Quantifying Uncertainty in Modular Foundation Models: A Case Study in Robot-Assisted Bite Acquisition

Anonymous Author(s)

Abstract

As robots increasingly rely on large foundation models for perception and planning, the ability to estimate uncertainty reliably on those foundation models becomes critical for safe human-robot interaction and effective human-in-the-loop failure recovery, particularly in assistive caregiving domains. While Vision-Language Models (VLMs) and Vision-Language-Action (VLA) models offer impressive generalization capabilities, they either lack native confidence measurements or produce uncalibrated scores prone to catastrophic overconfidence in failure modes. In this work, we introduce three model-specific uncertainty quantification strategies for a modular robotic bite-acquisition pipeline consisting of a food detector (GPT-4o), a bounding box selector (GroundingDINO), a skill selector (GPT-4o), and a skill parameter selector (RT-1). Leveraging data collected from using full pipeline food acquisition, we demonstrate that these tailored strategies substantially reduce overconfidence in failure modes compared to uncalibrated baselines. Our results highlight the necessity of architecture-specific uncertainty quantification to ensure safe and reliable failure recovery in foundation-model-based human-robot interaction systems.

CCS Concepts

• Computer systems organization → Robotics; **• Computing methodologies → Artificial intelligence**; **• Human-centered computing →** *Human computer interaction (HCI).*

Keywords

Human-Robot Interaction, Foundation Models, Uncertainty Quantification, Failure Recovery, Robot-Assisted Feeding

ACM Reference Format:

Anonymous Author(s). 2026. Quantifying Uncertainty in Modular Foundation Models: A Case Study in Robot-Assisted Bite Acquisition. In *Proceedings of ACM/IEEE International Conference on Human-Robot Interaction (HRI '26).* ACM, New York, NY, USA, 4 pages.

1 Introduction

The integration of large foundation models into robotic pipelines has enabled systems to operate in unstructured environments with impressive generalization capabilities. Vision-Language Models (VLMs) and Vision-Language-Action models (VLA) allow robots to interpret open-set instructions and generate complex plans without task-specific training [2, 4, 8]. However, in safety-critical domains, such as robot-assisted caregiving, the ability of a robot to perform a task is secondary to its ability to recognize when it cannot. A robot that fails is inconvenient; a robot that fails while confident in its success poses a safety risk and causes user mistrust. To ensure safety and enable effective human-in-the-loop failure recovery, these systems must have reliable and calibrated measures of their own uncertainty.

Reliable uncertainty quantification in modular robotic pipelines [1, 5] is non-trivial because different model classes uses different training methodologies and therefore exhibit fundamentally different statistical behaviors. VLMs like GPT-4o output token log-probabilities as confidence scores [4]; open-set detectors such as GroundingDINO output text-image alignment scores [8], while VLAs such as RT-1 provide no native confidence estimates at all [2]. In practical systems, these modules may be a combination of open-source models with full architectural access and proprietary models available only through API calls. This limits visibility into model internal representations or uncertainty heads, requiring uncertainty estimation methods to rely on model outputs rather than model internals. While recent work has explored uncertainty in end-to-end VLA models [6, 7] or LLM planners [9, 10], evaluating model uncertainty within multi-stage, modular architectures remains an open challenge. Relying on raw model output often yields uncalibrated estimates prone to catastrophic overconfidence, particularly in failure modes where the model hallucinates or acts out-of-distribution [3].

In this work, we conduct an empirical study of model-specific uncertainty quantification methods with a previously proposed modular bite-acquisition pipeline recently for assistive food acquisition. This framework decomposes the task into four sequential modules, Food Type Identification (M_1), Bounding Box Selection (M_2), Skill Selection (M_3), and Skill Parameter Selection (M_4), and includes a human-in-the-loop query policy that uses module-level confidence scores to decide when to ask the user for help. Critically, this prior work assumes that these confidence scores are reasonably calibrated.

Here, we directly investigate this assumption by benchmarking a set of customized uncertainty estimation strategies for each module against uncalibrated baselines, using a dataset collected from real-world pipeline executions.

We summarize our contributions as the following:

- A systematic analysis of the uncertainty characteristics across three foundation model architectures (GPT-4o, GroundingDINO, RT-1).
- Three model-specific, calibrated uncertainty methods corresponding to each module in the bite acquisition pipeline.
- An empirical evaluation of these uncertainty methods using data collected from real-world executions of the pipeline.

