SCOUT: A Lightweight Framework for Scenario Coverage Assessment in Autonomous Driving

Anil Yildiz*,†, Sarah M. Thornton†, Carl Hildebrandt†, Sreeja Roy-Singh† and Mykel J. Kochenderfer*

***Abstract*— Assessing scenario coverage is crucial for evaluating the robustness of autonomous agents, yet existing methods rely on expensive human annotations or computationally intensive Large Vision-Language Models (LVLMs). These approaches are impractical for large-scale deployment due to cost and efficiency constraints. To address these shortcomings, we propose SCOUT (Scenario Coverage Oversight and Understanding Tool), a lightweight surrogate model designed to predict scenario coverage labels directly from an agent's latent sensor representations. SCOUT is trained through a distillation process, learning to approximate LVLM-generated coverage labels while eliminating the need for continuous LVLM inference or human annotation. By leveraging precomputed perception features, SCOUT avoids redundant computations and enables fast, scalable scenario coverage estimation. We evaluate our method across a large dateset of real-life autonomous navigation scenarios, demonstrating that it maintains high accuracy while significantly reducing computational cost. Our results show that SCOUT provides an effective and practical alternative for large-scale coverage analysis. While its performance depends on the quality of LVLM-generated training labels, SCOUT represents a major step toward efficient scenario coverage oversight in autonomous systems.**

I. Introduction

Ensuring comprehensive *scenario coverage* is a fundamental challenge in evaluating the robustness and reliability of autonomous agents. Coverage analysis determines whether an agent has encountered a sufficient diversity of critical situations, particularly those involving potential failure modes or rare edge cases. However, existing methods for scenario coverage assessment typically rely on human-annotated data or high-fidelity simulation environments [1], both of which are expensive and infeasible to scale. The lack of a scalable, lightweight approach to coverage estimation hinders progress in evaluating empirical safety metrics for high-risk autonomous robotics applications [2].

This problem is particularly important in safety-critical applications such as autonomous driving, robotics, and embodied AI, where failures in underexplored scenarios can lead to catastrophic consequences [3]. Without an efficient way to oversee coverage distribution, agents risk encountering novel, high-risk situations in real-world deployments without sufficient prior exposure during training or validation. A scalable coverage estimation framework would enable proactive failure mitigation, targeted policy refinements, and improved robustness across a broader set of deployment conditions. Additionally, effective scenario coverage estimation is crucial for both public trust and regulatory compliance, as agencies increasingly require systematic validation of autonomous systems before deployment [4].

Traditional methods for scenario coverage estimation are computationally demanding and require heuristics and extensive human intervention [5]. Supervised learning approaches depend on costly human-annotated datasets, which are limited in scope and difficult to scale. Large Vision-Language Models (LVLMs) [6], [7] have emerged as a potential solution, offering automated labeling capabilities, but they are prohibitively expensive to run at large scales. Furthermore, direct inference from raw sensor observations is computationally inefficient and redundant, as most perception stacks already compute feature-space representations for downstream tasks. A naive approach relying solely on human or LVLM-based annotation is therefore impractical due to cost, scalability, and efficiency constraints.

Existing solutions fail to fully address the challenges of scalable and accurate scenario coverage estimation, including the high cost of human annotation, the computational expense of LVLM-based labeling, and the inefficiencies of processing raw sensor observations. While LVLMs can augment coverage labels, their computational cost makes them unsuitable for real-time or large-scale use. On the other hand, handcrafted metrics and heuristic-based methods lack the generalizability and adaptability required for diverse and evolving deployment environments. Prior work has not effectively leveraged the precomputed latent representations from perception pipelines, missing an opportunity to efficiently infer coverage without redundant computation. To overcome these shortcomings, a lightweight, scalable alternative is needed, one that avoids direct reliance on human annotation or LVLM inference while still maintaining high coverage labeling accuracy.

To this end, we introduce SCOUT (Scenario Coverage Oversight and Understanding Tool), a surrogate model that efficiently predicts scenario coverage labels using the latent feature representations already computed in an agent's perception stack. As illustrated in Fig. 1, we first fine-tune an LVLM on a small, human-labeled subset to generate reliable coverage labels at a larger scale. SCOUT is then distilled from this LVLM, learning to reproduce coverage labels directly from the agent's latent sensor features. Once trained, SCOUT eliminates the need for both human-annotated and LVLM-generated labels, enabling low-cost, high-speed scenario coverage estimation. By leveraging precomputed sensor representations, SCOUT ensures minimal computational overhead while maintaining high accuracy. However, as a

*Department of Aeronautics and Astronautics, Stanford University, 496 Lomita Mall, Stanford, CA 94305, {yildiz, mykel}@stanford.edu

†Nuro Inc., 1300 Terra Bella Ave, Mountain View, CA 94043, {ayildiz, sthornton, childebrandt, sreeja}@nuro.ai

