Estimating Confidence and Precision
in the NOSIA NOAA Value Tree
for Various Decision Scenarios
Using Beta Distributions

Louis Cantrell, PhD (Profitable Weather, LLC)
A white paper drafted for DOC/NOAA/NESDIS/SAE/RAAD/TPIO
April 2026 (version 6)

Introduction

One purpose of the National Oceanic and Atmospheric Administration (NOAA) value tree constructed in the NOAA Observing System Integrated Analysis (NOSIA) is to prioritize the impact of a portfolio of observing systems used by an interconnected network of products and services to address various outcomes and mission service areas. This information does not directly facilitate a decision but can serve as one building block to inform decisions.

Decisions that NOSIA is designed to inform are many. Some of them include:

- Which observing programs can NOAA defund under a constrained budget while maintaining the highest mission accomplishment across all of NOAA's missions?
- Which observing programs are critical for continuity of services?
- Which missions have the greatest sensitivity to observing program improvements and which observing programs should be targeted for those investments?

To answer these questions, we need to capture priorities that motivate the creation of NOAA's service activities, the complexity of NOAA's network of products and services that rely on observations. These services include human-in-the loop processes, regime-dependent performance, budget-driven resources, and evolving capabilities. More directly, we need to understand the relevance and performance of the network, observing program cost information, portfolio budget information, portfolio capability trades and observing system requirement satisfaction analyses. We posit here that the addition of uncertainty information can minimize risks in selecting an optimum observing portfolio to maintain and improve.

This way of thinking motivates our objective to model uncertainty formally and rationally within the constraints of the NOSIA elicitation framework that seeks to collect as much information as possible about the use of NOAA's observing system portfolio without overburdening our subject matter experts (SME). Without uncertainty, our value tree implicitly treats all inputs as equally well-known. Modeling uncertainty will allow us to distinguish between high-confidence capability and assumptions that may expose NOAA's mission services to hidden risk.

A portfolio prioritization and budget optimization with uncertainty can identify solutions that optimize performance and allow us to test decisions under real-world variability. Including uncertainty makes our decisions more defensible to leadership because we can show not only what choices they have but also what the relative scale of the risks are.

A single performance score cannot represent performance across all conditions. Uncertainty allows us to capture real operational variability. Without including uncertainty information in our assessments, we introduce a false precision to support portfolio investment decisions. By incorporating uncertainty information, we can move from status quo scoring to dynamic scenario analysis which can better estimate risks. Incorporating uncertainty will also improve the credibility of NOSIA because it will align the study with established best practices in other mission-critical decision environments.

If we move from scalar assessments of performance to probabilistic distributions of performance, we can obtain more information from our surveys. Collecting uncertainty information during product surveys can improving SME elicitation quality. Because it requires deeper insight, it disciplines subjectivity by capturing SME confidence, ambiguity, and regime dependencies that are not captured in scaler estimates of performance. Thinking about uncertainty also leads SMEs away from anchoring, confirmation bias, and framing effects. This is not adding subjectivity, instead, it structures the discussion with the SME it so that their confidence in the scores we collect can be measured and managed. Adding a confidence rating is a low-cost extension to our SME elicitation framework that could extract significantly more actionable information. Collecting confidence information also brings minimal additional burden on SMEs.

Sources of Uncertainty

Sources of uncertainty include but are not limited to these general categories:

- The environment
- The product
- The product data sources
- The SME
- The Model

Uncertainty of the environment refers to the natural earth process environment that NOAA addresses. This refers to all phenomena that motivates NOAA’s investments in products and services, infrastructure, observing system programs, research activities, and collaborations that serve as inputs to those products and services to meet various societal benefits, goals, missions, outcomes. These phenomena have a natural variance and uncertainty in their frequency, timing, duration, intensity, and location that NOAA’s products and services seek to mitigate. For example, tropical cyclones, fires, fish populations, and ocean currents occur whether NOAA addresses them or not, but generally their range of values are bounded.

Uncertainty of the product refers to the products and services NOAA develops and delivers to mitigate the uncertainty of the environment. A tropical cyclone forecast is designed to reduce the uncertainty of where, when, and how strongly a hurricane will impact a community. Coupled climate, ocean circulation, and fish population model are designed to reduce the uncertainty of where, when, and how much fish can be harvested without overfishing the resource. But the performance of these products and services are rarely perfect due to random noise and how they are structured. This implies there may be uncertainty introduced by random noise, how sensitive the product is to the variations in the environment, or how they are structured. They are often insufficient to meet the needs of every application of them which implies an uncertainty in their application or user suitability.

Uncertainty of the product data sources refers to the variance of the networked services, observations, and processes used as inputs to construct products and services. This could include data source sensitivity to the natural environment, the relevance of the data source to the product; data source attributes of performance such as geo-coverage, spatiotemporal resolution, latency, measurement accuracy, and availability or the ability to detect a phenomenon such as a frontal boundary, ecosystem tipping point, or risk of flood. There could be uncertainty in how calibrated the data source is to what it measures or how calibrated the product is to the data source.

Uncertainty of the SME refers to the variance of the subject matter expert. NOSIA uses expert elicitations from SMEs because empirical evidence is unsuitable or not available, which nevertheless introduces both rational and subjective uncertainties. This includes the SME's expertise, the SME's mental model of the product and its data sources, and the SME's mental model of a NOSIA product survey elicitation. Uncertainty also arises in the SME's adherence to heuristic biases such as anchoring, availability, representativeness, or confirmation. There is uncertainty in perspectives of the product. By that we mean the level of sensitivity and satisfaction may be widely different among users who depend on the product. There is uncertainty from the range of perspectives of different SMEs who may be surveyed about the product. There is also uncertainty in the perspective of the product or data source performance estimates between the requirement satisfaction for a single product or for a generalized agency wide measure of requirement satisfaction.

Uncertainty of the Model refers to the variance of the NOSIA NOAA Value Tree. Uncertainties introduced here include the degree of fidelity it has to meeting the objectives for the use of the model. For example, the model is designed to provide an optimum portfolio under a constrained budget. The model also introduces uncertainty in how well outcomes, services, and observing systems are sampled including their granularity; how well NOAA's objectives, services, and data sources are integrated into a representative graph of value; how well it models sensitivity to performance changes via its edge functions, and how well its responses are calibrated to real-world decisions.

Selecting a Probabilistic Distribution to Enhance Product Analysis

The NOSIA elicitation strategy uses a psychometric scale that measures product performance between 1 and 100 under various conditions. These conditions include the product status quo condition that involves all of its currently available data sources, and scenarios with the availability of all but one of each of its data sources.

Performance (Satisfaction) Scale		
100	**Ideal**	Meets all requirements and exceeds some
90	**Fully Satisfied**	Meets all requirements
80	**Good**	Meets all major requirements, with minor limitations
60	**Fair**	Meets most major requirements, with significant limitations
40	**Poor**	Fails to meet many major requirements, but provides some value
20	**Very Poor**	Fails to meet most major requirements, but provides minor value
1	**No Capability**	Provides no value

Table 1. Translating Product Performance into Numerical Values.

This elicitation approach does not account for the natural variability of the product's performance under the status quo budget condition. It is not entirely clear what a scalar status quo requirement satisfaction score can imply. Does a score of 80 mean that it is always meeting most requirements regardless of regime or application, or does it mean that 80% of the time it meets or exceeds requirements?

The environment has natural variability and therefore uncertainty. NOAA's products and services are motivated to reduce this uncertainty with information: e.g., when and where will hurricanes make landfall; how much fish can be safely harvested to preserve its availability; how will space weather affect communications; where will sea ice affect shipping lanes; etc.? These products and services depend on data source investments, which is the primary purpose of NOSIA's assessments. However, they also depend on processes, data source sampling, data assimilation, and other logistic considerations to deliver useful information. This motivates an estimation of product uncertainty that is not captured simply by eliciting performance under various data source availability scenarios.

By using a probability distribution, we can capture the SME's assessment of the product's uncertainty in maintaining performance. Variation in product performance is not sampled in our current elicitation approach. Examples of unsampled variations can arise from how the product is produced, from regional or seasonal performance variation, from data source performance variation, and from the product's sensitivity to the natural variability of the environment.

Converting performance scores to probability distributions supports a broader modeling strategy. Wide distributions in performance imply greater uncertainty or less precision than narrow distributions. Uncertainty can help us assess risk that scalers cannot. Probability distributions also let us combine scores from disparate SMEs when there is disagreement in performance. When modeling a constrained budget that limits NOAA's ability to invest in observing systems, we expect that as product performance declines, performance uncertainty will climb. Probability distributions let us model this uncertainty.

A Beta Distribution is useful for adapting to our elicitation and value modeling strategies.

- Because the Performance Satisfaction Scale can be transformed to between 0 and 1, it is amenable to the Beta Distribution which is also bounded in the same range.
- The Beta Distribution is easy to employ because it depends on only a few shape parameters.

- Performance scores can be represented as modes in the probability distribution, and its shape can be simply modified to accommodate variance metrics.
- We can later revise the variance of the distribution by evidence. For example, if the performance of the service varies more or less than the SME estimated, we can modify the distribution to better reflect that. This means the variance in the distributions could be empirically calibrated to forecast validations, to forecast usefulness or timeliness, or to specialized product performance studies such as Observing System Experiments (OSE).
- It may be necessary to represent service variance in multiple dimensions for the environmental regimes (e.g., Winter/Summer; benign, moderate, or extreme conditions), for dependence on observing system quality, or for organizational process or product generation processes (e.g., model Data Assimilation, forecaster skill, or available Days at Sea for sampling fisheries). The objective here is to make the uncertainty in the distribution empirically consistent.
- The Beta Distribution can be applied to Bayesian approaches. A SME's score can provide the mode and confidence as a "prior distribution" and an updated distribution estimated for future observation capabilities a "posterior distribution". This allows us to merge expert judgements and historical service behaviors with estimating how uncertainties will change from expected future observation capabilities.
- Products and services made available at NOAA have a cadence and sequential behavior; they are rarely single service products. Examples include frequent services such as weather observations, forecasts, and advisories, and infrequent services such as seasonal outlooks and annual fisheries reports. The performance of a sequence of NOAA services can help measure variability in the performance of the service and help calibrate a SME's assessment of variability in the performance of their product.

A Two-Question Script to Translate Status Quo Scores into a Beta Distribution

Instead of eliciting only a scalar, we ask the SME two specific questions during our NOSIA assessments:

1. The Mode (Status Quo Score).

    "On a scale of 1 to 100, what is the most likely performance score for this product right now?"

    Math parameter: This becomes the mode (m).

2. The Confidence Level

    "On a scale of 1 to 5, how confident are you in that exact score? (1 being very uncertain, 5 being absolutely certain)."

    Math parameter: Map this 1-5 rating to a hidden "Concentration Parameter" (κ). The higher the κ, the tighter and taller the probability curve.

    If uncertainty metrics are available, such as from Observing System Experiments, we can convert those metrics to concentration parameters.

    If the SME is unable to provide a confidence rating, we can default to $\kappa = 16$, implying "Moderately Confident".

A more detailed elicitation protocol is given in Appendix 1.

To translate these two answers into Beta Distribution shape parameters, α and β, we use a concentration parameter, κ, which represents the total "weight" or "sample size" of the expert's belief and the SME's status quo score as the mode, m, of the distribution. The shape parameters α and β represent the frequency of success and failure respectively. The mean of the distribution is the amount of success divided by all cases of success and failure. The expected value (mean) of the distribution is given by:

$$\mu = \frac{\alpha}{\alpha + \beta}$$

The mode of the Beta Distribution, which represents the SME's performance score, is given by:

$$m = \frac{\alpha - 1}{\alpha + \beta - 2}$$

The concentration parameter of the Beta Distribution is given by:

$$\kappa = \alpha + \beta$$

SME Confidence Rating	Meaning	Concentration Parameter (κ)
Min **κ**	Maximum Uncertainty	$2^1 = 2$
1	Very Uncertain (Wide-Spread)	$2^2 = 4$
2	Somewhat Uncertain	$2^3 = 8$
3	Moderately Confident	$2^4 = 16$
4	Highly Confident	$2^5 = 32$
5	Absolutely Certain (Tight Peak)	$2^6 = 64$
Max **κ**	Minimum Uncertainty	$2^7 = 128$

Table 2. Confidence rating vs. Concentration parameter

Once we have elicited the mode (m) and concentration parameter (κ) from the SME, we can compute the Beta Distribution shape parameters (α and β).

$$\alpha = m(\kappa - 2) + 1$$
$$\beta = (1 - m)(\kappa - 2) + 1$$

This mode only applies when $\alpha > 1$ and $\beta > 1$. The area under the probability curve is equal to 1.

Example:

Moderator Question: *"On a scale of 1 to 100, what is the most likely performance score for this product right now?"*

SME Answer: *"It performs at an 80 out of 100."*

Mode Parameter: $m = \frac{SQ}{100} = 0.8$

Moderator Question: *"What is your confidence in this estimate?"*
SME Answer: *"I've only seen limited data lately, so I'm somewhat uncertain. I'll give it a 2."*
Concentration Parameter: Confidence Rating 2 → $\kappa = 8$

Compute Beta Distribution shape parameters from m and κ:

$$\alpha = m(\kappa - 2) + 1 = 0.8(8 - 2) + 1 = 5.8$$
$$\beta = (1 - m)(\kappa - 2) + 1 = (1 - 0.8)(8 - 2) + 1 = 2.2$$

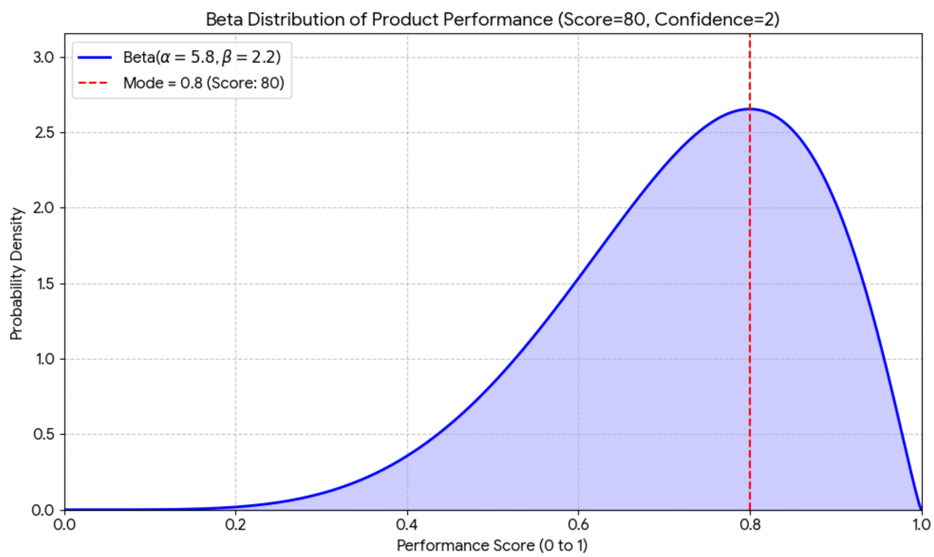

Figure 1. Beta Distribution elicited from a Status Quo score of 80: "Meets all major requirements with minor limitations", and a confidence score of 2: "Somewhat Uncertain".

