A Framework for Studying Behavior Using Deep Learning

Marcel Gietzmann-Sanders, Michael Courtney, Andrew Seitz, Curry Cunningham

University of Alaska Fairbanks

Introduction

Value of Probabilistic Modeling

Framing

This methodology centers on a two-stage cycle of discovery: (1) testing hypotheses about covariates that are believed to predict behaviors and (2) using the results to generate new hypotheses.

In the first stage, we model behaviors as decisions made from a set of choices, each associated with hypothesized covariates. Then, using machine learning, we build a predictive model to evaluate the likelihood of these selections based on the covariates.

In the second stage, we evaluate how the model shifts the likelihood of specific individuals or decision groups relative to baseline models. Improved performance reveals where the covariates more effectively predict behavior, while underperformance indicates missing covariates. This focus on areas of "greatest possible effect" provides case studies for generating new hypotheses, enabling iterative refinement of the process.

The keystone of this entire process becomes having a clear, quick means of producing machine learning models from our framing of decisions and choices and our covariates, an issue which we turn to next.

Theory

Standard probabilistic deep learning networks are typically framed as a classification problem, using categorical cross-entropy as the loss function (Oliver Durr (2020)). Each output neuron represents a potential choice, with the model predicting the probability of each choice being correct based on this loss formulation. For these choices, we provide the network with features encapsulating the relevant information. Training is then comprised of providing a series such decisions.

However, this formulation introduces a critical challenge: if there are N features per choice and M potential choices, the overall dimensionality of the input space becomes $N \cdot M$. Adding even a single feature increases the dimensionality by M not just 1.

This growth poses a significant challenge due to the "curse of dimensionality", where the amount of data required to effectively train models grows exponentially with the dimensionality of the input space (CITE).

Log-Odds Modeling

To address this issue, we propose an alternative framing. Instead of predicting the probabilities directly, we predict the log-odds ϕ_m for each choice and calculate the probability p_m using the softmax function:

$$p_m = \frac{e^{\phi_m}}{\sum_{m=1}^{M} e^{\phi_m}}$$

This approach reduces the feature space dimensionality to N and effectively increases the number of training examples by a factor of M.

We can implement this log-odds model using standard probabilistic deep learning techniques by replicating the log-odds computation across all M choices. The outputs are fed into a softmax layer with M units, where the layer's weights are set to the identity matrix and biases are set to zero. Using categorical cross-entropy as the loss function ensures compatibility with standard probabilistic deep learning while enabling us to train the log-odds weights and significantly reduce the problem's dimensionality.

Contrast Sampling

As M grows large, a practical issue arises: for each training example, most instances of the internal log-odds model would ideally report very low log-odds, resulting in low probabilities. Ideally, only one choice should produce $p_m = 1$. This is analogous to a class imbalance problem, where the model becomes prone to predicting the most common class (CITE).

To address this, we balance the training data. Instead of presenting the model with full decisions containing all M choices, we create training pairs, or contrasts, where each pair consists of one selected choice and one unselected choice. This approach is valid because the log-odds model focuses on the relative likelihood of choices, making the number of choices considered at any one time irrelevant.

The primary risk in using contrasts is introducing bias by disproportionately sampling certain combinations of choices. To mitigate this, we randomly sample pairs from each decision, ensure an equal number of contrasts per decision, and an equal number of decisions per individual. This preserves the balance across the training data and avoids skewing the model's predictions.

Alternative Approaches to Reducing Dimensionality

There are various strategies for reducing the dimensionality of one's data. One straightforward method is to limit the choices available. For instance if making decisions among a set of directions, one could reduce the precision of the angles allowed. While this approach reduces the model's specificity, it significantly simplifies the dimensionality of the problem, which is often a key challenge in probabilistic deep learning.

The methodology proposed, however, achieves dimensionality reduction without sacrificing specificity. That said, there are scenarios where some choices are unlikely to be relevant. In such cases, dimensions corresponding to these irrelevant choices can simply be discarded, further simplifying the model.

Another effective strategy to enhance the data set is leveraging the order invariance of choices in traditional probabilistic problem framing. Specifically, the order in which choices are presented to the model should not matter. For instance, whether a particular choice appears in the first or the thirteenth position should have no impact on the model's operation. This property allows for data augmentation by reordering choices.