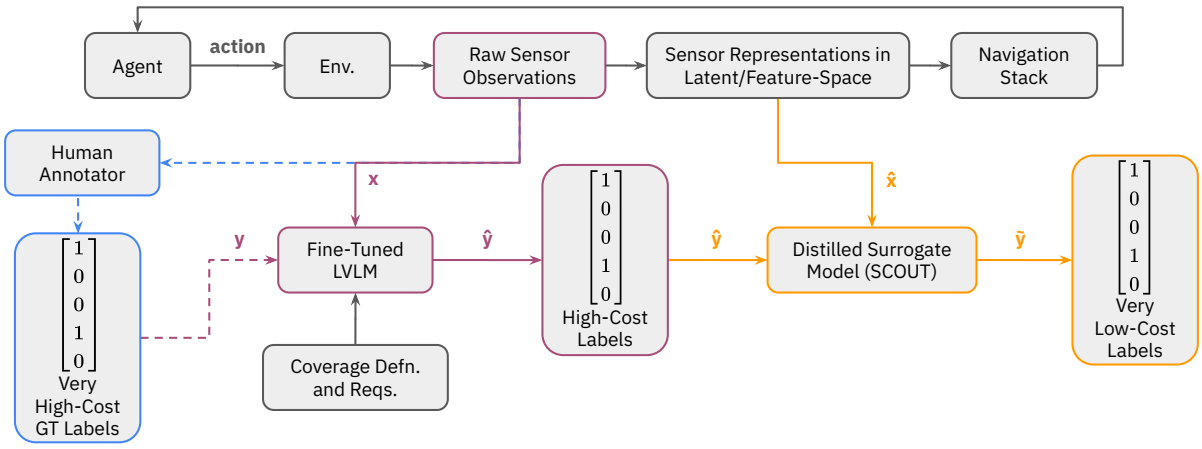

Fig. 1. Overview of the scenario coverage pipeline. The distilled surrogate model, SCOUT, predicts scenario coverage labels using precomputed sensor latent representations, which are inherently consumed by the agent's navigation stack. Due to the high costs incurred, only a small subset of data is annotated by humans to obtain ground-truth labels. To scale the labeling process, an LVLM is fine-tuned and later used to generate labels for a larger dataset, augmenting the training data. SCOUT, trained as a distilled surrogate model, learns to replicate the LVLM's labeling process, thereby enabling lightweight and scalable coverage estimation.

surrogate model, its accuracy depends on the quality and diversity of the LVLM-generated labels it is trained on, which may introduce biases if not properly managed. Despite this, SCOUT represents a significant step toward scalable and efficient scenario coverage oversight, making it a practical solution for large-scale autonomous system evaluation.

II. Related Work

Existing work on scenario coverage can be divided into the following four main categories.

a) Scenario-Based Test Coverage: A growing body of work focuses on formalizing and quantifying coverage for autonomous systems. PhysCov [8] introduces a physical test coverage metric by combining vehicle dynamics and sensor inputs to estimate the region of influence during test drives. GUARD [9] proposes a scalable, probabilistic approach to partition scenario parameter spaces without discretization, using Gaussian Processes and level set estimation. Similarly, parameter coverage [10] has been introduced to ensure that decision-relevant variables in autonomous driving systems are exercised across diverse simulations. These efforts highlight the growing importance of measuring test sufficiency for safety validation, but they often depend on simulation-specific abstractions or explicit environmental modeling, limiting their generalizability across platforms.

b) Surrogate Models for Coverage: Surrogate modeling has emerged as a powerful tool for reducing the computational cost of evaluating expensive environments. Deep surrogate-assisted methods like DSAGE [11] train learned predictors of agent behavior for efficient environment generation. Other works leverage Bayesian optimization with adaptive surrogate models [12] or employ neural and Gaussian process-based surrogates for optimizing robot swarm behaviors [13]. RI-SHM [14] introduces a surrogate model for mixed-variable coverage optimization problems. Our study, SCOUT, builds on these ideas, but rather than optimizing control or planning directly, it distills LVLM-labeled coverage indicators into a lightweight predictor that consumes precomputed sensor features, bridging perception and testing in a scalable way.

c) Classification Based Coverage: Several works have tackled the structure and classification of scenario spaces. Tree-based scenario classifications [15], [16] introduce a logic-based approach to categorize scenarios over time, enabling systematic analysis of scenario coverage. In parallel, coverage strategies in real-world environments have been studied through multi-agent systems [17], optimizing urban surveillance using UAVs under sensing and motion constraints. Autonomous monitoring approaches have also been deployed on embedded platforms for specialized detection tasks such as reckless driving [18]. While these approaches advance the field of scenario understanding and deployment practicality, SCOUT focuses specifically on labeling coverage itself from latent representations, offering a complementary capability that can plug into broader testing, classification, or deployment pipelines.

d) Language Model Driven Coverage: Recent work has explored the use of Large Language Models (LLMs) for coverage estimation and anomaly detection. LogGPT [6] uses an LLM to classify structured system logs via prompt-based reasoning. CSAM [7] integrates an LLM-inspired attention module into an object detection pipeline to improve anomaly detection in cluttered scenes. CoverUp [19] prompts LLMs to generate high-coverage regression tests by incorporating code and coverage gaps. These efforts demonstrate the growing role of language models in structured reasoning and coverage analysis. SCOUT builds on this direction, distilling LVLM outputs into a lightweight surrogate for scalable, real-time coverage estimation.