We can then pass these values into the value tree graph for more detailed risk analysis. The variance of this distribution is given by:

$$Var(X) = \sigma^2 = \frac{\alpha\beta}{\kappa^2(\kappa + 1)}$$

For the elicited example above, the mean and variance are given by:

$$\mu = \frac{\alpha}{\alpha + \beta} = \frac{5.8}{5.8 + 2.2} = 0.725$$
$$Var(X) = \sigma^2 = \frac{\alpha\beta}{\kappa^2(\kappa + 1)} = \frac{5.8 \times 2.2}{8^2(8 + 1)} \approx 0.022$$

We can apply this reasoning to the product's data sources as well. Figure 2 illustrates how the shape of the beta distribution varies for a constant mode with variable concentration parameters. Figure 3 illustrates how the shape of the beta distribution varies for a constant concentration parameter for various modes.

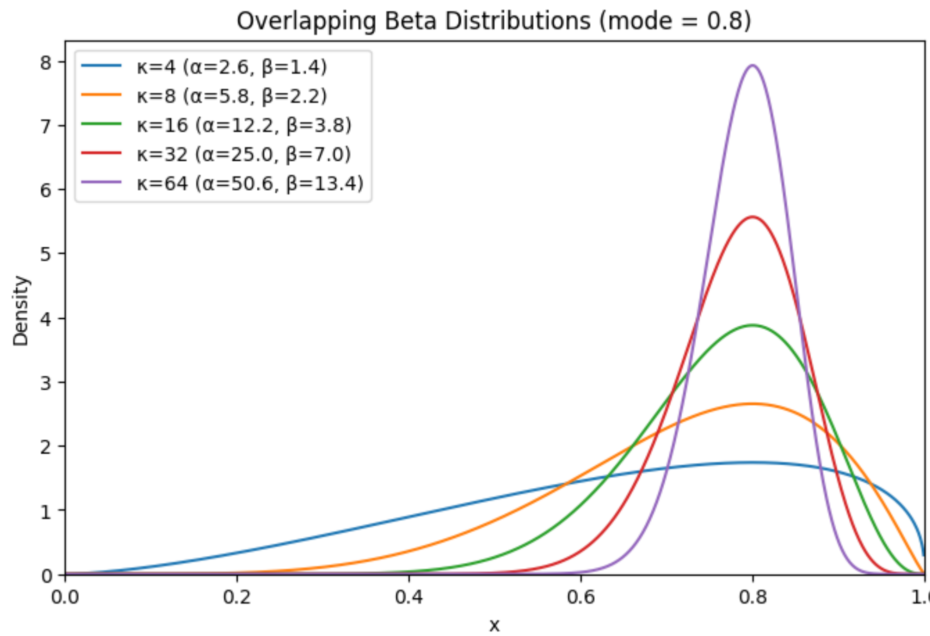

Figure 2. Variation of beta distribution for a mode of 0.8 (i.e., SQ = 80) under various concentration parameters representing confidence. Higher confidence produces a tighter distribution.

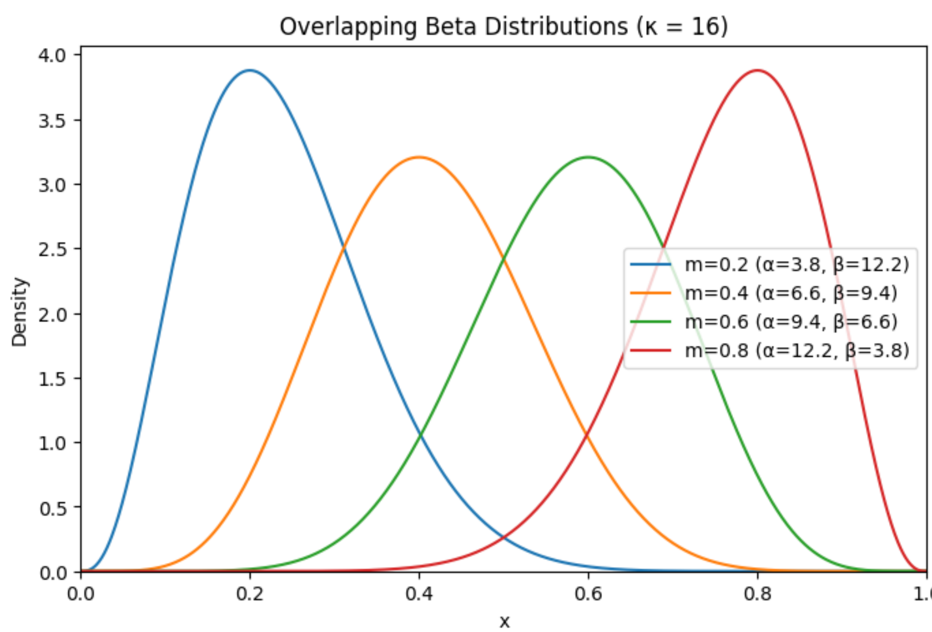

Figure 3. Variation of beta distributions for a concentration parameter of 16, “Moderately Confident” for various modes from status quo scores.

Unfortunately, single SME confidence scores conflate fundamentally different uncertainties and becomes hard to interpret or validate. However, our objective is to introduce probability density information into the value tree with as little impact to the elicitation workload on the SME as possible. What follows is an outline of the elicitation and modeling procedure.

Modeling Procedure

A product elicitation collects a status quo score that is designated the mode of the beta distribution. The elicitation also collects a single confidence rating. The confidence rating is converted to a concentration parameter. From the mode and the concentration parameter, we compute the remainder of the beta distribution shape parameters.

The elicitation collects a number of data sources and elicits product swing scores that represent the performance mode of the product when an individual data source is unfunded, i.e., the data source performance is swung to zero. Swing scores represent new modes for the product. A changing mode is assumed to result in changing uncertainty. For each of the data source scenarios, swing scores are converted to modes, then concentration parameters are estimated using a $\log_2$ procedure. From the modes and computed concentration parameters, we can estimate the remining beta distribution shape parameters for these swing scenarios.

Next, the elicitation collects the performance and a confidence rating of each data source. These scores are converted to modes and concentration parameters and subsequently used to calculate the remaining distribution shape parameters for the performance of the data sources. After the elicitation, the modes of the status quo product score, the swing scores, and the data source performance scores are used in NOSIA's edge calculation procedures to establish a functional relationship among the modes.

Once the functions are created, we can test the value tree with scenarios that either defund a data source or improve its performance. The returned score from this function is a new mode that depends on the performance of the data sources. In our modeling framework, changes in performance occur independent of uncertainty. For example, if a product changes from a mode of 0.8 to a mode of 0.6 because one of its data sources were defunded, the performance mode delta of 0.2 has no contribution from its prior or posterior values of uncertainty. However, our intuition is that changing a data source's performance will change each data source's uncertainty and ultimately change the uncertainty of the product.

Changes in Uncertainty with Changes in Mode

Our intuition is that as a product loses data sources resulting in a decreasing mode, then we expect the uncertainty to increase. Similarly, if the product's data sources improve, then we expect the variance to decrease as its performance increases. To demonstrate the logic of this, consider a shipping lane product that relies on observations of sea surface temperature and the presence of ice in a shipping lane. The natural environment's cloud coverage, SST, and ice presence has some natural variability that exists regardless of whether or not it is observed. We assume that for this product an observation of SST is not available if ice or clouds are present. The product will perform well if there are no clouds, so the performance of the product has some variability depending on cloudiness. The shipping lane product's uncertainty is also driven by the uncertainty of the two observation types it does have. If the SST observation became unfunded, then the uncertainty mitigated by the SST observation is no longer transferred to the product and the product become more uncertain and the product performs poorer. In other words, this introduces an intuitive connection between performance changes and uncertainty changes.

We expect uncertainty to be a maximum when the mode approaches zero. This would then represent the maximum uncertainty of the environment without the benefit of the product to make an informed assessment of it. We expect uncertainty to be a minimum when the mode is nearing 1. This of course is an assumption because it is entirely possible for perfectly performing data sources to be used by an imperfectly assimilating process. But for the purposes of a value trade space, we believe this assumption is rational. To capture variation in uncertainty from changes in performance, we need to first explore the range of possible values for the mode, the concentration parameter, and the variance.

If the performance mode is approaching a minimum of zero, we expect the concentration parameter to be approaching a minimum of 2. Values less than 2 produce multi-modal conditions which we seek to avoid.

$$
\begin{aligned}
m_{min} &\approx 0 \\
\kappa_{min} &\approx 2^1 \\
\alpha_{min} &= m_{min}(\kappa_{min} - 2) + 1 \approx 1 \\
\beta_{min} &= (1 - m_{min})(\kappa_{min} - 2) + 1 \approx 1 \\
\sigma_{max}^2 &= \frac{\alpha_{min}\beta_{min}}{\kappa_{min}^2(\kappa_{min} + 1)} \approx \frac{1}{12} = 0.0833
\end{aligned}
$$

If the performance mode is approaching a maximum of 1, we expect the concentration parameter to be approaching a maximum of 2^7. This is an arbitrarily selected maximum value for kappa to support our formulation.

$$
\begin{aligned}
m_{max} &\approx 1 \\
\kappa_{max} &\approx 2^7 \\
\alpha_{max} &= m_{max}(\kappa_{max} - 2) + 1 \approx 100 \\
\beta_{max} &= (1 - m_{max})(\kappa_{max} - 2) + 1 \approx 100 \\
\sigma_{min}^2 &= \frac{\alpha_{max}\beta_{max}}{\kappa_{max}^2(\kappa_{max} + 1)} \approx \frac{1}{2{,}113{,}536} \approx 4.7 \times 10^{-7}
\end{aligned}
$$

These values imply a parameterized variance in the range that spans five orders of magnitude: $\sigma^2 \in [4.7 \times 10^{-7}, 8.33 \times 10^{-2}]$

We need to make the concentration parameter κ a function of the mode and of the SME's assigned confidence level. A derivation of the spline we have chosen is given in Appendix 2. The procedure defines a parameter γ_0 for an anchor point (m_0, κ_0). When the value tree computes m_{new} using its graph edge functions, we than apply the spline to solve for a new κ_{new} given the new m_{new}.

$$
\begin{aligned}
\gamma_0 &= \frac{ln\left(\frac{log_2(\kappa_0) - 1}{6}\right)}{ln(m_0)} \\
\kappa_{new} &= 2^{1+6{m_{new}}^{\gamma_0}}
\end{aligned}
$$

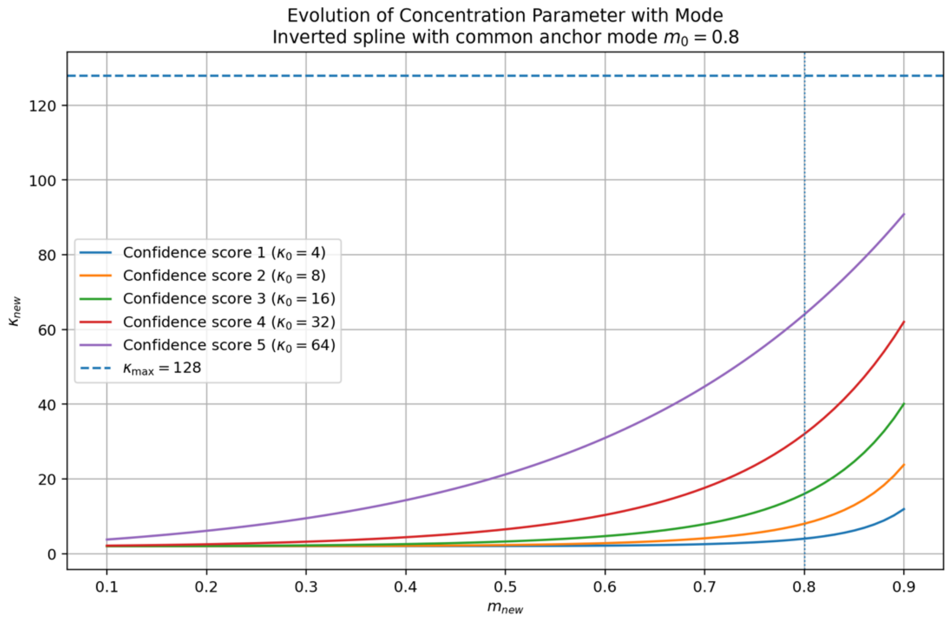

Figure 4. Demonstration Plot of variation in κ_{new} from m_{new} for different confidence scores set using a spline anchor of (m_0) = (0.8), which represents an initial status quo score of 80.

Figure 4 illustrates how the spline will adjust the concentration parameter when a performance mode changes from a prior to a posterior. The spline is more sensitive to prior concentration parameters that are high. Next, we demonstrate in more detail how the elicitation collects the parameters for the functional relationship between a product and its data sources using beta distributions. We elicit the following information depicted in Figure 3. The survey collects information about the product's status quo performance and confidence in that score, the data sources that the product relies on, how the product's performance is expected to change if the data source was defunded (i.e., swing score), how well the data source meets the performance requirements of the SME, and how confident the SME is in the data source performance score.

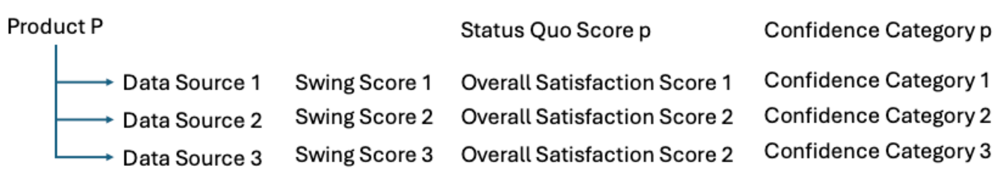

Figure 3. Data collected during SME elicitation for a product P and three data sources.

