Ref. Ares(2026)5995264 - 12/06/2026





SOPRANO

Socially-Acceptable and Trustworthy Human-Robot Teaming for Agile Industries

Project Number 101120990

D3.4 Technologies for Trustworthy and Responsible AI II

Version 1.0
14 April 2026
Final

Public Distribution

CEA and University of York

Project Partners: **AEGIS, ATB, CEA, Centro Ricerche Fiat, Circular Economy Foundation, E-TERRY, F6S, FORTH, Harokopio University of Athens, IFADO, Jade University, KUKA Assembly & Test, Netcompany-Intrasoft, NTUA, PROFACTOR, Technology Transfer Systems, The Open Group, University of Bremen, University of York**

Every effort has been made to ensure that all statements and information contained herein are accurate, however the SOPRANO Project Partners accept no liability for any error or omission in the same.

© 2026 Copyright in this document remains vested in the SOPRANO Project Partners.

PROJECT PARTNER CONTACT INFORMATION

AEGIS Nikos Kostalas Humboldtstrasse 25 38106 Braunschweig, Germany E-mail: n.kostalas@aegisresearch.eu	**ATB** Sebastian Scholze Wiener Strasse 1 28359 Bremen, Germany E-mail: scholze@atb-bremen.de
CEA Nikolaos Matragkas NanoInnov Bat. 862 PC 174 91191 Gif-sur-Yvette Cedex, France E-mail: nikolaos.matragkas@cea.fr	**Centro Ricerche Fiat** Felice Tauro Strada Torino 50 10043 ORBASSANO, Italy E-mail: felice.tauro@crf.it
Circular Economy Foundation Ioannis Basdekis Rue Breydel 34-36-40 1040 Brussels, Belgium E-mail: ioannis.basdekis@circular.foundation	**E-TERRY** Fabian Rösler Neuwerkstrasse 50 99084 Erfurt, Germany E-mail: fabian.roesler@e-terry.de
F6S Gergana Dimitrova 77 Lower Camden Street Dublin D02 XE80, Ireland E-mail: gergana@f6s.com	**FORTH** Manolis Lourakis N Plastira Str 100 70013 Heraklion, Greece E-mail: lourakis@ics.forth.gr
Harokopio University of Athens Iraklis Varlamis Eleftheriou Venizelou 70 176 71 Athens, Greece E-mail: varlamis@hua.gr	**IFADO** Felix Dreger Ardeystrasse 67 44139 Dortmund, Germany E-mail: dreger@ifado.de
Jade University George Athanassiou Friedrich Paffrath Strasse 101 26389 Wilhelmshaven, Germany E-mail: georgios.athanassiou@jade-hs.de	**KUKA Assembly & Test** Georgios Karantinakis Uhthoffstrasse 1 28757 Bremen, Germany E-mail: georgios.karantinakis@kuka.com
Netcompany-Intrasoft John Soldatos 2B Rue Nicolas Bové 1253 Luxembourg, Luxembourg E-mail: john.soldatos@netcompany-intrasoft.com	**NTUA** Maria Pateraki Heroon Polytechniou 9, Zographou Campus 157 80 Athens, Greece E-mail: mpateraki@mail.ntua.gr
PROFACTOR Sharath Akkaladevi Im Stadtgut D1 4407 Steyr-Gleink, Austria E-mail: sharath.akkaladevi@profactor.at	**Technology Transfer Systems** Antonio Avai Via Francesco d'Ovidio, 3 20131 Milan, Italy E-mail: avai@ttsnetwork.com
The Open Group Scott Hansen Rond Point Schuman 6, 7th Floor 1040 Brussels, Belgium E-mail: s.hansen@opengroup.org	**University of Bremen** Nico Hochgeschwender Bibliothekstrasse 1 28359 Bremen, Germany E-mail: ni_ho@uni-bremen.de
University of York Simos Gerasimou Deramore Lane York YO10 5GH, United Kingdom E-mail: simos.gerasimou@york.ac.uk	

DOCUMENT CONTROL

Version	Status	Date
0.1	Initial draft with outline	18 December 2025
0.5	Contributions by partners	23 March 2026
0.7	Initial review comments	26 March 2026
0.8	Updates for final reviews	27 March 2026
0.9	Updates addressing review comments	10 April 2026
1.0	Final QA version	14 April 2026

TABLE OF CONTENTS

1 Introduction **1**

1.1 Overview . . . 1

1.2 Safety Challenges . . . 1

1.2.1 Challenges from AI Internal factors . . . 1

1.2.2 Challenges from AI External factors . . . 2

1.3 Ethics Challenges . . . 2

1.3.1 Identification of ethical issues beyond immediate development concerns . . . 2

1.3.2 Early integration of ethical and regulatory frameworks . . . 2

1.3.3 Making ethical and regulatory referentials accessible and actionable . . . 3

1.3.4 Balancing qualitative ethical reflection with measurable project accountability . . . 3

1.4 Relationship with other Work Packages and Deliverables . . . 3

1.5 Structure of the Document . . . 4

2 Responsible AI & Ethics [RAI] **5**

2.1 Design and Implementation . . . 5

2.1.1 Design of the LPERL evaluation tool . . . 5

2.1.2 Using the LPERL framework to assess ERLs . . . 8

2.2 Results . . . 14

2.3 Discussion and Next Steps . . . 17

3 AI Trustworthiness [AIT] **19**

3.1 Design and Implementation . . . 19

3.1.1 Explainability-driven Testing for DNNs . . . 19

3.1.2 Improved DNN Testing through Data Augmentation . . . 26

3.1.3 Uncertainty Estimation of AI Models . . . 30

3.2 Preliminary Results . . . 34

3.2.1 Experimental Setup . . . 34

3.2.2 Experimental Results . . . 36

3.3 Integration . . . 44

3.3.1 Uncertainty Estimation . . . 44

3.4 Discussion and Next Steps . . . 50

4 Conclusion **51**

A Progress Since D3.2 **52**

A.1 Ethics Readiness Levels for industrial AI robots 52

A.1.1 Block of indicators for human/robot interaction 52

A.1.2 Validation of the evaluation tool 52

A.1.3 Ethics evaluation of three use cases and conception of interview method 52

A.1.4 ERL publication and GitHub repository 53

A.1.5 Ethics evaluation of ten new SOPRANO use cases 53

A.2 AI Trustworthiness 53

A.2.1 Explainability-driven Testing for DNNs 53

A.2.2 Uncertainty Estimation 54

LIST OF FIGURES

1 Indicator structure within LPERL. At the root level, all questions have to be answered. But second- and third-order questions at level 2 and 3 are conditioned to the response given at the root level. Answering "no" to the root question leads to a closure of this question and all subquestions are skipped. 7

2 View of the LPERL tool interface. Depending on the answer given to the first question (the root question), subquestions will be shown or hidden. Here, questions III.3.1, III.3.2 and III.3.3 are shown only because the answer 'yes' has been selected to question III.3. The current score of the block (which is the AI Act block here) is shown on the top-right corner and is immediately updated when responses are selected. 8

3 Example of detailed evaluation report showing the ALTAI block. Current score is displayed after each root indicator to help developers understand where they lose most points and focus on these indicators. The notes (empty here) contain context-specific information filled in by the ethics expert during the evaluation session. 10

4 Example of detailed evaluation report showing the GDPR block. The notes (empty here) contain context-specific information filled in by the ethics expert during the evaluation session. . . 11

5 Example of detailed evaluation report showing the AI Act block. The notes (empty here) contain context-specific information filled in by the ethics expert during the evaluation session. 12

6 Example of detailed evaluation report showing the Robotics block. The notes (empty here) contain context-specific information filled in by the ethics expert during the evaluation session. 13

7 View of the first page of the LPERL tool interface. While the lines on top permit one to navigate between the four different blocks of indicators, the evaluation process starts with question I.1. At first, all answers are set by default so that the perfect score of 4 points is achieved. The score of the current block is always visible on the right-hand side of the screen. 15

8 View of the first page of the LPERL tool interface. Now that the response to the root question has been changed to 'yes', subquestions pop out to check on the mitigation of risks. Here, some points are lost on question I.1.1. Once the evaluation moves on to the next root question, these points can never been won back. 15

9 Results obtained on the 13 use cases in SOPRANO. Overall ethics readiness level ranges from 1.540 to 3.365, however any comparison between projects is strongly discouraged because of contextual dependencies. SOPRANO projects are widely different, with some being much more ethically challenging than others. This table is meant to provide the initial evaluation results in a synthetic form, however the only meaningful comparison would be a longitudinal study comparing two or several iterative evaluations at different stages of the same project, e.g. four to six months apart. 17

10 WISDOM workflow for determining the importance-based testing adequacy of a DL system. . 19

11 Attributions methods showcase: house finch (from ImageNet) with VGG16 model. 21

12 LeNet-5 pruning showcase 22

13 Overview of the data augmentation pipeline. 27

14 Overview of the MC-CP method, specifically for classification. 30

15 Overview of the BB-CP method for the detection task. 33

16 A perturbation example using ImageNet dataset, illustrating the original image (left), along with the perturbed (middle), where (a) shows a sample from U_{R+O} and (b) a sample from U_{I+O}, and the pixel difference between the original and the perturbed (right). 36

17 Normalized coverage changes for VGG16-CIFAR10 adversarial subsets under PGD, FGSM, and CW attacks across error ratios (1–10%). Higher values reflect greater sensitivity to adversarial examples. 39
18 The Pruning performance for LeNet FC layer-1. 40
19 Image of a cat used as an input for augmentation. 41
20 Image of a ship used as an input for augmentation. 44
21 Qualitative results of the augmentations (airplane, automobile, bird, deer and dog) 45
22 Qualitative results of the augmentations (frog, horse, ship and truck) 46
23 Qualitative results of the augmentations (airplane, automobile bird, deer and dog) 47
24 Qualitative results of the augmentations (frog, horse, cat and truck) 48
25 Qualitative results of Timans and BB-CP.) 49
26 Qualitative results of C-CP and BB-CP.) 50
27 Consensus voting in WISDOM. We conduct the mask pruning to the neurons that cause less loss ($\Delta\mathcal{L}_n$ in Equation 1). WISDOM aggregates rankings from multiple attribution methods via weighted voting to produce a single list of important neurons (Equation 2, Equation 3, Equation 4). 54

EXECUTIVE SUMMARY

This deliverable provides an overview of the prototype technologies for responsible and trustworthy AI developed within the context of the SOPRANO project. The components implemented aim to define and evaluate responsible AI models from a legal, privacy, and ethical perspective, and to address the following questions: how can the trustworthiness of AI models be enhanced? and how can their uncertainty be measured?

In particular, initially, we position the work carried out thus far within the broader research landscape of trustworthy AI, covering aspects like Deep Learning testing, Data Augmentation for DNNs and Uncertainty estimation in DNNs. Next, we present the foundational work conducted on responsible AI and Ethics (RAI) that enables the definition and evaluation of Responsible AI technologies, using the Legal, Privacy and Ethics Readiness Levels (LPERL) in analogy with Technology Readiness Levels (TRL). We then introduce our work on AI Trustworthiness (AIT) which focuses on Deep Neural Networks testing by leveraging explainability concepts, followed by a data augmentation technique underpinned by Generative AI and a technique for uncertainty quantification. All techniques are complemented by preliminary quantitative and qualitative results across a diverse set of case studies and datasets, demonstrating the applicability of our prototype implementations. These developments support the objectives defined in the SOPRANO Description of the Action (DoA) by strengthening trust, safety, and reliability in Multi-Human Multi-Robot (MH-MR) collaboration scenarios, particularly in relation to measurable trustworthiness, validation of AI robustness, and the operationalisation of ethical and regulatory compliance within real-world use cases.

Progress since deliverable D3.2 *Technologies for Trustworthy and Responsible AI I* are specified in Annex A

This deliverable establishes the baseline for the work on trustworthy AI within the SOPRANO project, as well as positions the next steps for enhancing the implemented modules, increasing their technology readiness levels, and integrating the modules with the technologies within each of the industrial Pilots.

1 Introduction

1.1 Overview

This document summarizes the activities concerning responsible AI and AI trustworthiness of Deep Learning (DL) models, employed for the SOPRANO project. In particular, we focus on DL models used for any downstream task related to the Multi-Human Multi-Robot (MH-MR) collaboration systems, including, but not limited to, object detection and object classification.

In the context of responsible AI (RAI), our work comprises a prototype ethical evaluation tool to support the assessment of the ethical readiness of SOPRANO components from design to deployment. Likewise, in the context of AI trustworthiness (AIT), our work includes the development of a prototype module to test DL models to assess their prediction confidence by considering a white-box analysis of the neurons of the DL model. Furthermore, extensible testing of these models can be performed with a data augmentation module, which creates synthetic visual data, aiming to boost the prediction metrics and enlarge the prediction space, by incorporating additional data instances, which were not included in the primary training dataset. Finally, an uncertainty estimation module is being developed to quantify the uncertainty of the predictions, given a DL model and the data provided at runtime. These capabilities are designed to be directly applicable within SOPRANO use cases, ensuring that evaluation, uncertainty estimation, and data augmentation processes remain aligned with real-world operational requirements.