III. Scenario Coverage in Real-World Driving

Ensuring *scenario coverage* is paramount in evaluating and improving the reliability of autonomous systems deployed

in real-world environments. Coverage refers to the degree to which a dataset encompasses all of the possible environmental conditions, behaviors, and hazards in the system's operational design domain. In safety-critical contexts, insufficient coverage reduces confidence that the system can safely handle encounters with novel or underrepresented situations on public roads or other complex domains.

A. Safety-Critical Relevance

By training and evaluating a system against diverse operational conditions, including rare and hazardous events, confidence that the system will safely handle hazardous situations during deployment is improved. However, it is usually impractical to immerse the agent to every possible event. However, it is usually impractical or unscalable to immerse the agent to *every* unsafe event. Therefore, lightweight and reliable tools are needed to oversee and verify scenario coverage.

B. Definition and Scope of Coverage

At a high level, coverage can be defined as the *breadth* and *depth* of scenarios that a system has experienced or been validated against. This includes:

- High-frequency events, such as straightforward lane-keeping or car-following maneuvers.
- Low-frequency but high-severity events, often termed *edge cases*, such as sudden pedestrian intrusions or multi-vehicle collisions on busy highways.
- Environmental variations, including weather, lighting, and road types.
- Behavioral interactions between vehicles, cyclists, and pedestrians.

C. Conflicts in the SHRP2 Naturalistic Driving Study

An illustrative framework for understanding the kinds of scenarios critical to coverage analysis comes from the second Strategic Highway Research Program (SHRP2) [20], [21]. SHRP2 identifies *conflicts* as events or situations in which:

- A driver or automated system faces an *elevated risk* of collision, road departure, or other hazard.
- There is a *noticeable interaction* or potential interaction between traffic participants (e.g., a close following distance or crossing paths at an intersection).

During the study, vehicles were instrumented to capture detailed driver behavior and roadway conditions, observing a wide range of conflict scenarios. SHRP2 introduced a taxonomy of possible conflicts, such as *run-off-road incidents*, *rear-end near-collisions*, *turning across or into traffic*, and *head-on approaches*, each tied to specific driving contexts and geometries. Through extensive data collection, SHRP2 documented patterns of driver response, near-crash indicators, and collision avoidance behaviors. This categorization has made a formal definition of the scope of scenario coverage for onroad vechicles.

Many conflict types in the SHRP2 taxonomy correspond to rare but critical scenarios. Consequently, driving datasets often exhibit inherent class imbalance, reflecting true event frequency rather than collection bias.

TABLE I
CRASH TYPOLOGY DEFINITIONS [21] FOR DRIVING COVERAGE.

Label	Description	Count
Group I. Single Driver		
A	Right roadside departure	5
B	Left roadside departure	5
C	Forward impact	6
Group II. Same Trafficway, Same Direction		
D	Rear end	17
E	Forward impact	10
F	Angle/sideswipe	6
Group III. Same Trafficway, Opposite Direction		
G	Head-on	4
H	Forward impact	10
I	Angle/sideswipe	4
Group IV. Change Trafficway, Vehicle Turning		
J	Turn across path	8
K	Turn into path	10
Group V. Intersect Paths		
L	Straight paths	6
Group VI. Misc		
M	Backing, etc.	5

D. Implications for Autonomous Vehicles

The SHRP2 conflict taxonomy was originally developed to study human driver behavior and roadway safety in naturalistic settings. The taxonomy, consisting of a total of 95 conflicts, is summarized in Table I, where its primary goal is to capture and categorize critical traffic interactions that contribute to crash risk in everyday driving. Example descriptions from Table I are depicted in Fig. 2. The upper example in Fig. 2, "*traction loss*", is one of the 5 descriptions that fall under Group I.B. The lower example, *"avoid collision with object"*, is one of the 10 descriptions that fall under Group II.E. E.g. The driver in the latter case makes a maneuver to avoid a collision with an onroad object that results in a forward impact with another vehicle in the same trafficway and direction.

In this work, we extend the use case of SHRP2 to the domain of autonomous vehicles (AVs) by leveraging its well-defined categories as a structured protocol for evaluating scenario coverage. By mapping AV behavior against this taxonomy, we assess whether autonomous systems are exposed to a sufficiently diverse set of realistic and safety-critical driving scenarios, grounded in empirical observations from human driving behavior.

To facilitate this mapping, we implement a human-supervised annotation process depicted in Fig. 3. Raw forward-facing camera footage is first segmented into shorter clips (i.e. *scenes*) if an *interesting event* (e.g. includes an interaction that may correspond to one or more SHRP2-defined conflicts) occurs within the raw footage. These scene instances, typically lasting around 10 seconds each, are detected using other onboard sensor data (e.g. pose, distance to collision, deceleration). Scenes are then passed to human

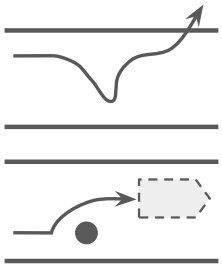

Fig. 2. Depictions of example conflicts [20].