Product Status Quo Score: SQ_0

$$m_0 = SQ_0/100$$

Product Confidence Category: CC_0

$$\kappa_0 = f(CC_0): table \text{ (Elicited)}$$

Product Distribution Shape Parameters

$$\alpha_0 = f(\kappa_0, m_0) = m_0(\kappa_0 - 2) + 1$$
$$\beta_0 = f(\kappa_0, m_0) = (1 - m_0)(\kappa_0 - 2) + 1$$
$$\mu_0 = f(\alpha_0, \beta_0) = \frac{\alpha_0}{\alpha_0 + \beta_0}$$
$$\sigma_0{}^2 = f(\alpha_0, \beta_0, \kappa_0) = \frac{\alpha_0 \beta_0}{\kappa_0{}^2(\kappa_0 + 1)}$$

Data Source Swing Scores: SS_i

$$m'_i = SS_i/100$$

We do not elicit confidence metrics for swing scores to obtain the swing score concentration parameter (κ'_i); instead, we compute them based on the status quo mode (m_0), status quo concentration parameter (κ_0), and swing score mode (m'_i) using the spline.

$$\gamma_0 = \frac{ln\left(\frac{log_2(\kappa_0) - 1}{6}\right)}{ln(m_0 - 1)}$$
$$\kappa'_i = 2^{1+6(m'_i - 1)^{\gamma_0}}$$

We then collect the remaining data source distribution shape parameters for the swing scores.

$$\alpha'_i = f(\kappa'_i, m'_i) = m'_i(\kappa'_i - 2) + 1$$
$$\beta'_i = f(\kappa'_i, m'_i) = (1 - m'_i)(\kappa'_i - 2) + 1$$
$$\mu'_i = f(\alpha'_i, \beta'_i) = \frac{\alpha'_i}{\alpha'_i + \beta'_i}$$
$$\sigma_i{}^2 = f(\alpha'_i, \beta'_i, \kappa'_i) = \frac{\alpha'_i \beta'_i}{\kappa'_i{}^2(\kappa'_i + 1)}$$

Data Source Overall Satisfaction Scores: OSS_i

$$m_i = OSS_i/100$$

Data Source Confidence Categories: CC_i

$$\kappa_i = f(CC_i): table \text{ (Elicited)}$$

Data Source distribution shape parameters

$$\alpha_i = f(\kappa_i, m_i) = m_i(\kappa_i - 2) + 1$$
$$\beta_i = f(\kappa_i, m_i) = (1 - m_i)(\kappa_i - 2) + 1$$

$$\mu_i = f(\alpha_i, \beta_i) = \frac{\alpha_i}{\alpha_i + \beta_i}$$

$$\sigma_i{}^2 = f(\alpha_i, \beta_i, \kappa_i) = \frac{\alpha_i \beta_i}{\kappa_i{}^2(\kappa_i + 1)}$$

Next, we use NOSIA's Interval Preserving Symmetric Extended Average – Power version (IPSEA-P) to functionally relate the mode of the product to the modes of its data sources. The derivation of the IPSEA-P function is given in Cantrell (2026).

Solve for s_2 using numerical methods, $s_2 = f(n, m_0, m'_i)$:

$$\left(\frac{n-1}{n}\right) m_0{}^{s_2} - \frac{1}{n}\sum_{i}^{n} m'_i{}^{s_2} = 0 \pm \in$$

Solve for $s_1$using numerical methods, $s_1 = f(\mu_0, m'_i, m_i)$:

$$\sum_{i}^{n} \frac{m^{s_2} - m'_i{}^{s_2}}{m'_i{}^{s_1}} = 1 \pm \in$$

Solve for w_i, the relevance of the data source, $w_i = f(w_i, m_0, m'_i)$:

$$w_i = \frac{m_0{}^{s_2} - m'_i{}^{s_2}}{m_i{}^{s_1}}$$

Solve for the return score mean value for the product, $m_p = f(n, w_i, m_i, s_1, s_2)$:

$$m_p = \left(\sum_{i}^{n} w_i m_i{}^{s_1}\right)^{\frac{1}{s_2}}$$

This function gives us the ability to model how the mode of a product's performance changes when the performance of its data sources change. We call this new mode a "return mode".

If we assume the variance of the product changes as it undergoes changes to its data source modes and its return mode, we can estimate the product's concentration parameter and then compute the remaining distribution parameters.

We estimate the return concentration parameter (κ_p) from the return mode (m_p), the status quo mode (m_0), and the status quo concentration parameter (κ_0).

$$\gamma_0 = \frac{ln\left(\frac{log_2(\kappa_0) - 1}{6}\right)}{ln(m_0 - 1)}$$

$$\kappa_p = 2^{1+6(m_p-1)^{\gamma_0}}$$

The remaining distribution shape parameters can now be computed.

$$\alpha_p = m_p(\kappa_p - 2) + 1$$
$$\beta_p = (1 - m_p)(\kappa_p - 2) + 1$$
$$\mu_p = \frac{\alpha_p}{\alpha_p + \beta_p}$$
$$\sigma_p{}^2 = \frac{\alpha_p\beta_p}{\kappa_p{}^2(\kappa_p + 1)} \approx \left(\frac{\alpha_p\beta_p}{\kappa_p{}^3}\right)$$

Finally, we can solve for the C = 90% confidence interval for the distribution of performance modeled for the product, compute the information gain and precision for various scenarios, and finally make violin plots and statements about scenarios.

Other Sources of Variance

One challenge with this framework is how uncertainty (variance: $\sigma_p{}^2$) propagates from inputs to outputs when the relationship is not purely statistical but mediated by human or algorithmic processes. Our procedure uses an elicitation approach that collects a confidence rating on a scale of 1 to 5 that translates the rating to a concentration parameter (κ) on a scale of 2^1 to 2^7. These elicited values are independently collected for the product and the product's data sources.

Because the product score is produced through a human-elicited functional relationship, the product's uncertainty is a blend of at least three things:

1. <u>Uncertainty in the input scores</u>. Confidence scoring procedure may be insufficiently granular.
2. <u>Attenuation or amplification by the production process</u>. Product production process may well or poorly assimilate, correct, or filter the variance of its data sources.
3. <u>Uncertainty in the SME's mental model of the product as a whole</u>. Product variance sensitivity to data source variance. Product confidence assessment may well or poorly accommodate redundancy or synergy, positive and negative correlation or regime sensitivity in the performance of its data sources. SME product and data source confidence may be over or under-estimated.

This means the product-level confidence rating should not be expected to equal some direct aggregation of the source-level confidence ratings. However, disagreement between them can be diagnostically useful. We can think of product uncertainty as having two components:

$$\sigma_p{}^2 = \sigma_{propagated}{}^2 + \sigma_{structural}{}^2$$

where:

$\sigma_{propagated}{}^2$ is the part of the product's variance inherited from its uncertain inputs.
$\sigma_{structural}{}^2$ is the extra uncertainty from the process itself:

- Data assimilation, filtering, input variance accommodation
- Regime dependence
- Unmodeled interactions (redundancy, synergy)
- Hidden covariances
- Lack of SME clarity about how the product behaves
- Adherence to agency requirement satisfaction metrics
- Variations in user-assessed confidence in performance

Because this decomposition is useful, we attempt next to estimate both terms. We first estimate the amount of variance reduced by the product.

$$\Delta{\sigma_0}^2 = {\sigma_{max}}^2 - {\sigma_0}^2$$

The term ${\sigma_{max}}^2$ represents the uncertainty of the product-dependent environment without the benefit of the product.

Next, we examine the amount of variance enhanced by the removal of each data source. For this we examine the change in variance from the status quo mode to the swing score mode.

$$\Delta{\sigma'_i}^2 = {\sigma'_i}^2 - {\sigma_0}^2 = w'_i{\sigma_i}^2$$

$$w'_i = \frac{{\sigma'_i}^2 - {\sigma_0}^2}{{\sigma_i}^2}$$

Where:

$\Delta{\sigma'_i}^2 \equiv$ increase in product variance due to loss of data source i.
${\sigma'_i}^2 \equiv$ variance computed from the mode of the swing score for data source i.
${\sigma_0}^2 \equiv$ variance computed from the mode of the product status quo score.
${\sigma_i}^2 \equiv$ variance computed from the mode of data source i.
$w'_i \equiv$ the relative weight of the variance of data source I computed from the above known values.

The propagated variance is then the weighted average of the change in variance from the status quo mode to the swing score mode.

$${\sigma_{propagated}}^2 = \frac{\sum\left(w'_i\Delta{\sigma'_i}^2\right)}{\sum w'_i}$$

The estimate for the structural variance is then given by:

$${\sigma_{structural}}^2 = {\sigma_p}^2 - {\sigma_{propagated}}^2$$

Confidence could be different things to different SMEs. It may reflect epistemic uncertainty (noise), algorithmic reliability, observational coverage, personal familiarity, etc. Accompanying the elicited performance score confidence metrics, we elicit the following information:

"What process do you employ to manage your confidence in this product status quo score?", or *"What do you think contributes most to your confidence assessment?"*

Examples of process might include data assimilation, algorithm design, QC, temporal smoothing, filtering, operational expertise, etc. We simply capture this information as a SME comment on their chosen confidence index.

The following function then describes the information contributed by the product to reduce the uncertainty of the environment that the product serves.

$$I_p = \frac{1}{\Delta\sigma_0{}^2} = \frac{1}{\sigma_{max}{}^2 - \sigma_0{}^2} = \frac{1}{\sigma_{max}{}^2 - \sigma_{propagated}{}^2 - \sigma_{structural}{}^2}$$

Another challenge in our framework is capturing error correlation among input data sources. If input data sources share correlated errors, the effective variance contribution can be different from the weighted sum of independent variances. Variance terms that move together (positively or negatively correlated) can offset each other resulting in an overall variance reduction when they are negatively correlated and a variance increase when they are positively correlated.

"To what extent do errors in data sources tend to occur together?"

This could guide the SME to place correlated data sources in product survey capability groups to help us compute the group's variance value. This group then adopts the weighted average of its child modes and the weighted average of its variances scores with an additional term. We could then introduce a covariance term in our equation for the variance of the capability group.

To include an estimation of Covariance, we need to elicit a scaling term for how much data sources reinforce each other. One bounded scale could be between [0,1] where 0 indicates complete independence in performance and 1 indicates they always move together. We then need to know if the performance is correlated (+1) or anti-correlated (-1).

For example, we can ask:

"How well related is the performance of your data sources on a scale of 0 to 1: 0=independent, 0.25=weekly related, 0.50=moderately related, 0.75=strongly related, and 1.00=always move together.?"

We then follow up with:

"Are these data source performance scores positively correlated or anti-correlated?"

This lets us assign a value to $\bar{\rho}_g$, which represents the average correlation coefficient for a group. For example, if data sources are moderately anti-correlated, we would use $\bar{\rho}_g = -0.50$. The equation for the variance of the group then becomes:

$$Var(group) = \sigma_{group}{}^2 = \left(1 - \bar{\rho}_g\right)\sum_{i=1}^{n} w_i^2\sigma_i^2 + \bar{\rho}_g\left(\sum_{i=1}^{n} w_i\sigma_i\right)^2$$

Where $w_i \equiv$ weight of child, $\sigma_i \equiv$ child variance, and $\bar{\rho}_g \equiv$ correlation coefficient for the group. When $\bar{\rho}_g = 0$, we recover the case in which all data sources in the group act independently. Notice that an anti-correlation ($\bar{\rho}_g < 0$) decreases the variance of the group.

Another latent correlation factor in our framework is the treatment of redundancy. Redundancy reduces variance. This could guide the SME to place redundant data sources in product survey capability groups to help us compute the group's variance value. We could then introduce a redundancy term to our variance function that reduces the variance depending on its redundancy with other inputs, e.g., *Var(group) = weighted average of Var(group children) x (1-R)*. We use the reward term *(1-R)* because we want the variance to decrease from its original value if there is some estimation of redundancy. The term *(1-R)* can be estimated by a ratio of the number of effectively redundant children to the number of actual children. For example, if there are three data sources, and if two of them are duplicates of the other being fully redundant, then the *(1-R) = (1-(2/3)) = 1/3*. For a more continuous redundancy definition, suppose we simply ask:

> "On a scale of 0 to 1, how would you characterize the redundancy of this group of data sources with 0 representing no redundancy and 1 representing fully duplicative?".

We can then compute the redundancy term as follows:

$$\sigma_{gwr}{}^2 = \sigma_g{}^2(1 - R) = \left(\sigma_g{}^2\right)\frac{1 + (N-1)(1-r)}{N}$$

Where:

- $R \equiv$ Redundancy
- $N \equiv$ number of children in group
- $r \equiv$ redundancy index in the range [0,1]
- $\sigma_g{}^2 \equiv$ group variance with covariance adjusted values
- $\sigma_{gwr}{}^2 \equiv$ group variance with redundancy and covariance adjusted variance

Another latent factor in our framework is "Conditional Variance". We could consider characterizing variance terms that represent performance in different regimes in time and space. We can collect this information as a SME comment on conditional variance for further study.

> *"Other than data source availability, under what conditions do these data sources most strongly affect product performance?"*

Assessing SME Experience

Our expert elicitations in NOSIA include questions about the experience level of our SME. Two SMEs giving the same answer do not provide the same information if one has 20 years operational experience and one is new to the system. Without experience metadata, our uncertainty model treats them as identical observations. Self-identification of experience level can capture familiarity with this specific product, confidence in this specific judgment and recency and relevance of experience. It permits the model to capture under-confidence typical of novices, overconfidence typical of intermediate, and calibrated confidence typical of experts. (O'Hagan, 2006) Currently, elicitations collect SME experience by asking the following question:

> "How many years of experience do you have?"

Past motivation for collecting this metadata was simply to demonstrate the combined years of experience from hundreds of SMEs contributing thousands of years' experience; however, this data contributes little to characterizing the uncertainty of the information collected. It also lacks definition or precision. For example, the SME could assign years since graduation in the field, since exposure to the product, since employed by the agency responsible for the product, or since assigned responsibility for the production or quality control of the product. Our intuition is that collecting the information gives NOSIA "confidence in the SME's confidence". To make better use of this type of information, we propose using four categories of experience for self-identification and then using the categories to adjust the concentration parameter that reflect typical confidence biases attributed to each category.