2 Overview of the Prior Modular Bite-Acquisition Pipeline

We build on a previously proposed modular bite-acquisition architecture for assistive feeding. The system decomposes the task into four sequential perception-action modules:

HRI '26, Edinburgh, Scotland, UK
2026.

M_1***: Food Type Identification.*** A GPT-4o vision-language model processes the plate RGB image $z_{\text{RGB}} \in \mathbb{R}^{H \times W \times 3}$ and identifies a candidate set of food labels $\mathcal{L} = \{l_1, \ldots, l_K\}$, where K is the number of unique food categories present. The module outputs the label with the highest model confidence:

$$M_1(z_{\text{RGB}}) \rightarrow l^*.$$

This module employs an iterative prompting strategy. It first queries GPT-4o to estimate the count of items (N), then executes N sequential classification calls, instructing the model to identify the k-th item while avoiding previously found items. The module filters duplicates and outputs the label l^* with the highest confidence from the candidate set.

M_2***: Bounding Box Selection.*** Given the selected label l^* and the image z_{RGB}, GroundingDINO predicts a set of bounding boxes $\mathcal{B}(l^*) = \{b_1, \ldots, b_J\}$ corresponding to detected instances of food type l^*:

$$M_2(z_{\text{RGB}}, l^*) \rightarrow \mathcal{B}(l^*).$$

M_3***: Skill Selection.*** A second GPT-4o module predicts the optimal high-level manipulation skill (e.g., skewering, scooping, twirling) given the food label and its bounding box:

$$M_3(l^*, b_i) \rightarrow a_i^h,$$

where $a_i^h \in \mathcal{A}_h$ denotes the selected high-level skill.

M_4***: Skill Parameter Selection.*** A modified regression-based RT-1 refines the chosen skill into low-level motion parameters $a_i^l = (x_1, y_1, x_2, y_2) \in \mathbb{R}^4$ (e.g., start/end points), where each parameter is normzlied to $[0, 1]$. Note that unlike the standard RT-1, which discretizes actions into tokens and outputs a probability distribution over a vocabulary, this architecture replaces the tokenization head with a continuous regression head.

$$M_4(a_i^h, b_i) \rightarrow a_i^l.$$

Human-in-the-Loop Query Policy. A human-in-the-loop query policy determines when to proceed autonomously or to request user for help based on the confidence signals from M_1 to M_4, and this algorithm assumes access to reasonably calibrated module-level confidence signals.

3 Uncertainty Quantification Methods

Our primary goal is to obtain calibrated confidence measures, where the predicted probability correlates reliably with the likelihood of task success. Because the pipeline employs fundamentally different foundation model architectures, a unified uncertainty metric is insufficient. Instead, we introduce tailored quantification strategies for each module's output.

3.1 M_1, M_3: The Sink-Token Strategy

The Food Identification M_1 and Skill Selection M_3 use GPT-4o to classify visual inputs into a discrete label set. Let $\mathcal{L} = \{l_1, ..., l_K\}$ denote the set of canonical food or skill labels. GPT-4o produces a log-probability vector over $\mathcal{L}$ via a Softmax distribution:

$$p(l_i|x) = \frac{\exp(z_i)}{\sum_{j=1}^{K} \exp(z_j)}$$

where z_i are the model logits for input x.

In practice, the raw log-probability score, however, is often overconfident, especially in ambiguous or out-of-distribution cases. To overcome this, we augment the label set with an explicit "sink" token, representing the "None of the above" option and providing a destination for probability mass when evidence for all labels is weak.

$$\mathcal{L}' = \mathcal{L} \cup \{l_{sink}\}$$

GPT-4o now outputs logits over $\mathcal{L}'$, and the calibrated confidence score is defined as:

$$c(x) = \begin{cases} 0, & l^* = l_{\text{sink}}, \\ \exp(\log p(l^*|x)), & \text{otherwise}, \end{cases}$$

3.2 M_2: The Competing Confidence Score

The bounding box selection module M_2 uses GroundingDINO that outputs a sigmoid-alignment score for each text label and candidate bounding box. Unlike Softmax-based classifiers, the sigmoid scores are not competitive to each other: the score for one label is computed independently of others. For instance, a high alignment score for "Cantaloupe" does not suppress the score for a visually similar "Honeydew." This makes it hard to decide the decision boundary. A "0.54" score doesn't necessarily mean the model is confident, and a "0.32" doesn't necessarily mean the model is uncertain.

To address this, we introduce a competing-confidence scoring method that explicitly measures how well a bounding box aligns with the target label relative to all other possible labels. To do this, we introduce a two-round evaluation.