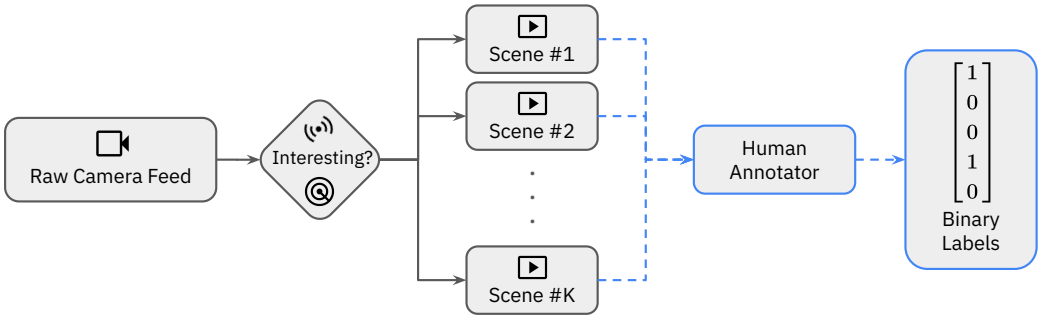

Fig. 3. Extraction pipeline of scenes including conflict(s). Raw camera recordings are split the into smaller scenes (∼10 seconds) if they include an *interesting* interaction. A human then annotates them with respect to their context.

annotators, who label them with binary indicators reflecting the presence or absence of conflict categories (Table I). In this pipeline, expert-designed heuristic-based tools that rely on onboard sensor data may also be leveraged to speed up the manual annotation process.

This annotation process provides high-quality coverage labels that serve as ground truth for our initial training phase of SCOUT. In doing so, we effectively translate the SHRP2 taxonomy into a practical tool for quantifying AV scenario coverage, allowing us to assess which types of high-risk interactions are underrepresented in a given driving dataset.

IV. Training Schemas of LLMs and LVLMs

Large language models (LLMs) have transformed natural language processing by enabling sophisticated understanding, generation, and reasoning. We denote an LLM by p_Φ, which processes a tokenized input sequence $x = \big(x^{(1)}, x^{(2)}, \ldots, x^{(n)}\big)$ and models its probability via the factorization

$$\begin{aligned} p_\Phi(x) &= p_\Phi\big(x^{(1)}, x^{(2)}, \ldots, x^{(n)}\big) \\ &= \prod_{i=1}^{n} p_\Phi\big(x^{(i)} \mid x^{(1)}, \ldots, x^{(i-1)}\big) \end{aligned} \tag{1}$$

to capture the likelihood of each token $x^{(i)}$ given all preceding tokens in the sequence.

Language models can be extended to include visual embeddings as an additional input modality, giving rise to large vision-language models (LVLMs). In an LVLM, the sequence x may contain tokens from multiple modalities or can include feature tokens extracted from visual inputs (e.g., images). Although the overall modeling framework in Eq. (1) remains the same, the LVLM incorporates auxiliary components to encode and fuse these multimodal features.

Formally, an LVLM comprises modality encoders $E^{(k)}$, a multimodal fusion module F, a language model f, and an output projector P. Given encoded representations from each modality, the fused embedding is passed through the language model to produce the next-token distribution. At each step i, we can write

$$\begin{aligned} p_\Phi(x) &= p_\Phi\big(x^{(1)}, \ldots, x^{(n)}\big) \\ &= P\Big\{ f\Big[F\big(E^{(1)}(x^{(1)}), \ldots, E^{(n)}(x^{(n)})\big)\Big]\Big\} \end{aligned} \tag{2}$$

where $x^{(k)}$ denotes the input features (e.g., image embeddings, text tokens) from modality k. The fusion module $F(\cdot)$ integrates these modality-specific representations into a unified embedding, $f(\cdot)$ refines it in the language modeling space, and $P(\cdot)$ maps the resulting latent representation to a probability distribution over the next token.

A. Pre-training

Pre-training is crucial for enabling LLMs and LVLMs to learn broad linguistic and multimodal representations from large-scale data. For the LLM setting, let $\{x_i\}_{i=1}^{N}$ be a collection of unlabeled training sequences, where each sequence $x_i = \big(x_i^{(1)}, \ldots, x_i^{(T_i)}\big)$ has length T_i. We denote the model parameters by Φ. The goal of pre-training is to maximize the log-likelihood of each token given its preceding context:

$$\Phi^* = \arg\max_{\Phi} \sum_{i=1}^{N} \sum_{j=1}^{T_i} \log\, p_\Phi\big(x_i^{(j)} \mid x_i^{(j-c)}, \ldots, x_i^{(j-1)}\big) \tag{3}$$

where c is the context window size that determines how many previous tokens inform the prediction at each position j. This formulation also extends naturally to multimodal sequences in LVLMs, where some tokens $x_i^{(j)}$ can represent visual or other modality-specific embeddings where E, F, f, and P are used as shown in Eq. (2).