1.2 Safety Challenges

The fast development of AI continues to provide major economic and social benefits. With the increasing deployment and extensive use of AI in safety-critical and high-impact domains such as transportation, banking, healthcare, security, and entertainment, managing risks associated with incorrect or uncertain model behaviour becomes essential. Addressing these challenges requires systematic evaluation methodologies that enable the quantification and monitoring of model behaviour under varying operational conditions.

In the context of MH-MR collaboration, responsible AI and AI trustworthiness are challenged by many internal and external factors, including the availability of labelled training and testing data, the variability of the data employed capturing various scenarios/domains, as well as the interpretation and response to user inputs.

1.2.1 Challenges from AI Internal factors

AI internal factors relate to the various problems and challenges that arise from the Machine Learning (ML) models themselves, which may occur during development, training, deployment, or even maintenance. These challenges comprise the generalization across different scenarios related to the downstream task at hand, as well as the prediction efficiency in terms of achieving high prediction metric scores. Such challenges are directly linked to the availability and diversity of labelled training and testing data used during the design time or runtime task concerning the ML models.

In our work, we propose a data augmentation technique which aims to create new synthetic visual data capturing cases, i.e., scenarios and configurations, which were not primarily included in the training/testing datasets. These extra instances can be used to enhance the training data, to systematically evaluate and strengthen model behaviour with respect to defined performance and robustness indicators, i.e., prediction accuracy. The synthetic data can also be used to test the generalisation capabilities of the AI model and to guide possible model re-training.

1.2.2 Challenges from AI External factors

AI external factors relate more to humans and their interactions with the environment and or the multi-robot system. Humans can be ambiguous when performing a task and act with uncertainty. Thus, human user input can be challenging to process, accurately interpret, and act/respond to it. Human-robot collaboration can lead to catastrophic situations if human behaviour is not modelled appropriately. For safe human-robot collaboration, human agents interacting with multi-robot systems need to obtain feedback from the robotic agents. This feedback should include information about the certainty of the robotic agent's actions. Hence, the need for a tool that accurately quantifies the uncertainty of predictions for tasks relating to MH-MR collaboration is evident.

The runtime reliability evaluation technique, which is one of the technologies described in this deliverable report, aims to address this issue by estimating robust prediction intervals with the complete predictive distribution, yielding an uncertainty estimate.

1.3 Ethics Challenges

In SOPRANO, ethical considerations are addressed as an integral part of technological development rather than an external addition. Engineering activities already engage with ethical dimensions, particularly through concerns related to system performance, reliability, and safety, which are central to industrial innovation. However, the deployment of AI-driven systems and collaborative robotics also raises broader ethical and societal questions that are sometimes less visible within technical development processes. Supporting development teams in identifying and addressing these wider dimensions in a structured and practical manner, while enabling their integration into development environments that require measurable progress and accountability, therefore constitutes an important objective of the Responsible AI activities in the project. Within SOPRANO, these challenges are addressed through structured methodologies and supporting tools that enable their integration into AI development and evaluation workflows.

In this context, four main challenges motivate the approach developed within SOPRANO.

1.3.1 Identification of ethical issues beyond immediate development concerns

In practice, ethical considerations are not absent from engineering activities, but they may remain limited to dimensions that are directly visible during technical development. Other ethical aspects, related for instance to organizational impacts, responsibility distribution or long-term human interaction with autonomous systems, may be less readily identified.

A first challenge therefore consists in supporting development teams in moving from a partial or implicit ethical awareness towards a more comprehensive identification of ethical dimensions relevant to AI deployment in industrial contexts.

1.3.2 Early integration of ethical and regulatory frameworks

Ethical and regulatory frameworks relevant to AI systems already provide important guidance for responsible innovation. In practice, however, their use within engineering development may occur late in projects or remain primarily associated with compliance considerations addressed close to deployment stages.

A key challenge is therefore to support development teams in integrating these existing frameworks earlier in design processes, fostering an ethics-by-design approach that helps anticipate potential issues without positioning the project as providing legal compliance assessments.

1.3.3 Making ethical and regulatory referentials accessible and actionable

Although numerous ethical guidelines, standards and assessment frameworks already exist, they are not always widely known or easily usable within engineering practice. Even when identified, translating such documents into concrete development decisions remains difficult under industrial time constraints. Providing accessible information and practical support enabling teams to engage with these referentials in their daily development activities therefore constitutes a third challenge.

1.3.4 Balancing qualitative ethical reflection with measurable project accountability

Ethical analysis typically relies on contextual, qualitative and deliberative approaches that resist simple quantification. However, large collaborative research and innovation projects require structured reporting, measurable progress indicators and comparable assessment methods enabling accountability and longitudinal monitoring.

A further challenge therefore lies in balancing the need to preserve nuanced ethical reflection with the necessity of producing operational indicators and assessment tools that can be applied consistently across technical developments. Developing approaches that maintain ethical depth while enabling practical monitoring and evaluation constitutes an important objective of the methodology developed in SOPRANO.

1.4 Relationship with other Work Packages and Deliverables

This deliverable interacts with a number of work packages and deliverables, a list of which is provided in this subsection.

- **Relevant Work Packages:**
  - **WP1 Framework Requirements and Use Cases:** description of the use cases, the testing environments, and the Pilots' technologies.
  - **WP2 Industrial Context Sensing and Collaborative Task Execution:** provides AI model and training data of these models, from other Partners whose components interact with the components described in this deliverable.
  - **WP5 Integration, Deployment, and Industrial Evaluations:** comprises the integration of the components along with the Pilots' technology, to perform tests on real-world environments.
- **Relevant Deliverables:**
  - **D1.1 Project Requirements:** outlines the high-level requirements that guide the development of responsible and trustworthy AI components developed in WP3.
  - **D1.2 Evaluation Plan:** description of the evaluation datasets and metrics for validating and evaluating the developed components.
  - **D1.3 EP Requirements Framework Requirements for MH-MR:** identification of the component's requirements to achieve trustworthiness in MH-MR scenarios.
  - **D3.4 Technologies for Trustworthy & Responsible AI II:** final prototype of the technologies described in this deliverable.
  - **D5.1 Architecture Definition Report:** description of the architecture and how each component will interact with the rest, ensuring consistency and interoperability.
  - **D5.2 Integrated Technology I:** initial integration with the Pilots' technologies.
  - **D5.3 Use Cases Evaluation I:** initial evaluation of the use cases using the developed prototype tools.
  - **D5.4 Analysis of MH-MR Collaboration in EPs:** (MH–MR) evaluation of Kuka, CRF and E-TERRY use cases examining the workers experience.

1.5 Structure of the Document

This document is structured as follows. Section 2 introduces the RAI component, presenting both the design of the LPERL evaluation tool, its implementation, the results obtained on SOPRANO's use cases and a discussion regarding the proper use of the tool and understanding of the results. Next, Section 3 presents the AIT component alongside an experimental evaluation. The document concludes with Section 4, where closing remarks and future directions for the technologies are outlined. Progress since Deliverable D3.2 are detailed in Annex A.

2 Responsible AI & Ethics [RAI]

2.1 Design and Implementation

2.1.1 Design of the LPERL evaluation tool

In the context of the SOPRANO project, we adopt the Responsible Research and Innovation (RRI) approach [17, 18] to define and evaluate Responsible AI technologies through the Legal, Privacy and Ethics Readiness Levels (LPERL), in analogy with Technology Readiness Levels (TRL) [33]. LPERL address the maturity of ethical design, privacy practices, and data governance at the system level [48]. They were initially developed in the MultiRATE project and subsequently adapted by CEA to the SOPRANO use cases.

The recent emergence of AI regulation has brought forward specific ethics-by-design concerns that go beyond privacy alone, including human autonomy, fairness, responsibility, human supervision, transparency, and related dimensions, as described by the High-Level Expert Group (HLEG) for Trustworthy AI in the ALTAI Assessment List [9] and further reflected in Recital 27 of the AI Act.

LPERL consist of a set of indicators designed to assess the maturity of an AI application in terms of its ability to address ethical risks, implement sound privacy practices, and anticipate legal requirements. These dimensions are interrelated, forming a coherent set of responsible use practices that reflect an application's ethical, privacy, and legal maturity. The indicators are used during structured interviews with one or several developers, both to raise awareness on a range of topics and to assess the current maturity of the project.

The evaluation methodology for LPERL is based on a modular approach that combines different frameworks. In designing this methodology, we chose to rely on core European frameworks for ethical and privacy regulation and guidance. In particular, the ALTAI framework is useful for introducing engineers and developers to Responsible AI, as it provides a structured yet manageable set of seven key areas:

1. **Human Agency and Oversight**
    - Human Agency and Autonomy.
    - Human Oversight.
2. **Technical Robustness and Safety**
    - Resilience to Attack and Security
    - General Safety
    - Accuracy
    - Reliability, Fall-back plans and Reproducibility
3. **Privacy and Data Governance**
    - Privacy
    - Data governance
4. **Transparency**
    - Traceability
    - Explicability
    - Communication
5. **Diversity, Non-discrimination and Fairness**

- Avoidance of Unfair Bias
- Accessibility and Universal Design
- Stakeholder Participation

6. **Societal and Environmental Well-being**
   - Environmental Well-being
   - Impact on Work and Skills
   - Impact on Society at large or Democracy
7. **Accountability**
   - Auditability
   - Risk Management

By combining these frameworks and adapting them to the specific needs of industrial applications, we develop a comprehensive and practical LPERL evaluation methodology. The structure of the evaluation is based on individual modules tailored to each application. The assessment includes an "ALTAI" block, an "AI Act" block, a "Personal Data" block, and a "Robotics" block. These modules are designed to translate the content and requirements of EU frameworks into practical, accessible, and actionable indicators.

In SOPRANO use cases, AI robots are systematically assessed using at least three modules: the "ALTAI", "AI Act", and "Robotics" blocks. The "Personal Data" or "GDPR" module is only activated when relevant, i.e., when personal data are processed, which is not always the case. Conversely, in projects that are entirely software-based, the "Robotics" module would not be used. This constitutes a first level of adaptiveness of the LPERL evaluation tool. By selecting only relevant modules, the evaluation process is streamlined, reducing its duration and helping interviewees remain focused. In practice, more than half of the time (7 out of 13 use cases), one entire block of indicators is skipped.

Our objective is to define a metric that captures the ethical readiness level of an application, and we do so independently for each module. The score initially starts at 4 points and evolves within the interval [0,4]. Within each module, which contains on average 35 indicators, we define a tree-like structure that organises relationships between indicators. In most cases, three levels of indicators are distinguished:

1. Root questions correspond to the highest level of indicators and typically identify a potential risk. When such a risk is established, it leads to a penalty (i.e., a loss of points). **Ex. Robotics Q5. Is there human-robot interaction?**
2. If answered positively, the root question activates a second-order question aimed at assessing the mitigation of the identified risk. **Ex. Robotics Q5.1. Are the robot's reactions and decisions clearly communicated to humans?** These indicators allow points to be recovered.
3. Finally, if mitigation measures are in place, a third level of indicators may be triggered to assess their robustness. **Ex. Robotics Q5.1.1. Does the robot provide real-time updates on its actions and status during collaboration?** These indicators either confirm or invalidate the recovery of points from the previous step.

The score evolves in real time, allowing the interviewee to observe the impact of each response and to understand the dynamics of risk identification, mitigation, and verification. Although this three-level structure is the most common, other types of indicator relationships also exist. For instance, a root question identifying a risk may be followed by a second-order question focusing on a sub-risk. In such cases, if the sub-risk is not present, points may be partially recovered.

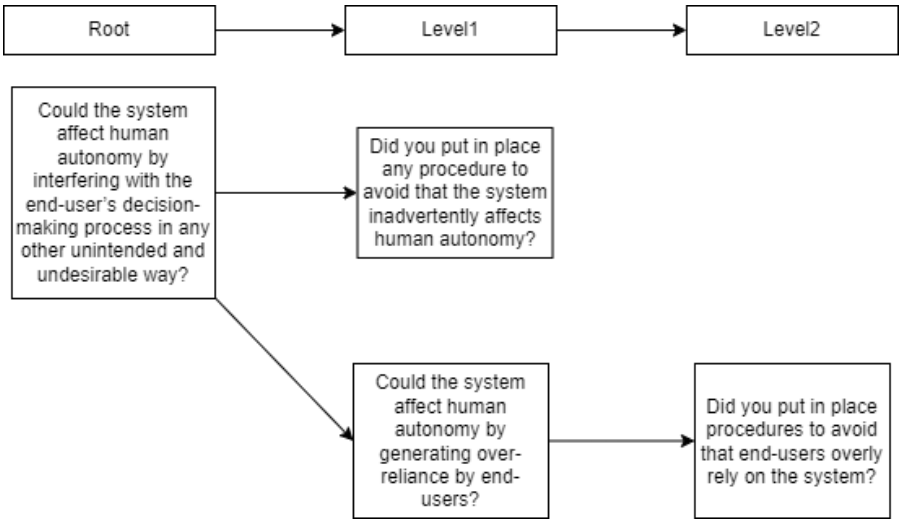


Figure 1: Indicator structure within LPERL. At the root level, all questions have to be answered. But second- and third-order questions at level 2 and 3 are conditioned to the response given at the root level. Answering "no" to the root question leads to a closure of this question and all subquestions are skipped.

