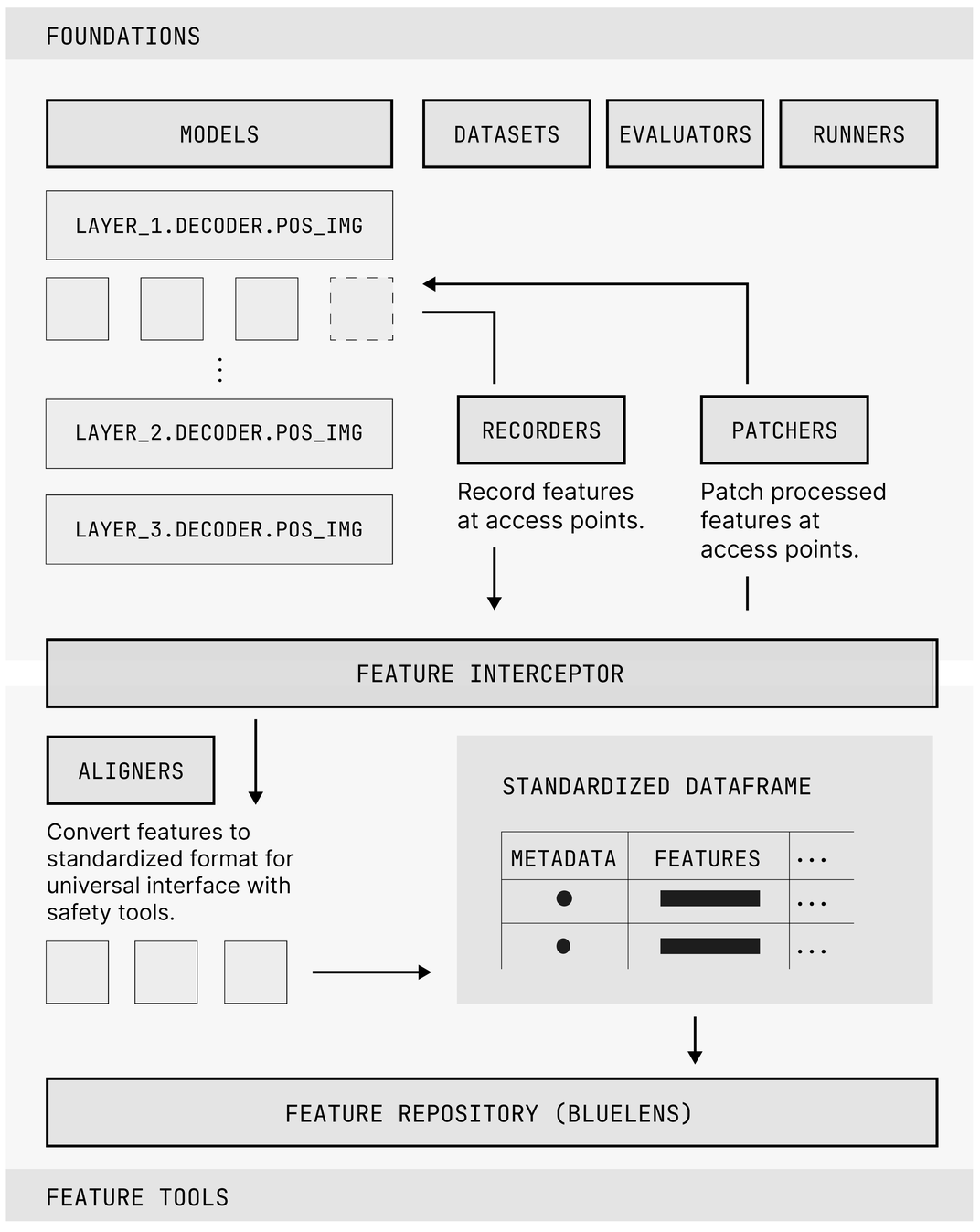
# BlueGlass — A Framework for Composite AI Safety

Harshal Nandigramwar [1, 2], Qutub Syed [1], Kay-Ulrich Scholl [1]

[1] Intel Labs , Germany [2] University of Stuttgart, Germany

University of Stuttgart
Germany




TL;DR

Existing methods in AI safety are fragmented and address complimentary aspects of model behavior. Comprehensive safety of models necessitates composition of diverse tools and techniques across multiple functional dimensions — a paradigm we term as **Composite AI Safety**.