B. Fine-tuning

Fine-tuning adapts a pre-trained model to specific tasks or domains. Previous work explores various techniques, such as prompt tuning [22], instruction tuning [23], reinforcement learning from human feedback (RLHF) [24], and LoRA (low-rank adaptation) [25]. In our study, we employ LoRA, which freezes the original model weights and introduces small, trainable matrices $\mathbf{A}_l \in \mathbb{R}^{d\times r}$ and $\mathbf{B}_l \in \mathbb{R}^{r\times d}$ into each Transformer [26] layer l, where $r \ll d$. The adapted weights are computed as

$$\mathbf{W}_l' = \mathbf{W}_l + \mathbf{A}_l\,\mathbf{B}_l, \tag{4}$$

where $\mathbf{W}_l \in \mathbb{R}^{d\times d}$ are the original (frozen) parameters. LoRA thereby focuses the training on a small subset of parameters $\mathbf{A}_l$ and $\mathbf{B}_l$, which is especially beneficial when fine-tuning large models on specialized tasks.

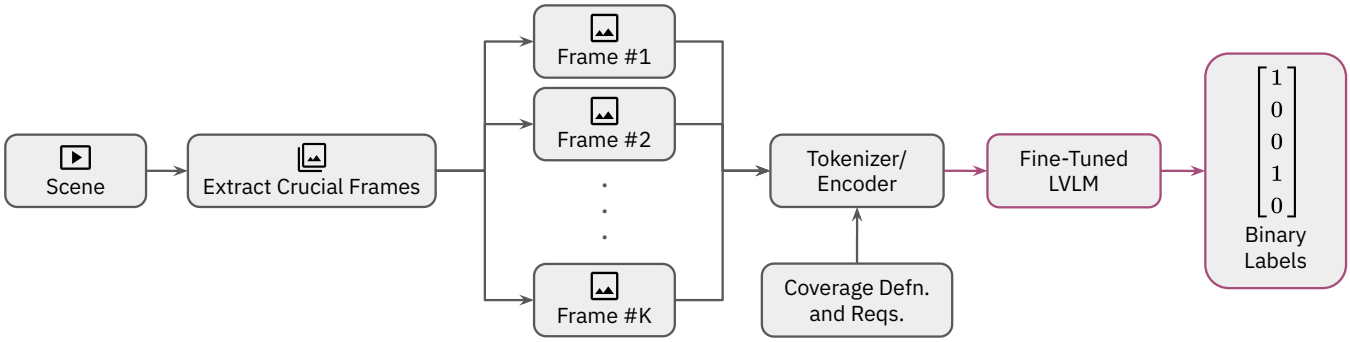

Fig. 4. Overview of the scenario coverage label generation pipeline. A driving scene is first processed to extract visually informative frames. Each frame is then encoded, alongside the tokenized/encoded scenario coverage information, and passed through the LVLM. The output is a binary label for each conflict definition, whether they exist within the scene or not.

V. Scenario Coverage Oversight and Understanding Tool

Figure 1 illustrates the overall pipeline to determine scenario coverage. The process begins with an autonomous agent interacting with its environment through an action and collecting raw sensor observations (LiDAR, radar, camera, etc.). These observations are then transformed into a latent feature representation, which is a standard component of modern navigation stacks. Rather than directly processing raw data, Scenario Coverage Oversight and Understanding Tool (SCOUT) leverages these precomputed representations, ensuring computational efficiency. Training SCOUT is a two-step process.

1) Fine-tuning an LVLM to generate a larger training dataset from a small amount of human-labeled (or heuristic based automated labels) ground truth data.
2) Using the upscaled dataset, train a distilled surrogate model to determine coverage properties for a different input modality, sensor latent representations, rather than their expensive raw versions.

A. Fine-Tuning our LVLM using Human Annotations

Our primary objective is to train SCOUT, the distilled surrogate model. However, due to the small size of human annotated data available, we first need to augment high-quality but low-cost data. To do so, we use a pre-trained large vision-language model (LVLM) to that takes in the same modality x as human annotators do, and automatically annotates scenes in a similar fashion. Human labeled data y is then used to fine-tune the LVLM to improve its prediction accuracy. It is important to note that using the LVLM to generate new labels on unseen scenes is still an expensive operation. However, it is cheaper compared to manual hand labels, and thus, is a reasonable alternative to enlarge the data for downstream training.

Figure 4 outlines the pipeline used to generate scenario coverage labels using an LVLM. The pipeline begins by consuming a scene that was clipped earlier, as shown in Fig. 3. From this scene, we extract a small number of *crucial frames* (see [27], [28]) which are key snapshots that visually capture the main essence of the interactions or conflicts present in the scene. Each of these extracted frames was then passed through an encoder, which transforms the visual inputs into a format suitable for downstream processing by the LVLM.

Simultaneously, coverage definitions and requirements, as described in Table I, are tokenized and encoded to be passed as a prompt to the LVLM. We prompt the model to return a `Yes` or a `No` for each conflict definition, which we convert to a binary vector labels $\hat{y}$ afterwards, one for each coverage category predefined in Table I. These labels serve as labels for training the lightweight surrogate model (SCOUT) downstream.

To fine-tune the LVLM, we use LoRA, where the target predictions during training are `Yes` or `No`, given the consumed ground truth labels provided by a human annotator (Fig. 3).

B. Training a Distilled Surrogate Model

Once a sufficient volume of scene-level labels has been generated via the fine-tuned LVLM, we proceed to train SCOUT, our distilled surrogate model. The objective of SCOUT is to replicate the LVLM's labeling capability while operating on a more efficient input modality: precomputed sensor latent representations. These latent features, already computed as part of the agent's navigation stack, are typically available in real-world autonomy pipelines and avoid the need to reprocess raw sensor observations.

