KNOW WHERE YOU'RE UNCERTAIN WHEN PLANNING WITH MULTIMODAL FOUNDATION MODELS: A FORMAL FRAMEWORK

Neel P. Bhatt [*1] **Yunhao Yang** [*1] **Rohan Siva** [1] **Daniel Milan** [1] **Ufuk Topcu** [1] **Zhangyang Wang** [1]

ABSTRACT

Multimodal foundation models offer a promising framework for robotic perception and planning by processing sensory inputs to generate actionable plans. However, addressing uncertainty in both perception (sensory interpretation) and decision-making (plan generation) remains a critical challenge for ensuring task reliability. We present a comprehensive framework to disentangle, quantify, and mitigate these two forms of uncertainty. We first introduce a framework for uncertainty *disentanglement*, isolating *perception uncertainty* arising from limitations in visual understanding and *decision uncertainty* relating to the robustness of generated plans.

To quantify each type of uncertainty, we propose methods tailored to the unique properties of perception and decision-making: we use conformal prediction to calibrate perception uncertainty and introduce Formal-Methods-Driven Prediction (FMDP) to quantify decision uncertainty, leveraging formal verification techniques for theoretical guarantees. Building on this quantification, we implement two targeted *intervention* mechanisms: an active sensing process that dynamically re-observes high-uncertainty scenes to enhance visual input quality and an automated refinement procedure that fine-tunes the model on high-certainty data, improving its capability to meet task specifications. Empirical validation in real-world and simulated robotic tasks demonstrates that our uncertainty disentanglement framework reduces variability by up to 40% and enhances task success rates by 5% compared to baselines. These improvements are attributed to the combined effect of both interventions and highlight the importance of uncertainty disentanglement which facilitates targeted interventions that enhance the robustness and reliability of autonomous systems. Webpage, videos, demo, and code: https://uncertainty-in-planning.github.io.

1 INTRODUCTION

Accurate perception and the generation of actionable plans are essential for effective robotic perception and planning (Song et al., 2023; Shah et al., 2022; Yang et al., 2023; 2022; Ichter et al., 2022; Bhatt et al., 2023; Wang et al., 2024). Recent advancements in multimodal foundation models have equipped robots with the ability to process visual data and generate corresponding textual plans (Wu et al., 2023; Gu et al., 2023; Mu et al., 2023). While multimodal models provide integrated perception and planning capabilities, these two components — perception and decision-making — each contribute distinct forms of uncertainty that affect overall performance. Without disentangling these sources, any observed failures or inconsistencies in model outputs remain difficult to diagnose, as it is unclear whether they stem from inaccuracies in visual perception or limitations in planning.

Disentangling uncertainty allows for targeted improvements: enhancing perception uncertainty focuses on refining sensory interpretation and visual recognition, whereas addressing decision uncertainty improves the robustness and alignment of generated plans to task specifications. By separately identifying and mitigating these uncertainties, we may enable targeted interventions that enhance model reliability and adaptability in dynamic, real-world environments.

This paper aims to bridge this largely unaddressed gap (Ye et al., 2023; Vazhentsev et al., 2022; Ren et al., 2023) by introducing a novel framework to distinguish and quantify the uncertainty associated with each constituting module, thereby enabling targeted model refinement and improved execution reliability. Specifically:

- ***Perception uncertainty*** arises from the model's limitations in visually understanding and reliably interpreting sensory data, such as recognizing and localizing objects. For instance, in a cluttered room, high perception uncertainty can manifest in the robot's inability to accurately detect and localize obstacles in its path.

- ***Decision uncertainty*** pertains to the model's planning capabilities, specifically its capacity to generate a sequence of actions that aligns with the task requirements.

[*]Equal contribution [1]The University of Texas at Austin, United States. Correspondence to: Neel P. Bhatt <npbhatt@utexas.edu>, Yunhao Yang <yunhaoyang234@utexas.edu>.

Proceedings of the 8th MLSys Conference, Santa Clara, CA, USA, 2025. Copyright 2025 by the author(s).

Using the same navigation example, decision uncertainty would reflect the model's ability to formulate a plan that avoids all obstacles and successfully reaches the target destination. Importantly, decision uncertainty can persist even when visual perception is perfect, highlighting the need for separate evaluation and mitigation of each uncertainty source.

Contributions. This paper advances multimodal foundation models in robotic planning through three key contributions:

- **Uncertainty Disentanglement Framework**: We introduce a novel framework to disentangle *perception uncertainty* and *decision uncertainty* within multimodal foundation models, addressing the unique challenges each type of uncertainty presents in robotic planning. This disentanglement allows us to identify and treat these uncertainties separately, enabling targeted improvements in model robustness.
- **Novel Quantification of Each Uncertainty**: To estimate these uncertainties, we propose novel methods tailored to each type. For perception uncertainty, we leverage *conformal prediction* to calibrate the model's visual confidence, providing a probabilistic measure of the model's accuracy in object recognition. For decision uncertainty, we introduce *Formal-Methods-Driven Prediction* (FMDP), which uses formal verification techniques alongside conformal prediction to assess the likelihood that generated plans will satisfy task-specific requirements, offering a robust means to validate planning outputs.
- **Targeted Interventions via Active Sensing and Automated Refinement**: Building on this uncertainty quantification, we implement a two-part improvement strategy. First, our active sensing mechanism dynamically initiates re-observation of high-uncertainty scenes, enhancing visual input quality and reducing the propagation of perceptual errors. Second, we introduce an uncertainty-aware fine-tuning procedure that refines the model on low-uncertainty samples, improving its consistency in meeting task specifications while eliminating the need for human annotations.

We empirically validate the proposed frameworks through experiments in ground robot navigation, both in real-world environments and using Carla simulations (Dosovitskiy et al., 2017). Unlike conventional refinement frameworks, which lack the precision to identify and address specific uncertainty sources, our disentangled approach achieves a significant reduction in variability of model outputs — up to **40%** — and enhances the probability of satisfying task specifications by up to **5%**. These enhancements are a direct result of the synergistic combination of our targeted interventions and highlight the importance of distinguishing between sources of uncertainty in autonomous systems, ultimately improving their reliability and adaptability.

1.1 Related Work

Multimodal foundation models, such as GPT-4 (Achiam et al., 2023) and LLaVA (Liu et al., 2023b;a), have advanced robotic systems by jointly processing text and images to generate actionable plans. Existing uncertainty quantification methods (Ye et al., 2023; Vazhentsev et al., 2022; Shen et al., 2021; Ren et al., 2023; Hori et al., 2023) treat these models as monolithic, providing an aggregate measure of uncertainty without distinguishing between perception and decision components. This results in a "black box" evaluation, where any assessment of task failure or risk lacks insight into whether uncertainty originates from perception limitations or decision-making flaws.