The conditional structure makes it unlikely that all indicators will be triggered during an interview, which constitutes a second, more fine-grained level of adaptiveness of the LPERL tool. For example, if a root question identifying a risk is answered negatively, all subsequent sub-questions are automatically skipped, and the evaluation proceeds to the next root question. As a result, although the full framework includes around 140 indicators across four blocks, interviews typically last less than one hour, including the presentation of LPERL and an initial informal discussion. This is particularly important given the time constraints faced by development teams.

The weights, i.e., the distribution of penalties and bonuses associated with each response, have been determined based on an assessment of risks, as well as iterative testing of the methodology on multiple use cases. The magnitude of penalties reflects both the scope and the severity of potential consequences. For instance, in the example of human-robot interaction (**Q5**), the associated risks are broad, justifying a penalty of up to one full point. Mitigation questions then allow this penalty to be progressively compensated. Safety-related questions are associated with even higher penalties, reaching up to 1.450 points, reflecting their critical importance. Conversely, more specific risks with limited consequences lead to smaller penalties. In a later phase, the LPERL tool was applied to both real and synthetic use cases to validate and adjust these weightings.

Once defined, weights must be normalised to ensure consistency of the scoring system. Because LPERL define a metric, it is necessary that the mathematical properties of ERL comply with metric requirements. We therefore impose the constraint that, for a block composed of N root questions with associated penalties $P_1, ..., P_N$, we have $\sum_{i=1}^{N} P_i = 4$, ensuring that the final score lies within the interval [0,4], with both bounds reachable. This condition is important if, for instance, the LPERL questionnaire evolves in between two evaluation sessions of a given project, say a year apart. If, because the law evolves, or because other evaluations reveal that an indicator should be added regarding a missing issue, we shouldn't keep the same weights and add some potential more penalty now associated with the new indicator. Indeed, this could lead to a decreased score in the later evaluation, even though some progress could have been made on some of the previously defined indicators. As a consequence, any change that occurs in the number of indicators, or weights associated with them must always lead to a renormalization of the whole block.

At the level of mitigation indicators, it is essential that any penalty can be fully compensated if appropriate measures are implemented. Thus, for a given penalty P_i, if $M_1, ..., M_K$ denote the mitigation scores, we impose $\sum_{j=1}^{K} M_j = P_i$. The idea behind this condition is that projects facing more difficulties, namely raising more ethical issues, shouldn't been penalized *just because they are more risky*. The crucial point relies in the proper mitigation of risks and this is reflected by the fact that a more ethically challenging project should be

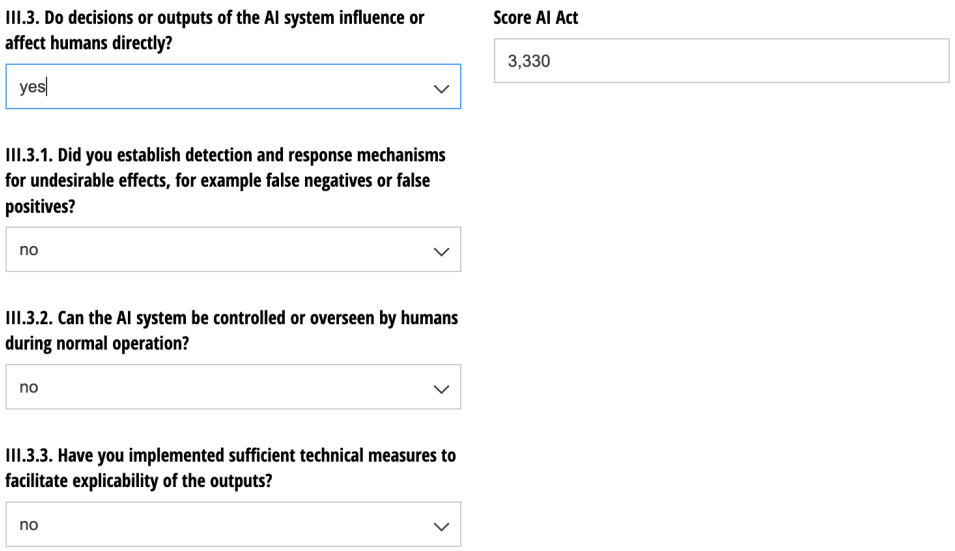


Figure 2: View of the LPERL tool interface. Depending on the answer given to the first question (the root question), subquestions will be shown or hidden. Here, questions III.3.1, III.3.2 and III.3.3 are shown only because the answer 'yes' has been selected to question III.3. The current score of the block (which is the AI Act block here) is shown on the top-right corner and is immediately updated when responses are selected.

Question-Level	Guided Questions
6	If your component uses ML, is it trained on datasets involving humans or human characteristics?
6.1	When selecting data for training, did you consider human diversity?
6.1.1	Have you measured diversity indicators using quantitative metrics?
6.1.2	Have you mitigated biases in the data found during testing?
6.2	Are there mechanisms in place for users to report fairness issues or discrimination?

Table 1: Indicators for bias in the AI Act LPERL module

able to score an ERL just as high as any other less challenging project. At a more general level, this condition of a whole mitigation of risks is necessary if we wish to offer the possibility to reach a perfect score.

2.1.2 Using the LPERL framework to assess ERLs

During the evaluation, one ethicist reads the questions together with one or several developers, who provide binary answers ("yes" or "no") while being supported in interpreting the questions and clarifying terminology when needed. Indicators are dynamically activated or deactivated based on the tree structure, and the score evolves until the end of the module (see Figure 2).

At the end of the evaluation, each module yields a score (typically ALTAI, AI Act, Robotics, and, when applicable, GDPR). The overall Ethics Readiness Level (ERL) is defined as the minimum of these scores. This choice is motivated by the need to highlight weaknesses rather than average them out. For example, a project scoring highly in most modules but poorly in data protection would still receive a low ERL, thereby encouraging targeted corrective action.

We show in Tables 1, 2 and 3 a few examples of how a root question is fully developed with mitigation indicators, as well as further indicators to check on robustness of the mitigation measures.

As innovation in AI involves a diverse range of professionals, from computer scientists to legal experts, different disciplinary languages are used to address ethical issues. The questions in the LPERL tool may therefore not always be entirely self-explanatory, though in case of repeated difficulties observed with several teams,

Question-Level	Guided Questions
1	Are users informed about the capabilities and limitations of the robot?
1.1	Are potential risks and failures explained to users?
1.2	Is this information readily available and easy to understand?
2	Are there mechanisms to ensure the traceability of the robot's decisions?
2.1	Are there procedures for users to inquire about the causes of the robot's decisions?
2.2	Is technical logging designed to enhance the explicability of the robot's decisions?
2.3	Is there a framework for investigating incidents involving the robot?
2.3.1	Are there guidelines for corrective actions based on the investigation outcomes?
3	Is there a clear chain of responsibility for decisions made by the robot?
3.1	Are roles and responsibilities of humans and the robot clearly documented?

Table 2: Transparency and responsibility indicators in the Robotics LPERL module

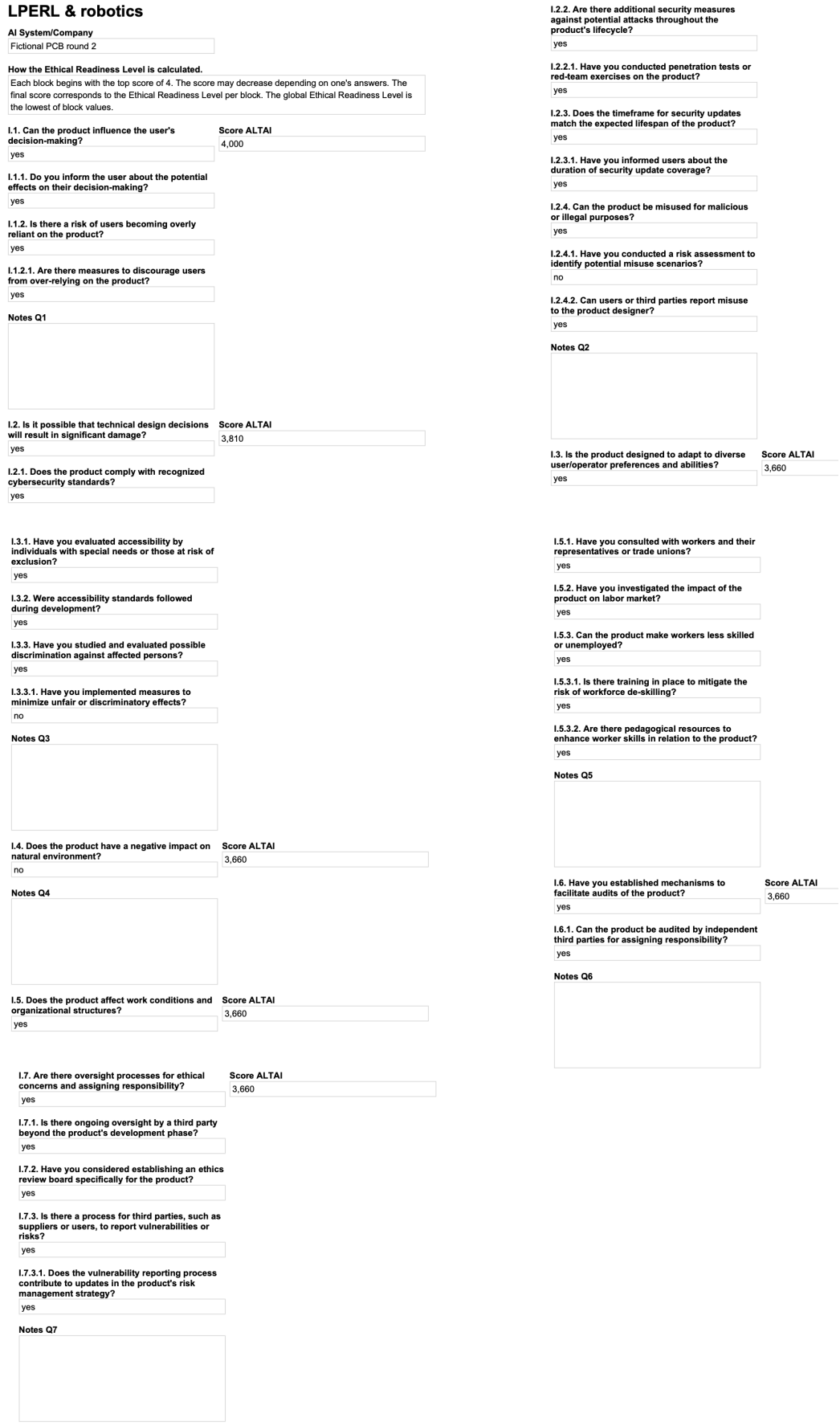
Question-Level	Guided Questions
4	Are the safety protocols based on industry standards and best practices?
4.1	Are safety protocols regularly reviewed and updated?
4.2	Have you conducted safety tests across a range of operating conditions?
4.2.1	Were tests conducted to assess safety under extreme or unpredictable conditions?
4.3	Are measures in place to ensure the robot does not cause physical harm to users?
4.3.1	Are sensory components in place to detect and avoid user contact?
4.3.1.1	Are there soft-touch or impact-mitigation features incorporated in the design?
4.4	Are there fail-safe mechanisms to protect users in case of component failure?
4.4.1	Do the fail-safe mechanisms activate automatically and reliably?
4.5	Are emergency stop features easily accessible and reliable?
4.5.1	Are users trained on how to activate these emergency stop mechanisms?

Table 3: Safety indicators in the Robotics LPERL module

we also work on rephrasing to avoid ambiguities or other comprehension problems. Effective operationalisation requires fostering dialogue between stakeholders to improve mutual understanding and convergence of interpretations. While the checklist provides a shared reference framework, face-to-face discussion remains essential. Dialogue has long been recognised as a cornerstone of ethics, and meaningful progress in AI ethics often emerges from interactions between ethicists and developers [1].