In essence, for each training example, $M!$ (factorial of the number of choices) augmented examples can be created. While this exponentially increases the size of the data set as the feature space grows, it provides a mechanism to generate additional training examples from the same data.

The issue with this approach is that as your augmented data size grows to match the needs of the greater dimensionality so too does the time required for training. So while the problem remains theoretically possible, the exponential uptick in time complexity poses a significant practical issue. The proposed approach does not suffer from this issue and remains linear in time with the amount of data provided.

Application

Data

We consider a series of tracks from 111 Chinook salmon (*Oncorhynchus tshawytscha*) caught and monitored between 2013 and 2022 (CITE). These tracks were obtained from pop-up satellite archival tags which collect temperature, light level, and depth information at specified (sub day) intervals. This data is then passed through a proprietary algorithm from Wildlife Computers to determine likely longitude and latitude during each day of of monitoring (Computers (2024)).

Environmental data was derived from the Global Ocean Biogeochemistry Hindcast dataset (10.48670/moi-00019) and the Global Ocean Physics Reanalysis (10.48670/moi-00021) from the E.U. Copernicus Marine Service Information. Net primary production (mg/m3/day) and mixed layer thickness (m) were aggregated per Uber h3 resolution 4 cell in the Northern Pacific.

Formulation

The resolution 4 Uber h3 cell containing each salmon location was identified and then, assuming a maximum travel distance of 100km (centeroid to centeroid) all adjacent cells within the 100km were identified as choices (including the currently occupied cell). In general this represented ~ 19 choices per decision with the intention being to predict the probability of moving to any particular cell. Training data was derived by identifying the actual cell moved to.

Features

Movement heading in radians and distance to the centroid of the choice cell were computed and then mixed layer thickness and net primary production were joined to the choices on cell and day.

Distance was normalized to a range of 0-1 by division by 100, while mixed layer thickness and net primary production were both log-scaled and then centered at zero.

Contrast Sampling

After inspecting the distribution of number of choices per salmon and number of choices per decision, we decided on random sampling (with replacement) 200 decisions per individual and 19 choices per decision.

Over a test/validation/training split of 40, 71 this resulted in 345,800 contrasts of which 269,800 were used in training and the rest in validation.

Note that only 14,200 training examples would've been available to a traditional probabilistic approach representing a large increase in the number of available training examples.

Training

Three sets models were trained, one including only distance, one with both distance and movement heading, and one with all four features. Note that while the feature dimensions of these models are 1, 2, and 4 respectively, given the maximum number of choices per decision seen was 33 they dimensionality of a standard probabilistic model would've been 33, 66, and 132 representing a large reduction in the dimensionality of our feature spaces.

Architectures/hyperparameters for the log-odds component of the model were parametrized in the following ways:

Component	Options
Layers	3, 4
Units per Layer	24, 32
Batch Size	10000
Learning Rate	0.0005

With 5 models trained for each combination. Models were trained in Keras using an Adam optimizer for 100 epochs.

Models for each set of features we selected on the basis of the loss over the validation set of contrasts.

Results

In training the metric of interest is the contrasts normalized log probability (C-NLP) - the average log probability per contrast. For an equivalent metric over all of the decisions we also computed the average log probability per decision for each individual and then computed an average over those across individuals (in order to not favor individuals with many decisions). This is the D-NLP reported in the table below.

Model	Train C-NLP	Val C-NLP	Train D-NLP	Val D-NLP
No Model	-0.693	-0.693	-2.944	-2.944
Model 1	-0.172	-0.154	-1.336	-1.223
Model 2	-0.156	-0.150	-1.281	-1.200
Model 3	-0.147	-0.146	-1.248	-1.180

"No Model" assumes all decisions are equally likely, Model 1 is the distance only model, Model 2 adds the movement heading, and Model 3 adds the net productivity and mixed layer thickness features.