Such aggregate uncertainty scores complicate model refinement since they obscure the root cause of performance issues, hindering targeted improvements (Rudner et al., 2024; Yu et al., 2022; Jang et al., 2024). For instance, while log-likelihood-based uncertainty estimation methods (Srivastava et al., 2022; Hendrycks et al., 2020) use conformal prediction techniques (Vovk et al., 2005; Angelopoulos et al., 2023; Balasubramanian et al., 2014) to calibrate confidence in plan correctness, they often rely on human-labeled ground truth for accuracy. This dependence can limit scalability and adaptability in dynamic environments.

Our framework addresses these limitations by disentangling uncertainty into perception and decision components, enabling targeted interventions to reduce each type of uncertainty independently. The proposed FMDP estimates decision uncertainty without requiring extensive human labeling. This disentanglement and targeted quantification significantly enhance model robustness and scalability through targeted uncertainty mitigation.

2 Preliminary Background

Formal Methods in Decision-Making Formal methods provide tools for modeling, analyzing, and verifying decision-making problems represented mathematically. By representing model outputs in a symbolic representation such as a Kripke structure, $\mathcal{A}$, these methods provide formal verification of the model's output against task specifications (Browne et al., 1988).

We use temporal logic to express the task specifications. A temporal logic specification, ϕ, constrains the temporal ordering and logical relationship between sequences of events and actions. Subsequently, we perform model checking, a process to determine whether the Kripke structure satisfies the specification, denoted as $\mathcal{A} \models \phi$, which provides a

Notation	Meaning	First Def.
M	foundation model	Sec. 3
$\mathcal{T}, \mathcal{I}$	text, image space	Sec. 3
I or I_i	a single image	Sec. 3
T or T_i	generated text instruction	Sec. 3
V	vision encoder	Sec. 3.1
H	projection head	Sec. 3.1
c	softmax confidence score	Sec. 3.1
y or y_i	image object label	Sec. 3.1
u_p	percep. uncertainty score	Def. 1
t_p	percep. threshold	Sec. 5.1
P_T or P_{T_i}	textual task description	Sec. 4
Y or Y_i	a set of observed objects	Sec. 4.1
u_d	decision uncertainty score	Def. 2
t_d	decision threshold	Sec. 4.2

Table 1: A summary of important notations.

formal guarantee associated with this verification.

Conformal Prediction Through the theory of conformal prediction, we can estimate the uncertainty of model predictions and obtain statistical guarantees on the correctness of these predictions (Vovk et al., 2005).

Formally, consider a space of (data, truth label) pairs $\mathcal{X} \times \mathcal{Y}$. We have a *calibration set* $S_C = \{(x_1, y_1), ..., (x_n, y_n)\}$ with n elements from the space $\mathcal{X} \times \mathcal{Y}$. These pairs are independent and identically distributed (i.i.d.). The goal is to find a *prediction band* $\hat{C} : \mathcal{X} \to \{\text{subsets of } \mathcal{Y}\}$. For example, $\hat{C}(x_i) = \{y_1, ..., y_i\} \subset \mathcal{Y}$.

Given a new test pair (x_{n+1}, y_{n+1}) sampled from $\mathcal{X} \times \mathcal{Y}$, we first make the following assumption:

Assumption 1. (x_{n+1}, y_{n+1}) is sampled from a distribution that is identically distributed with the distribution from which we sampled $S_C = \{(x_1, y_1), ..., (x_n, y_n)\}$.

Let $\epsilon \in [0, 1]$ be a score reflecting uncertainty. If assumption 1 holds, we expect the prediction band to meet the property

$$\mathbb{P}\left[y_{n+1} \in \hat{C}(x_{n+1})\right] \geq 1 - \epsilon. \tag{1}$$

The objective is to find $\hat{C}$ such that the probability of the ground truth label y_{n+1} belonging to the set $\hat{C}(x_{n+1})$ is bounded by $1 - \epsilon$. For instance, if $\hat{C}(x_{n+1})$ consists of two elements, then the probability of at least one of the elements matching the ground truth label is $1 - \epsilon$.

Table of Notations For convenience, we present all notations used in the rest of the paper in Table 1.

3 Perception Uncertainty

A multimodal foundation model, $M : \mathcal{I} \times \mathcal{T} \to \mathcal{T} \times [0, 1]$, takes an image observation I in the image space $\mathcal{I}$, and a task description P_T in the text space $\mathcal{T}$ as inputs. It then produces a textual instruction $T \in \mathcal{T}$ for the given task and a softmax confidence $c \in [0, 1]$. M consists of a vision encoder $V : \mathcal{I} \to \mathbb{R}^d$ that maps input images from the image space to a d-dimensional embedding space. The vision encoder outputs a d-dimensional vector which is sent as an input to a text generator for producing textual instructions. We define *perception uncertainty* in the vision encoder's output as below:

Definition 1. Let I be an image containing multiple objects, a *perception uncertainty score* u_p for I is a theoretical lower bound on the probability of correctly identifying all objects in the image. A lower score implies higher uncertainty.

Problem 1. Given a foundation model M, a task description $P_T \in \mathcal{T}$ with an image $I \in \mathcal{I}$, estimate the *perception uncertainty score* u_p for I.

3.1 Conformal Confidence-Uncertainty Calibration

As ad-hoc confidence scores returned by machine learning models, such as softmax, do not accurately reflect prediction accuracies (Guo et al., 2017), it is essential to calibrate confidence scores to prediction accuracies for uncertainty estimation. However, we ask for a theoretical lower bound on the accuracy according to Def. 1, which existing calibration methods cannot provide (Guo et al., 2017; Ding et al., 2021). Conformal prediction provides confidence intervals or sets for predictions guaranteeing a specified error rate (Vovk et al., 2005). We leverage the theory of conformal prediction for confidence-uncertainty calibration.

We first learn a *projection head* $H : \mathbb{R}^d \to \mathbb{R}^k$ that maps image embeddings to a k-dimensional space where vector $v \in \mathbb{R}^k$ represents the softmax confidence scores over k object classes. We denote the confidence score for class $l \in [0, k]$ by v_l. The class associated with the highest confidence score is the model's *prediction*.

Consider a calibration set $\{I_i, y_i\}_{i=1}^n$, where $I_i \in \mathcal{I}$ and $y_i \in [0, k]$ is the true class label for the object of interest in I_i, and assume the calibration set is i.i.d. with the images we from evaluation tasks. We connect the vision encoder to the projection head $H \circ V : \mathcal{I} \to \mathbb{R}^k$ and obtain a set of *nonconformity scores* $S_{nc} = \{1 - H(V(I_i))_{y_i}\}_{i=1}^n$. Intuitively, a nonconformity score is the sum of the softmax confidences of wrong predictions.