To facilitate this methodology, we introduce **BlueGlass**, an open source framework providing a scalable, efficient and unified interface for integrating and orchestrating safety workflows. We demonstrate the capabilities of this framework through three case studies spanning across varying model classes, datasets, tasks and safety approaches.

KEY FEATURES OF BLUEGLASS

- Asset (models, datasets, evals) pooling from various sources, such as HuggingFace, Detectron2, MMDetection, etc, and access through a unified interface defined via lightweight *adaptation* files.
- Comprehensive feature management system supporting collection of model internals, caching strategies, efficient storage, loading and processing. Provides an interface (called recorders, patchers) for model-, task- and framework-agnostic model internals handling.
- Modular and standardized interfaces across components enabling flexibility, extensibility and composability. Scalable (CPU to multi-GPU) and efficient (in memory, time) processing.
- Common analysis, interpretability and explainability tools, along with presets, recipes and pre-extracted model internals provided via remote feature repository (BlueLens).

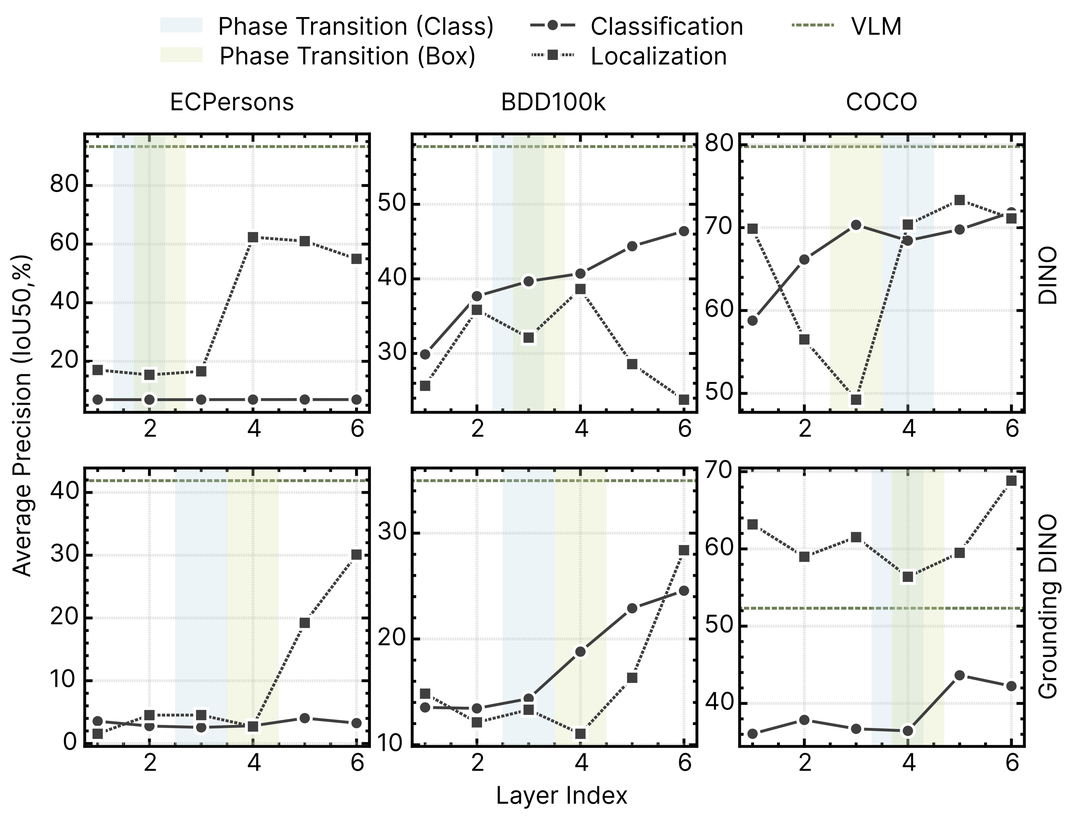
CASE STUDY #1

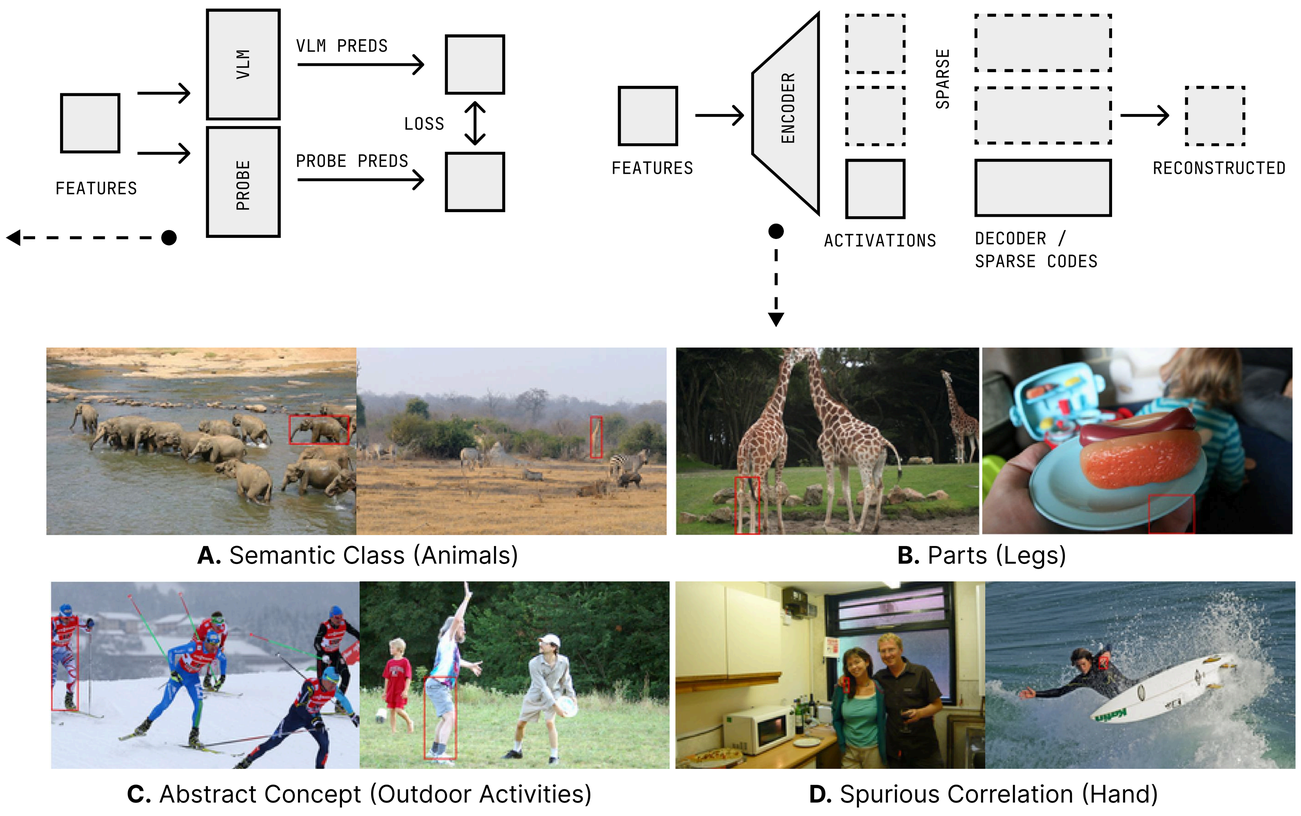
## Distributional Evaluations

*Can vision-language models perform object detection? If so, what are their failure modes and how do they compare against traditional vision-only detectors?*