Round 1 (Target Detection). Query GroundingDINO with the target label l^*, producing a bounding box b^* and its text-image alignment score $s_1(b^*, l^*)$

Round 2 (Competing Labels). Query all other labels

$$\mathcal{L}_{\neg l^*} = \mathcal{L} \setminus \{l^*\},$$

producing competing scores

$$s_c(b_j, l), \quad l \in \mathcal{L}_{\neg l^*} \quad b_j \in \mathcal{B}(l).$$

For each bounding box b_j, we first consider it a meaningful competitor only if it overlaps the target box b^* with Intersection over Union (IoU) above a threshold τ (we used 0.95), ensuring that only visually similar foods at the same location compete.

We then identify the strongest competitor:

$$s_{\max} = \max_{\substack{l \in \mathcal{L}_{\neg l^*} \\ b_j \in \mathcal{B}(l)}} s_c(b_j, l).$$

Competing-Confidence Score. We define the competing-confidence for box b^* as:

$$c_{\text{comp}} = s_1(b^*, l^*) - s_{\max}.$$

and this is the calibrated confidence score for M_2.

3.3 M_4: Monte-Carlo Dropout

The skill-parameter module (M_4) uses a modified RT-1 model to predict a 4-dimensional continuous action vector representing the start and end points of the intended motion. Unlike M_1 and M_2, this module is trained as a regressor rather than a classifier, meaning it does not produce token probabilities or any native confidence scores.

To estimate uncertainty, we apply Monte-Carlo (MC) Dropout to the modified RT-1. For each cropped input image, RT-1 is evaluated T times (we use $T = 16$) with dropout enabled at a rate of 0.2 in the intermediate layers. Each stochastic forward pass yields a 4-dimensional action vector a_t. The predictive mean $\bar{a}$ is computed as the average across samples, and the predictive uncertainty is quantified using the variance of these samples. A high predictive variance indicates that RT-1 is unstable or unsure about the correct skill parameters, whereas a low variance reflects confident and consistent outputs.

$$\mathrm{Var}_{\mathrm{MC}}(a) = \frac{1}{T} \sum_{t=1}^{T} \|a_t - \bar{a}\|^2 . \tag{1}$$

4 Evaluation and Results

We evaluate our uncertainty quantification strategies using a dataset collected from real-world executions of the bite-acquisition pipeline during a user study. From the interaction logs, we extracted 316 execution samples, each containing whole-plate RGB images and corresponding model outputs. For each module (M_1 to M_4), we manually annotated the ground-truth correctness of the prediction.

To assess calibration quality, we employ two complementary metrics. Our primary metric is the *Confidence Gap* ($\Delta\mu$), defined as the difference between the mean confidence of correct predictions μ_{correct} and mean confidence of incorrect predictions $\mu_{\mathrm{incorrect}}$:

$$\Delta\mu = \mu_{\mathrm{correct}} - \mu_{\mathrm{incorrect}}.$$

A larger $\Delta\mu$ indicates stronger discriminative ability, reflecting clearer separation between reliable predictions and failures.

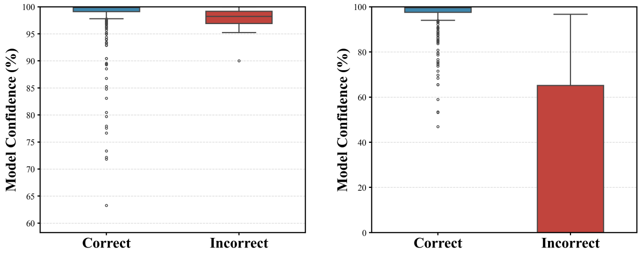
Because mean separation alone does not account for variability in the confidence distributions, we also compute the *Discriminability Index* (d'):

$$d' = \frac{|\mu_{\mathrm{correct}} - \mu_{\mathrm{incorrect}}|}{\sqrt{\frac{1}{2}\left(\sigma^2_{\mathrm{correct}} + \sigma^2_{\mathrm{incorrect}}\right)}}.$$

where $\sigma_{\mathrm{correct}}$ means the confidence variance of correct predictions and $\sigma_{\mathrm{incorrect}}$ means the confidence variance of incorrect predictions. A higher d' indicates more statistically robust separation, penalizing methods that achieve large confidence gaps only at the cost of high prediction noise or instability.