Our objective for including SME experience is to correct for bias and overconfidence. Our application is as follows:

- the SME gives a mode for product performance
- the SME gives a confidence rating, which we convert to a concentration parameter, kappa, representing the SME's stated uncertainty around that mode,
- we then use experience category to adjust that kappa, because experience is an estimate of how much trust to place in the SME's own confidence judgment.

Experience only modifies the estimate of dispersion, not the central estimate or mode. We propose asking the following question:

> "How would you classify your experience level with this product: Novice, Intermediate, Advanced, or Expert?"

Based on our intuition, the four categories with definitions and adjustments are:

- Novice:
  - limited direct experience with the product or service
  - strongest reduction in kappa ($c(E) = -0.25\kappa_{elicited}$)
- Intermediate:
  - some operational familiarity, but not sustained responsibility
  - moderate reduction in kappa ($c(E) = -0.15\kappa_{elicited}$)

- Advanced:
  - substantial direct experience and repeated application
  - no change in kappa ($c(E) = 0.00\kappa_{elicited}$)
- Expert:
  - recognized authority with deep operational and/or developmental knowledge
  - slight increase in kappa ($c(E) = 0.05\kappa_{elicited}$)

$$\kappa_{effective} = \kappa_{elicited} + c(E)$$

We also need to limit the effective concentration parameter to remain within our parameterized boundaries for kappa.

$$\kappa_{effective} = min\big(\kappa_{max},\ max(\kappa_{min},\ \kappa_{elicited} + c(E))\big)\text{, or}$$
$$\kappa_{effective} = min\big(2^7, max(2^1, \kappa_{elicited} + c(E))\big)$$

Connecting Products in the Graph

Ultimately, these products and their data sources are networked into a knowledge graph. These edges use the Piecewise Stineman Spline (PWSS) (Cantrell, 2025) to link the mode performance score of a product to the mode performance score of the product as data source.

We use the PWSS to compute and link these modes. As node modes change, their computed concentration parameter changes, and subsequently their shape parameters change. The resulting network not only propagates changes in data source performance but also propagates uncertainty throughout the network.

The implementation of a PWSS method in NOSIA uses a variation of the Stineman function (Stineman, 1980) that defines the boundary slopes $(x_0, y_0) = (0,0)$ and $(x_1, y_1) = (1,1)$ passing through an intermediate point: (x, y).

$$m_0 = \frac{y - xy}{x - xy}$$
$$m_1 = \frac{1}{m_0}$$
$$y = x + \frac{(m_0 x - x)(1 + m_1(x - 1) - x)}{(m_0 x - x) + (1 + m_1(x - 1) - x)}$$

where:

$y \equiv$ parent node performance mode value
$x \equiv$ child node performance mode value
$m_0 \equiv$ slope of the spline at the point $(x_0, y_0) = (0,0)$
$m_1 \equiv$ slope of the spline at the point $(x_1, y_1) = (1,1)$

Computing Variance in the Mission Hierarchy

The NOAA Value Tree consists of three main layers. At the top sits the mission value hierarchy, which relies on the weighted averages of key products and services. These key outputs are drawn from the middle layer, a densely interconnected network of products and services. This entire network is built upon the tree's foundational "leaf nodes", which is a diverse portfolio of observing systems, databases, and other raw data sources.

We use this graph to compute the performance of nodes, and from the discussion above, we have represented performance as probability distributions with modes representing performance. In the mission value hierarchy portion of the tree, we compute modes from a weighted average of its child modes. At this level of the value tree, our procedure seeks to aggregate the modes using simple weighted averages that will result in only one mode instead of permitting mixed modes from child inputs to the mission hierarchy. This is because modes are computed everywhere in the graph before we compute the variance about each mode.

Our process computes a single mode for each child key product, and then we demand that the aggregation of those child distributions produces a weighted average of their modes. However, each child key product brings with it an uncertainty in its performance probability distribution. It is these uncertainties that we wish to aggregate to apply to the aggregated node representing a group of key products. We seek to aggregate the variance of the child mode distributions using the same weights that are used for aggregating the modes.

At this level in the value tree, our procedure to compute the aggregated variance is as follows:

- Compute the aggregated mode (m_{agg}) value from the modes of the child products (m_i) using a weighted average.

$$m_{agg} = \frac{\sum w_i m_i}{\sum w_i}$$

- Compute the aggregated variance ($\sigma^2_{agg\_a}$) value from the variance of the child products (σ_i^2) using a weighted average.

$$\sigma_i^2 = \frac{\alpha_i \beta_i}{(\alpha_i + \beta_i)^2 (\alpha_i + \beta_i - 1)}$$
$$\sigma^2_{agg\_a} = \frac{\sum w_i \sigma_i^2}{\sum w_i}$$

- Compute an additional aggregated variance value ($\sigma^2_{agg\_b}$) to represent uncertainties for the group arising from a dispersion of its child modes. The term S is a scaling factor that minimizes this variance contribution if the aggregated concentration parameter nears 2^1 and maximizes this contribution if the aggregated concentration parameter nears 2^7.

$$L_{agg} = \frac{\sum w_i log_2(\kappa_i)}{\sum w_i}$$

$$S = \frac{L_{agg} - 1}{6}$$
$$\sigma^2_{agg\_b} = S\left(\frac{\sum w_i(m_i - m_{agg})^2}{\sum w_i}\right)$$

- Compute the total variance of the aggregated node.

$$V = \sigma^2_{agg\_b} + \sigma^2_{agg\_b}$$

- Now that the variance is computed with the penalty for distributed modes, we extract the aggregated concentration parameter (κ_{agg}) and shape parameters (α_{agg}, β_{agg}) using numerical methods to solve for the real roots of a cubic function of κ_{agg}.

$$\kappa_{agg}{}^3(V) + \kappa_{agg}{}^2(V - W) - \kappa_{agg}(-4W + 1) - (4W - 1) = 0$$
$$W = m_{agg}(1 - m_{agg})$$

- Finally, solve for the aggregated shape parameters α_{agg} and β_{agg}.

$$\alpha_{agg} = m_{agg}(\kappa_{agg} - 2) + 1$$
$$\beta_{agg} = \kappa_{agg} - \alpha_{agg}$$

A more detailed discussion of the aggregation procedure development is given in Appendix 3.

Treating Dispersion in User Scores

In addition to assessing uncertainty in product performance arising from differences between a product and the environment, another important source of uncertainty arises from differences between a producer and a user metric. When a SME assesses their estimate of product performance in meeting their customer's needs, that is only one score of performance. We have found in our value tree that users of the product may provide a wide range of product performance measures for the same product.

For example, we may survey a sea surface temperature (SST) product from a satellite such that the SST product SME believes the product performance has a mode at 0.8 with little variability because it is consistent in measurement accuracy and update rate. A user that depends on the SST may find that the product performs even better, say a score of 0.9 with even less variability, because they only use it for outlooks and they are less sensitive to its variance. Another user who produces a tropical cyclone warning may find that the SST product performs poorly, say a score of 0.5, because its measurement accuracy is not good enough to accurately forecast the movement of a tropical cyclone. In other words, there is variance in the user's scores from the source's score and confidence estimates.

The dispersion of product user scores about the source score, and the departure of the user's satisfaction score from the generalized agency requirement satisfaction score are contributors of uncertainty and should be accounted for. In these cases, we use the similar methods of aggregation that we applied to the mission hierarchy with a few caveats. To treat dispersion of user scores, we apply the following procedure.

- The values of source score mode and concentration parameter, m_{ss} and κ_{ss}, are already known from the return score for the product. From these values, we compute the beta distribution shape parameters, α_{ss} and β_{ss}, and then σ_{ss}^2, the baseline variance of the return score.

- All user scores m_{usi} and concentration parameters κ_{usi} are assumed to be equally weighted w_i. Then solve for aggregated user scores.

$$L_{usagg} = \frac{\sum log_2(\kappa_{usi})}{N}$$
$$S = \frac{L_{usagg} - 1}{6}$$
$$\sigma_{usagg}^2 = S\left(\frac{\sum(m_{usi} - m_{ss})^2}{N}\right)$$

- Compute the total variance of the aggregated user score variance with the penalty for distributed modes.

$$V = \sigma_{ss}^2 + \sigma_{usagg}^2$$

- Next, solve for updated aggregated concentration parameter of the source score (κ_{ssagg}) and shape parameters (α_{ssagg}, β_{ssagg}) using numerical methods to solve for the real roots of a cubic function of κ_{ssagg}.

$$\kappa_{ssagg}{}^3(V) + \kappa_{ssagg}{}^2(V - W) - \kappa_{ssagg}(-4W + 1) - (4W - 1) = 0$$
$$W = m_{ss}(1 - m_{ss})$$

- Finally, solve for the updated source score's aggregated shape parameters α_{ssagg} and β_{ssagg}.

$$\alpha_{ssagg} = m_{ss}(\kappa_{ssagg} - 2) + 1$$
$$\beta_{ssagg} = \kappa_{ssagg} - \alpha_{ssagg}$$

Another source of uncertainty arises from the difference between a SME's subjective assessment of performance with respect to a particular product and an objective requirement satisfaction score that simply measures the capability of an observing system against a particular organizationally defined requirement that motivated investment in the observing system's capabilities. Such requirements may be independent of the product-specific requirement for using the observing system. For example, weather warnings require high refresh rates for observations, but outlooks are less sensitive to refresh rate. If the requirement is stated in a way that only supports the more infrequent refresh rate for the outlook, the observing capability that meets the need for weather warning will over-estimate the value of the capability to support warnings. On the other hand, if the requirement is stated to support weather warnings, it will over-estimate its value to support outlooks. The warning SME may have low confidence in a scaler performance score because it could vary minute by minute, while the outlook SME may have much higher confidence and less sensitivity to its variability over the span of weeks or months.

The departure of the user's satisfaction score from the generalized agency requirement satisfaction score (RSS) is a contributor of uncertainty that can also be accounted for. To treat the uncertainty arising from the departure of the data source performance score from a generalized RSS, we don't apply a method similar to that above because there is no concentration parameter estimated for the RSS. Instead, we add a penalty or reward parameter to the known variance of the data source satisfaction score by recomputing the concentration parameter as if the mode of the data source shifted to half the distance between the original mode of the data source and the mode represented by the RSS.

- We estimate the adjusted return concentration parameter (κ_{padj}) from the difference between the RSS mode (m_{RSS}), the return mode (m_p), and the returned concentration parameter (κ_p).

$$\gamma_0 = \frac{ln\left(\frac{log_2(\kappa_p) - 1}{6}\right)}{ln(m_p - 1)}$$

$$m_{padj} = \frac{m_{RSS} + m_p}{2}$$

$$\kappa_{padj} = 2^{1+6(m_{padj}-1)^{\gamma_0}}$$

- If $m_{RSS} > m_p$, then $\kappa_{padj} > \kappa_{padj}$ and uncertainty will be reduced from that reported by the SME.

Measuring Risk: Bayesian Credible Interval

We can treat our model with prior (status quo) and posterior (swing scores) using Bayesian approaches. A Bayesian credible interval is a range within a posterior probability distribution that contains an uncertain parameter with a specific, high probability (e.g., 90%). It directly quantifies the probability that the true parameter lies within the interval, based on the SME's assessed posterior and prior performance conditions. We next compute the Bayesian credible interval so that we can make statements like: *"We are 90% confident that the parent product's performance sits between 62 and 84 when data source X becomes unfunded."*

If we want a 90% confidence interval, we use the Percent Point Function (PPF) that is available in python and in Excel. This chops off the extreme tails of our uncertainty cutting of the lowest 5% (p_{lower}) and highest 5% (p_{higher}). For any confidence level C (where C is a decimal like 0.90), the percentiles are:

$$p_{lower} = \frac{1 - C}{2}$$

$$p_{upper} = 1 - \frac{1 - C}{2}$$

We then use numerical methods to compute the interval. There is no closed equation that we can employ. Python and Excel modules that compute a Bayesian Credible Interval are given in Appendix 4.

Measuring Risk: Violin Plots

To visually summarize probability distribution changes under observing system portfolio trades side-by-side, we have chosen violin plots. These plots are amenable to the bounded range of the beta distribution (0 to 1), and they align the widest part of the "violin" with the mode.

Mission outcome nodes in the mission hierarchy portion of the value tree will use aggregated modes (performance) and variances (uncertainty). One example is the mission outcome node called "Tropical Weather Warnings", which depends on warnings from the National Hurricane Center for the Atlantic theater and from the Joint Typhoon Warning Center for the Pacific theater. Our objective for using plots is to visually demonstrate how changes in performance and uncertainty result from various scenarios in the NOAA value tree. For a mission outcome node like "Tropical Weather Warnings," leadership doesn't just need to know the average performance; they need to intuitively grasp the confidence and the tail risks (e.g., the probability of a catastrophic warning failure due to degraded satellite data).

Violin plots appear to be a more intuitive way of interpreting these changes in mission outcome nodes. Their interpretation is simple. The widest part of the violin moves up or down the Y-axis across scenarios to depict changes in performance. Vertical stretching or flattening depicts changes in uncertainty. To highlight the mode, our presentation of these plots overlays a horizontal line across the widest part of the bulge. Since aggregated beta distributions are bounded between 0 and 1, our presentation also locks the Y-axis of the plots to exactly [0, 1] (or 0% to 100%). This gives leadership a constant frame of reference for how close to "perfect" or "failed" the mission outcome is for each scenario.

An example of a violin plot for a scenario involving defunding a data source is given in Figure 6. A python module to compute a violin plot is given in Appendix 4.

Measuring Risk: Confidence and Precision

Our intuition in designing the above heuristic procedure is to define changes propagated in the NOAA value tree by various scenarios. We have chosen two such measures of change. The first is a measure of information gain or loss, which measures confidence. The second is a measure of precision. Precision defines how tightly packed the probability mass is. In the NOAA value tree, we can measure the confidence and precision information contributed to each mission node by its children via a scenario. A scenario may be cancelling an observing program or improving one. Our procedure is simple.