We denote the input to SCOUT as $\hat{x}$, which corresponds to the latent sensor representation for a given driving scene. SCOUT is trained to predict a binary coverage label vector $\tilde{y}$ that approximates the high-cost label $\hat{y}$ produced by the LVLM.

Training SCOUT requires no additional human annotations or LVLM inference once the distillation process is complete. This makes SCOUT an ideal tool for continuous monitoring of scenario coverage during policy evaluation, dataset curation, or simulation-based testing. While SCOUT's predictions are inherently bounded by the accuracy of the LVLM it distills, we find that the model maintains high fidelity under realistic conditions, enabling scalable coverage oversight without sacrificing semantic resolution.

VI. Experiments and Results

A. Dataset, Annotation and Class (Im-)Balance

We evaluate on a real-world driving corpus of 90,000 scenes (5–15 seconds each) collected from an operational fleet of autonomous vehicles. A random 10,000 scenes were labeled by an expert-designed majorly-automated annotator, producing binary indicators for the 68 conflict types defined by the SHRP2 taxonomy (for Groups II–V in Table I). We treat these labels as the human-labelled ground truth for this study; the same pipeline would remain unchanged if fully human annotations were substituted. The remaining 80,000 scenes were labelled by our fine-tuned LVLM.

In the entire dataset, 45 out of the 68 conflict categories exhibit a class imbalance worse than 70/30, which is a distribution intentionally preserved to reflect real-world long-tail event scenarios. Our dataset is split as follows.

- **Human-labelled split**: 8,000 train / 2,000 test (used *only* for LVLM training).
- **LVLM-labelled split**: 56,000 train / 12,000 val / 12,000 test (used for SCOUT and baseline training).

Every scene in the dataset is processed to extract approximately 5 to 8 keyframes (Fig. 4) using Katna [29]. Latent sensor representations are obtained from the agent's perception stack.

B. Model and Training Details

a) LVLM (Teacher): The pre-trained LVLM used for labeling is based on Gemma-3-12B [30]. The model is fine-tuned through LoRA [25] on SHRP2-aligned coverage definitions (Table I) to generate multi-label binary outputs for each scene. Unsloth [31] is used to reduce memory usage during fine-tuning.

b) SCOUT (Distilled Surrogate Model): SCOUT is a residual [32] fully connected neural network (Residual FCNN) with a cross-self-attention mechanism [26]. The model processes a sequence of latent embeddings and an attention mask using a multi-head cross-attention layer, followed by mean pooling. The pooled vector passes through three residual blocks, then a projection layer with batch normalization and dropout, before generating multi-label predictions via a sigmoid output. This design balances semantic expressivity and efficiency, supporting broad generalization with low inference cost.

SCOUT is trained using binary cross-entropy loss over the LVLM-generated labels:

$$\mathcal{L}_{\text{BCE}} = -\sum_{i=1}^{G}\sum_{j=1}^{C}\left(y_{ij}\log\hat{y}_{ij} + (1-y_{ij})\log(1-\hat{y}_{ij})\right) \tag{5}$$

where G and C are the amounts of coverage categories, and counts in each category, respectively.

C. Evaluating LVLM Agreement with Human Labels

To validate the quality of our fine-tuned LVLM, we compare its scenario coverage predictions against a manually annotated set of 2,000 scenes labeled by domain experts. This benchmark allows us to assess how well the LVLM aligns with human understanding of the SHRP2 conflict taxonomy.

We compute the following evaluation metrics:

- **Precision, Recall, and F1 Score** for each coverage category.
- **Macro-Averaged F1 Score**, reflecting balanced performance across all categories, regardless of class imbalance.
- **Exact Match Rate**, the proportion of scenes where the LVLM's predicted label vector exactly matches the human annotation.
- **Per-label Agreement Rate**, measuring the percentage of individual category labels that match human annotations across all scenes.

TABLE II

Evaluation of LVLM-predicted scenario coverage labels against human annotations.

Categories (accumulated)	Precision	Recall	F1 Score
Group II. Same TW, Same Dir.	0.91	0.88	0.89
Group III. Same TW, Opp. Dir.	0.85	0.79	0.82
Group IV. Change TW, Veh. Turn.	0.87	0.75	0.80
Group V. Intersect Paths	0.83	0.86	0.84
Macro Avg.	0.86	0.82	**0.84**
Exact Match Rate		76.2%	
Label Agreement Rate		84.5%	

The quantitative results are summarized in Table II, which reports per-group precision, recall, and F1 scores. Despite the class imbalance inherent in the training dataset, the LVLM is able to demonstrate a macro averaged F1 score of 0.84 across different groups, indicating strong, balanced performance across the 68 SHRP2 conflict categories. Moreover, the model achieves an exact match rate and per-label agreement of 76.2% and 84.5%, respectively, with human annotators. These findings indicate that the LVLM provides supervision of sufficient fidelity to serve as a teacher model for subsequent distillation. We next investigate how closely SCOUT can replicate this performance while operating on sensor-space embeddings directly.