The LPERL structure aligns with an anticipatory approach to ethics, in which ethical issues are addressed before full deployment [19]. By mapping evaluation results onto readiness levels, the tool provides a structured assessment of ethical maturity, helping developers identify areas for improvement and supporting informed decision-making. The methodology offers a high degree of alignment with European principles and regulatory frameworks. By integrating elements from ALTAI, the AI Act, and GDPR into a unified evaluation approach, it reflects established norms while making them accessible and actionable for development teams. In doing so, it provides a comprehensive framework for assessing AI systems in terms of ethical, legal, and societal considerations.

From a developer point of view, the ethical assessment may be fruitful because issues that were not very visible in their project can be identified, and this is also a sometimes unique occasion to discuss some aspects that are more societal than technical. But in order to consolidate this new perspective, it is also important that they can also later reflect and refer to a document produced after the evaluation. We then systematically send to interviewees a document synthesizing our questions, their answers and the evolution of the score after each root indicator. This permits them to have a very precise trace of our exchanges and to track the evolution of ERLs for each block of indicators. In Figure 3, we show the part of this document relative to the first block of indicators (ALTAI).



LPERL & robotics

AI System/Company
Fictional PCB round 2

How the Ethical Readiness Level is calculated.
Each block begins with the top score of 4. The score may decrease depending on one's answers. The final score corresponds to the Ethical Readiness Level per block. The global Ethical Readiness Level is the lowest of block values.

I.1. Can the product influence the user's decision-making? — **Score ALTAI** 4,000
yes

I.1.1. Do you inform the user about the potential effects on their decision-making?
yes

I.1.2. Is there a risk of users becoming overly reliant on the product?
yes

I.1.2.1. Are there measures to discourage users from over-relying on the product?
yes

Notes Q1

I.2. Is it possible that technical design decisions will result in significant damage? — **Score ALTAI** 3,810
yes

I.2.1. Does the product comply with recognized cybersecurity standards?
yes

I.2.2. Are there additional security measures against potential attacks throughout the product's lifecycle?
yes

I.2.2.1. Have you conducted penetration tests or red-team exercises on the product?
yes

I.2.3. Does the timeframe for security updates match the expected lifespan of the product?
yes

I.2.3.1. Have you informed users about the duration of security update coverage?
yes

I.2.4. Can the product be misused for malicious or illegal purposes?
yes

I.2.4.1. Have you conducted a risk assessment to identify potential misuse scenarios?
no

I.2.4.2. Can users or third parties report misuse to the product designer?
yes

Notes Q2

I.3. Is the product designed to adapt to diverse user/operator preferences and abilities? — **Score ALTAI** 3,660
yes

I.3.1. Have you evaluated accessibility by individuals with special needs or those at risk of exclusion?
yes

I.3.2. Were accessibility standards followed during development?
yes

I.3.3. Have you studied and evaluated possible discrimination against affected persons?
yes

I.3.3.1. Have you implemented measures to minimize unfair or discriminatory effects?
no

Notes Q3

I.4. Does the product have a negative impact on natural environment? — **Score ALTAI** 3,660
no

Notes Q4

I.5. Does the product affect work conditions and organizational structures? — **Score ALTAI** 3,660
yes

I.5.1. Have you consulted with workers and their representatives or trade unions?
yes

I.5.2. Have you investigated the impact of the product on labor market?
yes

I.5.3. Can the product make workers less skilled or unemployed?
yes

I.5.3.1. Is there training in place to mitigate the risk of workforce de-skilling?
yes

I.5.3.2. Are there pedagogical resources to enhance worker skills in relation to the product?
yes

Notes Q5

I.6. Have you established mechanisms to facilitate audits of the product? — **Score ALTAI** 3,660
yes

I.6.1. Can the product be audited by independent third parties for assigning responsibility?
yes

Notes Q6

I.7. Are there oversight processes for ethical concerns and assigning responsibility? — **Score ALTAI** 3,660
yes

I.7.1. Is there ongoing oversight by a third party beyond the product's development phase?
yes

I.7.2. Have you considered establishing an ethics review board specifically for the product?
yes

I.7.3. Is there a process for third parties, such as suppliers or users, to report vulnerabilities or risks?
yes

I.7.3.1. Does the vulnerability reporting process contribute to updates in the product's risk management strategy?
yes

Notes Q7

Figure 3: Example of detailed evaluation report showing the ALTAI block. Current score is displayed after each root indicator to help developers understand where they lose most points and focus on these indicators. The notes (empty here) contain context-specific information filled in by the ethics expert during the evaluation session.

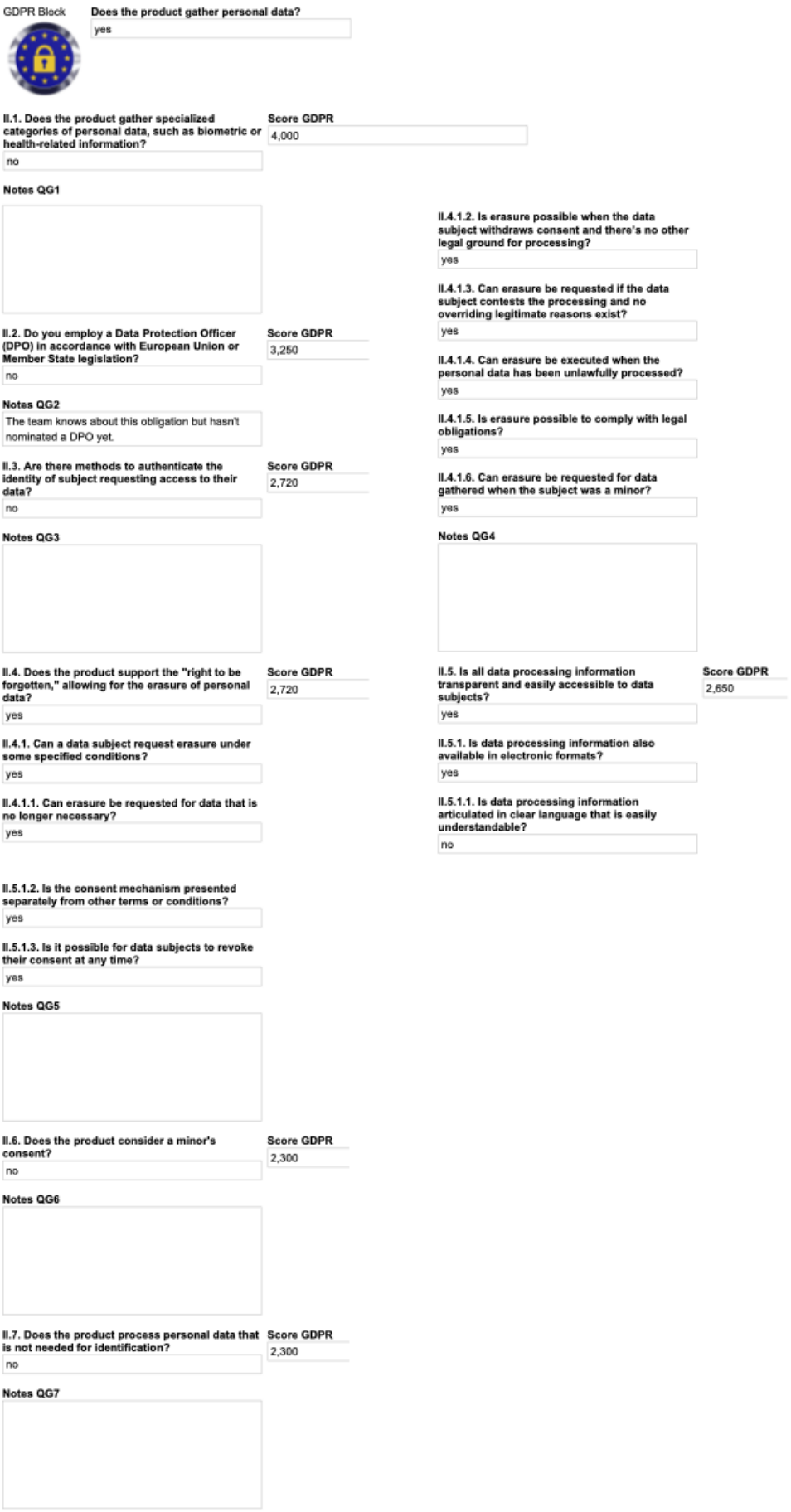


Figure 4: Example of detailed evaluation report showing the GDPR block. The notes (empty here) contain context-specific information filled in by the ethics expert during the evaluation session.



Figure 5: Example of detailed evaluation report showing the AI Act block. The notes (empty here) contain context-specific information filled in by the ethics expert during the evaluation session.

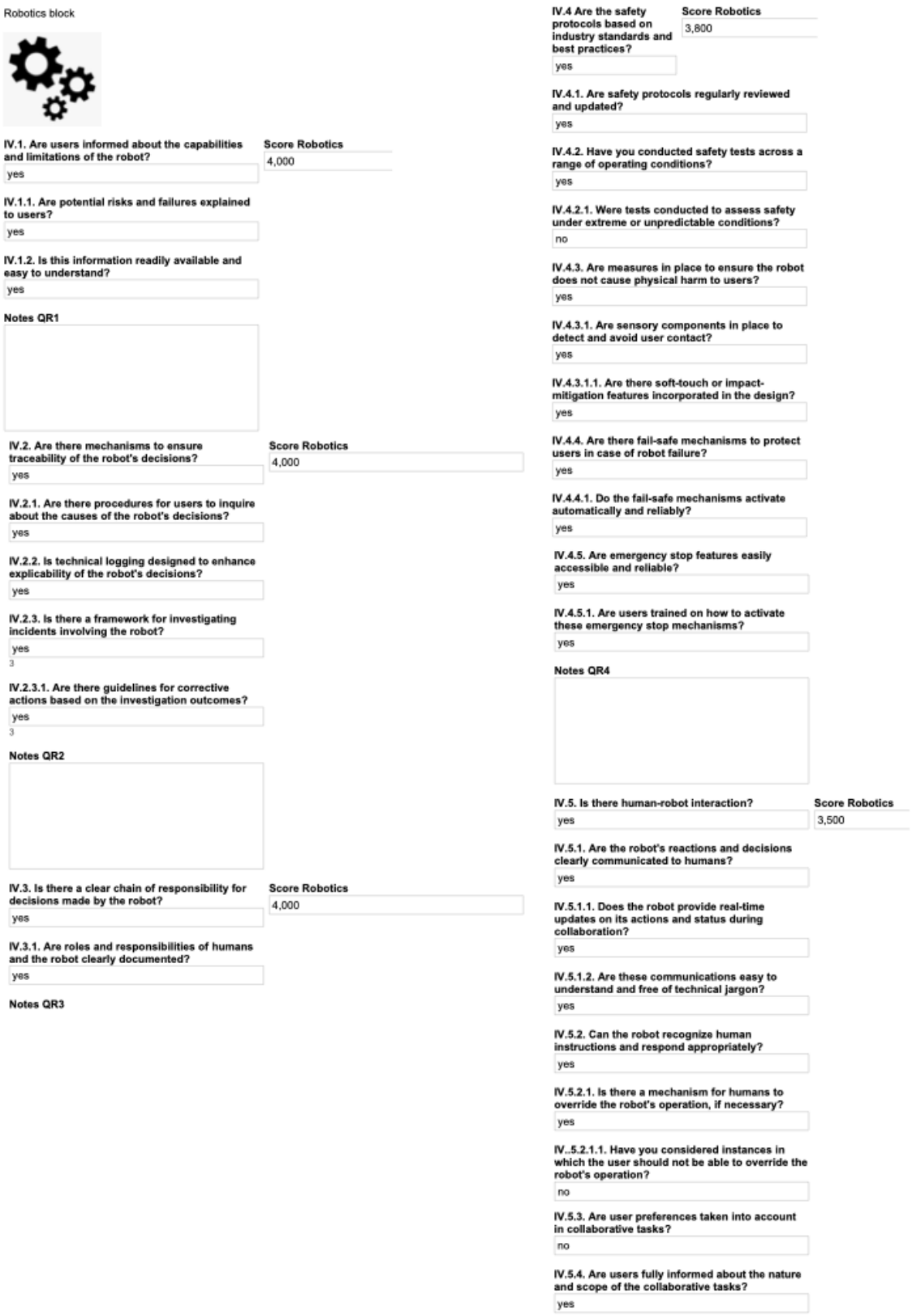


Figure 6: Example of detailed evaluation report showing the Robotics block. The notes (empty here) contain context-specific information filled in by the ethics expert during the evaluation session.

In Figures 4, 5 and 6, the other blocks of indicators are shown, thus completing the synthetic document that developers keep and can later refer to.

2.2 Results

The validation process for the evaluation of Ethics Readiness Levels using the LPERL tool for Responsible AI started in late 2024 and continued during the first months of 2025. The LPERL tool used in SOPRANO was implemented on the online platform `cognitoforms.com`. The current implementation, as of March 2026, can be accessed at `https://www.cognitoforms.com/CEA7/LPERLRobotics`.