We use the below function of the shape parameters to compute the entropy of a beta distribution.

$$h(X) = ln\big(B(\alpha,\beta)\big) - (\alpha-1)\psi(\alpha) - (\beta-1)\psi(\beta) + (\alpha+\beta-2)\psi(\alpha+\beta)$$

Where $ln\big(B(\alpha,\beta)\big)$ is the natural logarithm of the Beta function. $\psi(\alpha)$ and $\psi(\beta)$ are Digamma functions (the derivative of the logarithm of the Gamma function) of α and β. The Digamma functions measure how sensitive the distribution is to changes in α and β. The more negative the entropy, the more information we have, and the more certain we are about the mission outcome.

Because these formulas use the natural logarithm (ln), the raw output is in units called “nats”. Because we want to measure information in bits (base 2), we simply divide the final result by $ln(2)$. We use a python module to compute the natural logarithm of the Beta function and the Digamma functions given in Appendix 4.

Our scenario risk analysis procedure involves nine steps.

1. *Define the entropy of the mission outcome expected at its highest performing and most certain state.* If we are going to evaluate an improvement to an observing system, then this is the state with the observing system improved. If we are going to evaluate an observing program cancellation, then this is the status quo state before the cancellation.

2. *Define the entropy of the mission outcome at its lowest performing and most uncertain state from the scenario.* This is the opposite state of the highest state.

3. *Compute the Information Gain for the scenario (Bits)*

$$Information\ Gain\ (scenario) = h(lowest) - h(highest)$$

   Because Beta entropy is negative, subtracting the more-negative lowest state entropy from the less negative highest state entropy results in a positive number of bits gained.

4. *Normalize to a percentage of information gain from the absolute lowest state, or maximum possible entropy, where $h = 0$.* To do this, we need to define the maximum possible entropy. Maximum entropy represents a state of complete, maximum uncertainty. If we know absolutely nothing about a system, every possible outcome is equally likely. This is best described with a uniform distribution where $\alpha = 1, \beta = 1$. The maximum entropy for a scale of 0 to 1 is exactly zero. For a continuous uniform distribution bounded between an initially uncertain state, a, and a final more certain state, b, the differential entropy $h(X)$ or information gain (IG) is calculated as:

$$h(X) = -\int p(x) log_2\big(p(x)\big) dx$$

   where X is the beta distribution that is a function of α, β, and κ.

$$IG\ =\ h(\alpha_a, \beta_a, \kappa_a) - h(\alpha_b, \beta_b, \kappa_a)$$

   Since our value tree uses a 0 to 1 scale we can compute the absolutely lowest uncertainty state:

$$h(X) = log_2(a - b) = log_2(1 - 0) = log_2(1) = 0 \text{ bits}$$

   The normalized information gain is then:

$$IG_{normalized} = \frac{h(\alpha_a, \beta_a, \kappa_a) - h(\alpha_b, \beta_b, \kappa_a)}{h(\alpha_a, \beta_a, \kappa_a) - 0}$$

Python functions that will compute entropy, beta_entropy_bits.py, are given in Appendix 4.

5. *Communicate the result of the Information Gain.* This metric only evaluates confidence, not performance.
    a. Example: "In our fully funded baseline, our systemic uncertainty is extremely low (highly negative entropy). If we drop to the constrained budget, our uncertainty expands. We are not just forecasting a lower performance score; our models are actually 5% less confident in what the outcome will be."
    b. Example: "In our fully funded baseline, our systemic uncertainty is moderately high. If we invest in improving this observing program, our models are 5% more confident in their outcomes."

6. *Compute the Kullback-Leibler (KL_scenario) Divergence of the scenario.* It too is computed with the natural logarithm of the Beta function and the Digamma function. It has four terms:

$$D_{KL} = (P \parallel Q) =$$
$$ln\left(\frac{B(\alpha_q, \beta_q)}{B(\alpha_p, \beta_p)}\right) +$$
$$(\alpha_p - \alpha_q)\psi(\alpha_p) +$$
$$(\beta_p - \beta_q)\psi(\beta_p) +$$
$$(\alpha_q - \alpha_p + \beta_q - \beta_p)\psi(\alpha_p + \beta_p)$$

The best-case scenario state is defined by α_p and β_p and the degraded state is defined by α_q and β_q. A python implementation of the KL divergence function is given in Appendix 4.

$D_{KL} = (P \parallel Q)$ is measuring how much the degraded state Q differs from the best-case state P, considering both performance (mode) and uncertainty (spread). The first term is a global shape/normalization adjustment. It describes how different the overall shapes of the two distributions are. This term grows larger if the shapes are more different. The second term describes how much the weight of high-performance outcome has shifted between the two states. The third term describes how much the weight of the low performance outcome has shifted between the two states. Finally, the fourth term describes the overall concentration between the two states, or how much the uncertainty of the two states has changed. The KL divergence is larger for large shape differences, more shifting between the states, and a larger shift change in uncertainty.

7. *Compute the Kullback-Leibler (KL_total_loss) Divergence of a total loss*. For this value, use the best-case scenario state defined by α_p and β_p and the maximum degraded state defined by $\alpha_q = 1$ and $\beta_q = 1$.

8. *Normalize to a "Percentage of Uncertainty Resolved".* This is simply the ratio:

$$\text{Percentage of Uncertainty Resolved} = \frac{KL\_scenario}{KL\_total\_loss} x100\%$$

9. *Communicate the result of Percentage of Uncertainty Resolved.* This measures total improvement in accuracy. It too ignores performance. Example: "Adding a microwave sounder to the satellite architecture resolves 5% of our systemic uncertainty for Tropical Weather Warnings."

An Application of the Uncertainty Estimation Procedure

To demonstrate the decision value of the proposed procedure, we evaluate its application to an observing system portfolio investment decision: improving a single observing program to enhance the tropical cyclone warning mission outcome. This outcome is one of more than 175 outcomes addressed by NOAA across mission service areas including long-term environmental monitoring, a weather-ready nation, healthy oceans, and resilient coastal communities and economies.

The decision is to consider an annualized modest budget increase of $300M in NOAA's observing system portfolio budget that manages hundreds of observing programs total more than $2B annually. The focus of the investment will be to improve either a radar network or a satellite-based microwave sounder. To demonstrate support for the decision, we apply the procedure to a collection of four SME surveys that trace the relevance and performance of a portion of NOAA's observing system portfolio through products and services to support tropical cyclone warnings. All SME's are assumed to be self-identified as with "Advanced Experience".

The National Weather Service is responsible for managing resources necessary to support the Tropical Cyclone (TC) mission area. TC Warnings originate from two business units: National Hurricane Center (NHC), which is responsible for the Atlantic and Eastern Pacific theaters, and the Joint Typhoon Warning Center (JTWC), which is responsible for the remainder of the Pacific theater. Resources supporting TC warnings for these theaters differ greatly and this is reflected in the SME surveys below in Table 3.

Surveys represent elicitations from four SMEs: NWS/OCOO, which manages NWS Operations; NWS/NHC, which manages the NHC; DoD/JTWC, which manages the JTWC jointly with NWS; and NWS/NCEP, which manages numerical weather prediction models. The NWS/OCOO SME identified which services are relevant to the mission area outcome. NHC and JTWC SMEs identify which products and observing systems are necessary to support their agency's TC Weather Warnings. Because an intermediate product for a satellite-based hurricane model was identified in the surveys for the TC Weather Warnings, we included it in the value network. In addition to listing inputs, these surveys include elicitation of status quo scores, swing scores, data source performance scores, and we introduce confidence scores.

Survey 1							
Mission Outcome Name	Business Unit	Mission Outcome ID					
Tropical Cyclone Weather Warnings	NWS/OCOO	TCWxWrngs					
Key Product Name	Business Unit	Key Product ID	Key Product Weight				
National Hurricane Center Hurricane Warning	NWS/NHC	NHCTCWxWrng	60				
Joint Typhoon Warning Center Typhoon Warning	NWS/JTWC	JTWCTCWxWrng	40				
Survey 2							
Product Name	Business Unit	Product ID	Status Quo Score	Confidence Score	Kappa		
Joint Typhoon Warning Center Typhoon Warning	DOD/JTWC	JTWCTCWxWrng	60	4	32		
Product Data Source Name	Business Unit	Product Data Source ID	Swing Score	kappa	Performance Score	Confidence Score	Kappa
Satellite-Based Tropical Cyclone Model	NWS/NCEP	TCModel	40	Splined from SQ Score	60	3	16
Satellite - Himawari Advanced Himawari Imager	NESDIS	Himawari(AHI)	40	Splined from SQ Score	70	4	32
Satellite - Microwave Sounder	NESDIS	JPSS(ATMS)	40	Splined from SQ Score	70	4	32
Survey 3							
Product Name	Business Unit	Product ID	Status Quo Score	Confidence Score	Kappa		
National Hurricane Center Hurricane Warning	NWS/NHC	NHCTCWxWrng	80	3	16		
Product Data Source Name	Business Unit	Product Data Source ID	Swing Score	Kappa	Performance Score	Confidence Score	Kappa
Model - Satellite-Based Tropical Cyclone Model	NWS/NCEP	TCModel	70	Splined from SQ Score	60	3	16
Satellite - Microwave Sounder	NESDIS	JPSS(ATMS)	50	Splined from SQ Score	70	4	32
Satellite - GOES Advanced Baseline Imager	NESDIS	GOES(ABI)	74	Splined from SQ Score	80	4	32
Buoy - Coastal Weather Buoys	NWS/NDBC	CWB	76	Splined from SQ Score	80	4	32
Aircraft - Hurricane Hunters	NASA	WP-3D	70	Splined from SQ Score	95	5	64
Radar - Coastal NEXRAD	NWS/OBS	NEXRAD	60	Splined from SQ Score	60	2	8
Survey 4							
Product Name	Business Unit	Product ID	Status Quo Score	Confidence Score	Kappa		
Model - Satellite-Based Tropical Cyclone Model	NWS/NCEP	TCModel	60	4	32		
Product Data Source Name	Business Unit	Product Data Source ID	Swing Score	Kappa	Performance Score	Confidence Score	Kappa
Radar - Coastal NEXRAD	NWS/OBS	NEXRAD	58	Splined from SQ Score	60	2	8
Satellite - Microwave Sounder	NESDIS	JPSS(ATMS)	25	Splined from SQ Score	70	4	32
Satellite - GOES Advanced Baseline Imager	NESDIS	GOES(ABI)	50	Splined from SQ Score	80	4	32
Satellite Himawari Advanced Himawari Imager	NESDIS	Himawari(AHI)	50	Splined from SQ Score	70	4	32

Table 3. Data collected from four SME surveys from NWS/OCOO, DOD/JTWC, NWS/NHC, and NWS/NCEP.

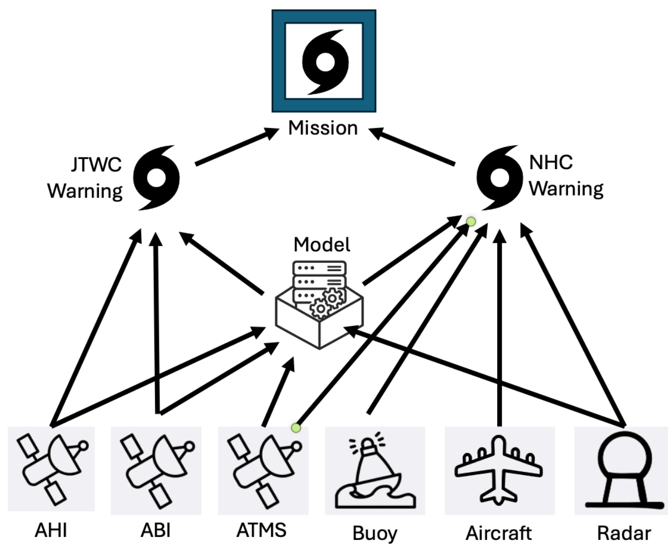

Figure 5. Graph comprising a mission outcome (top), key products to the outcome, an intermediate product, and a portfolio of observing programs.

The primary graph calculation tools for this procured called Portfolio Analysis Machine (PALMA™) are licensed by the MITRE Corporation and made available to NOAA. The uncertainty analysis is post-processed from PALMA output. Outputs without confidence information include observing system portfolio ranking, measures of observing system impact to the outcome, and a budget-vs-benefit table for considering expanded and constrained budgets relevant to the outcome. Additional measures made possible with the above procedure include risk measurements for the relative information gain and relative change in precision expected by either investing $300M in the radar network or in the satellite-based microwave sounder.

Results

Type	Mission Outcome	Key Product	Key Product	Intermediate Product
Name	Tropical Cyclone Weather Warnings	National Hurricane Center Hurricane Warning	Joint Typhoon Warning Center Typhoon Warning	Model - Satellite-Based Tropical Cyclone Model
ID	TCWxWrngs	TCWxWrng NHC	TCWxWrng JTWC	TCModel NCEP
STATUS QUO				
Baseline Performance Score	72.00	80.00	60.00	60.00
Portfolio % Impact				
Himawari(AHI)	2.8%	1.3%	5.6%	17.6%
JPSS(ATMS)	48.6%	45.0%	54.8%	61.4%
GOES(ABI)	19.8%	9.2%	38.4%	17.5%
CWB	3.3%	5.2%	0	0
WP-3D	8.3%	13.0%	0	0
NEXRAD	17.2%	26.4%	1.2%	3.5%
CHOOSE ATMS+ FOR INVESTMENT				
New Performance Score	84.69	88.04	79.68	80.63
% Improvement	17.6%	10.0%	32.8%	34.4%
CHOOSE NEXRAD+ FOR INVESTMENT				
New Performance Score	85.91	91.27	77.87	81.47
% Improvement	19.3%	14.1%	29.8%	35.8%

Table 4. Performance Based Assessment.

The “Baseline” portion of Table 4 shows the impact of the portfolio elements. Outcome-wide and among all surveyed products, the microwave sounder has the greatest impact followed by the GOES program and the NEXRAD radar program. Products show modest sensitivity to Partner Geostationary Satellite, Buoys, and Aircraft.

Observing Program scenarios in Table 4 show dramatic improvements from an annualized investment in either the microwave sounder program (17.6% Outcome-wide) or in the coastal radar program (19.3% Outcome-wide). Improvements are similar in size among the key products and the intermediate model product.

From Table 4, AHI has a measurable impact on NHC TC Warnings and NEXRAD has a measurable impact on JTWC Warnings even though they were not explicitly surveyed as their data sources. This impact flows from the product’s dependence on the model which does depend on AHI and NEXRAD.