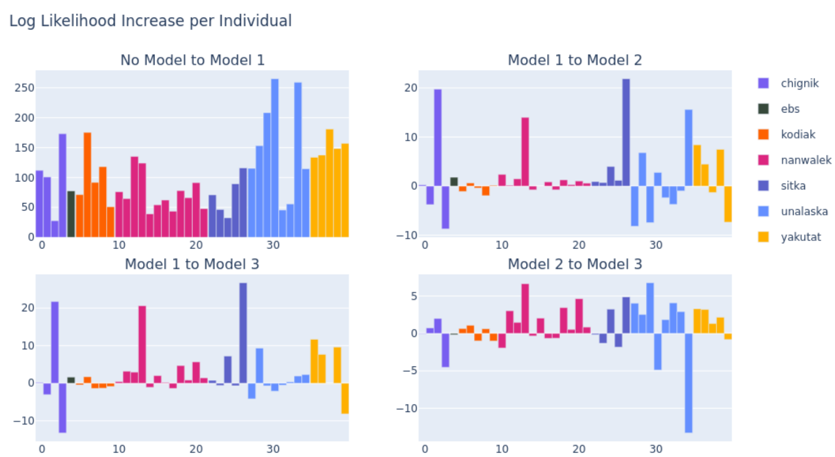

Figure 1: Log likelihood increase (unnormalized) per individual of each model change.

Figure 2: Log likelihood increase (unnormalized) per individual of each model change in cases where the selection involved no movement.

Figure 3: Log likelihood increase (unnormalized) per individual of each model change in cases where the selection involved movement.

Figure 1 shows the unnormalized changes in log likelihood for each individual colored by the region in which they were initially tagged as we go from one model to the next. Figures 2 and 3 break this down for decisions that resulted in no movement and movement respectively. Only individuals in the validation set were considered.

The appendix contains a series of mapped examples (figures 4-9) from each of these regions (besides EBS) that shows how the log likelihood per decision (when moving) changed in moving from model 1 to model 2 and then from model 2 to model 3. As in the above, only individuals in the validation set were considered.

Finally as much of the behavior seems to be modulated by movement distances (or lack of movement) a summary table of empirical likelihood of movement of a specific distance is given (taken over all individuals).

Distance Bin (km)	Likelihood of Selection	# Training Decisions
No Movement	65.6%	3163
Up to 50km	32.5%	1567
50km to 100km	1.9%	92

Discussion

Model One: Distance as a Key Feature

Starting with Model One, we observe a significant improvement in log-likelihood across all individuals in the validation set. This outcome is intuitive, given that the most probable action for these salmonids is to remain in place. This behavior highlights the importance of distance as a feature, as it dramatically reduces the likelihood of alternative actions. Beyond staying in place, most movement is limited to a range of 0 to 50 kilometers, with only a small percentage extending further.

This demonstrates that a baseline model treating all locations as equally likely is a gross oversimplification of salmon behavior. Introducing distance as a feature alone vastly improves the model's performance, as it aligns more closely with observed patterns.

Model Two: Movement Heading

Model Two adds movement heading as a feature, offering further improvement, though not as dramatic as Model One's addition of distance. Movement heading enables the model to learn general directional trends in salmon migration, which are particularly influenced by the geography and location of the fish.

Examining changes in log-likelihood between Models One and Two, we see a general pattern: southeasterly movements gain likelihood, while northern, northwesterly, and northeasterly movements often see decreases. This aligns with the data, as the farthest-moving fish predominantly migrate southeast. However, the inclusion of movement heading introduces variability—while most fish experience an increase in likelihood, some see decreases due to the model imposing a general directional preference that may not align with all individual behaviors.

Model Three: Environmental Features

To refine the model further, we incorporate environmental features such as net primary productivity and mixed layer thickness in Model Three. These features provide information about productivity in the fish's habitat. Their inclusion results in another improvement in log-likelihood, particularly for fish previously associated with less-likely movements (e.g., westward or northward). This suggests a correlation between environmental productivity and movement behavior.

However, as with movement heading, the addition of these features benefits some individuals while diminishing the likelihoods for others. This variability reflects the nuanced interactions between salmonid behavior and environmental conditions, reinforcing the importance of continuing to explore exemplar cases where the model performs poorly.