We empirically estimate a probability density function (PDF) f_{nc} from the nonconformity scores. For an image I_{n+1} from the test task (beyond the calibration set), given an error bound ϵ, we can find corresponding confidence c^* such

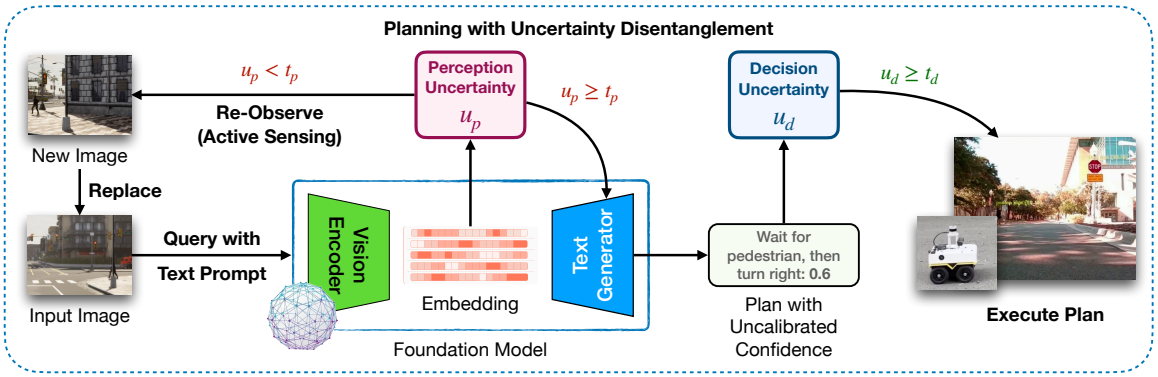

Figure 1: Our planning framework disentangles perception and decision uncertainty, triggering the active sensing intervention. The framework improves the robustness of generated plans by reducing the propagation of perceptual inaccuracies.

that $\hat{C}(I_{n+1}) = \{l \ : \ H(V(I_{n+1}))_l > 1 - c^*\}$ satisfies $\mathbb{P}\left[y_{n+1} \in \hat{C}(I_{n+1})\right] \geq 1 - \epsilon$ (Vovk et al., 2005).

Conversely, given a c^*, we can compute the corresponding error bound $\epsilon = 1 - \int_0^{c^*} f_{nc}(x)dx$. Given a confidence vector $v \in \mathbb{R}^k$ of image I_{n+1}, let $c^* = 1 - sort(v)_{-2}$, where $sort(v)_{-2}$ is the second-greatest value in v. The prediction band becomes

$$\begin{aligned}\hat{C}(I_{n+1}) &= \{l \ : \ H(V(I_{n+1}))_l > 1 - c^*\} \\ &= \{l \ : \ H(V(I_{n+1}))_l > sort(v)_{-2}\} \\ &= \{argmax(H(V(I_{n+1})))\}.\end{aligned} \tag{2}$$

The prediction band contains a single element, which is the prediction with the maximum confidence. Let $y_{n+1} \in [0, k]$ be the ground truth object label. By the theory of conformal prediction, there exists an $\epsilon \in [0, 1]$ such that

$$\begin{aligned}&\mathbb{P}\left[y_{n+1} \in \hat{C}(I_{n+1})\right] = \\ &\mathbb{P}\left[y_{n+1} = argmax(H(V(I_{n+1}))\right] \geq 1 - \epsilon,\end{aligned} \tag{3}$$

where $1 - \epsilon$ is the perception uncertainty score per Def. 1.

We know that c^* is the $1 - \epsilon$ quantile of S_{nc}. Thus, $1 - \epsilon = \int_0^{c^*} f_{nc}(x)dx$. Then,

$$\mathbb{P}\left[y_{n+1} = argmax(H(V(I_{n+1}))\right] \geq 1 - \epsilon = \int_0^{c^*} f_{nc}(x)dx. \tag{4}$$

Hence, we can calibrate a confidence vector v into a perception uncertainty score

$$u_p = \int_0^{c^*} f_{nc}(x)dx, \text{ where } c^* = 1 - sort(v)_{-2}. \tag{5}$$

4 Decision Uncertainty

Existing frameworks depend on human-annotated datasets to assess how well a generated plan aligns with human annotations, limiting their scalability (Malinin et al., 2017; Jiang et al., 2021). These approaches are limited in their ability to adapt to specific task requirements, resulting in less effective uncertainty estimates when applied to diverse or specialized scenarios (Ren et al., 2023). We use formal verification techniques alongside conformal prediction to eliminate the need for human annotations and improve adaptability to various task domains.

Recall that a foundation model $M : \mathcal{I} \times \mathcal{T} \to \mathcal{T} \times [0, 1]$ returns a text-based instruction to complete a task and a confidence score between 0 and 1. For example,

```
<Image> <Task description>
<Instruction>
Does the instruction satisfy the rules?
<Specifications>
Y/N
```

The inputs to M are in blue and the outputs are in red. The confidence refers to the softmax score of token 'Y' (Yes).

Given a set Φ of temporal logic specifications, we expect the text-based instructions generated by M to satisfy all specifications. However, due to the black-box nature of M, we cannot provide a guarantee. Thus, we aim to estimate a *decision uncertainty score* for each generated instruction.

Definition 2. Let Φ be a set of temporal logic specifications and T be a text-based instruction. A *decision uncertainty score* u_d for T is a numeric value estimating the probability that T satisfies all specifications $\phi \in \Phi$.

We formulate our problem as:

Problem 2. Given a foundation model M, a set Φ of specifications, a task description $P_T \in \mathcal{T}$, and an image $I \in \mathcal{I}$, estimate the *decision uncertainty score* u_d for instruction T, where $T, c = M(P_T, I)$.

4.1 Formal-Methods-Driven Prediction

We present **formal-methods-driven prediction (FMDP)**, which uses formal verification techniques alongside confor-

mal prediction, to solve Problem 2. For each task description P_T and input image I, FMDP aims to find a prediction band $\hat{C}(M, P_T, I) \subseteq \{T \mid T, c = M(P_T, I)\}$ such that

$$\mathbb{P}\left[T \in \hat{C}(M, P_T, I) : \forall_{\phi \in \Phi}\ \textbf{pEnc}(T, AP, Y) \models \phi\right] \geq 1 - \epsilon \tag{6}$$

for a given error tolerance $\epsilon \in [0, 1]$. Y is the set of objects from I, **pEnc** is an algorithm converting text to automaton, defined in Alg. 1. AP is the set of atomic propositions that are used to formulate every $\phi \in \Phi$. Eq. 6 implies that the probability that "at least one of the instructions obtained using the prediction band $\hat{C}$ satisfies the constraints defined by Φ" is at least $1 - \epsilon$.

Text-Based Instruction to Automaton Problem 2 requires verifying the textual outputs from M against the specifications. As we cannot directly verify texts against a logical specification, we develop an algorithm that expresses text-based instruction in a formal representation. The algorithm takes a textual instruction T, a set AP of atomic propositions, and a set Y of objects observed from them as inputs. It returns a *Kripke structure* (Browne et al., 1988) to represent the instruction. We show the details in Alg. 1.

Definition 3. A Kripke structure $\mathcal{A} = (Q, Q_0, \delta, \lambda)$ is a tuple consisting of a set Q of states, a set Q_0 of initial states, transitions $\delta \subseteq Q \times Q$, and a label function $\lambda : Q \to 2^{AP}$.