Annualized JPSS	$1.0B per year	1000 $M
Annualized JPSS+ (30% more)	$1.3B per year	1300 $M
Annualized NEXRAD	$0.3B per year	300 $M
Annualized NEXRAD+ (100% more)	$0.6B per year	600 $M
Annualized NASA Hurricane Hunters (50%)	$0.01B per year	10 $M
Annualized CWB	$0.05B per year	5 $M
Annualized Himawari MOA	$0.001B per year	1 $M
Annualized GOES-R	$0.4B per year	480 $M

Table 5. Notional Estimates of Annualized Portfolio Cost used in Efficient Frontier Calculation.

Portfolio Chosen (all other options unchanged)	Cost of Total Portfolio	% Change in Performance

JPSS(ATMS), NEXRAD	$1796M	Baseline
JPSS(ATMS)+, NEXRAD	$2096M	+17.6%
JPSS(ATMS), NEXRAD+	$2096M	+19.3%

Table 6. Return on Investment on Mission Outcome.

To compute Normalized Information Gain and KL Divergence, we first estimate the Mission Outcome's change in kappa from its PALMA computed change in mode from its initial status quo state to its final states (investment in either a microwave sounder or a radar). To estimate kappa, we use the spline developed earlier. From the changes in mode and concentration parameter, we then compute the Outcome's beta distribution shape parameters. Those parameters then serve as inputs to the Normalized Information Gain and Normalized KL Divergency functions.

Observing System Investment Scenario	System Type	% Information Gain on Outcome	% Precision Improvement on Outcome
MW Sounder +	Satellite	33%	33%
Radar +	Radar	5%	29%

Table 7. Risk Assessment

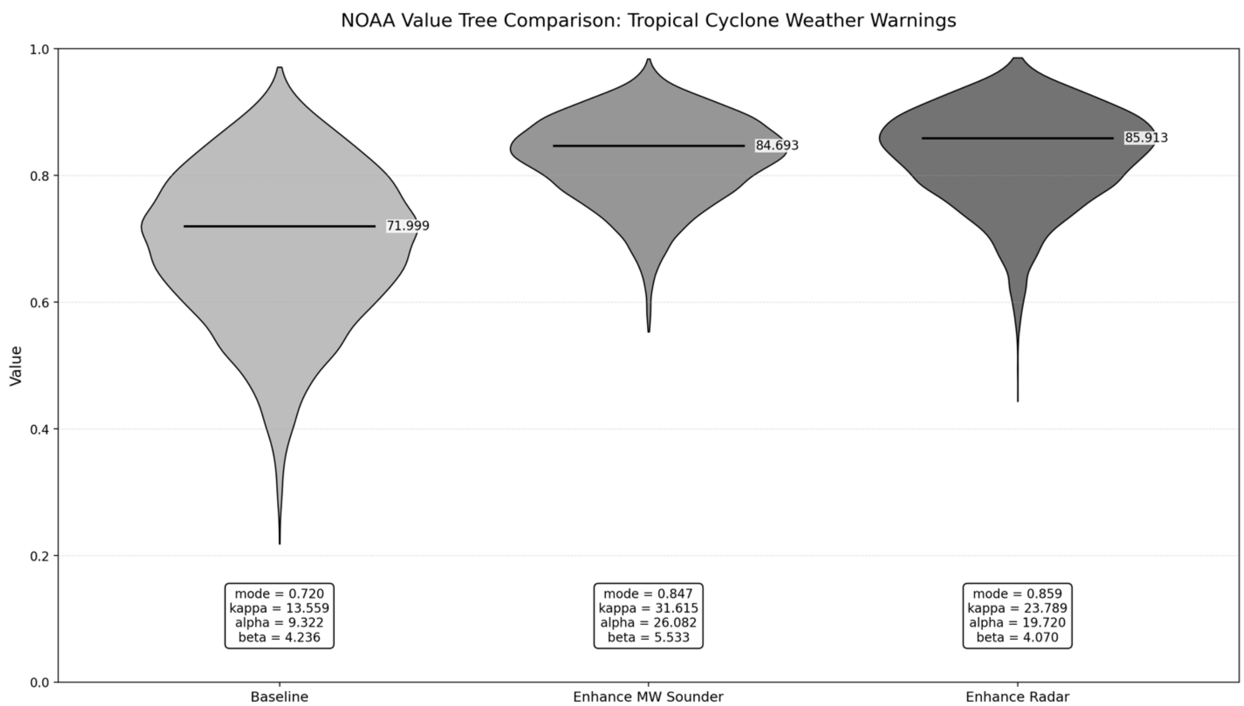

Figure 6. Comparison of violin plots of baseline (left) mission outcome, and the outcome for scenarios that either invest in a Microwave Sounder (center) or in a Radar (right).

DISCUSSION

In the notional scenario above, NEXRAD has a much higher refresh rate than JPSS(ATMS) but limited geo-coverage. Some uncertainties introduced by NEXRAD might include data assimilation for the hurricane model, uncertainty in retrieved values for low level precipitation rate and near surface winds, nearness to coast contributing to beam blockages, and varying resolution from beam broadening at long distance and limited detection distance. A radar investment may be designed to resolve some of these uncertainties by providing more range, detection at lower levels, and mitigating surface interference, or by supporting radar in the Pacific Theater.

JPSS(ATMS) has much more geo-coverage than NEXRAD but less refresh. Some uncertainties introduced by JPSS(ATMS) includes data assimilation for the hurricane model that handles both theaters may be optimized for the Atlantic theater than for the Pacific theater, uncertainty introduced by regimes (e.g., El Niño /La Niña years), uncertainty in the temperature and moisture profile from limited band retrievals, and cloud coverage obscuring the ocean surface temperatures ahead of the storm. A microwave sounder investment may be designed to resolve some of the uncertainties by providing a higher refresh rate, or by increasing band resolution.

These factors are different than performance attributes expected of the observing systems to meet product requirements such as geographic coverage, horizontal and vertical resolution, measurement accuracy, data latency, and refresh rates. In this comparison scenario, updating the Radar with the increased budget has a slight advantage over updating the microwave sounder with increases in performance, information gain (reducing uncertainty), and improving the precision of TC Weather Warnings overall.

Surveys of observing system dependence can also better characterize uncertainties in scoring both observing system options by increasing the granularity of the survey to accommodate regime dependence, considering the frequency of occurrence of tropical cyclones, defining performance for the observation types retrieved from each observing system (e.g., radar retrieves rain rate, wind direction and speed for various levels), basing performance scores on more objective measures such as forecast validation statistics. Enhancements to surveys can better capture attributes of the mission outcome that could be most sensitive to improvements in observing system performance, confidence, and precision.

REFERENCES

- **NOSIA:**
  1. Helms, D., M. Austin, L. Mccullouch, RC. Reining, A. Pratt, R. Mairs, et al. (2016). NOAA Observing System Integrated Analysis (NOSIA-II) Methodology Report. NOAA Tech. Rep. NESDIS 147, 86 pp., https://doi.org/10.7289/V52V2D1H
  2. Cantrell, L. (2026) IPSEA [Internal White Paper, NOAA/NESDIS that develops the IPSEA function]. https://docs.google.com/document/d/1_CS-tDQ-ktN8CC37i7Qkz00obIryccN6/
- **Beta Distribution, Beta Function, and Digamma Functions**:
  3. Johnson, Norman L.; Kotz, Samuel; Balakrishnan, N. (1995). "Chapter 25: Beta Distributions". Continuous Univariate Distributions Vol. 2 (2nd ed.). Wiley. ISBN 978-0-471-58494-0.
  4. Abramowitz, M.; Stegun, I. A., eds. (1972). Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables (reprint 10th ed.). New York: Dover.
- **Stineman Spline**:
  5. Stineman, R.W. (1980) A Consistently Well-Behaved Method for Interpolation. Creative Computing, 6, 54-57.

6. Cantrell, L. (2025) The Stineman and MITRE Splines useful in NOSIA. [Internal White Paper, NOAA/NESDIS that develops a focused version of the Stineman Spline for NOSIA]

- **Entropy, Information Gain, and KL Divergence**:
  7. Verdugo Lazo, A. C. G.; Rathie, P. N. (1978). "On the entropy of continuous probability distributions". IEEE Trans. Inf. Theory. 24 (1): 120–122. doi:10.1109/TIT.1978.1055832
- **Eliciting Uncertainty**
  8. A. O'Hagan, C. E. Buck, A. Daneshkhah, J. R. Eiser, P. H. Garthwaite, D. J. Jenkinson, J. E. Oakley, and Tim Rakow (2006). Uncertain Judgements: Eliciting Experts' Probabilities. Wiley. 323 pp.
- **Wikipedia Topics**
  Beta Distribution
  Entropy
  Information Gain
  KL Divergence

APPENDIX 1: Elicitation

An Elicitation Strategy

NOSIA and EOA use expert elicitation to construct their respective value trees.

According to The National Academies of Sciences, Engineering, and Medicine: "an expert is defined not merely by years of experience, but by the structure of their knowledge and their performance capabilities within a specific domain." [https://www.nationalacademies.org/read/9853/chapter/5]

Elicitation Strategy

- Supports Observing System Portfolio Relevant Decision-Making
- Suitable for Wide Range of Subject Matter Experts (SME) Within and Across Partner Agencies
- Repeatable Elicitation Process
- Consistent Metrics
- Flexible to Address Various Domains
- Adequate Empirical Sampling
- Collections Planned and Accommodated
- Collections Amenable to Data Stewardship and Interrogation, and Knowledge Graphs
- Supports Rational Modeling Approaches
- Recognizes Individual SME Subjectivity
- Measures Uncertainty and Variation
- Transparency
- Verifiability

Elicitation Technique Considerations

- Preplan Elicitation
    - What to survey [products, mission outcomes, agency requirements]
    - When to survey [schedule planning with focus on decision input deliverable]
    - Who to survey [SME selection]
    - How to survey [Elicitation Protocol]
    - Where to survey [In person, virtual, chatbot]
    - Accommodations: feedback, training, and flexible schedules
- Unambiguous questions
    - Generally applicable for various agencies, domains, data types
        - Spans federal agencies [NOAA, USGS, NASA, CI's]
        - Spans geophysical domains [GCMD Topics]
        - Spans missions, outcomes, products, observing systems, databases, research activities, other data sources (e.g., grant programs, laboratory / fieldwork)
    - Clear definitions; glossary for reference and clarification
- Simple quantities of elicited parameters
    - Integers precision 0 to 100 with categorical help

    - Values between 0 and 1
    - Limited use of special scales (e.g., [1, 6])
    - Limited use of simple categories (categories <= 7 members)
- Consistent scales, categories and units of elicited data
    - Performance Satisfaction Scale [0, 100]
    - Categories of Expertise [Novice, Intermediate, Sub-Expert, Expert]
    - Categories of Relevance [Very High, High, Moderate, Low]
    - Requirement Gap Assessment Scale [0, 1]
    - Confidence Scale [1, 6]
    - Correlation Scale [0, 1], [+, -]
    - Redundancy Scale [0, 1]
    - Requirement Attribute Units {varies with attribute and may be from a look up table, an integer, or a continuous variable with appropriate precision}
- Appropriate level of node granularity
    - Product granularity
    - Requirement granularity
    - Data Source granularity
    - User granularity
- Expert selection
    - Focus on necessary experience and skills
    - SME self-assigns expertise category
    - Sufficient SME willingness and availability to participate
- Elicitation training
    - Moderator Training to proctor the elicitation
    - SME Training to understand and participate in the elicitation
    - Training to minimize heuristic biases
- Elicitation aggregation
    - Use individual elicitation first (product, requirement, mission outcome)
    - Aggregate from consensus if multiple SMEs for the same node (e.g., Delphi process)
    - Aggregate using consistently applied rollup rules if grouping heterogeneous or homogeneous node groups
- Models of elicited data
    - Node Types in value tree
    - Elicited Parameter types
    - Computed Parameter types (clearly distinguished from elicited parameters)
    - Edge Function types that use elicited and computed parameters
        - Weighted averages, IPSEA-P, PWSS
        - Maximum or minimum of set
        - On/Off
        - Copy or Pass-through (aka Ghost)
        - Special trade groups [together, group1]

The Elicitation Protocol

1. **Modes** (Status Quo Score, Swing Score, and Overall Satisfaction Score).

Performance (Satisfaction) Scale		
100	**Ideal**	Meets all requirements and exceeds some
90	**Fully Satisfied**	Meets all requirements
80	**Good**	Meets all major requirements, with minor limitations
60	**Fair**	Meets most major requirements, with significant limitations
40	**Poor**	Fails to meet many major requirements, but provides some value
20	**Very Poor**	Fails to meet most major requirements, but provides minor value
1	**No Capability**	Provides no value

Table A1.1. Translating Product Performance into Numerical Values.

Status Quo Score:
"On a scale of 1 to 100, what is the most likely performance score for this product right now?"
Math parameter: mode (m_0).

Swing Score:
"On a scale of 1 to 100, what is the most likely performance score for this product without this data source?"
Math parameter: mode (m'_i).

Overall Satisfaction Score:
"On a scale of 1 to 100, what is the most likely performance score for this data source in meeting your requirements for this product?"
Math parameter: mode (m_i).

2. **Confidence Levels:**

SME Confidence Rating	**Meaning**	**Concentration Parameter (κ)**
Min **κ**	Maximum Uncertainty	$2^1 = 2$
1	Very Uncertain (Wide-Spread)	$2^2 = 4$
2	Somewhat Uncertain	$2^3 = 8$
3	Moderately Confident	$2^4 = 16$
4	Highly Confident	$2^5 = 32$
5	Absolutely Certain (Tight Peak)	$2^6 = 64$
Max **κ**	Minimum Uncertainty	$2^7 = 128$

Table A1.2. Confidence rating vs. Concentration parameter

“On a scale of 1 to 5, how confident are you in that exact score? (1 being very uncertain, 5 being absolutely certain)."

Elicit this for Status Quo Score and Overall Satisfaction Score. Do not elicit for Swing Score. Math parameter: Map this 1-5 rating to a hidden "Concentration Parameter" (κ). The higher the κ, the tighter and taller the probability curve. If uncertainty metrics are available, such as from Observing System Experiments, we can convert those metrics to concentration parameters. If the SME is unable to provide a confidence rating, we can default to $\kappa = 16$, which comes from a “moderately confident” SME Confidence Rating of 3.