Iterative Refinement Through Exemplars

At each stage, we identify individuals with the lowest log-likelihood improvements compared to their baselines. These exemplars provide valuable insights into potential missing covariates or unmodeled behaviors. By iteratively incorporating new features and examining their effects, we uncover diverse behavioral patterns and refine the model's predictive capability.

Throughout this process, the value of contrast as a guide is evident. Improvements in normalized log-likelihood consistently align with better decision predictions, demonstrating that the model is effectively trained to assess the odds of observed behaviors. This contrast-driven approach allows for incremental learning and refinement, ensuring the model captures key patterns while maintaining flexibility to incorporate new information.

This methodology proves powerful in understanding salmonid movement by enabling rapid integration of new features, establishment of revised baselines, and identification of exemplar behaviors for further analysis. By iteratively refining the model, we continue to discover and represent the complex ecological and behavioral dynamics underlying salmonid movement.

Computational Performance and Efficiency

The computational performance of these log-odds models proved to be efficient and straightforward. All models were trained in the cloud using a two-core, 4GB machine, with training times ranging from 15 to 30 minutes over 100 epochs. The variation in training time depended on the complexity of the model and the number of features included.

This efficiency scales well in a cloud environment, as parallelizing model training across multiple machines is both feasible and straightforward. By distributing the workload, all models for various subsets of the data can be trained simultaneously, maintaining the same 15 to 30-minute window for training and arriving at a new baseline.

Within the proposed workflow—where a baseline model is built based on hypothesized important features, and poorly performing individuals are identified as exemplars for further investigation—the computational time spent on training is a minor component. The iterative process of identifying features, gathering new data, and refining the model dominates the time and energy expenditure.

As such, the short training time ensures that computational efficiency is not a bottleneck, allowing for rapid iteration and refinement.

Given the combination of short training times, scalability in cloud environments, and ease of integration into the iterative process, these log-odds models are highly efficient tools for data exploration and feature discovery. The author concludes that their use is well-suited to workflows requiring repeated refinement and baseline adjustments.

References

Computers, W. (2024). Minipat.

Oliver Durr, Beate Sick, E. M. (2020). *Probabilistic Deep Learning.* Manning Publications.