Figure 7 shows the beginning of an evaluation on a synthetic use case. Note that the top four lines permit one to navigate between the different screens of the ALTAI block (line 1), the GDPR block (line 2), the AI Act block (line 3) and the Robotics block (line 4). Each of these blocks is comprised of five to seven screens that will be navigated during the interview. After typing the name of the system and of the company, the grading system is quickly described and the interview starts with question I.1. By default, all root questions are answered in the negative so that no point is lost. However, this will change when real answers are chosen in the interface.

In Figure 8, the answer has been turned to a 'yes' after the interviewees have decided that their product could be said to influence a user's decision-making. This response made new subquestions of two different kinds pop up. The first one, Q.I.1.1, is a mitigation question but here, it does not permit the product to win back points because it's not implemented yet. The second one, Q.I.1.2, is a subrisk question. It permits one to isolate a narrower issue from the root question like here, asking if users can become *overly* reliant on the product. This question typically does not get answered without some discussion around what over reliance may mean in general and in the current context. Here, it has been decided that this is not the case, that the type of reliance that developers admit their future users may have is manageable and unproblematic. For instance, they don't see that their users may lose a capacity that they originally have because the product won't permit them to exercise it. As a consequence, the only lost points here are due to the lack of information highlighted in Q.I.1.1, which is not a serious issue. The team is already aware of the necessity to inform users about their robot's influence and they asked to take note that they're not yet at the stage of having proper users.

At the bottom of each page of the questionnaire (each page corresponds to one root indicator, further developed into all the other indicators from other layers), there is always the possibility to take free notes, which is one of the ways to encourage reflection beyond yes or no answers.

In May and June 2025, evaluations were scheduled with SOPRANO's first three use cases: E-Terry, CFR and Kuka. In preparation for these evaluations, CEA's ethicist visited E-Terry and Kuka in April 2025 in order to become familiar with the projects, technologies, environments, and team members involved. Beyond collecting data on the ethical preparedness of these projects and obtaining ERLs, these first evaluations were also useful for better understanding some of the challenges raised by the LPERL methodology itself. In particular, it became clear that questions of the form "Have you implemented X" were sometimes difficult to answer when X had not yet been technically implemented, but had already been considered and planned by the team. This ambiguity was gradually clarified so as to encourage developers to provide a literal answer to the question, rather than answering on the basis of plans or intentions. In most cases, however, the evaluation process runs smoothly, with the ethicist accompanying the team, presenting the tool and the objectives of the evaluation, and supporting discussion whenever needed.

Unsurprisingly, the results collected showed that the three use cases were already well advanced from an ethical point of view. Indeed, these projects were also relatively advanced from a technological perspective, and they had been selected in SOPRANO because they displayed promising qualities not only in technical terms, but also and above all as societal projects aiming to relieve workers from painful, difficult, or less meaningful tasks.

In the months that followed, SOPRANO's Open Call led to the inclusion of ten new use cases, with applications ranging from logistics and waste management to home delivery of prescription drugs and precision

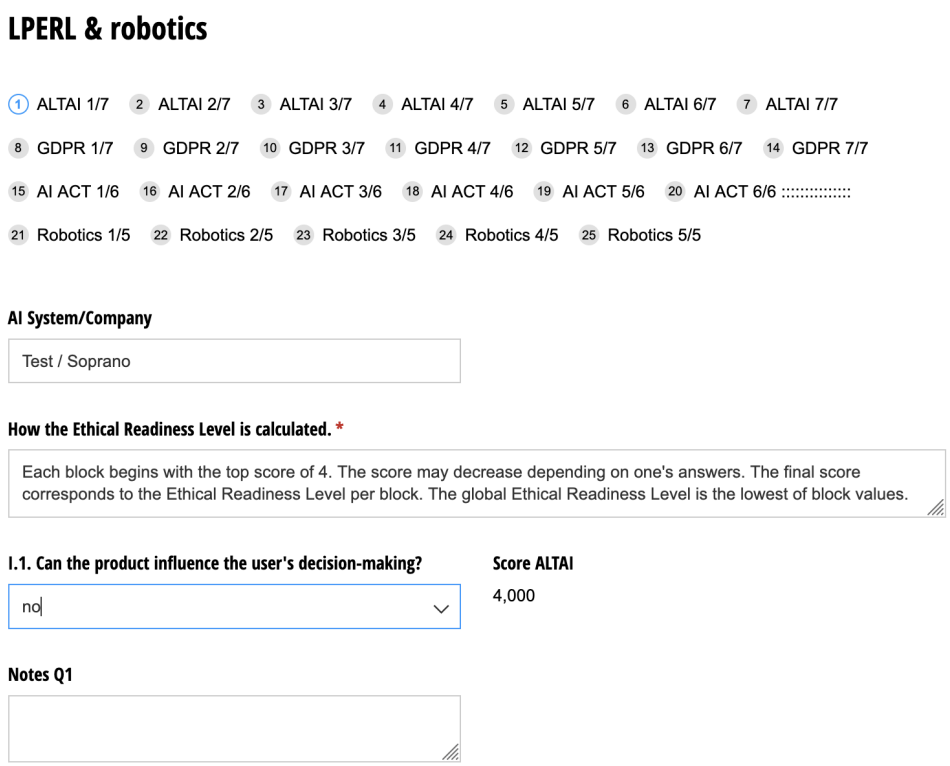


Figure 7: View of the first page of the LPERL tool interface. While the lines on top permit one to navigate between the four different blocks of indicators, the evaluation process starts with question I.1. At first, all answers are set by default so that the perfect score of 4 points is achieved. The score of the current block is always visible on the right-hand side of the screen.

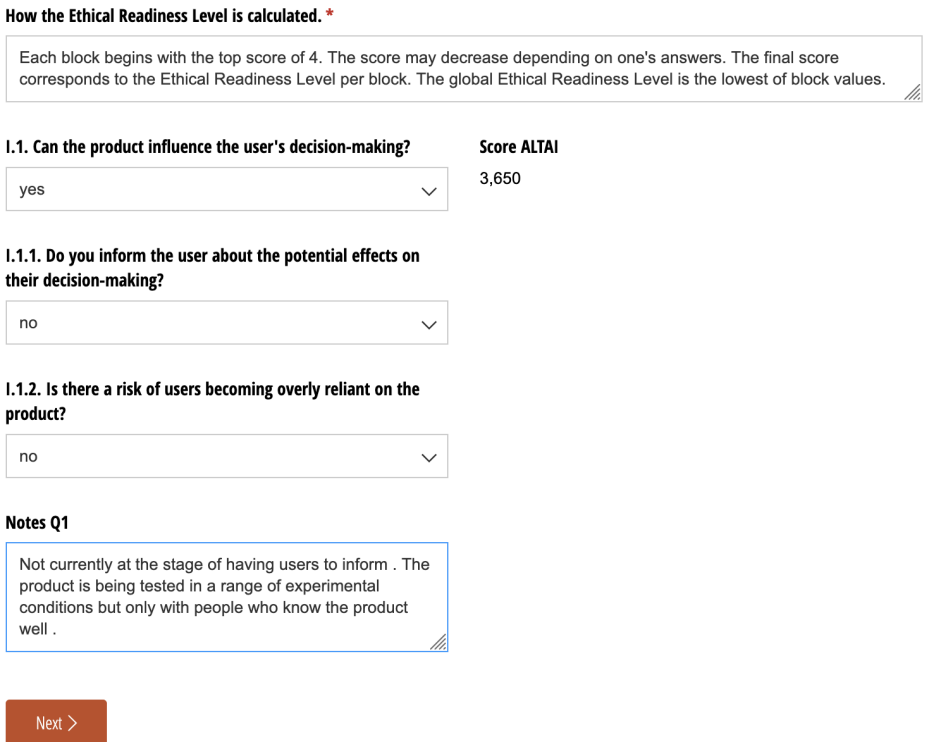


Figure 8: View of the first page of the LPERL tool interface. Now that the response to the root question has been changed to ‘yes’, subquestions pop out to check on the mitigation of risks. Here, some points are lost on question I.1.1. Once the evaluation moves on to the next root question, these points can never been won back.

manufacturing. In preparation for this intensive phase of evaluation, the interview format was standardised with a five-minute presentation of the tool, covering both the point system and the tree-like structure of indicators, as well as the objectives and meaning of the evaluation. Our aim was to create a constructive and relaxed atmosphere in which developers would feel comfortable rather than scrutinised, despite the grading-like appearance of the process. While this had come naturally with the original use cases, because we had met the teams in person beforehand, no such opportunity existed with the ten new use cases scattered across Europe. Trust therefore had to be established online during a first and only meeting.

This was not only the best way to obtain reliable answers, but also a necessary condition for creating the right environment for dialogue and free expression, without participants fearing that they might say the "wrong" thing. More often than not, developers would apologise for not being knowledgeable in ethics, whether as a philosophical discipline or as a set of practices and guidelines related to their activities. Providing a safe space for reflection and discussion around ethics and responsible innovation was therefore an absolute priority.

From this perspective, it was also essential to downplay the importance of the "grading" dimension of the evaluation and to reassure developers that the score itself was not the central aspect of the process. To be sure, the score is useful for several reasons. It helps structure the interview and gives a degree of concreteness to missing elements. When the impact of a technology has not been fully assessed, when cybersecurity has not yet been ensured, or when developers have not considered environmental impact, ERLs make these gaps visible and symbolise the idea of consequences through the loss of points. At the same time, this remains a form of gamification of ethical assessment, which could in principle exist even without scoring. ERLs help participants remain focused and take the exercise seriously; they also provide a rough indication of a project's ethical maturity. However, one should not over-interpret the idea that an ERL offers an exact representation of ethical preparedness. It is important to emphasise that the core value of this evaluation process lies less in the final score itself than in the opportunity to identify vulnerabilities and possible areas for improvement along the way. Raising awareness of the many dimensions reflected in the indicators is the most important effect that can be expected from these sessions, especially at this relatively early stage of development.

Figure 9 shows the scores obtained by each use case for each block of indicators, as well as the corresponding ERL. Again, the fact that the new use cases selected through the Open Call had been chosen not only for their technological promise but also for a certain degree of ethical maturity contributed to the generally positive results. Scores ranged overall from 1.540 to 3.365, with an average of 2.56 and a standard deviation of 0.6. The observed differences were mainly explained by two factors: first, the overall stage of development of the project, with teams already having products on the market generally being more advanced in ethical terms as well; and second, the type of application, with robots intended to operate in closed and secure environments tending to raise fewer issues than robots designed to function in more open settings, potentially including interaction with the general public.

The caution with which we have advocated to explain ERLs to project developers that we interviewed must be repeated here. On several occasions, developers were curious of other projects' scores and sometimes anxious to know how they compare. However, we told them that this comparison meant nothing and this is something fundamental to keep in mind when looking at Table 9. Knowing the ERLs of two projects, with one lower than the other has no meaning in general. Only in the extreme cases of very low vs. very high scores can something be said about relative ethical preparedness. But for scores ranging as they do here, there is no point deeming one or the other project ethically better or more advanced. Firstly, ERLs are constructed so as to highlight the block of lowest score, encouraging targeted corrective action. With this feature, two projects sharing an equal number of "defects" will have very different ERLs if one of them has those defects mostly concentrated in one block because this block will score low and the general ERL will dip down too. On the other hand, the same number of defects distributed across three or even four blocks of indicators will result in much more smoothed out ERLs per block, thus leading to a higher general ERL.

Secondly, the number of issues that a project is facing completely depends upon the project's ambitions, scope and environment. Hence, as we evoked earlier, a project in which a robot is intended to operate in an open setting and potentially even interact with the general public may be very advanced and well-thought, it faces

	ALTAI	GDPR	AI ACT	ROBOTICS	LPERL
E-Terry	3.610	2.720	3.280	3.670	2.720
CFR	2.330	N/A	2.915	3.050	2.330
Kuka	2.520	N/A	1.915	3.600	1.915
Caboto/COMPASS	1.680	1.895	2.535	1.820	1.680
ECEON/AUTOLOAD	3.860	4.000	3.600	3.365	3.365
Robotico/FRACTAL	3.350	N/A	2.995	1.540	1.540
Robenso/HR-PICKAIR	4.000	N/A	2.995	4.000	2.995
3D-Components AS/FAROS	3.400	N/A	3.390	3.190	3.190
Hiro Robotics/REWARD	3.250	2.720	3.940	3.900	3.250
ANT Maschinen/YARDMATE	3.740	N/A	2.495	3.700	2.495
Unparallel/DESCANT	2.230	2.260	3.165	3.110	2.230
J&W SRL/YOUNIC	3.550	2.480	3.850	4.000	2.480
Essential Robotics/ESSENTIAL	3.740	N/A	3.430	3.040	3.040

Figure 9: Results obtained on the 13 use cases in SOPRANO. Overall ethics readiness level ranges from 1.540 to 3.365, however any comparison between projects is strongly discouraged because of contextual dependencies. SOPRANO projects are widely different, with some being much more ethically challenging than others. This table is meant to provide the initial evaluation results in a synthetic form, however the only meaningful comparison would be a longitudinal study comparing two or several iterative evaluations at different stages of the same project, e.g. four to six months apart.

so many more challenges than even an average project in which a robot will work in a secure and closed environment that it will be very hard for the former to score higher than the latter.