3. **Confidence Sources:**

“What process do you employ to manage your confidence in this product status quo score?”, or *“What do you think contributes most to your confidence assessment?”*

Examples of process might include data assimilation, algorithm design, QC, temporal smoothing, filtering, operational expertise, etc. We simply capture this information as a SME comment on their chosen confidence index.

4. **Error Correlation among group data sources:**

“To what extent do errors in these data sources tend to occur together?”

5. **Performance Correlation among group data sources:**

“How well related is the performance of your data sources on a scale of 0 to 1: 0=independent, 0.25=weekly related, 0.50=moderately related, 0.75=strongly related, and 1.00=always move together.?”

We then follow up with:

“Are these data source performance scores positively correlated or anti-correlated?”

6. **Redundancy:**

“On a scale of 0 to 1, how would you characterize the redundancy of this group of data sources with 0 representing no redundancy and 1 representing fully duplicative?”.

7. **Conditional Variance:**

“Other than data source availability, under what conditions do these data sources most strongly affect product performance?”

APPENDIX 2: Derivation of the $\kappa - m$ spline.

Here, we propose a spline function based on the following six inputs comprising the maximum and minimum values for mode m and concentration parameter κ and for the spline's anchor point:

$m_{min} = 0$	m	$m_{max} = 1$
$\kappa_{min} = 2^1$	κ	$\kappa_{max} = 2^7$

Table A2.1 K—M Spline inputs.

A well-behaved spline function will evaluate the mode monotonically from 0 to 1 while the concentration parameter varies logarithmically from 2^1 to 2^7.

For arbitrary linear mode endpoints:

$$t = \frac{m - m_{min}}{m_{max} - m_{min}}$$

For arbitrary log-linear concentration parameter endpoints:

$$\log_2(u) = \frac{\log_2(\kappa) - \log_2(\kappa_{min})}{\log_2(\kappa_{max}) - \log_2(\kappa_{min})}$$

The family of splines we seek:

$$\log_2(u) = t^\gamma$$

$$t^\gamma = \frac{\log_2(\kappa) - \log_2(\kappa_{min})}{\log_2(\kappa_{max}) - \log_2(\kappa_{min})}$$

$$\log_2(\kappa) - \log_2(\kappa_{min}) = \left(\log_2(\kappa_{max}) - \log_2(\kappa_{min})\right)t^\gamma$$

$$\log_2(\kappa) = \log_2(\kappa_{min}) + \left(\log_2(\kappa_{max}) - \log_2(\kappa_{min})\right)t^\gamma$$

$$\log_2(\kappa) = \log_2(\kappa_{min}) + \left(t^\gamma \log_2\left(\frac{\kappa_{max}}{\kappa_{min}}\right)\right)$$

$$\log_2(\kappa) = \log_2(\kappa_{min}) + \left(\log_2\left(\frac{\kappa_{max}}{\kappa_{min}}\right)^{t^\gamma}\right)$$

$$\log_2(\kappa) = \log_2\left(\kappa_{min}\left(\frac{\kappa_{max}}{\kappa_{min}}\right)^{t^\gamma}\right)$$

$$\kappa = \kappa_{min}\left(\frac{\kappa_{max}}{\kappa_{min}}\right)^{t^\gamma}$$

$$\kappa = \kappa_{min}\left(\frac{\kappa_{max}}{\kappa_{min}}\right)^{t^\gamma}$$

For our case: $\kappa_{min} = 2$, $\kappa_{max} = 128$, $m_{min} = 0$, $m_{max} = 1$:

$$\kappa = \kappa_{min}\left(\frac{\kappa_{max}}{\kappa_{min}}\right)^{t^{\gamma}} = 2\left(\frac{128}{2}\right)^{t^{\gamma}} = 2(64)^{t^{\gamma}}$$
$$t = \frac{m - m_{min}}{m_{max} - m_{min}} = \frac{m - 0}{1 - 0} = m$$
$$\kappa = 2(64)^{m^{\gamma}} = 2(2^6)^{m^{\gamma}} = 2^1 \cdot 2^{6m^{\gamma}} = 2^{1+6m^{\gamma}}$$
$$z = f(m) = 1 + 6m^{\gamma}$$
$$\kappa = 2^z = 2^{1+6m^{\gamma}}$$
$$log_2(\kappa) = log_2(2^z) = z = 1 + 6m^{\gamma}$$
$$\frac{log_2(\kappa) - 1}{6} = m^{\gamma}$$
$$ln(m^{\gamma}) = ln\left(\frac{log_2(\kappa) - 1}{6}\right)$$
$$\gamma ln(m) = ln\left(\frac{log_2(\kappa) - 1}{6}\right)$$
$$\gamma = \frac{ln\left(\frac{log_2(\kappa) - 1}{6}\right)}{ln(m)}$$

This is a useful because it bounds κ between 2 (lowest confidence) and 128 (highest confidence), is monotone with increasing m between 0 and 1, is curved, and is controlled by only one spline shape parameter, γ.

$$\gamma_0 = \frac{ln\left(\frac{log_2(\kappa_0) - 1}{6}\right)}{ln(m_0)}$$

Once the parameter γ is defined for these endpoints and anchor point (m_0, κ_0), we can compute m_{new} using the NOSIA graph edge functions, then apply the spline to solve for a new κ_{new} given the new m_{new}.

$$\kappa_{new} = 2^{1+6{m_{new}}^{\gamma_0}}$$

Variance is proportional to m(m-1), which means the variance or uncertainty decreases as we naturally increase the mode at constant concentration parameter, but as performance improves, however, our spline formulation assumes uncertainty decreases faster than the intrinsic beta variance would suggest because we expected it to reach a maximum value as we approached maximum performance.

APPENDIX 3: Derivation of Aggregated Variance

Compute the aggregated mode (m_{agg}) value from the modes of the child products (m_i) using a weighted average.

$$m_{agg} = \frac{\sum w_i m_i}{\sum w_i}$$

Compute the aggregated variance ($\sigma^2_{agg\_a}$) value from the variance of the child products (σ_i^2) using a weighted average.

$$\sigma_i^2 = \frac{\alpha_i \beta_i}{(\alpha_i + \beta_i)^2(\alpha_i + \beta_i - 1)}$$

$$\sigma^2_{agg\_a} = \frac{\sum w_i \sigma_i^2}{\sum w_i}$$

Compute an additional aggregated variance value ($\sigma^2_{agg\_b}$) to represent uncertainties for the group arising from a dispersion of its child modes. The term S is a scaling factor that minimizes this variance contribution if the aggregated concentration parameter nears 2^1 and maximizes this contribution if the aggregated concentration parameter nears 2^7. It comes out of the gamma function developed in APPENDIX 2.

$$L_{agg} = \frac{\sum w_i log_2(\kappa_i)}{\sum w_i}$$

$$S = \frac{L_{agg} - 1}{6}$$

$$\kappa_{agg} = 2^{L_{agg}}$$

$$\sigma^2_{agg\_b} = S\left(\frac{\sum w_i \left(m_i - m_{agg}\right)^2}{\sum w_i}\right)$$

Compute the total variance of the aggregated node.

$$V = \sigma^2_{agg\_b} + \sigma^2_{agg\_b}$$

To derive the cubic function of κ, we start with the equations for concentration, mode, and variance.

$$\kappa = \alpha + \beta$$
$$m = \frac{\alpha - 1}{\alpha + \beta - 2}$$

$$V = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

Next, we define α and β in terms of m and κ.
Solve for α.

$$m = \frac{\alpha-1}{\alpha+\beta-2} = \frac{\alpha-1}{\kappa-2}$$
$$\alpha = m(\kappa-2)+1$$

Solve for β

$$\kappa = \alpha+\beta$$
$$\beta = \kappa-\alpha = \kappa-m(\kappa-2)-1$$
$$\beta = \kappa-m(\kappa-2)-1-1+1$$
$$\beta = (\kappa-2)-m(\kappa-2)+1$$
$$\beta = (\kappa-2)(1-m)+1$$

Substitute α and β into the variance denominator.

$$V = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$
$$V = \frac{\alpha\beta}{\kappa^2(\kappa+1)}$$

Substitute α and β into the variance numerator.

$$\alpha\beta = [m(\kappa-2)+1][(\kappa-2)(1-m)+1]$$
$$\alpha\beta = (m\kappa-2m+1)(\kappa-\kappa m-2+2m+1)$$
$$\alpha\beta = (m\kappa-2m+1)(\kappa-\kappa m-1+2m)$$
$$\alpha\beta = m\kappa^2-m^2\kappa^2-m\kappa+2m^2\kappa-2m\kappa+2m^2\kappa+2m-4m^2+\kappa-\kappa m-1+2m$$
$$\alpha\beta = m\kappa^2-m^2\kappa^2-m\kappa+2m^2\kappa+2m^2\kappa-2m\kappa+\kappa-\kappa m-4m^2+2m+2m-1$$
$$\alpha\beta = [m\kappa^2-m^2\kappa^2]+[-m\kappa+2m^2\kappa+2m^2\kappa-2m\kappa+\kappa-\kappa m]+[-4m^2+2m+2m]-1$$
$$\alpha\beta = [m\kappa^2-m^2\kappa^2]+[-m\kappa-2m\kappa-\kappa m+2m^2\kappa+2m^2\kappa+\kappa]+[-4m^2+2m+2m]-1$$
$$\alpha\beta = [m\kappa^2-m^2\kappa^2]+[-4m\kappa+4m^2\kappa+\kappa]+[4m-4m^2]-1$$
$$\alpha\beta = \kappa^2[m-m^2]+\kappa[-4m+4m^2+1]+[4m-4m^2]-1$$
$$\alpha\beta = \kappa^2[m-m^2]-\kappa[4m-4m^2]+\kappa+[4m-4m^2]-1$$
$$\alpha\beta = \kappa^2 m(1-m)-4\kappa m(1-m)+\kappa+4m(1-m)-1$$

Let $W = m(1-m)$

$$\alpha\beta = \kappa^2 m(1-m)-4\kappa m(1-m)+\kappa+4m(1-m)-1$$
$$\alpha\beta = \kappa^2 W-4\kappa W+\kappa+4W-1$$

Group terms by power of κ.

$$\alpha\beta = \kappa^2(W) + \kappa(-4W + 1) + (4W - 1)$$

Substitute numerator into variance equation.

$$V = \frac{\alpha\beta}{\kappa^2(\kappa + 1)}$$
$$V = \frac{\kappa^2(W) + \kappa(-4W + 1) + (4W - 1)}{\kappa^2(\kappa + 1)}$$
$$V = \frac{\kappa^2(W) + \kappa(-4W + 1) + (4W - 1)}{\kappa^3 + \kappa^2}$$

Rearrange into cubic polynomial.

$$V(\kappa^3 + \kappa^2) = \kappa^2(W) + \kappa(-4W + 1) + (4W - 1)$$
$$V\kappa^3 + V\kappa^2 - \kappa^2(W) - \kappa(-4W + 1) - (4W - 1) = 0$$
$$\kappa^3 V + \kappa^2(V - W) - \kappa(-4W + 1) - (4W - 1) = 0$$

Solve for κ_{agg} using numerical methods.

$${\kappa_{agg}}^3 V + {\kappa_{agg}}^2(V - W) - \kappa_{agg}(-4W + 1) - (4W - 1) = 0$$
$$W = m_{agg}(1 - m_{agg})$$

Finally, solve for the aggregated shape parameters α_{agg} and β_{agg}.

$$\alpha_{agg} = m_{agg}(\kappa_{agg} - 2) + 1$$
$$\beta_{agg} = \kappa_{agg} - \alpha_{agg}$$

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
APPENDIX 4_v6
Created on Fri, Apr 4, 2026

@author: Louis Cantrell, PhD
Profitable Weather, LLC
"""

import math
import numpy as np
from scipy.special import betaln, digamma

def compute_gamma(kappa_0, m_0):
    """
    Computes gamma for the kappa spline function
    """
    return math.log((math.log2(kappa_0) - 1) / 6) / math.log(m_0)

def compute_kappa_new(kappa_0, m_0, m_new):
    """
    Computes new kappa using a spline
    """
    gamma_0 = compute_gamma(kappa_0, m_0)
    return 2 ** (1 + 6 * (m_new ** gamma_0))

def compute_alpha(kappa, m):
    """
    Computes alpha
    """
    return m * (kappa - 2) + 1

def compute_beta(kappa, m):
    """
    computes beta
    """
    return (1 - m) * (kappa - 2) + 1

def compute_variance(kappa, m):
    """
    Computes variance
    """
    alpha = compute_alpha(kappa, m)
    beta = compute_beta(kappa, m)
    return (alpha * beta) / (kappa**2 * (kappa + 1))

def compute_params(kappa, m):
    """
    Computes alpha, beta, and variance
    """
    alpha = compute_alpha(kappa, m)
    beta = compute_beta(kappa, m)
    variance = (alpha * beta) / (kappa**2 * (kappa + 1))
    return alpha, beta, variance

def beta_entropy_bits(alpha, beta, kappa):
    """
    Computes the Entropy of the distribution in bits
    """
    return (
        math.lgamma(alpha) +
        math.lgamma(beta) -
        math.lgamma(kappa)
        - (alpha - 1) * digamma(alpha)
        - (beta - 1) * digamma(beta)
        + (kappa - 2) * digamma(kappa)
    ) / math.log(2)

def compute_beta_kl_divergence(alpha_p, beta_p, alpha_q, beta_q):
    """
```

```
    Computes the KL Divergence (Info. Gain) in bits
    from Beta distribution Q to Beta distribution P.
    Parameters
    ----------
    P = Baseline (truth)
    Q = Degraded (approximation)
    Returns
    -----
    KL Divergence value in bits.
    """
    # Calculate the log-beta difference
    # Note: betaln computes the natural log of the beta function
    log_beta_diff = betaln(alpha_q, beta_q) - betaln(alpha_p, beta_p)
    # Calculate the digamma terms
    digamma_p_sum = digamma(alpha_p + beta_p)
    term1 = (alpha_p - alpha_q) * digamma(alpha_p)
    term2 = (beta_p - beta_q) * digamma(beta_p)
    term3 = (alpha_q - alpha_p + beta_q - beta_p) * digamma_p_sum
    # Sum it all up for nats
    kl_nats = log_beta_diff + term1 + term2 + term3
    # Convert nats to bits
    kl_bits = kl_nats / np.log(2)
    # Prevent small floating point errors from showing up as negative zero
    return max(0.0, kl_bits)

def aggregate_betas_scaled(alpha_list, beta_list, weight_list):
    """
    Aggregates modes, concentration parameters, alpha, and beta from child modes.
    """
    alphas = np.array(alpha_list, dtype=float)
    betas = np.array(beta_list, dtype=float)
    weights = np.array(weight_list, dtype=float)
    # 1. Normalize weights
    w = weights / np.sum(weights)
    # 2. Compute individual child parameters
    modes = (alphas - 1) / (alphas + betas - 2)
    kappas = alphas + betas
    variances = (alphas * betas) / ((kappas**2) * (kappas + 1))
    # 3. Compute Aggregated Mode
    M_agg = np.sum(w * modes)
    # 4. Compute Penalty Scaling Factor (S) in log2 space
    L_agg = np.sum(w * np.log2(kappas))
    S = max(0.0, min(1.0, (L_agg - 1.0) / 6.0))
    # 5. Compute Aggregated Variance
    V_agg = np.sum(w * variances) + S * np.sum(w * (modes - M_agg)**2)
    # 6. Solve cubic for kappa_agg
    W = M_agg * (1 - M_agg)
    coeffs = [
        V_agg,
        V_agg - W,
        4 * W - 1,
        1 - 4 * W
    ]
    roots = np.roots(coeffs)
    valid_roots = [r.real for r in roots if np.isclose(r.imag, 0) and r.real > 2]
    if not valid_roots:
        raise ValueError("No valid real root found for kappa_agg > 2.")