Appendix: Mapped Examples

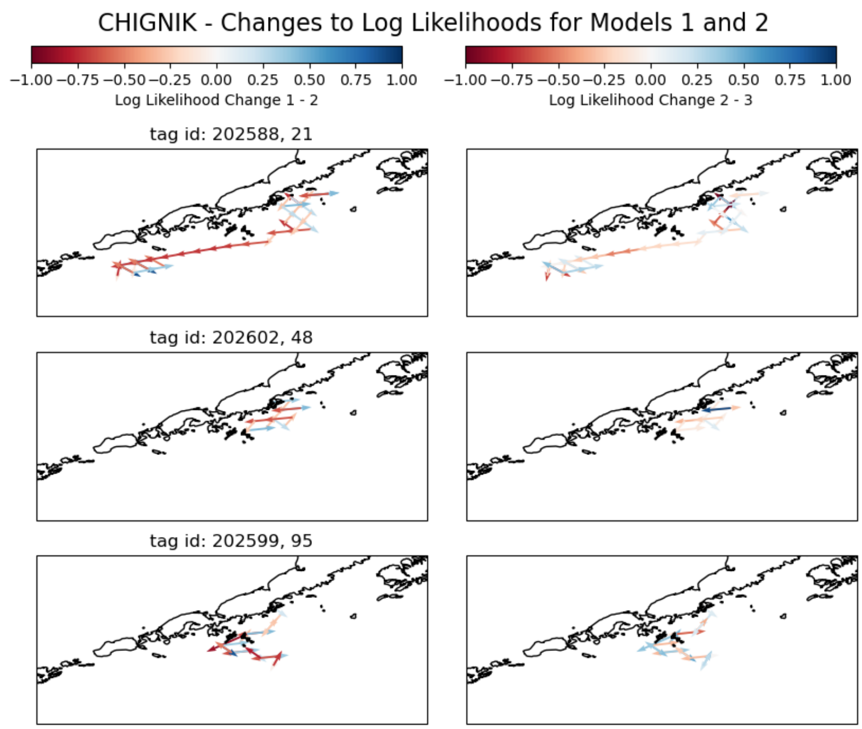

Figure 4: Log likehood changes per decision - Chignik

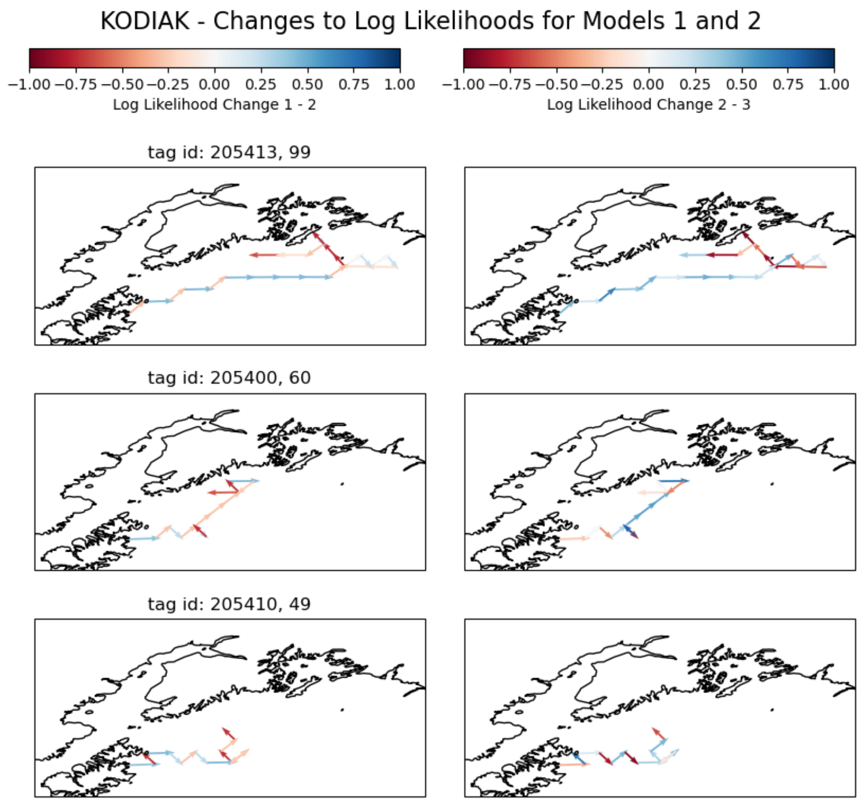

Figure 5: Log likehood changes per decision - Kodiak