So we show these results while asking the reader to resist the temptation to compare scores across lines because we believe that no meaning can be attached in such comparison. We show them for the sake of having a synthetic way of keeping scores for each project in SOPRANO but we believe that the main real value of ERL is to keep track of progress *for a given project.*

At this stage, we have completed one round of Responsible AI evaluation for each and every use case in SOPRANO. Beyond the scores themselves, the process has clearly helped draw the attention of teams to these issues and has been beneficial in identifying areas for improvement across all use cases. All teams received a PDF containing all indicators (i.e., the questions), together with the evolution of the score after each indicator. They can therefore easily identify where points were lost and work more specifically on those aspects.

Given the nature of ethical assessments, what matters most is the possibility to track progress some months apart. Therefore, we intend to add another round of assessments for the first three use cases in SOPRANO but we find it also important that the other ten new use cases can also get a second assessment. Only this will reveal how developers have tackled ethical issues and what are the remaining issues.

2.3 Discussion and Next Steps

The implementation of the evaluation process across SOPRANO pilots has not only provided insights into the ethical maturity of the systems under development, but also into the evaluation methodology itself. In particular, it has highlighted a fundamental tension inherent to ethical appraisal of technological systems: balancing the dialogical and context-dependent nature of ethical reflection with the need for structured, manageable and time-efficient assessment processes.

Ethical evaluation cannot be reduced to a simple set of indicators without risking a “box-ticking” approach that would undermine its purpose. At the same time, purely open-ended discussions are difficult to integrate into engineering workflows and project timelines. The approach developed in SOPRANO seeks to navigate this tension. While it relies on structured indicators and produces aggregated scores, it is not intended to replace qualitative reflection, but rather to support and structure it.

This balance requires careful facilitation. In particular, the initial presentation of the evaluation tool to development teams plays a crucial role. Beyond explaining its technical aspects, it establishes the conditions

for meaningful engagement. First, participants must be encouraged to go beyond binary responses and to articulate their reasoning in context. The use of qualitative notes associated with each indicator is essential in this regard, as it allows the evaluation to capture nuances that cannot be expressed through "yes/no" answers alone. Creating a setting in which participants feel comfortable reflecting critically on their own work is a key condition for the success of the process.

Second, the role of scoring must be carefully framed. There is a fine line between using scores as a structuring element and reducing ethical evaluation to a purely quantitative exercise. Conversely, presenting scores as purely illustrative or gamified would deprive them of their practical value. In the approach developed here, Ethics Readiness Levels (ERLs) do not carry absolute meaning and are not intended for comparison across unrelated projects. However, they do provide meaningful indications of relative ethical maturity within a given project, and are particularly valuable when used for longitudinal monitoring over time.

Based on this experience, one important conclusion is that ethical evaluations should be conducted iteratively throughout the lifecycle of a project. In contexts similar to SOPRANO, performing at least two evaluations, approximately one year apart, would allow for meaningful tracking of progress and support the integration of ethical considerations into development processes. While this was not fully achievable within the timeframe of the present task, it represents a clear direction for future implementations.

Another significant observation concerns the format of evaluation sessions. While most evaluations were conducted with a single representative from the development teams and already yielded valuable insights, sessions involving multiple team members consistently led to richer and more dynamic discussions. In these cases, questions prompted collective reflection, with participants engaging with each other's perspectives rather than providing immediate individual answers. This not only improved the quality of the evaluation, but also facilitated the internal dissemination of ethical considerations within the teams, thereby increasing the likelihood of follow-up actions.

In future implementations, it is therefore recommended to involve multiple members of development teams in evaluation sessions whenever possible. This contributes both to the depth of the ethical analysis and to its effective integration within the project.

These findings confirm that ethical evaluation, when properly integrated into development practices, can act not only as an assessment tool, but also as a driver for reflection, coordination and continuous improvement within engineering teams.

3 AI Trustworthiness [AIT]

3.1 Design and Implementation

Our work on AI trustworthiness (AIT) focuses on analysing AI models and using this understanding to improve its capabilities. To this end, the AIT component comprises (i) a module for testing deep neural networks by considering the internal architectures of these AI models and leveraging explainability concepts (Section 3.1.1); (ii) a module for augmenting datasets to improve generalization (Section 3.1.2); and (iii) a module for estimating the runtime uncertainty of the employed deep neural networks (Section 3.1.3).

3.1.1 Explainability-driven Testing for DNNs

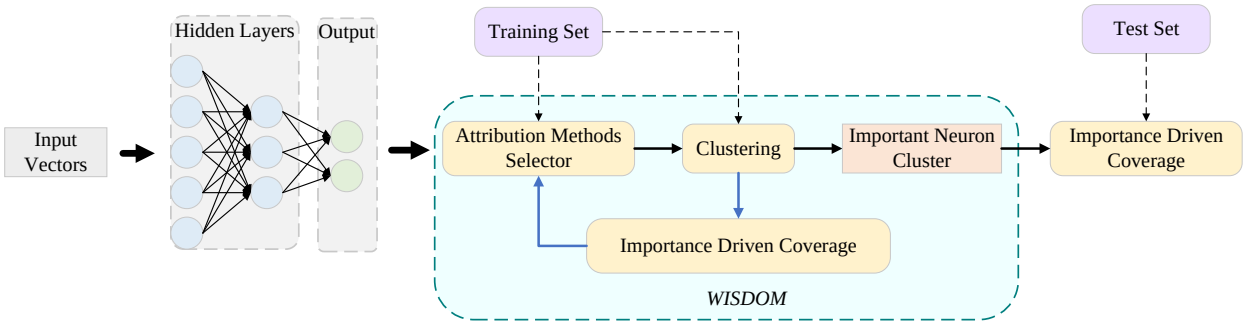


Figure 10: WISDOM workflow for determining the importance-based testing adequacy of a DL system.

In the context of explainability-driven testing, we are extending DeepImportance [15] to introduce a *white-box* testing system called WISDOM. Figure 10 presents the high-level workflow of WISDOM, which accepts a pre-trained deep learning (DL) model and training dataset as inputs and applies an *Important Neuron Identification* to automatically select an appropriate attribution method. This method identifies the most critical neurons that contribute substantially to the model's decision-making process. Next, WISDOM performs a quantization step, generating an optimally determined set of neuron activation clusters that captures how the behaviour of these key neurons varies in response to inputs from the training set. Finally, WISDOM applies a dynamic clustering approach to all relevant neurons to evaluate the test set's coverage adequacy. This phase, known as *Importance-driven Coverage*(IDC), also guides the selection of attribution methods within the system.

Neuron Importance and Attributions Selector

The purpose of importance analysis is to identify neurons within a DL system that are key contributors to its decision-making process. Various approaches have been developed to determine neuron importance, as explored in previous research [15]. Evaluating each neuron's contribution to the model's output, based on its activation in a given layer, is known as an attribution method. Table 4 provides a summary of the attribution methods implemented in WISDOM.

Different attribution methods may exhibit varying preferences for identifying important neurons. In Figure 11, we present a heatmap of a *house finch* classified by a pre-trained VGG16 model. The dark green regions indicate the features deemed important by each attribution method based on Table 4, highlighting the areas that are likely to influence the model's predictions. Since different attribution methods may vary in precision when identifying important neurons — exhibiting preferences based on DL architecture and input changes — we present a framework, WISDOM, specifically designed to select the optimal attribution methods based on the

Attribution Methods	Abbr.	Approach	Preferred Domains
Activation	la	Forward Propagation-based	General neural networks with certain activation function
LRP [3]	lrp	Backward Propagation-based	Image data with Linear/Convolution layers, general neural networks
Conductance [11]	lc	Backward Propagation-based	General neural networks, especially for understanding neuron contributions
Internal Influence [36]	ii	Backward Propagation-based	General neural networks, focusing on internal layer contributions
DeepLift [53]	ldl	Backward Propagation-based	General neural networks, including image and text data
GradientShap [49]	lgs	Backward Propagation-based	General neural networks, combining gradients with SHAP values
Integrated Gradients [56]	lig	Backward Propagation-based	General neural networks, applicable to various data types
Feature Ablation	lfa	Perturbation-based	General neural networks, applicable to various data types

Table 4: Attribution Methods considered by WISDOM

given inputs and target layer. WISDOM supports multiple approaches for assessing neuron importance within deep neural networks, offering diverse perspectives that contribute to a more comprehensive coverage analysis.

To measure the importance of a neuron n in layer L, we use the model's loss difference $\Delta\mathcal{L}_n$ (i.e., degradation in model performance) to guide us in choosing the optimal attribution method:

$$\Delta\mathcal{L}_n = \mathcal{L}(f_{\theta\setminus\theta_k^m}) - \mathcal{L}(f_\theta), \tag{1}$$

where $f_{\theta\setminus\theta_k^m}$ represents the a pruning operator that set the top-k neurons' weight tensors chosen from attribution method m (i.e., $\theta^m[0,\ldots,k]$) into zeros. We then conduct the $\Delta\mathcal{L}_m$ as our voting weights across all the attribution set $\mathcal{A}$:

$$w_m = \frac{\Delta\mathcal{L}_m}{\sum_{m'\in\mathcal{A}}\Delta\mathcal{L}_{m'}}. \tag{2}$$

Using these weights, we have a unified importance score for each neuron n via a weighted scoring mechanism:

$$S(n) = \sum_{m\in\mathcal{A}} \mathcal{R}^m(n)\cdot w_m, \tag{3}$$

where $\mathcal{R}^m(n)$ represents the attribution score in neuron n with attribution method m.

Neurons that consistently appear at the top of many rankings (especially from high-weight methods) will accumulate a larger score. In addition, neurons not in the top-k for a method receive a score of zero. After computing $S(n)$ for all neurons, we obtain a final aggregated ranking list rank* by sorting neurons in descending order of the consensus score $S^*(n)$ (see Eq. 3). We then select the k highest-ranked neurons from rank* as the set of *important neurons* identified by WISDOM:

$$S^*(n) = TopK(rank^*) \tag{4}$$

$S^*(n)$ provides a composite score reflecting the neuron's significance across multiple perspectives, guiding IDC and further testing. In WISDOM, the initial training iterations serve as a *warm-up* stage, during which each

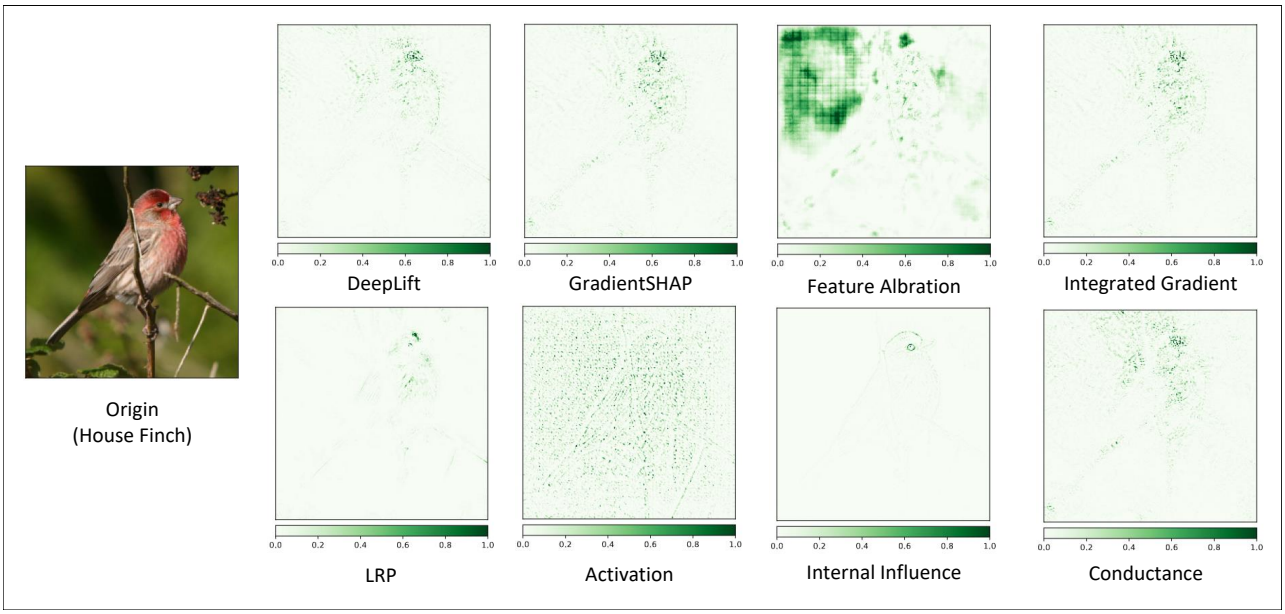


Figure 11: Attributions methods showcase: house finch (from ImageNet) with VGG16 model.