D. SCOUT Performance

SCOUT is trained using the 80,000 scenes labeled by the fine-tuned LVLM. Table III shows SCOUT's performance across the same conflict categories, for the same 2,000 scenes the LVLM is tested on in Table II. SCOUT is able to achieve a macro averaged F1 score of 0.80, only a 0.04 drop from the

TABLE III

SCOUT performance across scenario coverage categories.

Category (accumulated)	Precision	Recall	F1 Score
Group II. Same TW, Same Dir.	0.89	0.85	0.87
Group III. Same TW, Opp. Dir.	0.81	0.76	0.78
Group IV. Change TW, Veh. Turn.	0.79	0.72	0.75
Group V. Intersect Paths	0.80	0.84	0.82
Macro Avg.	0.82	0.79	**0.80**

fine-tuned LVLM's performance. Despite being trained on a different modality of input and machine-annotated labels, the distillation process successfully transfers the LVLM's nuanced coverage reasoning into a much smaller surrogate model.

E. Ablation Study

We conduct an ablation study to test the impact of different components of SCOUT, by removing one component at a time, and reporting the new macro averaged F1 scores. We also benchmark against an ℓ_2-regularized logistic regression model (LogReg) on the same latent features. These results are shown in Table IV.

TABLE IV
IMPACT OF DESIGN CHOICES ON SCOUT.

Variant	Macro Avg. F1	Δ vs. Full
LogReg	0.58	− 0.22
10 k training set (instead of 80 k)	0.70	− 0.10
No cross-attention	0.75	− 0.05
No dropout	0.77	− 0.03
Two residual blocks (instead of three)	0.78	− 0.02
Full SCOUT	**0.80**	

SCOUT's performance degrades when key components like cross-attention or residual depth are removed, showing their importance for accurate scenario prediction. Using the LVLM to scale up training data proves essential, significantly boosting the surrogate model's effectiveness.

F. Inference Efficiency

Table V compares the inference costs across different methods. Experiments were conducted on an RTX A6000. SCOUT achieves a massive speedup over both a human annotator as well as a fine-tuned LVLM while having the fraction of memory usage, making real-time coverage monitoring feasible on-board the vehicle.

TABLE V
INFERENCE COST COMPARISONS.

Model	Avg. Time	VRAM Usage
Human Annotator	10–15 min	–
Fine-Tuned LVLM (Gemma-3-12B)	69.4 s	42.7 GB
SCOUT (Distilled Surrogate)	**7.3 s**	**1.6 GB**
LogReg	2.4 s	0.4 GB

G. Qualitative Example

Figure 5 shows a scene where a safety-critical conflict is correctly predicted. In the scene, a motorbike recklessly enters the four-way intersection the ego vehicle is attempting to cross. The motorbike illegally crosses at a red light, causing a near-collision with the ego vehicle. Both vehicles are then able to come to a stop on time before an accident. For this interaction we correctly identify that the scene contains a conflict that belongs to a category in Group V in Table I.

VII. Conclusion and Future Work

We presented SCOUT, a lightweight surrogate model for estimating scenario coverage in autonomous vehicles using latent features already computed by the agent's navigation stack. By distilling labels from a fine-tuned LVLM trained on SHRP2-aligned human annotations, SCOUT enables scalable and efficient coverage estimation without relying on costly inference or human labeling.

Experiments across 90,000 real-world driving scenes demonstrate that **(i)** distillation preserves most of the LVLM's predictive power, **(ii)** SCOUT significantly outperforms classical surrogates, and **(iii)** improvements are statistically robust despite major class imbalance. Future work will incorporate temporal localization and semi-supervised self-training.

Acknowledgments

The research reported in this work was supported by Nuro Inc. through the real-world driving scenes provided. We also thank Daniel Bonny, Joel Bernier, Kyle Foss, Al Ricciardelli and Ananth Kini for their valueable insights and contributions.

References

[1] Z. Zhong, Y. Tang, Y. Zhou, V. d. O. Neves, Y. Liu, and B. Ray, "A survey on scenario-based testing for automated driving systems in high-fidelity simulation," *arXiv preprint arXiv:2112.00964*, 2021.

[2] H. Choset, "Coverage for robotics–a survey of recent results," *Annals of Mathematics and Artificial Intelligence*, vol. 31, pp. 113–126, 2001.

[3] E. Khalastchi and M. Kalech, "On fault detection and diagnosis in robotic systems," *ACM Computing Surveys (CSUR)*, vol. 51, no. 1, pp. 1–24, 2018.

[4] M. Fisher, V. Mascardi, K. Y. Rozier, B.-H. Schlingloff, M. Winikoff, and N. Yorke-Smith, "Towards a framework for certification of reliable autonomous systems," *Autonomous Agents and Multi-Agent Systems*, vol. 35, pp. 1–65, 2021.

[5] W. Ding, C. Xu, M. Arief, H. Lin, B. Li, and D. Zhao, "A survey on safety-critical driving scenario generation—a methodological perspective," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, pp. 6971–6988, 2023.

[6] J. Qi, S. Huang, Z. Luan, S. Yang, C. Fung, H. Yang, D. Qian, J. Shang, Z. Xiao, and Z. Wu, "Loggpt: Exploring chatgpt for log-based anomaly detection," in *IEEE International Conference on High Performance Computing & Communications, Data Science & Systems, Smart City & Dependability in Sensor, Cloud & Big Data Systems & Application*, 2023.