In particular, the algorithm first parses the textual instruction into a set of verb and noun phrases using semantic parsing (Kamath & Das, 2018). Next, we create an initial state q_0 whose label is the intersection of the set of atomic propositions and the set of observed objects. Then, we create a set of states, where each state corresponds to a phrase, and connect them linearly. Finally, we create a "final" state indicating the completion of the task. We present a step-by-step running example in Appendix B.

Then, we verify this structure against the specification through a model checker (Baier & Katoen, 2008). The model checker takes a logical specification and a Kripke structure as inputs and returns a binary signal indicating whether this structure satisfies the specification. We consider the textual instruction to satisfy the specification if and only if its corresponding Kripke structure passes the model-checking step.

Calibration Set and Nonconformity Distribution Once we enable the verification of the generated instructions, we can use the verification outcomes to formulate a calibration set for estimating a nonconformity distribution.

Given a set $\{I_j, Y_j\}_{j=1}^n$ of images with observed objects, e.g., the objects for the second image in Fig. 4 are {truck, car}, we supply each image I_j with a task description $P_{T_j} \in \mathcal{T}$ to M and obtain an instruction T_j and a confidence score c_j: $T_j, c_j = M(P_{T_j}, I_j)$. By repeating this step for all images, we obtain our calibration set $\{(T_j, c_j, Y_j)\}_{j=1}^n$.

Algorithm 1: Natural Language to Kripke Structures

input textual instruction T, atomic proposition set AP, set Y of observed objects
output $(Q, q_0, \delta, \lambda)$
$Ph = \{Ph_1, Ph_2, ...\}$ = **parse**(T)
Q, δ = [q_0], [] {Define a set of states and transitions. q_0 denotes initial states}
$\lambda(q_0) = Y \cap AP$ {The initial state's label is the observed objects from the image}
for Ph_i in Ph
  Q.append(q_i), δ.append((q_{i-1}, q_i)),
  $\lambda(q_i) = \{p \in AP : p \in Ph_i\}$
end for
Q.append(q_{done}), δ.append($(q_{|Ph|}, q_{done})$),
δ.append(q_{done}, q_{done})), $\lambda(q_{done}) = \emptyset$

We then obtain a set S_{nc} of *nonconformity scores*

$$S_{nc} = \{1 - c_j \ :\ \exists_{\phi \in \Phi}\ \textbf{pEnc}(T_j, AP, Y_j) \models \phi\}_{j=1}^n . \tag{7}$$

We use S_{nc} to estimate a PDF f_{nc} for the nonconformity distribution.

After we get the set of nonconformity distributions, for a given error bound $\epsilon \in [0, 1]$, we can find a prediction band $\hat{C}(M, P_T, I) = \{T \mid T, c = M(P_T, I) \text{ and } c \geq 1 - c^*\}$ that satisfies Eq. 6, where c^* is the $1 - \epsilon$ quantile of S_{nc}.

4.2 Estimating Decision Uncertainty Score

In this section, we use FMDP to estimate a decision uncertainty score for each output from the foundation model, i.e., solve Problem 2.

Given a new image I_{n+1} and a task description $P_{T_{n+1}}$, the foundation model outputs an instruction T_{n+1} and a confidence c_{n+1}. We disregard the instructions whose confidence in satisfying the specifications is below 0.5. Then, we compute the decision uncertainty score u_d using the nonconformity distribution f_{nc}:

$$u_d = \int_0^{c_{n+1}} f_{nc}(x)dx. \tag{8}$$

Proof. Consider a prediction band $\hat{C}(M, P_T, I) \subseteq \{T \mid T, c = M(P_T, I) \text{ and } c \geq 1 - c^*\}$ that satisfies Eq. 6. For each ϵ, there is a corresponding value of c^*. Let $c^* = c$, $\hat{C}(M, P_T, I)$ will contain only one element, which is the instruction T. By the theory of conformal prediction, there exists an ϵ such that

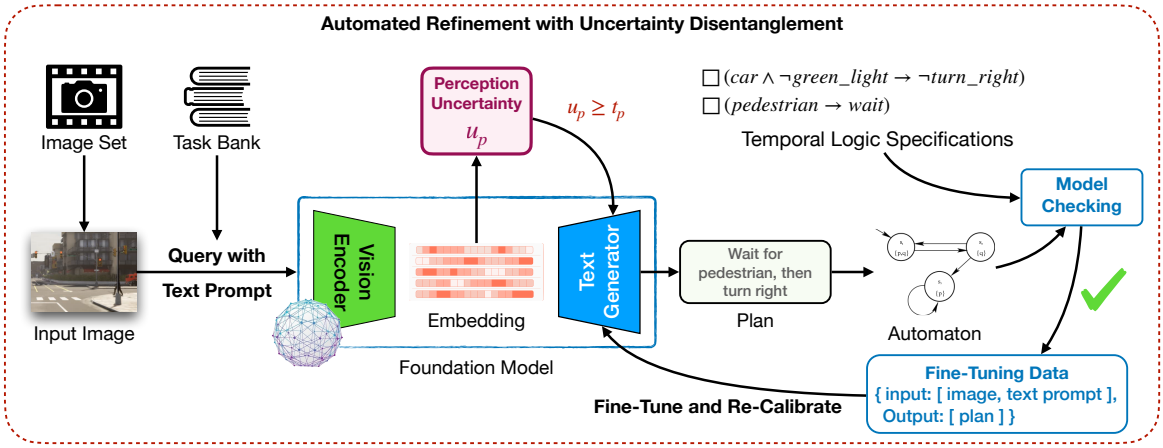

Figure 2: Our automated refinement framework generates high-certainty training data and fine-tunes the foundation model to improve its ability to generate plans that comply with task requirements.

$$\mathbb{P}\left[T : \forall_{\phi \in \Phi}\ \mathbf{pEnc}(T, AP, Y) \models \phi\right] \geq 1 - \epsilon.$$

The left-hand side is our decision uncertainty score u_d according to Def. 2. Therefore, if we find the value of ϵ, we can get the lower bound u_d. In conformal prediction, c^* is the $1-\epsilon$ quantile of S_{nc}. Inversely, $\epsilon = 1 - \int_0^{c^*} f_{nc}(x)dx$. Hence, u_d is bounded by $1 - \epsilon = \int_0^{c^*} f_{nc}(x)dx$. Conservatively, we consider $u_d = \int_0^{c_*} f_{nc}(x)dx$. $\square$

Justification on Separating Decision Uncertainty Formal verification can be computationally expensive and time-consuming when the number of atomic propositions grows. The complexity increases linearly with the number of states in the constructed Kripke structure and exponentially with the number of atomic propositions contained in the specifications (Baier & Katoen, 2008). Therefore, we do not directly verify the Kripke structures during planning. Instead, we obtain the calibration set and the nonconformity distribution offline, then estimate the decision uncertainty online. Such estimation (via Eq. 8) is $O(1)$, which significantly reduces the complexity and turnaround time for planning.

In the planning stage, we estimate a decision uncertainty score u_d and compare it against a predefined *decision threshold* t_d. We only execute the plan if $u_d \geq t_d$. Hence, we can guarantee the probability of safe plan execution in the target environment. We demonstrate the procedure in Fig. 1.