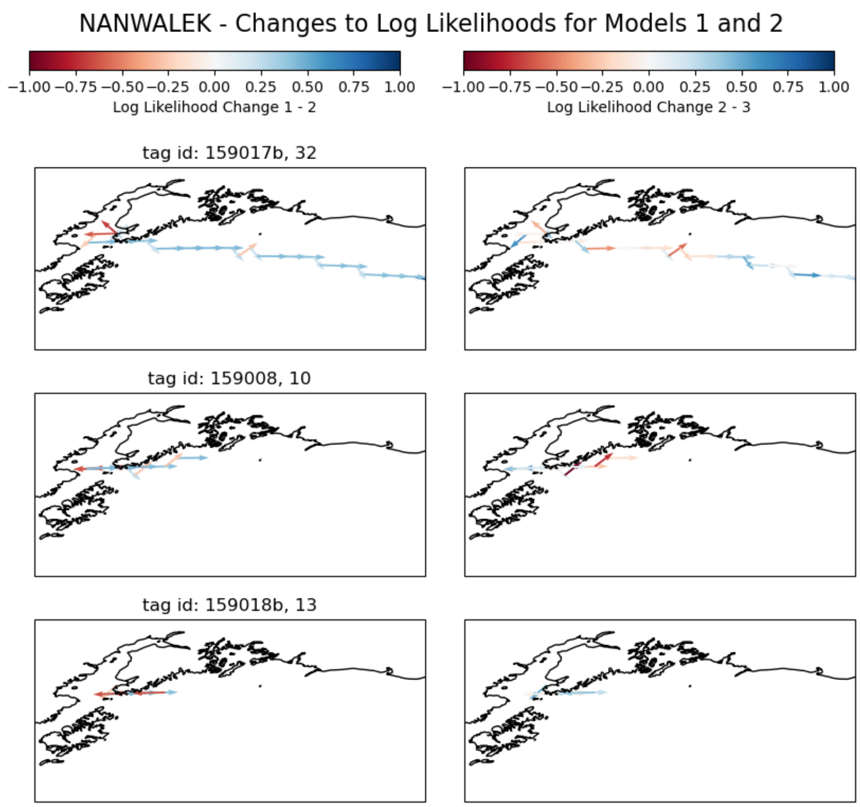

Figure 6: Log likehood changes per decision - Nanwalek

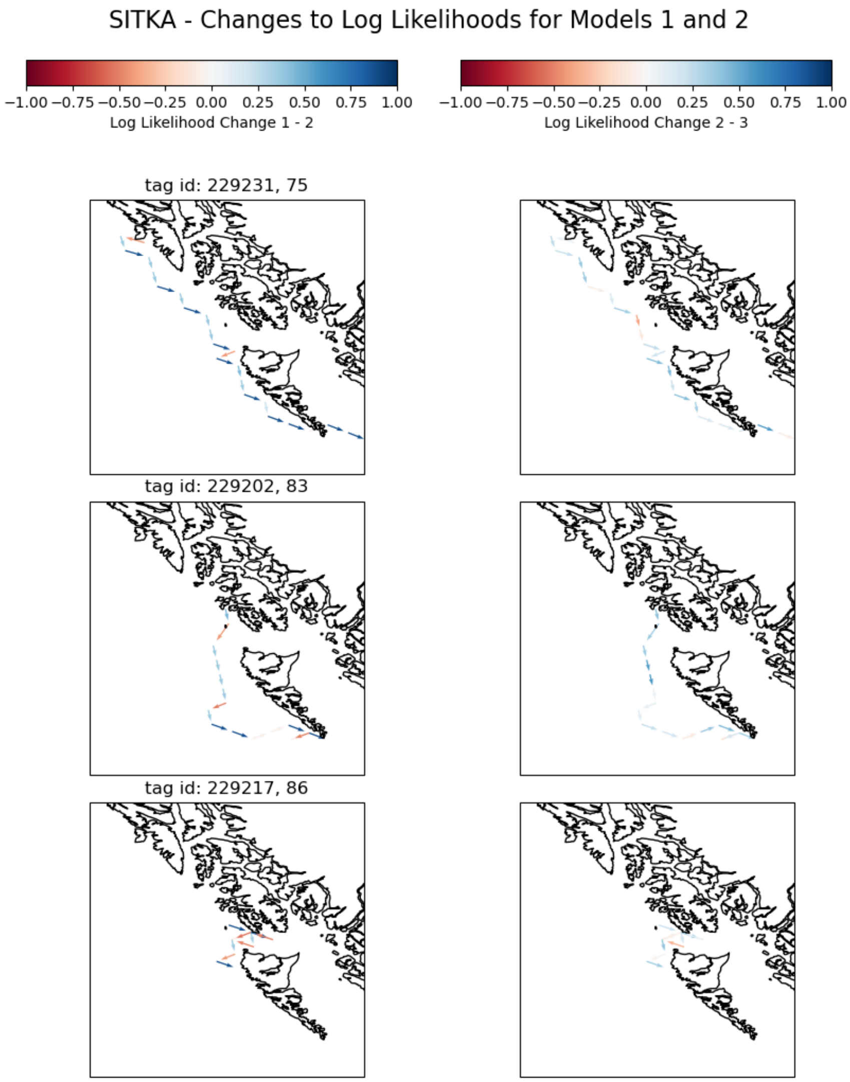

Figure 7: Log likehood changes per decision - Sitka