4.1 M_1 Uncertainty Results Analysis

As shown in Figure 1 below, the baseline confidence distribution provides almost no separation between success and failure cases: incorrect predictions receive nearly the same confidence as correct ones. This is reflected quantitatively in the small Confidence Gap ($\Delta\mu = 0.010$) and the very low discriminability index ($d' = 0.247$), indicating that the baseline signal is essentially non-diagnostic. After applying our Sink Token normalization, the confidence distribution becomes sharply bimodal, with incorrect predictions receiving substantially reduced scores. This shift produces a dramatically larger separation between the two distributions ($\Delta\mu = 0.708$) and more than ten times increase in discriminability ($d' = 2.62$). Together, these results show that our approach yields a significantly more meaningful and reliable uncertainty signal for the Food Identification module M_1.



(a) Distribution of M_1 Baseline Confidence: Correct vs. Incorrect

(b) Distribution of M_1 Calibrated Confidence: Correct vs. Incorrect

Figure 1: Calibration effects on M_1 Food Identification

Table 1: Calibration statistics for Food Identification GPT-4o (M_1) on 316 real-world images. Higher is better for $\Delta\mu$ and d'.

	Correct ($\mu \pm \sigma$)		Incorrect ($\mu \pm \sigma$)		$\Delta\mu^{\uparrow}$	$d'^{\uparrow}$
Method	Mean	Std Dev	Mean	Std Dev		
Baseline	0.982	0.048	0.972	0.031	0.010	0.247
Calibrated	0.964	0.081	0.256	0.373	**0.708**	**2.62**

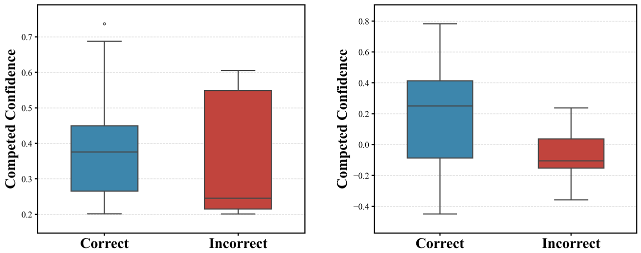
4.2 M_2 Uncertainty Results Analysis

For GroundingDINO, we evaluate our calibrated competing confidence scores against the raw model output. The baseline method exhibits substantial distributional overlap ($d' = 0.290$), with incorrect bounding boxes receiving alignment scores ($\mu = 0.340$) close to the correct ones ($\mu = 0.383$). This suggests that raw alignment behaves more like a visual similarity score than a confidence measure.

Our calibration approach shifts the incorrect distribution downward to $\mu = -0.05$. The improvement in Confidence Gap and Discriminability Index indicates that the calibrated score suppresses hallucinated boxes that match texture but lack semantic grounding.

Table 2: Calibration statistics for Bounding Box Detection (M_2) on 316 real-world images. Higher is better for $\Delta\mu$ and d'.

	Correct ($\mu \pm \sigma$)		Incorrect ($\mu \pm \sigma$)		$\Delta\mu^{\uparrow}$	$d'^{\uparrow}$
Method	Mean	Std Dev	Mean	Std Dev		
Baseline	0.383	0.131	0.340	0.161	0.043	0.293
Calibrated	0.198	0.290	-0.05	0.160	**0.248**	**1.06**



(a) Distribution of M_2 Baseline Confidence: Correct vs. Incorrect

(b) Distribution of M_2 Calibrated Confidence: Correct vs. Incorrect

Figure 2: Calibration effects on M_2 Bounding Box Detection

4.3 M_3 Uncertainty Results Analysis

The Skill Selection module (M_3) exhibits perfect performance across all 316 execution samples. This outcome is expected given the structure of the task: M_3 implements a deterministic mapping from the food category (provided by M_1) to a predefined skill type. Once the food label is known, the required skill is essentially a rule-based lookup with minimal ambiguity. As a result, M_3 produced no incorrect predictions in our dataset.

This confirms that uncertainty modeling at the skill-selection level in this pipeline is unnecessary, and effort should be concentrated on upstream perception modules (M_1, M_2) and downstream continuous-parameter regression (M_4), where uncertainty meaningfully impacts system performance.

4.4 M_4 Uncertainty Results Analysis

We next evaluate the MC Dropout variance metric for the RT-1 policy (M_4). Unlike earlier modules, this regression task has no native baseline confidence score, so our analysis focuses on whether predictive variance serves as a meaningful proxy for uncertainty.

