduce categorical cross entropy – a loss function whose negative measures the likelihood of the observed data given the predictions. Deep learning was selected as such models can learn highly non-linear, combinatorial patterns (Durr, 2020).

Broadly, models were built using the following steps. First, we retrieved Chinook salmon movement tracks from prior research in the Gulf of Alaska and Eastern Bering Sea (Seitz, 2024). The position and depth data were then transformed to provide samples for a classification model and split into training and validation sets. Environmental data was retrieved from the Copernicus Marine Service and temporal data derived from “ephem” and “suntimes” packages in Python. These data were then aggregated, scaled, and joined to the movement tracks to provide model features. Given the high dimensionality of this feature space, a novel dimensionality reduction technique was applied. Models were trained over a variety of increasingly comprehensive feature sets and the final model selected using loss over the validation set. The following paragraphs describe each of these steps in detail.

**Details of Model Training**

Chinook salmon movement tracks (Seitz, 2024), consisting of daily estimated positions and depths were used to train and test our models. Fish (62-100cm fork length) were caught and tagged near Dutch Harbor, AK and Chignik, AK (n=tbd), Homer, AK (n=tbd), Kodiak, AK (n=tbd), Yakutat, AK (n=tbd), Sitka, AK (n=tbd), and in the Eastern Bering Sea (n=tbd). In the EBS they were caught via midwater trawl or by hook whereas all other catches happened by hook and line (Seitz, 2023). Fish were then monitored using pop-up satellite archival tags that collect temperature, ambient light intensity, and depth information at specified (sub day) intervals during deployment and then release from the fish, surface, and transmit data over satellite. While the data is sampled at a sub-minute interval, the central tendency of depth occupancy is summarized per 15-minute period before being uploaded to conserve battery power. If the tag is recovered, the full dataset is available. To obtain location estimates, the data from the tag is passed through a proprietary hidden Markov model from Wildlife Computers that uses the tag’s data to estimate the likely paths taken by the fish during tag deployment (Wildlife Computers, 2025). An estimate of location can then be derived as the central tendency of those paths on a particular day. In total 7,532 observation days were collected across the fish. This data was especially well suited to our problem as it has the distinct advantages of being fisheries independent and giving a comprehensive view of depth occupancy throughout tag deployment no matter where the fish may go.

After collecting these tag tracks, observations were prepared for use in classification. To prevent over-representing the precision of the uploaded data, we sampled depth measurements from a normal distribution centered at the 15-minute depth summary with a standard deviation equal to 10% of the summary. We then assigned each sample to a depth bin. Depth bins were divided into increments of 25, 50, 75, 100, 150, 200, 250, 300, 400, and 500 meters. Each number indicates the upper bound of a bin and does not include depths from shallower bins. These depth bins were then normalized to a decimal between 0 and 1 with bin 25 corresponding to 0.1, 50 to 0.2, and so on. Finally, fish positions were aggregated to a fixed, hexagonal spatial grid - resolution 4 Uber H3 cells – in order to not over-represent the accuracy of the estimated positions.

Environmental conditions were derived from the Global Ocean Biogeochemistry Hindcast (10.48670/moi-00019) and Global Ocean Physics Reanalysis (10.48670/moi-00021) datasets provided by the E.U. Copernicus Marine Service. Data were drawn for each day between January 2013 and January 2023 in the Northern Pacific and values were aggregated to means per day, depth bin (see above), and Uber H3 cell at resolution 4. Environmental conditions chosen were chlorophyll, net primary production, nitrate, oxygen, phosphate, silicate, bottom elevation, mixed layer thickness, salinity, temperature, and north and easterly current velocity. These were then joined to the fish tracks on location and time.

Seasonality, lunar cycle, and diel features were derived using the datetimes associated with each sample along with the “ephem” and “suntimes” packages in Python for lunar and diel features respectively. In all cases, time was represented by taking the cycle in question, decomposing it to radians (0 at the beginning of the cycle and at the end of the cycle), and then providing the sine and cosine of that feature to the models. Specifically, we constructed features indicating the number of days through the year (seasonality), progression through the day or night (cosines at 0 for sunrise and sunset, sines positive during the day and negative at night), and a final set of sines and cosines indicating progress through the lunar cycle.

To ensure all covariates were on the same scale, chlorophyll, net primary production, and mixed layer thickness were log scaled and then all environmental features were rescaled to be between 0 and 1. Given the range of sines and cosines are -1 to 1, no further normalization was needed for the temporal features.

Next, we made a modification to the typical probabilistic deep learning classification problem to reduce the dimensionality of our feature set. Given we have features for each depth bin, selecting N covariates results in a feature space of dimension 10N (N features across 10 depth bins). The 10x multiplier is an issue because the volume of data required to fit a model effectively can grow exponentially with the dimensionality of the feature space (Verleysen, 2005). Instead, we trained a model that predicts the log odds of occupancy in a specific depth bin. As only one depth bin’s features are needed in this model, our dimensionality drops to N. To train this model using categorical cross entropy we copied the log-odds model weights across all choices, passed depth bin in as a feature to each choice, and then passed the output of each model through a softmax activation layer whose weights are an untrainable identity matrix. This results in a model that still predicts probabilities per depth bin but has an effective dimensionality of N + 1 thereby increasing our odds of a good fit.

One issue with this approach is that with 10 depth bins most instances of the internal log-odds model are encouraged to report very low log-odds. This is equivalent to a class imbalance problem. We can rebalance the data by taking advantage of the fact that the log-odds of one choice is independent of the others. As a result, we can down-sample the number of choices in any decision to just two and achieve a balance the positive and negative classes. To ensure we capture the variety in choices, we then resampled the same decisions repeatedly to get different pairs of selected and unselected choices. We call this process contrast sampling as we are sampling selected vs unselected contrasts. In this specific case we decided on a random sample (with replacement) of 5,000 decisions per individual and 10 choices per decision. Over a validation/training split of 39/72 this resulted in 5,550,000 contrasts of which 3,600,000 were used in training and the rest in validation.

Models were implemented in Keras and trained for 100 epochs using a batch size of 200,000 and an Adam optimizer with a learning rate of 0.001. A grid search was performed over 2 and 3 hidden layers and 24 and 32 units per hidden layer and the model with the lowest validation over the contrast set was then selected. Amazon Web Service’s Batch Fargate service was used for compute and training was performed on instances with 2 vcpus and 4GB memory.

To facilitate feature selection and investigate the predictive power of different feature sets, several models were trained over feature sets of increasing complexity. All feature sets were subsets of the environmental and temporal features discussed above. The model with the lowest loss over the contrast validation set was selected as the final model. Then, starting from no features, the final model’s features were incrementally added to build additional models used to investigate the importance of each set of features. Each incremental feature was selected according to whichever caused the maximal drop in loss. For each of these models, in order to make comparisons, categorical cross entropy was computed over the validation and training sets for both the contrasts and the original decisions.