To provide likelihoods per depth bin we applied probabilistic deep learning (Durr, 2020) to classify fish into bins. In our posing of the problem, each training example corresponded to one fish at a specific location and time. A fish's "classification" was defined as the depth bin it occupied, and features were derived from the environmental and/or temporal context associated with each depth bin at that position. Categorical cross entropy was used as the loss function as it maximizes the cumulative likelihood of our model’s predictions over the data. These combination of choices results in a classifier that predicts the likelihood of depth bin occupancy per fish given local context. We selected deep learning as our modeling framework as it has proven very capable of learning non-linear, combinatorial patterns (Durr, 2020).

Building the models proceeding along the following steps. First, we retrieved and sanitized Chinook salmon movement tracks from prior research in the Gulf of Alaska and Eastern Bering Sea (Seitz, 2024). That data was then transformed into targets for classification 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 and the models were trained. Models were trained over a variety of increasingly comprehensive feature sets and selected based on loss over the validation set. The following paragraphs describe each of these steps in detail.

As target data for our classification, we selected the Chinook salmon movement tracks (Seitz, 2024) as they have 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. These tracks were obtained 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 data is aggregated to a 15-minute period before being uploaded to conserve battery power. The data is then passed through a proprietary algorithm from Wildlife Computers that determines likely longitude and latitude during each day of monitoring (Wildlife Computers, 2025).

Fish were caught and tagged near Dutch Harbor, AK, 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 and line whereas all other catches happened by hook and line (Seitz, 2023). Fork length varied from 62-100cm. In total this amounted to 7,532 observation days across all fish.

In comparing those data with the aggregated data received over satellite, it was determined that the measurements uploaded only provide a sense of the central tendency of the fish during that time. Therefore, to prevent over-representing the precision of the uploaded data we sampled depth measurements from a normal distribution centered at the uploaded depth with a standard deviation equal to 10% of the uploaded depth.

Tagging data was then standardized for use in classification. We assigned each measurement to a depth bin in increments of 25, 50, 75, 100, 150, 200, 250, 300, 400, and 500 meters. Each bin indicates its upper bound and does not include depths from shallower bins. This gave us an assigned depth bin for every 15-minute interval during tag deployment for each of the 111 fish used in the study. 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 Uber H3 cells at resolution 4 in preparation for joining to environmental data.

Environmental context was 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 statistics were aggregated to means per day, depth bin (see above), and Uber H3 cell at resolution 4. Statistics 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.

Temporal features were derived using the “ephem” and “suntimes” packages in Python. The former was used to compute day/night features and the latter to compute lunar cycle features. Seasonality was derived using the date associated with each sample. 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 prevent issues during training, 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’ll 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. The model with the lowest validation over the contrast set. Amazon Web Service’s Batch Fargate service was used for computing 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. 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.