As shown in Table 3, incorrect predictions consistently exhibit higher variance ($\mu = 0.096$) than correct ones ($\mu = 0.079$). Although the gap is small ($\Delta\mu = 0.017$), the low variability makes this difference statistically significant, as reflected by the discriminability index ($d' = 0.827$).

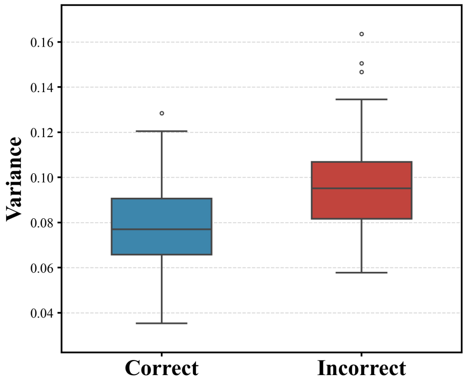


Figure 3: Distribution of M_4 Calibrated Confidence: Correct vs. Incorrect

Table 3: Calibration statistics for Skill Parameter Selection (M_4) on 316 real-world images. Higher is better for $\Delta\mu$ and d'.

	Correct ($\mu \pm \sigma$)		Incorrect ($\mu \pm \sigma$)		$\Delta\mu\uparrow$	$d'\uparrow$
Method	Mean	Std Dev	Mean	Std Dev		
Calibrated	0.079	0.019	0.096	0.022	**0.017**	**0.827**

5 Discussion

Across the four modules of our bite-acquisition pipeline, this work demonstrates three principled strategies for transforming raw model outputs into calibrated confidence signals: (1) a Sink-Token strategy for VLM classification, (2) a Competing Confidence strategy for GroundingDINO, and (3) Monte Carlo Dropout variance for continuous regression-based model.

Importantly, calibrated confidence is not an end in itself; rather, it provides the foundational signal required to build algorithms that determine *when* the robot should act autonomously versus *when* it should query or defer to a human. While our work establishes the uncertainty measures needed to support such decision-making, the design of the decision boundary is outside the scope of this paper. In real deployments, these decisions must incorporate task criticality, user-specific preferences, cognitive load, failure costs, and broader considerations of shared autonomy.

References

[1] Rohan Banerjee, Rajat Kumar Jenamani, Sidharth Vasudev, Amal Nanavati, Katherine Dimitropoulou, Sarah Dean, and Tapomayukh Bhattacharjee. 2025. To Ask or Not To Ask: Human-in-the-loop Contextual Bandits with Applications in Robot-Assisted Feeding. In *International Conference on Robotics and Automation (ICRA)*.

[2] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, et al. 2022. RT-1: Robotics Transformer for Real-World Control at Scale. *arXiv preprint arXiv:2212.06817* (2022).

[3] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Trans. Inf. Syst.* 43, 2, Article 42 (Jan. 2025), 55 pages. doi:10.1145/3703155

[4] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. GPT-4o System Card. *arXiv preprint arXiv:2410.21276* (2024).

[5] Rajat Kumar Jenamani, Priya Sundaresan, Maram Sakr, Tapomayukh Bhattacharjee, and Dorsa Sadigh. 2024. FLAIR: Feeding via Long-Horizon Acquisition of Realistic dishes. In *Robotics: Science and Systems (RSS)*.

[6] Ulas Berk Karli, Tetsu Kurumisawa, and Tesca Fitzgerald. 2025. Ask Before You Act: Token-Level Uncertainty for Intervention in Vision-Language-Action Models. In *Second Workshop on Out-of-Distribution Generalization in Robotics at RSS 2025*.

[7] Ulas Berk Karli, Ziyao Shangguan, and Tesca Fitzgerald. 2025. INSIGHT: INference-time Sequence Introspection for Generating Help Triggers in Vision-Language-Action Models. *arXiv preprint arXiv:2510.01389* (2025).

[8] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. 2023. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. *arXiv preprint arXiv:2303.05499* (2023).

[9] James F Mullen Jr and Dinesh Manocha. 2024. Towards Robots That Know When They Need Help: Affordance-Based Uncertainty for Large Language Model Planners. *arXiv preprint arXiv:2403.13198* (2024).

[10] Shiyuan Yin, Chenjia Bai, Zihao Zhang, Junwei Jin, Xinxin Zhang, Chi Zhang, and Xuelong Li. 2025. Towards Reliable LLM-based Robots Planning via Combined Uncertainty Estimation. In *Advances in Neural Information Processing Systems (NeurIPS)*.