    kappa_agg = valid_roots[0]
    # 7. Final aggregated shape parameters
    alpha_agg = M_agg * (kappa_agg - 2) + 1
    beta_agg = (1 - M_agg) * (kappa_agg - 2) + 1
    return (M_agg, V_agg, kappa_agg, alpha_agg, beta_agg, S)

def draw_violin_plots_x3(m,k,title):
    import numpy as np
    import matplotlib.pyplot as plt
    from scipy.stats import beta as beta_dist
    # Parameters
    plots = [
        {"title": "Baseline", "m": m[0], "kappa": k[0]},
```

```
    {"title": "Enhance MW Sounder", "m": m[1], "kappa": k[1]},
    {"title": "Enhance Radar", "m": m[2], "kappa": k[2]},
]
# Compute alpha and beta from mode m and concentration kappa
for p in plots:
    m = p["m"]
    k = p["kappa"]
    alpha = m * (k - 2.0) + 1.0
    beta = (1.0 - m) * (k - 2.0) + 1.0
    p["alpha"] = alpha
    p["beta"] = beta
# Generate samples for violin plots
rng = np.random.default_rng(42)
samples = [
    beta_dist.rvs(p["alpha"], p["beta"], size=6000, random_state=rng)
    for p in plots
]
# Create figure
fig, ax = plt.subplots(figsize=(14, 8))
parts = ax.violinplot(
    samples,
    positions=[1, 2, 3],
    widths=0.75,
    showmeans=False,
    showmedians=False,
    showextrema=False,
    vert=True,
)
# Grey shades only
grey_shades = ["#bdbdbd", "#969696", "#737373"]
for i, body in enumerate(parts["bodies"]):
    body.set_facecolor(grey_shades[i])
    body.set_edgecolor("black")
    body.set_linewidth(1.0)
    body.set_alpha(1.0)
# Axes and labels
ax.set_ylim(0.0, 1.0)
ax.set_xlim(0.4, 3.6)
ax.set_xticks([1, 2, 3])
ax.set_xticklabels([p["title"] for p in plots], fontsize=11)
ax.set_ylabel("Value", fontsize=12)
ax.set_title(
    "NOAA Value Tree Comparison: Tropical Cyclone Weather Warnings",
    fontsize=15,
    pad=18,
)
# Add mode lines and annotations
for i, p in enumerate(plots, start=1):
    m = p["m"]
    k = p["kappa"]
    a = p["alpha"]
    b = p["beta"]
    # Horizontal mode line
    ax.hlines(m, i - 0.26, i + 0.26, colors="black", linewidth=1.8)
    ax.text(
        i + 0.29,
        m,
        f"{m*100:.3f}",
        va="center",
        ha="left",
        fontsize=10,
        bbox=dict(facecolor="white", edgecolor="none", pad=0.4, alpha=0.85),
    )
    # Parameter box
    param_text = (
        f"mode = {m:.3f}\n"
        f"kappa = {k:.3f}\n"
        f"alpha = {a:.3f}\n"
        f"beta = {b:.3f}"
    )
    ax.text(
```

```
            i,
            0.06,
            param_text,
            ha="center",
            va="bottom",
            fontsize=10,
            bbox=dict(facecolor="white", edgecolor="black", boxstyle="round,pad=0.35"),
        )
    # Light horizontal guides
    ax.grid(axis="y", linestyle="--", linewidth=0.5, alpha=0.5)
    plt.tight_layout()
    output_path = "noaa_value_tree_beta_violin_comparison.png"
    plt.savefig(output_path, dpi=200, bbox_inches="tight")
    plt.close(fig)
    print(output_path)

def main():
    """
    Main module
    """
    #--------------------
    #load the known data from PALMA output
    products = ["NHCTCWxWrng", "JTWCTCWxWrng"] #NHC and JTWC products
    rtnScr_sq = [79.99989075,59.99876288] #Status Quo Scores for NHC and JTWC products
    rtnScr_atms = [88.03501536,79.68026463] #ATMS+ Return Scores for NHC and JTWC products
    rtnScr_radar = [91.2727356,77.87295946] #NEXRAD+ Return Scores for NHC and JTWC products
    rtnScr_outcome_sq = 71.9994396 #PALMA computed Baseline Outcome at Outcome
    rtnScr_outcome_atms = 84.69311507 #PALMA computed improvement from ATMS+ at Outcome
    rtnScr_outcome_radar = 85.91282515 #PALMA computed improvement from NEXRAD+ at Outcome
    wgts = [60,40] #Outcome weights for NHC and JTWC products
    kappa_sq = [16,32] #Status Quo kappas for NHC and JTWC products

    #--------------------
    #Initialize values for calculator
    modes_sq = [x / 100 for x in rtnScr_sq] #Modes for Status Quo Scores
    modes_atms = [x / 100 for x in rtnScr_atms] #Modes for ATMS+ Return Scores
    modes_radar = [x / 100 for x in rtnScr_radar] #Modes for NEXRAD+ Return Scores
    mode_outcome_atms = rtnScr_outcome_atms / 100
    mode_outcome_radar = rtnScr_outcome_radar / 100
    mode_outcome_sq = rtnScr_outcome_sq / 100 #Mode of sq outcome
    #Diagnose
    print(f"mode_outcome_initial: {mode_outcome_sq:.5f}")
    print(f"mode_outcome_atms: {mode_outcome_atms:.5f}")
    print(f"mode_outcome_radar: {mode_outcome_radar:.5f}")

    #--------------------
    #Compute status quo parameter values
    alpha_sq = [0,0]
    beta_sq = [0,0]
    var_sq = [0,0]
    for i in range(len(kappa_sq)):
        alpha, beta, variance = compute_params(kappa_sq[i], modes_sq[i])
        alpha_sq[i] = alpha
        beta_sq[i] = beta
        var_sq[i] = variance
        #Diagnose
        print("Status Quo", products[i])
        print(f"  kappa: {kappa_sq[i]:.3f}")
        print(f"  mode: {modes_sq[i]:.3f}")
        print(f"  alpha: {alpha:.3f}")
        print(f"  beta:, {beta:.3f}")
        print(f"  variance: {variance:.3f}")

    #--------------------
    #Compute new parameter values for ATMS+
    alpha_atms = [0,0]
    beta_atms = [0,0]
    kappa_atms = [0,0]
    var_atms = [0,0]
    for i in range(len(kappa_atms)):
        kappa_0 = kappa_sq[i]
```

```
        m_0 = modes_sq[i]
        m_new = modes_atms[i]
        kappa_new = compute_kappa_new(kappa_0,m_0,m_new)
        kappa_atms[i] = kappa_new
        alpha, beta, variance = compute_params(kappa_new, m_new)
        alpha_atms[i] = alpha
        beta_atms[i] = beta
        var_atms[i] = variance
        #Diagnose
        print("ATMS+", products[i])
        print(f"  kappa: {kappa_atms[i]:.3f}")
        print(f"  mode: {modes_atms[i]:.3f}")
        print(f"  alpha: {alpha:.3f}")
        print(f"  beta:, {beta:.3f}")
        print(f"  variance: {variance:.3f}")

    #--------------------
    #Compute new parameter values for NEXRAD+
    alpha_radar = [0,0]
    beta_radar = [0,0]
    kappa_radar = [0,0]
    var_radar = [0,0]
    for i in range(len(kappa_radar)):
        kappa_0 = kappa_sq[i]
        m_0 = modes_sq[i]
        m_new = modes_radar[i]
        kappa_new = compute_kappa_new(kappa_0,m_0,m_new)
        kappa_radar[i] = kappa_new
        alpha, beta, variance = compute_params(kappa_new, m_new)
        alpha_radar[i] = alpha
        beta_radar[i] = beta
        var_radar[i] = variance
        #Diagnose
        print("NEXRAD+", products[i])
        print(f"  kappa: {kappa_radar[i]:.3f}")
        print(f"  mode: {modes_radar[i]:.3f}")
        print(f"  alpha: {alpha:.3f}")
        print(f"  beta:, {beta:.3f}")
        print(f"  variance: {variance:.3f}")

    #--------------------
    #Aggregate status quo children for outcome
    a = alpha_sq
    b = beta_sq
    w = wgts
    M_agg_sq,V_agg_sq,kappa_agg_sq,alpha_agg_sq,beta_agg_sq,S_sq = aggregate_betas_scaled(a, b,
w)
    #Diagnose
    print("Aggregated values for scenario Status Quo:")
    print(f"  Mode: {M_agg_sq:.3f}")
    print(f"  Kappa: {kappa_agg_sq:.3f}")
    print(f"  Alpha: {alpha_agg_sq:.3f}")
    print(f"  Beta: {beta_agg_sq:.3f}")
    print(f"  Penalty Scale: {S_sq:.3f}")

    #--------------------
    #Aggregate ATMS+ children for outcome
    a = alpha_atms
    b = beta_atms
    w = wgts
    M_agg_atms,V_agg_atms,kappa_agg_atms,alpha_agg_atms,beta_agg_atms,S_atms =
aggregate_betas_scaled(a, b, w)
    #Diagnose
    print("Aggregated values for scenario ATMS+:")
    print(f"  Mode: {M_agg_atms:.3f}")
    print(f"  Kappa: {kappa_agg_atms:.3f}")
    print(f"  Alpha: {alpha_agg_atms:.3f}")
    print(f"  Beta: {beta_agg_atms:.3f}")
    print(f"  Penalty Scale: {S_atms:.3f}")

    #--------------------
```

```
    #Aggregate NEXRAD+ children for outcome
    a = alpha_radar
    b = beta_radar
    w = wgts
    M_agg_radar,V_agg_radar,kappa_agg_radar,alpha_agg_radar,beta_agg_radar,S_radar =
aggregate_betas_scaled(a, b, w)
    #Diagnose
    print("Aggregated values for scenario NEXRAD+:")
    print(f"  Mode: {M_agg_radar:.3f}")
    print(f"  Kappa: {kappa_agg_radar:.3f}")
    print(f"  Alpha: {alpha_agg_radar:.3f}")
    print(f"  Beta: {beta_agg_radar:.3f}")
    print(f"  Penalty Scale: {S_radar:.3f}")

    #--------------------
    #Normalize the weights
    total = sum(wgts)
    wgts = [x / total for x in wgts]

    #--------------------
    #Compute normalized information gain and KL Divergence for ATMS+ scenario
    a0 = alpha_agg_sq
    b0 = beta_agg_sq
    k0 = kappa_agg_sq
    a1 = alpha_agg_atms
    b1 = beta_agg_atms
    k1 = kappa_agg_atms
    h_sq = beta_entropy_bits(a0,b0,k0)
    h_final = beta_entropy_bits(a1,b1,k1)
    normalized_information_gain_outcome_atms = (h_final - h_sq) / h_final
    result = normalized_information_gain_outcome_atms * 100
    KLD_sq_atms = compute_beta_kl_divergence(a0, b0, a1, b1)
    KLD_final_atms = compute_beta_kl_divergence(1, 1, a1, b1)
    norm_KLD_atms = (KLD_sq_atms / KLD_final_atms) * 100
    #Diagnose
    print(f"Normalized Information Gain Outcome for ATMS+: {result:.3f} %")
    print(f"Percentage of Uncertainty Resolved for ATMS+: {norm_KLD_atms:.3f} %")

    #--------------------
    #Compute normalized information gain and KL Divergence for NEXRAD+ scenario
    a0 = alpha_agg_sq
    b0 = beta_agg_sq
    k0 = kappa_agg_sq
    a1 = alpha_agg_radar
    b1 = beta_agg_radar
    k1 = kappa_agg_radar
    h_sq = beta_entropy_bits(a0,b0,k0)
    h_final = beta_entropy_bits(a1,b1,k1)
    normalized_information_gain_outcome_radar = (h_final - h_sq) / h_final
    result = normalized_information_gain_outcome_radar * 100
    KLD_sq_radar = compute_beta_kl_divergence(a0, b0, a1, b1)
    KLD_final_radar = compute_beta_kl_divergence(1, 1, a1, b1)
    norm_KLD_radar = (KLD_sq_radar / KLD_final_radar) * 100
    #Diagnose
    print(f"Normalized Information Gain Outcome for NEXRAD+: {result:.3f} %")
    print(f"Percentage of Uncertainty Resolved for NEXRAD+: {norm_KLD_radar:.3f} %")

    #--------------------
    #Draw comparison violin plots
    m = [M_agg_sq,M_agg_atms,M_agg_radar]
    k = [kappa_agg_sq,kappa_agg_atms,kappa_agg_radar]
    title = "NOAA Value Tree Comparison: Tropical Cyclone Weather Warnings"
    draw_violin_plots_x3(m,k,title)

if __name__ == "__main__":
    main()
```