[7] F. Tian, Y. Lu, F. Liu, G. Ma, N. Zong, X. Wang, C. Liu, N. Wei, and K. Cao, "Supervised abnormal event detection based on chatgpt attention mechanism," *Multimedia Tools and Applications*, vol. 83, no. 41, pp. 89 501–89 519, 2024.

[8] C. Hildebrandt, M. von Stein, and S. Elbaum, "Physcov: Physical test coverage for autonomous vehicles," in *ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2023.

[9] J. Tu, S. Suo, C. Zhang, K. Wong, and R. Urtasun, "Towards scalable coverage-based testing of autonomous vehicles," in *Conference on Robot Learning (CoRL)*, 2023.

Fig. 5. A driving scene depicting a near-collision between a motorcyclist and the ego vehicle, captured from the dashcam perspective. The white motorcycle runs a red light, cutting across the intersection and triggering a high-risk interaction. Frames progress from left to right, starting at the top left and ending at the bottom right.

[10] T. Laurent, S. Klikovits, P. Arcaini, F. Ishikawa, and A. Ventresque, "Parameter coverage for testing of autonomous driving systems under uncertainty," *ACM Transactions on Software Engineering and Methodology*, vol. 32, no. 3, pp. 1–31, 2023.

[11] V. Bhatt, B. Tjanaka, M. Fontaine, and S. Nikolaidis, "Deep surrogate assisted generation of environments," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[12] B. Lei, T. Q. Kirk, A. Bhattacharya, D. Pati, X. Qian, R. Arroyave, and B. K. Mallick, "Bayesian optimization with adaptive surrogate models for automated experimental design," *Npj Computational Materials*, vol. 7, no. 1, p. 194, 2021.

[13] D. H. Stolfi and G. Danoy, "Optimising robot swarm formations by using surrogate models and simulations," *Applied Sciences*, vol. 13, no. 10, p. 5989, 2023.

[14] T. Wu, C. Miao, Y. Zhang, F. Deng, and C. Chen, "A ranknet-inspired surrogate-assisted hybrid metaheuristic for expensive coverage optimization," *arXiv preprint arXiv:2501.07375*, 2025.

[15] T. Schallau, S. Naujokat, F. Kullmann, and F. Howar, "Tree-based scenario classification: A formal framework for coverage analysis on test drives of autonomous vehicles," *arXiv preprint arXiv:2307.05106*, 2023.

[16] T. Woodlief, F. Toledo, S. Elbaum, and M. B. Dwyer, "S3c: Spatial semantic scene coverage for autonomous vehicles," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, 2024, pp. 1–13.

[17] S. Patel, S. Hariharan, P. Dhulipala, M. C. Lin, D. Manocha, H. Xu, and M. Otte, "Multi-agent coverage in urban environments," *arXiv preprint arXiv:2008.07436*, 2020.

[18] T. Heo, W. Nam, J. Paek, and J. Ko, "Autonomous reckless driving detection using deep learning on embedded gpus," in *IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 2020.

[19] J. A. Pizzorno and E. D. Berger, "Coverup: Coverage-guided LLM-based test generation," *arXiv preprint arXiv:2403.16218*, 2024.

[20] J. M. Hankey, M. A. Perez, and J. A. McClafferty, "Description of the shrp 2 naturalistic database and the crash, near-crash, and baseline data sets," Virginia Tech Transportation Institute, Tech. Rep., 2016.

[21] L. Scofield, "Researcher dictionary for safety critical event video reduction data," Virginia Tech. Virginia Tech Transportation Institute, Tech. Rep., 2015.

[22] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," *arXiv preprint arXiv:2104.08691*, 2021.

[23] S. Zhang, L. Dong, X. Li, S. Zhang, X. Sun, S. Wang, J. Li, R. Hu, T. Zhang, F. Wu, *et al.*, "Instruction tuning for large language models: A survey," *arXiv preprint arXiv:2308.10792*, 2023.

[24] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, *et al.*, "Training a helpful and harmless assistant with reinforcement learning from human feedback," *arXiv preprint arXiv:2204.05862*, 2022.

[25] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, *et al.*, "Lora: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[27] Y. D. Xiao, C. X. Min, and R. M. Ke, "Method of video key frame extraction based on lightweight-SAM," in *International Conference on Big Data Technologies*, 2024, pp. 24–30.

[28] G. Yasmin, S. Chowdhury, J. Nayak, P. Das, and A. K. Das, "Key moment extraction for designing an agglomerative clustering algorithm-based video summarization framework," *Neural Computing and Applications*, vol. 35, no. 7, pp. 4881–4902, 2023.

[29] K. Lab, *Katna: Automated video keyframe extraction tool*, version 0.3.2, 2023.

[30] G. Team, A. Kamath, J. Ferret, S. Pathak, N. Vieillard, R. Merhej, S. Perrin, T. Matejovicova, A. Ramé, M. Rivière, *et al.*, "Gemma 3 technical report," *arXiv preprint arXiv:2503.19786*, 2025.

[31] D. Han, M. Han, and Unsloth team, *Unsloth*, 2023.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.