5 From Uncertainty Disentanglement to Intervention

Building on the uncertainty disentanglement, we next present two key interventions, *Active Sensing* and *Automated Refinement*, designed to reduce these uncertainties during model deployment. The active sensing mechanism leverages perception uncertainty scores to dynamically guide re-observations, ensuring propagation of high-confidence visual inputs for subsequent plan generation. Meanwhile, automated refinement systematically fine-tunes the model based on both perception and decision uncertainty, using high-certainty data to enhance the model's ability to generate task-compliant instructions.

Together, these approaches create an iterative cycle, where active sensing feeds high-quality data into the planning module, and automated refinement iteratively improves the model's overall reliability, ultimately leading to a robust planning pipeline that effectively minimizes uncertainty from the perception to the decision-making stage.

5.1 Active Sensing with Perception Uncertainty

We incorporate perception uncertainty scores into an active sensing process, using these scores as indicators to initiate re-observation when necessary. This approach allows assessment of whether the observed image meets a predefined certainty threshold, enhancing the reliability of perception outcomes and the subsequent text generation step.

Figure 1 illustrates the active sensing process. Given an image I and a text description P_T, the vision encoder V first extracts image embeddings. Following the steps in Section 3.1, we obtain a perception uncertainty score u_p. If u_p exceeds a specified *perception threshold* $t_p \in [0, 1]$, we proceed to text generation. Otherwise, the robot is instructed to re-observe the scene by collecting a new image through camera rotation, switching to an alternate camera view, or by recapturing another image after a short delay until the perception uncertainty score rises above t_p. This iterative process continues until the perception uncertainty score surpasses the threshold. This active sensing mechanism ensures that only images with low perception uncertainty are fed into the text generator, reducing the risk of generating

Algorithm 2: Procedure for Automated Refinement

input task bank B_T, image set S_I, foundation model M, atomic propositions set AP, specification set Φ, sample size N_S, perception threshold t_p
output foundation model M
data = [] {Initialize an empty fine-tuning data set}
while True
  $I \sim S_I, P_T \sim B_T$ {Sample I and P_T}
  Obtain perception uncertainty score u_p for I
  if $u_p < t_p$
    continue {Skip high-uncertainty images}
  $T, c = M(P_T, I)$ {Query foundation model}
  Extract set of objects Y from I
  $\mathcal{A} = \textbf{pEnc}(T, AP, Y)$
  if $\forall_{\phi \in \Phi}$ $\textbf{pEnc}(T, AP, Y) \models \phi$
    data.add((I, P_T, T))
    if size(data) $\geq N_S$
      break
Fine-tune M with the collected data

plans that fail due to perceptual inaccuracies.

5.2 Automated Refinement Using Both Uncertainties

However, even with precise visual inputs, foundation models may still produce instructions that fail to meet task specifications due to a lack of domain knowledge. To address this, we introduce an automated refinement process aimed at minimizing the likelihood of generating such specification-violating instructions.

Consider an *image set* containing task environment images and a *task bank* of text-based task descriptions. This data can be collected from high-fidelity simulations or real task executions. The proposed automated refinement procedure is outlined in Algorithm 2. In this process, we filter out images with high perception uncertainty, use the foundation model to generate instructions for each image-task description pair, verify the instructions against task specifications, and add only the valid instructions to the fine-tuning dataset.

The model is then fine-tuned in a supervised manner, where each image-task description pair serves as input, and the instruction is the expected output. If supervised fine-tuning is unavailable, we can generate positive and negative instruction pairs and apply direct preference optimization (DPO) (Rafailov et al., 2024). Positive instructions are those whose automaton representations satisfy all specifications. Figure 2 visually demonstrates this fine-tuning framework. This approach eliminates the need for human annotations or labels, allowing the model to be refined with virtually unlimited data samples until it reaches the desired performance.

Note: The refinement process occurs before grounding the planning pipeline in the task environment. Thus, we apply the FMDP described in Section 4.1 to obtain a nonconformity distribution after refinement, which is then used to estimate decision uncertainty scores during planning.

6 EXPERIMENTS

We perform simulated and real-world autonomous driving experiments to demonstrate **(1)** how we disentangle and quantify perception and decision uncertainty in driving tasks; **(2)** how the proposed active sensing mechanism enhances visual input quality and reduces perceptual errors; and **(3)** how the proposed uncertainty-aware automated fine-tuning framework refines the model to improve its capability to meet task specifications. In our experiments, we use LLaVA-1.5-7B (Liu et al., 2023a) for perception and plan generation; it consists of a CLIP-L-336px vision encoder (Radford et al., 2021) and a Vicuna-7B-v1.5 language generator (Zheng et al., 2023).

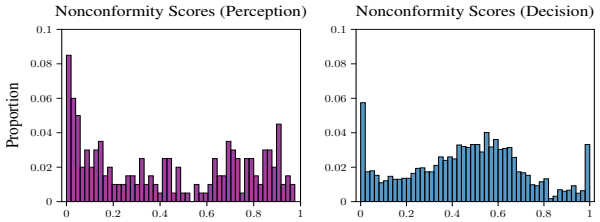

Figure 3: The first and second figures from left depict nonconformity distributions for perception and plan generation in simulated driving scenes (Carla) respectively.

6.1 Uncertainty Disentanglement and Quantification

Perception Uncertainty: We collect 542 images with ground truth labels from the Carla simulator, forming a *calibration set*. Then, we apply CLIP with a pre-trained projection head to classify objects such as stop signs, cars, and pedestrians. We classify a total of 3000 objects within the 542 images and collect their confidence scores. Then, we obtain a nonconformity distribution, as shown in Fig. 3, following the procedure outlined in Sec. 3.1.

Decision Uncertainty: We collect a *calibration set* containing 400 images from the Carla simulator—each associated with a task description—to obtain another nonconformity distribution (Fig. 3) using the procedure in Sec. 4.1 and 4.2.

For each image, we randomly select a task description from a set of tasks {*go straight, turn left, turn right, make a U-turn*} *at the* {*traffic light, stop sign*}. We define 10 safety-critical logical specifications for the driving tasks, some of which are listed below:

$\phi_1 = \Box(\text{red light} \rightarrow \neg\text{move forward})$,

$\phi_2 = \Box(\text{pedestrian} \rightarrow \text{wait})$,

$\phi_3 = \Box(\neg\text{stop sign} \wedge \neg \text{ traffic light } \rightarrow \text{move forward})$,

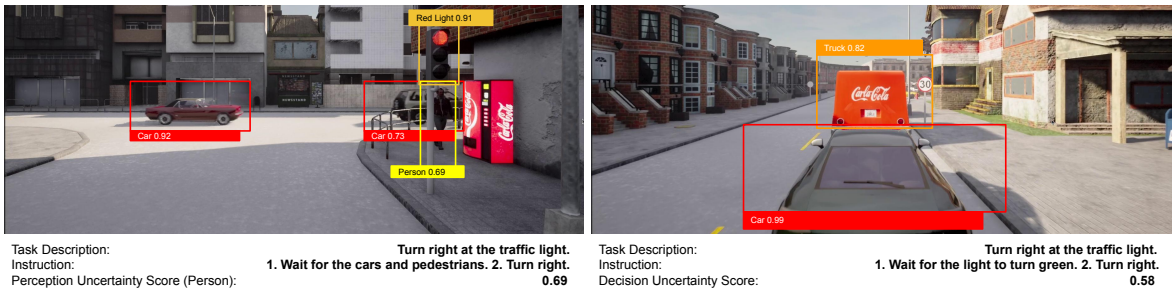

Figure 4: The first figure shows a scenario with high perception uncertainty due to the shadow and occlusion (perception uncertainty scores next to bounding boxes). The second figure shows a scenario with low perception uncertainty but high decision uncertainty, due to the inconsistency between the image and task description (traffic light is absent).