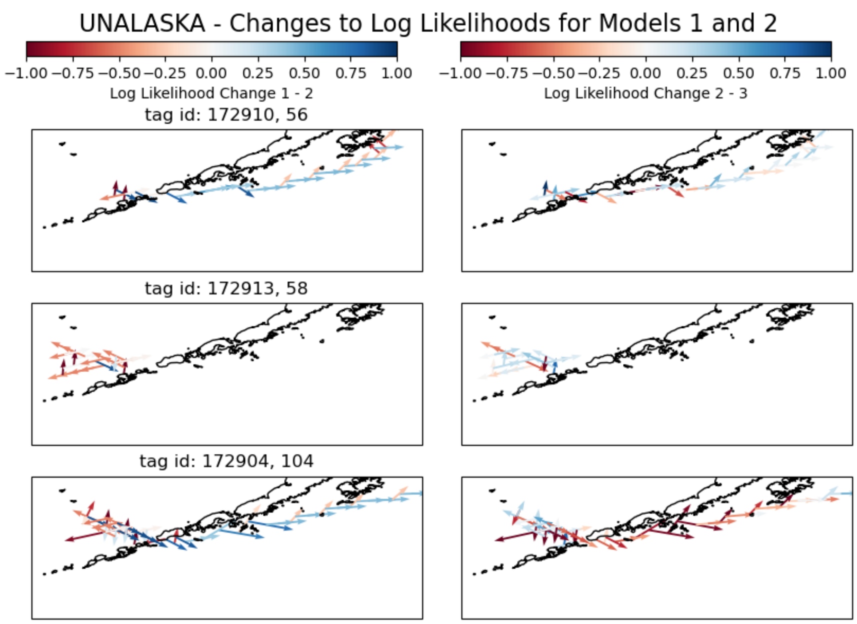

Figure 8: Log likehood changes per decision - Unalaska

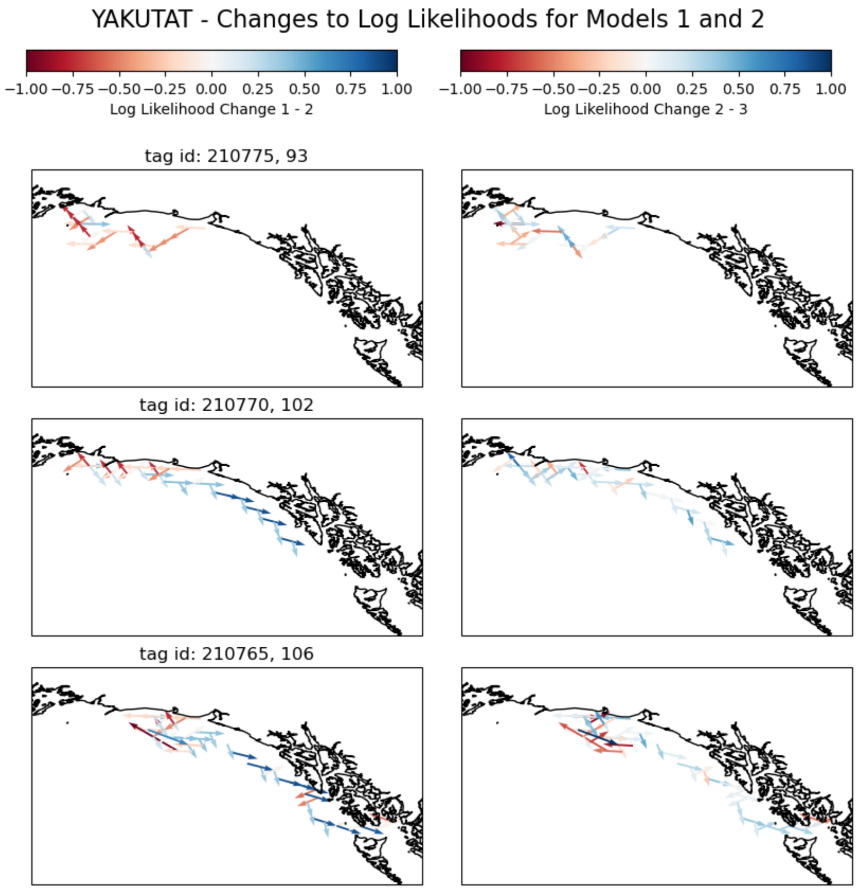

Figure 9: Log likehood changes per decision - Yakutat