Algorithm 1: Important Neurons Selector

Input: DNN model, training data X_{train}, test data X_{test}, attribution set $\mathcal{A}$, number of neurons k
Output: Top-n neurons' importance scores $S^*(n)$

1 **Function** `ImportantNeuronsSelection(`$model, layer, X_{train}$`)`**:**
2 **for** *each* $m : m \in \mathcal{A}$ **do**
3 $\mathcal{R}^m \leftarrow$ `GetAttributions`$(model, X_{train}, m)$;
4 $\text{rank}_m \leftarrow$ `RankNeurons`$(\mathcal{R}^m)$;
5 $S_k^m \leftarrow$ `TopK`(rank_m);
6 $\Delta\mathcal{L}_m \leftarrow$ `PruneAndEval`$(model, S_k^m)$;
7 $w_m \leftarrow$ `NormalizeWeights`$(\Delta\mathcal{L}_m)$; `// Calculate weight as normalized Loss diff, follow Eq. 2`
8 $\text{rank}^* \leftarrow$ `ConsensusRank`(rank_m, w_m); `// Follow the Eq. 3 to obtain the aggregated rank`
9 $S^*(n) \leftarrow$ `TopK`(rank^*);
10 **return** $S^*(n)$

attribution method is evaluated and assigned weights. Attribution methods with lower IDC scores (detailed in Section 11) are assigned lower weights. Subsequently, the optimal attribution methods are selected as the primary approach for important neuron analysis in later stages. Algorithm 1 provides a detailed overview of our approach. We first obtain all training attributions and conduct pruning to choose the optimal attribution method with the largest loss gain (line 2 - 6), this preprocessing was done *one-off*. Then the loss gain will be further used as ranking weights (line 7). Finally, consensus ranking was used and our building phase is over (line 8 - 9).

In `GetAttributions()` (line 3 in Algorithm 1), we extract relevant features from both the model architecture and the specific task. These features may include characteristics of individual neuron layers as well as input features. For instance, methods known to perform well with CNN layers, such as GradCAM, may receive higher weights for those layers. Finally, methods are ranked by predicted scores, and the top-performing method(s) are selected for the given context.

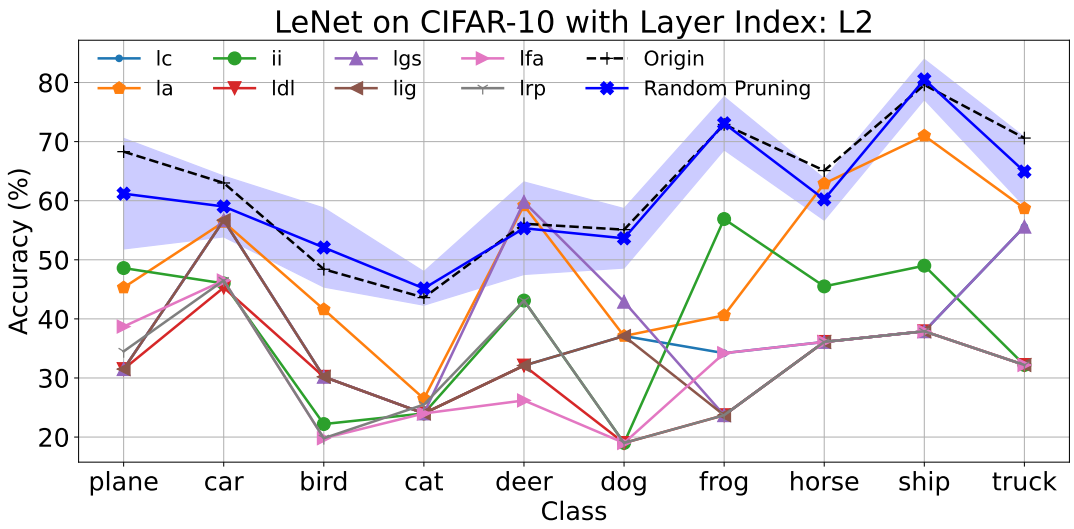


Figure 12: LeNet-5 second convolutional layer neuron (picking top-2 importance neurons) pruning performance with various attribution methods using CIFAR10 dataset.

As a motivating example, we demonstrate the situation in which Layer-wise Relevance Propagation (LRP) is chosen to operate within our system; other attribution methods are expected to function similarly. LRP is a backpropagation-based method developed to explain how a DNN output is generated by analyzing the contributions of individual neurons. LRP computes the "relevance" score for each neuron by propagating the final prediction score backwards through the network's layers. This approach aids in pinpointing which input components most significantly impact the network's decision, thereby offering a detailed layer-by-layer breakdown of the decision-making process. Given an input data $x \in Trainset$, the output $f(x)$ of our tested DNN model can be decomposed and the prediction score would be redistributed back to the neurons in previous layers in a manner that preserves the prediction. Mathematically, the propagation of relevance scores R_i at a given layer onto neurons of the lower layer is achieved by applying the rule:

$$R_i^l = \sum_k \frac{z_{ij}}{\sum_j z_{ij}} R_j^{l+1} \tag{5}$$

where z_{ij} represents the contribution of neuron i to neuron j in the forward pass and $\sum_j z_{ij}$ is the total contribution of all neurons in layer l to neuron j in layer $l+1$. R_j^{l+1} is the relevance of neuron j in the layer $l+1$. LRP serves as one of the attribution methods that help us identify important neurons of a given DNN model, and subsequently ensure that test sets adequately cover these critical components. Neurons with higher importance scores are deemed critical and require thorough testing to ensure the robustness of the DL system.

State-of-the-art testing adequacy criteria for DL systems, such as neuron coverage [45] and k-multisection neuron coverage [39], quantify test coverage solely based on neuron activations, without considering each neuron's contribution to the final decision. Consequently, a neuron may increase confidence in incorrect classes, a distinction that is not captured by these methods. Although advanced approaches like DeepImportance [15] incorporate the actual contributions of each neuron to a decision, they rest on the assumption that a chosen attribution method is consistently sensitive and valid across different datasets and models. This assumption, however, does not always hold. In Figure 12, we demonstrate that even within the same class of an image, different model layers can yield varying coverage rates. By leveraging the strengths of each attribution method, WISDOM establishes a robust foundation for enhanced testing adequacy.

Algorithm 2: Important Neurons Cluster

Input: $model, \mathcal{T}, num_m$

Output: Clustered vector Ψ of important neurons

1 $\Psi \leftarrow \emptyset$; `// Vector for clustered important neurons`
2 $c_{\text{opt}} \leftarrow 2$; `// Initial number of clusters`
3 $s_{\text{best}} \leftarrow -1$; `// Initialize the best Silhouette score`
4 **Function** `ChooseClusterConfigs(`Φ_n, C_{max}`)`**:**
5 **if** *using BO* **then**
6 $h^* \leftarrow$ `BOCluster(`$\Phi(X_{train}), T, \delta$`)` ; `// Bayesian optimization (Alg. 3)`
7 c_{opt}, ClusterMethod, $*args \leftarrow h^*$; `// Clustering configs from BO`
8 **else**
9 ClusterMethod $\leftarrow$ KMeans;
10 $*args \leftarrow$ KMeans Default;
11 **for** $c \leftarrow 2$ **to** C_{max} **do**
12 Cluster Labels $\leftarrow$ KMeans(Φ_n, c) ; `// Activation values into` c `clusters`
13 $s \leftarrow$ SilhouetteScore$(\Phi_n,$ Cluster Labels$)$; `// Calculate the Silhouette score`
14 **if** $s > s_{best}$ **then**
15 $s_{\text{best}} \leftarrow s$; `// Update the best Silhouette score`
16 $c_{\text{opt}} \leftarrow c$; `// Update the optimal number of clusters`
17 configs $\leftarrow c_{\text{opt}}$, ClusterMethod, $*args$;
18 **return** configs ; `// Return the clustering setups`
19 **for** *all* $n \in model$ **do**
20 $\Phi_n = (\phi(t, n)), t \in \mathcal{T}$; `//` n`-th neuron activation values`
21 configs = `ChooseClusterConfigs(`Φ_n, num_m`)`;
22 $\Psi_n =$ Cluster$(\Phi_n,$ configs$)$;
23 $\Psi = \Psi \cup \Psi_n$; `// Collect cluster vectors`
24 **return** Ψ

Neuron Clustering and Coverage

Once neuron importance is determined, the WISDOM framework further refines the testing process by clustering the activation values of the important neurons. This approach partitions the activation space into meaningful clusters, which represent different behavioral states of the neurons.

We implement neuron clustering primarily for the following reasons:

- Clustering based on semantically distinct features of each neuron provides a more meaningful partitioning than randomly segmenting activation values into equal widths.
- Clustering simplifies the analysis by reducing the complexity of examining individual activation values $\Phi(x, n)$ to analyzing combinations within each cluster.
- Identifying clusters for important neurons guides the allocation of testing resources to ensure that critical neuron regions are thoroughly tested.

Algorithm 2 presents the detailed implementation of our neuron clustering approach, utilizing *k-means* and applying the Silhouette Score to dynamically select the optimal number of clusters for each neuron. In Algorithm 2, function `ChooseClusterConfigs()` (lines 4–11) enhances cluster extraction by incorporating

Algorithm 3: `BOCluster`: Bayesian Optimization for Optimal Clustering

Input: Important neurons' activations $\Phi(X_{train})$, training set X_{train}, budget T, and threshold δ
Output: Clustering configuration h^*

Init($\mathcal{H}$, GP);
for $t = 1 \rightarrow T$ **do**
  $h_t \leftarrow \arg\max_{h \in \mathcal{H}} \mathrm{EI}(h)$; `// Acquisition: Expected Improvement`
  for ***for each*** *neuron* $n \in N_{top}$ **do**
    $\Psi_n(h_t) \leftarrow \texttt{Cluster}(\{\phi(x, n) : x \in X_{\text{train}}\},\ h_t)$
    $C_n(h_t) \leftarrow \{\texttt{Centroid}(\Psi_n^i) : 1 \leq i \leq |\Psi_n(h_t)|\}$; `// Per-neuron cluster representatives`
  $Cov(X_{train}) = \texttt{Coverage}\,(\mathcal{C}, \Phi(X_{train}) \in \Psi(X_{train}))$; `// Combination` $\mathcal{C}$ `with assignments` h_t
  $PC(X_{train}) = \texttt{corr}(Cov(X_{train}), Score(Y_{train}))$; `// Compute correlation using Eq. 8`
  $GP \leftarrow \texttt{Update}(h_t, PC(X_{train}))$ **if** $(\mathrm{EI}(x) < \delta)$ **then**
    break
$h^* = \arg\max_{h_t} PC(X_{train})$
return h^*

the Silhouette index, facilitating the rapid and dynamic selection of the optimal number of clusters for the important neuron sets. The Silhouette index is an internal clustering validation metric that evaluates the quality of a clustering structure independently of external information. An iteration through the important neurons (lines 12-16) begins by capturing each neuron's activation values (line 13) and with the Silhouette index, we establish the clusters such that $\Psi_n = \bigcup_{1 \leq i \leq c_{opt}} \Psi_n^i$, where Ψ_n^i is the vector containing the activation values for the i-th cluster (line 22). In this deliverable, k-means remains the default clustering method and serves as the baseline configuration for WISDOM (line 12). However, our approach is flexible and can accommodate various clustering algorithms, including density-based, grid-based, and hierarchical clustering methods. In Table 5, we additionally investigate a lightweight Bayesian Optimization stage (line 5-7 in Algorithm 2, and also see Algorithm 3) that tunes the clustering strategy and a small set of hyperparameters at design time. This optional tuning does not modify WISDOM itself, but refines the partitioning of important neuron activations so that the resulting clusters better reflect the behavior of deeper models. By testing the model against a test set that covers all possible combinations of these important neuron clusters, this coverage criterion can quantify how well the model's critical decision-making regions are being exercised.

Importance-driven Coverage

By identifying important neurons, we can measure the extent to which the test set covers these neurons, a metric we term *Importance-Driven Coverage*. IDC is an advanced testing criterion designed to assess the adequacy of DL models by focusing on the most critical components that influence a model's decision-making process. Unlike traditional coverage metrics, which often rely on neuron activation thresholds or distribution-based partitions, IDC emphasizes the significance of individual neurons or groups of neurons that contribute most to a model's output. This technique enhances the semantic diversity of a test set, ensuring it activates the most influential internal behaviours of the DL system. The semantic adequacy provided by IDC allows engineers to evaluate whether a model will respond appropriately across various scenarios, including adversarial inputs or corner cases. By concentrating on important neurons, IDC increases the test set's ability to uncover erroneous behaviours, boosting confidence in the model's deployment in real-world settings.