$\phi_4 = \Box(\text{green light} \wedge \neg\text{pedestrian} \rightarrow \bigcirc\neg\text{wait})$,

$\phi_5 = \Box((\text{stop sign} \wedge \neg\text{car} \wedge \neg\text{pedestrian}) \rightarrow \bigcirc\neg\text{wait})$,

where $\Box$, $\Diamond$, and $\bigcirc$ refer to always, eventually, and next, respectively. The detailed task descriptions for the foundation model and the complete set of logical specifications are provided in Appendix C.1 and C.2 respectively.

To avoid scenarios where the generated plans fail to meet the specifications due to language ambiguity or mismatching synonyms, we include in-context examples to constrain the sentence structure and vocabulary of the generated plans.

Quantification: Fig. 3 shows the nonconformity distribution we obtained. We can estimate a probability density function for each distribution. For every object detection and corresponding confidence score, we use Eq. 5 to calibrate the confidence into perception uncertainty scores. We present examples of the calibrated scores in Fig. 4 (next to the bounding boxes) for both types of uncertainty.

For every plan generated by the model, we follow the procedure to prompt the model at the beginning of Sec. 4 and obtain a confidence score and compute the decision uncertainty scores via Eq. 8. We present a scenario with high perception uncertainty and one with high decision uncertainty in Fig. 4. After uncertainty disentanglement and quantification, we can pinpoint the source of high uncertainty and tailor targeted interventions for mitigation.

6.2 Targeted Intervention 1: Active Sensing

We deploy the active sensing mechanism to the planning pipeline that uses a raw foundation model without fine-tuning, as described in Sec. 5.1. We set the threshold $t_p = 0.7$, i.e., a 70% percent probability (and above) of correct visual interpretation is acceptable. We also examine other thresholds and present results in the Appendix C.5.

On the simulated data, we observe a 35% reduction in decision uncertainty. Across a *test set* of 50 driving scenes from our ground robot and NuScenes data (Caesar et al., 2020), the active sensing mechanism reduces perception uncertainty by over 10% and subsequently reduces decision uncertainty by 10%. We present more results in Tab. 2.

We present qualitative results in Fig. 5, illustrating successful plan executions with the active sensing mechanism in simulated and real-world autonomous driving scenes.

6.3 Targeted Intervention 2: Automated Refinement

We investigate a complementary intervention that we cascade on top of active sensing, building upon its benefits. To improve the foundation model's capability to generate plans satisfying specifications, we apply our automated refinement framework to fine-tune the foundation model for autonomous driving tasks using a *train set* containing 800 images from the Carla simulator and the aforementioned 10 pre-defined logical specifications. We follow the steps outlined in Sec. 5.2 to automatically generate refinement data and fine-tune the foundation model's text generator.

Benchmark: To showcase the necessity of our uncertainty disentanglement framework, we select a refinement benchmark (Yang et al., 2024) that fine-tunes the same model using the dataset without uncertainty disentanglement.

This framework works as follows: (1) Create an empty set of fine-tuning data, (2) select an image with a task description and query the foundation model for a plan, (3) build automaton-based representation for the plan and verify it against the specifications, (4) add the plan to the set of fine-tuning data (without uncertainty disentanglement) if it passes all the specifications, (5) repeat step 1-5 until sufficient data is obtained, and (6) freeze the vision encoder and fine-tune the model.

In contrast to the benchmark, our framework first quantifies perception uncertainty and filters out high-uncertainty data, hence enhancing data quality.

Experiment Setting: We fine-tune two identical LLaVA models using our framework and the benchmark and present the training losses in Fig. 6. Our framework offers smoother convergence to a lower loss compared to the benchmark, indicating a higher potential for generating specification-

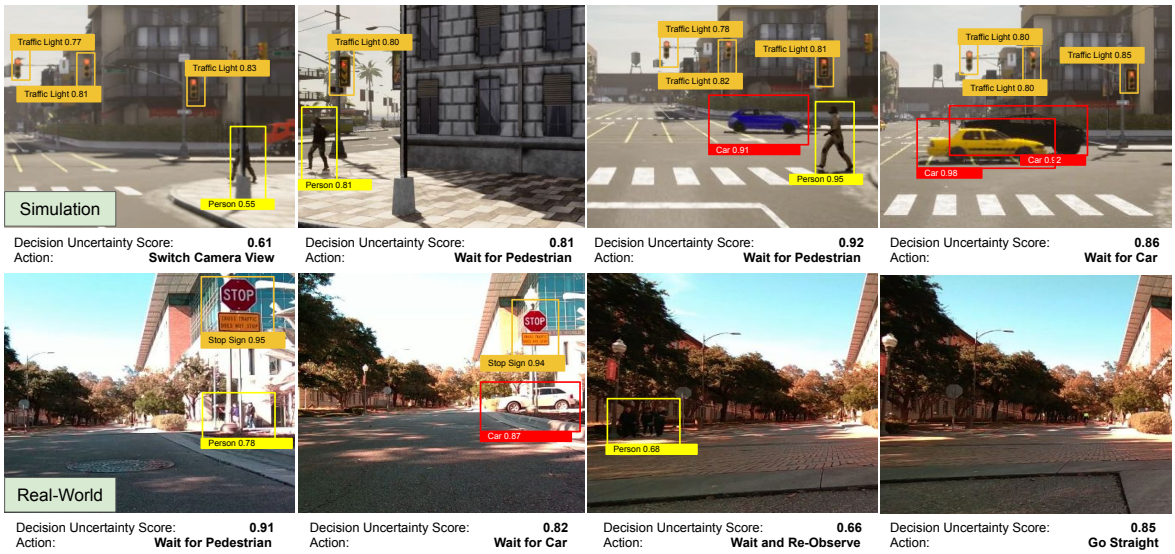

Figure 5: Illustration of autonomous driving tasks wherein our strategy presented in Fig. 1 satisfies all task specifications during plan execution in both simulations and real-world environments.

Table 2: Real-world driving results show that active sensing (AS) and automated refinement improve average perception and decision uncertainty scores in addition to the average and standard deviation of the likelihood of specification compliance. The raw model refers to the pretrained foundation model without any fine-tuning.