| Model | Attributes | | | | Funny Birds OD | | ECPersons | | VALERIE22 | | BDD100k | | COCO | | LVIS | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Type | Box | Size | FPS | AP | AR | AP | AR | AP | AR | AP | AR | AP | AR | AP | AR |
| *YOLO v8* | D | ✓ | **0.068** | **71.5** | 85.2 | 95.4 | 1.1 | 31.1 | 1.1 | 38.5 | 8.8 | 19.4 | 24.9 | 42.6 | 7.1 | 14.1 |
| Grounding DINO | C | ✓ | 0.172 | 8.3 | 87.3 | 91.2 | 22.1 | 46.5 | 15.2 | 50.8 | 23.8 | **59.4** | 48.5 | 77.2 | 14.2 | **53.2** |
| GenerateU | G | ✓ | 0.896 | 1.5 | 65.1 | 92.9 | 2.4 | 34.6 | 2.1 | 42.6 | 13.1 | 37.7 | 32.1 | 66.1 | **25.5** | 40.7 |
| Florence 2 Large | G | × | 0.822 | 2.9 | 87.9 | 93.0 | 1.6 | 30.7 | 1.3 | 43.5 | 11.7 | 25.5 | 40.1 | 55.2 | 2.3 | 0.3 |
| Gemini 2.0 Flash | G | × | † | † | 32.2 | 50.0 | 1.3 | 21.3 | 0.1 | 15.7 | 0.9 | 3.4 | 19.9 | 32.8 | 4.9 | 7.2 |
| *DINO-DETR (SFT)* | D | ✓ | 0.218 | 4.8 | **99.6** | **99.9** | **66.4** | **76.0** | **37.4** | **70.2** | **35.9** | 55.6 | **58.3** | **78.6** | 20.8 | 38.7 |

- **Method** - We perform a comprehensive evaluation of vision-language models (VLMs) representing varying architectural classes and compare them against zero-shot vision-only baseline and a fine tuned vision-only oracle via a novel, **label matching evaluator**, that handles the open-ended predictions of VLMs and mapping of baseline.
- **Insight - Localization-aided VLMs exhibit decent generalization across data domains and operational scenarios**. While these VLMs outperform all other models in open-vocabulary setting, fine-tuned vision-only models still remain dominant. **VLMs need geometric priors for improved object detection, particularly for dense cases.**




CASE STUDY #2

## Approximation Probes for Layer Dynamics

*How do vision-language models adapt their mechanism to perform zero-shot open-world object detection?*

- **Method** - We propose **approximation probes**, a variant of linear probes that approximate the final model prediction, measuring concept resolution.
- **Insight 1** - Concept evolution trajectories demonstrate a ***phase transition*** across layers, partitioning the evolution into **three phases**. This can be attributed to **hierarchical feature learning** and **composition of concepts.**
- **Insight 2** - As both VLM and vision-only model demonstrate similar layer dynamics, **the open-ended zero-shot capabilities are a consequence of the representational alignment between vision and language components.**

CASE STUDY #3

## Sparse Autoencoders for Concept Discovery

*Do vision-language models learn human-interpretable concepts for object detection? And if so, what concepts do they learn?*

- **Method** - We use TopK sparse autoencoder to decompose the residual stream features from layer 4 of Grounding DINO, and utilize the dataset attribution method to label them.
- **Insight 1** - **Sparse autoencoders find human-interpretable concepts in VLMs for object detection** across a wide spectrum of concept hierarchy, from object parts to abstract concepts such outdoor activities.
- **Insight 2** - Surprisingly, the analysis also **revealed several concepts which are responsible for hidden failures of model such as spurious correlations**. These concepts can be used to correct and mitigate model failures.