Planning Pipeline	Avg. Percp. Unc. Score	Avg. Dec. Unc. Score	Prob. of Satisfying Spec. (Avg)	Prob. of Satisfying Spec. (SD)
Raw Model w/o AS	0.842	0.279	—	—
Raw Model with AS	**0.936**	0.306	0.316	0.180
Fine-tuned Model (Benchmark) with AS	**0.936**	0.931	0.933	0.048
Fine-tuned Model (Ours) with AS	**0.936**	**0.955**	**0.959**	**0.025**

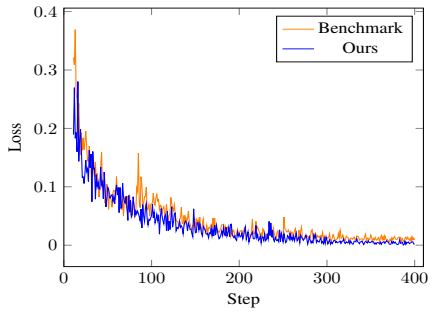

Figure 6: A comparison of the fine-tuning cross-entropy loss obtained using our framework and the benchmark.

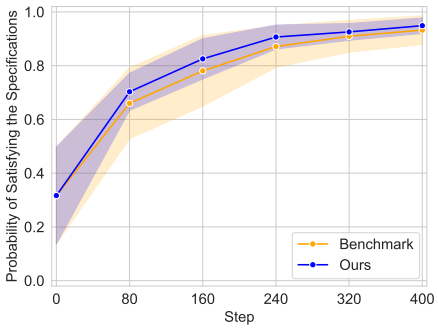

Figure 7: Probability of specification compliance during fine-tuning (color band shows standard deviation (SD)).

compliant plans. During fine-tuning, we save a checkpoint every 80 steps. Next, we use the *test set* with 50 scenes (same as Sec. 6.2) to validate the fine-tuned models. We use each checkpoint to generate plans given those images with task descriptions and record the probability that the plans satisfy the specifications.

Result and Analysis: Fig. 7 and Tab. 2 show the improvement in the model's ability to generate specification-compliant plans.

We observe over 50% improvement in the probability of generated plans satisfying the specifications after fine-tuning. Meanwhile, our framework outperforms the benchmark by improving this probability by up to **5%** while reducing the standard deviation by **40%**. The results indicate en-

Figure 8: Two examples where the foundation model fine-tuned via our refinement framework passes all the specifications while the benchmark fails. The first figure shows a scene with a red light and pedestrians crossing. The second figure shows a green light ahead.

hanced robustness and stability offered by our refinement framework, lowering uncertainty during planning. Uncertain visual observations increase the likelihood of generating plans that violate the specifications. For example, in the first image in Fig 4, uncertainty in observing the occluded pedestrian may result in a plan that overlooks the pedestrian, causing safety violations. Therefore, filtering out high-uncertainty images using our framework improves model performance consistency and scalability.

Sim2Real Transfer: Moreover, we fine-tune the model solely on simulated data and yet achieve over 95% probability of satisfying the specifications on real-world data, indicating an effective Sim2Real transfer of our uncertainty disentanglement framework and interventions. We present theoretical and empirical backing for this Sim2Real transfer capability in Appendix C.3.

Case Study: Fig. 8 shows two examples indicating the superior performance of our refinement framework compared to the benchmark. In the first example, we feed the first image in Fig. 8 along with a task description "turn right at the stop sign" to the foundation model and query for a plan. Note that this description is inconsistent with the image observation, as there is no stop sign in the image. We expect the foundation model to identify such inconsistency.

Prior to refinement, the raw model returns the following:

```
1. Pay attention to the stop sign.
2. Turn right onto the street.
```

The raw model completely ignores the inconsistency and instructs to turn right directly, which violates ϕ_2 in Sec. 6.1 (always wait for pedestrians). This can lead to severe consequences and is a dangerous action to execute.

The fine-tuned model using the benchmark returns:

```
1. There is no stop sign, move forward.
```

Compared to the raw model, it notices the image-description inconsistency. However, the decision to move forward violates ϕ_2 in Sec. 6.1. In contrast, the model fine-tuned using our framework successfully identifies the inconsistency and generates a plan that satisfies all the specifications:

```
1. Wait for the light to turn green.
2. Turn right.
```

In the second example, we feed the second image in Fig. 8 with a task description "go straight at the traffic light" to the foundation model. The model fine-tuned using the benchmark framework returns the following:

```
1. Wait for the light to turn green.
2. Move straight ahead.
```

It does not consider that the light is already green and decides to wait, violating ϕ_4. In contrast, the model fine-tuned using our framework mitigates this issue:

```
1. The traffic light is green.
2. Move forward.
```

In both examples, we demonstrate how our refinement framework outperforms the benchmark.

Decision Uncertainty in Plan Execution: The fine-tuned model does not achieve a 100% specification-satisfaction rate. We employ the decision uncertainty score to determine whether to execute the generated plans, as described at the end of Sec. 4.2. In the 50 scenes, 4 of the generated pans violate the specifications, and all of these plans have decision uncertainty scores below the pre-defined threshold $t_d = 0.7$. Consequently, restricting the execution of plans with uncertainty scores above t_d reduces the probability of specification violation from **8%** to **0%**, underscoring the importance of decision uncertainty during planning.

7 Conclusion

We presented a novel framework for enhancing multimodal foundation models in robotic planning. Our framework disentangles perception uncertainty in visual interpretation and decision uncertainty in plan generation. The proposed quantification methods, leverage conformal prediction for perception and FMDP for decision-making. The targeted interventions improve model robustness by employing active sensing and automated refinement. Empirical results from both real and simulated robotic tasks show that our framework reduces performance variability and improves task success rates, attesting to the value of distinct uncertainty interventions. Future work will extend this approach by considering additional uncertainty types and exploring broader interventions such as task description optimization, paving the way for robust and reliable autonomous systems.

Acknowledgements

This research was supported by the following: DARPA ANSR program under grant RTX CW2231110, DARPA TIAMAT program under grant HR0011-24-9-0431, and the Army Research Lab under grant ARL W911NF-23-S-0001.

REFERENCES

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Angelopoulos, A. N., Bates, S., et al. Conformal prediction: A gentle introduction. *Foundations and Trends® in Machine Learning*, 16(4):494–591, 2023.

Baier, C. and Katoen, J. *Principles of model checking*. MIT Press, 2008. ISBN 978-0-262-02649-9.

Balasubramanian, V., Ho, S.-S., and Vovk, V. *Conformal Prediction for Reliable Machine Learning: Theory, Adaptations and Applications*. Newnes, 2014.

Bhatt, N. P., Khajepour, A., and Hashemi, E. Mpc-pf: Socially and spatially aware object trajectory prediction for autonomous driving systems using potential fields. *IEEE Transactions on Intelligent Transportation Systems*, 24 (5):5351–5361, 2023.

Browne, M. C., Clarke, E. M., and Grümberg, O. Characterizing finite kripke structures in propositional temporal logic. *Theoretical computer science*, 59(1-2):115–131, 1988.

Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., and Beijbom, O. nuscenes: A multimodal dataset for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11618–11628, Seattle, WA, USA, 2020. Computer Vision Foundation / IEEE.

Cimatti, A., Clarke, E. M., Giunchiglia, E., Giunchiglia, F., Pistore, M., Roveri, M., Sebastiani, R., and Tacchella, A. NuSMV 2: An opensource tool for symbolic model checking. In Brinksma, E. and Larsen, K. G. (eds.), *Computer Aided Verification*, volume 2404 of *Lecture Notes in Computer Science*, pp. 359–364, Copenhagen, Denmark, 2002. Springer.

Ding, Z., Han, X., Liu, P., and Niethammer, M. Local temperature scaling for probability calibration. In *IEEE/CVF International Conference on Computer Vision*, pp. 6869–6879, Montreal, QC, Canada, 2021. IEEE. doi: 10.1109/ICCV48922.2021.00681. URL https://doi.org/10.1109/ICCV48922.2021.00681.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017.

Gnanadesikan, R. and Wilk, M. B. Probability plotting methods for the analysis of data. *Biometrika*, 55(1):1–17, 1968.

Gu, Q., Kuwajerwala, A., Morin, S., Jatavallabhula, K. M., Sen, B., Agarwal, A., Rivera, C., Paul, W., Ellis, K., Chellappa, R., et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. *arXiv preprint arXiv:2309.16650*, 2023.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Hori, K., Suzuki, K., and Ogata, T. Interactively robot action planning with uncertainty analysis and active questioning by large language model. *arXiv preprint arXiv:2308.15684*, 2023.

Ichter, B., Brohan, A., Chebotar, Y., Finn, C., Hausman, K., Herzog, A., Ho, D., Ibarz, J., Irpan, A., Jang, E., Julian, R., Kalashnikov, D., Levine, S., Lu, Y., Parada, C., Rao, K., Sermanet, P., Toshev, A., Vanhoucke, V., Xia, F., Xiao, T., Xu, P., Yan, M., Brown, N., Ahn, M., Cortes, O., Sievers, N., Tan, C., Xu, S., Reyes, D., Rettinghouse, J., Quiambao, J., Pastor, P., Luu, L., Lee, K., Kuang, Y., Jesmonth, S., Joshi, N. J., Jeffrey, K., Ruano, R. J., Hsu, J., Gopalakrishnan, K., David, B., Zeng, A., and Fu, C. K. Do as I can, not as I say: Grounding language in robotic affordances. In Liu, K., Kulic, D., and Ichnowski, J. (eds.), *Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 287–318. PMLR, 2022.

Jang, I., Kim, S., and Noh, S. Online fine-tuning with uncertainty quantification for offline pre-trained agents. In *NeurIPS 2024 Workshop on Fine-Tuning in Modern Machine Learning: Principles and Scalability*, 2024.

Jiang, Z., Araki, J., Ding, H., and Neubig, G. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977, 2021.

Kamath, A. and Das, R. A survey on semantic parsing, 2018.

Liu, H., Li, C., Li, Y., and Lee, Y. J. Improved baselines with visual instruction tuning, 2023a.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. In *NeurIPS*, 2023b.

Malinin, A., Ragni, A., Knill, K., and Gales, M. Incorporating uncertainty into deep learning for spoken language assessment. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 45–50, 2017.

Mu, Y., Zhang, Q., Hu, M., Wang, W., Ding, M., Jin, J., Wang, B., Dai, J., Qiao, Y., and Luo, P. Embodiedgpt: Vision-language pre-training via embodied chain of thought. *arXiv preprint arXiv:2305.15021*, 2023.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

Ren, A. Z., Dixit, A., Bodrova, A., Singh, S., Tu, S., Brown, N., Xu, P., Takayama, L., Xia, F., Varley, J., Xu, Z., Sadigh, D., Zeng, A., and Majumdar, A. Robots that ask for help: Uncertainty alignment for large language model planners. In *arXiv preprint arXiv:2307.01928*, 2023.

Rudner, T. G., Pan, X., Li, Y. L., Shwartz-Ziv, R., and Wilson, A. G. Fine-tuning with uncertainty-aware priors makes vision and language foundation models more reliable. In *ICML 2024 Workshop on Structured Probabilistic Inference {\&} Generative Modeling*, 2024.

Shah, D., Osinski, B., Ichter, B., and Levine, S. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In Liu, K., Kulic, D., and Ichnowski, J. (eds.), *Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 492–504. PMLR, 2022.

Shen, Y., Zhang, Z., Sabuncu, M. R., and Sun, L. Real-time uncertainty estimation in computer vision via uncertainty-aware distribution distillation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 707–716, 2021.

Song, C. H., Sadler, B. M., Wu, J., Chao, W., Washington, C., and Su, Y. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *IEEE/CVF International Conference on Computer Vision*, pp. 2986–2997. IEEE, 2023.

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

Vazhentsev, A., Kuzmin, G., Shelmanov, A., Tsvigun, A., Tsymbalov, E., Fedyanin, K., Panov, M., Panchenko, A., Gusev, G., Burtsev, M., et al. Uncertainty estimation of transformer predictions for misclassification detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8237–8252, 2022.

Vovk, V., Gammerman, A., and Shafer, G. *Algorithmic Learning in a Random World*, volume 29. Springer, 2005.

Wang, K., Li, J., Bhatt, N. P., Xi, Y., Liu, Q., Topcu, U., and Wang, Z. On the planning abilities of openai's o1 models: Feasibility, optimality, and generalizability. *arXiv preprint arXiv:2409.19924*, 2024.

Wu, Z., Wang, Z., Xu, X., Lu, J., and Yan, H. Embodied task planning with large language models. *arXiv preprint arXiv:2307.01848*, 2023.

Yang, Y., Gaglione, J.-R., Neary, C., and Topcu, U. Automaton-based representations of task knowledge from generative language models. *arXiv preprint arXiv:2212.01944*, 2022.

Yang, Y., Neary, C., and Topcu, U. Multimodal pretrained models for verifiable sequential decision-making: Planning, grounding, and perception. In *International Conference on Autonomous Agents and Multiagent Systems*, pp. 2011–2019, Auckland, New Zealand, 2023.

Yang, Y., Bhatt, N. P., Ingebrand, T., Ward, W., Carr, S., Wang, A., and Topcu, U. Fine-tuning language models using formal methods feedback: A use case in autonomous systems. In Gibbons, P. B., Pekhimenko, G., and Sa, C. D. (eds.), *Proceedings of the Seventh Annual Conference on Machine Learning and Systems*, Santa Clara, CA, USA, 2024. mlsys.org.

Ye, W., Ma, Y., and Cao, X. Uncertainty estimation in deterministic vision transformer, 2023.

Yu, Y., Kong, L., Zhang, J., Zhang, R., and Zhang, C. Actune: Uncertainty-based active self-training for active fine-tuning of pretrained language models. In *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 1422–1436, 2022.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36: 46595–46623, 2023.