Method	Hyperparameter	Range/Choices	Description
KMeans [26, 29]	`n_cluster`, `init`	[2, 10], {k-means++, random}	Number of clusters; init method
MiniBatchKMeans [50]	`n_cluster`, `batch_size`	[2, 10], [32, 512]	Number of clusters; mini-batch size
DBSCAN [12]	ε, `min_samples`	[0.1, 10.0], [3, 20]	Neighborhood radius; minimum samples
AgglomerativeClustering [4	`n_cluster`, `linkage`	[2, 10], {ward, average, complete}	Number of clusters; linkage strategy
MeanShift [8]	`bandwidth`	[0.1, 5.0]	Kernel bandwidth for shifting
SpectralClustering [43]	`n_clusters`, `affinity`	[2, 10], {rbf, `nearest_neighbors`}	Number of clusters; similarity metric
OPTICS [2]	`min_samples`, ξ	[3, 20], [0.01, 0.2]	Minimum samples; steepness threshold
Birch [64]	`threshold`, `n_clusters`	[0.1, 2.0], [2, 10]	Radius threshold; number of clusters
AffinityPropagation [13]	`damping`	[0.5, 0.95]	Damping factor for responsibility update

Table 5: Search space for clustering methods used in WISDOM.

From Equation 5, we assume that the tested layer L has relevance scores R_i, representing the *importance* score of each neuron i. Given a test set Y, we aim to ensure that Y adequately triggers combinations of activation value clusters for the neurons i, thereby achieving comprehensive coverage. This approach enables testing of the most influential neurons, offering confidence in the DL system's correct operation and reducing the risk of incorrect decisions. We define the Importance-Driven Neuron Cluster Combination (INCC) for each important neuron in layer L:

$$INCC = \prod_{n \in R_i} \left\{ \Psi_n^i \mid i = 1, 2, \ldots, K_n \right\} \tag{6}$$

INCC represents the set of cluster centroids for the neurons in layer L, where Ψ_n^i is the centroid of the i-th cluster for neuron n using *k-means*. K_n is the number of clusters for neuron n, determined dynamically via Silhouette score. The total number of possible combinations should therefore be $|INCC| = \prod_{n \in R_i} K_n$.

Given a test set $\mathcal{D}_{test}$, we define IDC as:

$$IDC(\mathcal{D}_{test}) = \frac{\left\{ INCC_j \mid \exists x \in \mathcal{D}_{test}, \forall V_n^i \in INCC_j, d(\Phi(x, n), V_n^i) \leq \epsilon \right\}}{|INCC|} \tag{7}$$

where $\Phi(x, n)$ represents the activation value of neuron n for the test sample x and V_n^i represents the cluster centroid for neuron n in a particular INCC combination $INCC_j$. We also specify a condition requiring that the *Euclidean Distance* between the activation value Φ and the cluster centroid V_n^i be within a small tolerance ϵ for the cluster centroid to be considered covered by the test sample.

In Table 5, we show all the clustering method together with their hyperparameters we have considered in WISDOM framework. By default, WISDOM adopts KMeans with the Silhouette [47] to automatically choose the number of clusters. However, we also offer a searching method that see all clustering methods as a black-box searching space. Such a combined search space across all clustering methods includes a mix of discrete choices and continuous hyperparameter ranges. With 10 methods and 1–2 hyperparameters each (many with 5-10 valid values), the total combinatorial space spans several hundred to a few thousand configurations, which

is large enough for us to effectively use an optimization like Bayesian Optimization (BO) [51, 4] to get a good configuration with a reasonable number of trials (≤ 30).

In WISDOM, we integrate BO to explore the search space to rapidly identify optimal regions, typically converging within fewer than 30 iterations given suitable initialization and acquisition functions. We utilize an Expected Improvement (EI) acquisition function in BO, guided by the Pearson correlation between the coverage rate $Cov(X)$ and the F1 scores $Score(Y)$:

$$PC(X) := \text{corr}(Cov(X), Score(Y)). \tag{8}$$

A higher positive correlation indicates that increased coverage aligns with accurate class predictions, thus reflecting a desirable clustering configuration. The optimization objective can therefore be expressed as:

$$h^* = \arg\max_{h \in \mathcal{H}} PC(X). \tag{9}$$

Since each evaluation of $PC(X)$ involves computationally expensive processes (forward passes and clustering operations), we treat $PC(X)$ as a black-box function modeled using a Gaussian Process (GP):

$$PC(X) \sim GP(\mu(x), k(x, x')) \tag{10}$$

We adopt a Matérn 5/2 kernel [5] for continuous variables and a Hamming kernel to encode all the categorical data into a *one-hot* vector. We leverage BOTorch [4] to fit the Gaussian Posterior (GP) by maximizing marginal log-likelihood and computing EI, defined as:

$$\text{EI}(x) = \mathbb{E}_{PC}\left[\max(0, PC(X) - PC(\text{best})\right] \tag{11}$$

For a fixed training set X_{train}, our BO pipeline is shown in Algorithm 3. The key idea for BO is to efficiently search the clustering configuration space by treating $PC(X)$ (Equation 10) as a black-box objective. At each iteration (line 2 in Algorithm 3), BO selects a candidate configuration (line 3) via EI (Equation 11), clusters neuron activations (line 5), computes coverage (line 6), and evaluates its correlation with predictive performance (line 7). The Gaussian Process surrogate is updated accordingly (line 8), enabling rapid convergence to a configuration that maximizes the alignment between coverage and model accuracy.

Finally, in the testing phase, according to Equation 7, each testing input feature is mapped to at most one particular neuron combination. The IDC score increases only when the mapped INCC element tests a new feature set not already covered by the existing test suite; otherwise, the score remains unchanged. Achieving a high IDC score thus demonstrates the diversity of the test suite and indicates that critical neurons have been activated, increasing confidence in the DL system's reliability. Additionally, IDC uniquely measures how effectively multiple inputs with semantically diverse features can activate INCC.

3.1.2 Improved DNN Testing through Data Augmentation

We propose a data augmentation technique, SEMAUG, which is based on coverage-guided fuzzing (CGF) and enhances the visual data quality and semantic diversity of the input dataset. The proposed model employs an image dataset to be trained and outputs synthetic images, which are semantically meaningful. A fidelity score is used to analyze the output synthetic images in terms of photo-realism. The selected images are later added to the original dataset to enlarge the dataset feature space.

The newly created images can be employed for enhancing the model's performance metrics, by incorporating them in the original dataset and re-training the model on the augmented dataset. They can also be exploited for testing the DNN's robustness by evaluating its performance in newly created image instances, which depict

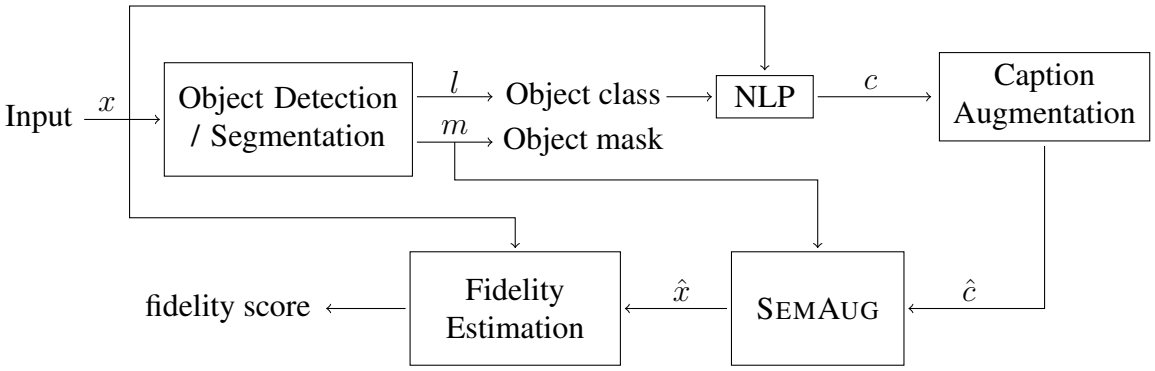


Figure 13: Overview of the data augmentation pipeline.

previously-unseen scenarios/configurations. Figure 13 illustrates an overview of the SEMAUG framework. The output of the augmentation pipeline is a fidelity score, which is deployed at Equation 13 to assess if the augmented image produced is valid or not, based on the threshold value δ, described later in this section. Firstly, an image sample is parsed through an object detector/segmentation component, which outputs the predicted object class as well as the object mask. The object class with the original image are parsed through an *NLP* module, aiming to produce a caption for the object of interest. This caption is sequentially parsed from a *Caption Augmentation* module, producing an augmented caption. The augmented caption with the detected object mask are employed as an input in the SEMAUG component to produce an augmented image based on the augmented caption. A similarity comparison is performed between the augmented and the original image by considering a fidelity estimation measure, which produces a fidelity score.

Semantic Data Augmentation

The generation of synthetic image data, using a semantic-based coverage-guided fuzzing method for DNN models, can be considered as the problem of adding perturbations ϵ on the original input sample x, which output a new sample x' that is semantically within the same data domain as the input sample x. More specifically, the objective is to minimize the overall perturbation ε and maximize the adequacy criterion λ, i.e., the coverage score $Cov(x')$ of the synthetic sample, so that $x' \in X$ holds. Thus, the objective function is defined as,

$$E_{x \sim A}\left[\min_{\varepsilon} X(x+\epsilon), \max_{\lambda} Cov(x')\right] \tag{12}$$

where A represents the augmentation technique of a DNN model, which outputs synthetic samples.

The optimization problem stated in Equation 12 is solved by deploying a text-conditioned generative AI model [23], which enables editing specific parts of a sample, i.e., image, by providing a mask and a text prompt to generate augmented seeds automatically. More specifically, diffusion model inpainting can be performed by sampling and replacing a region, i.e., mask, of an image sample with a sample from the training dataset. The text-conditional aspect of our approach provides a mechanism to control and adjust through textual prompts the mutations performed on the original image sample.

SEMAUG uses an input image sample and automatically outputs a mask and a prompt for every input. Mask R-CNN [21] is employed for producing segmentation masks m_i of detected instances of objects x_i corresponding to a class c_i. Natural language processing [55] is used for generating a prompt employing the detected object class and extracting a named entity from the input description y_i and replace it with a newly selected named entity. The generative AI model exploits this information and replaces the masked region of the input image with the target object, thus generating a synthetic image sample.

Fidelity Score	Domain
MSE	pixel-wise
PSNR	pixel-wise
FID	feature distribution
SSIM	lumination, contrast, and structure features
LPIPS	learned perceptual features
VIF	information-theoretic

Table 6: Fidelity scores and the domain they are applied on.

Fidelity Score Estimation

To evaluate the photo-realism of the synthetically generated images, a visual and textual fidelity score is employed, allowing us to keep only the outputs for which their fidelity score exceeds a certain pre-defined threshold value δ. Equation 13 describes this criterion,

$$\text{isValid}(\hat{x}, Q) = \begin{cases} True, & \text{if } f(\hat{x}, Q) \geqslant \delta \\ False, & \text{otherwise} \end{cases} \tag{13}$$

where f denotes the fidelity assessment technique, $\hat{x}$ represents the augmented image, and Q is the fidelity. The fidelity can be any of the fidelity scores, or any combination of them. They provide a quantitative measure, identifying the acceptable deviation between the original and the augmented images. δ denotes the threshold of the acceptable deviation and it is defined empirically as well as it is fidelity- and domain-specific. The fidelity scores employed for evaluating the similarity between the original and the augmented images are summarized in Table 6, and are further explained in this subsection.

Mean Squared Error (MSE). MSE is the L_2 norm of the arithmetic difference between two images, which is computed based on Equation 14,

$$MSE(x, \hat{x}) = \frac{1}{n \cdot m \cdot c} \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{z=1}^{c} (x_{ijz} - \hat{x}_{ijz})^2 \tag{14}$$

where, x and $\hat{x}$ represent the two images, and n, m, and z denote the width, height, and channels of the images.

Peak Signal-to-Noise Ratio (PSNR). PSNR is employed to compare images on the pixel-level, with higher PSNR values indicating higher similarity. The PSNR value between an original and an augmented image is computed as,

$$PSNR(x, \hat{x}) = 10 \cdot log_{10} \frac{max(x)^2}{MSE(x, \hat{x})} \tag{15}$$

where, x and $\hat{x}$ represent the original and the augmented image, respectively. $max()$ computes the maximum value of the argument provided, in the case where the argument is an image then the function returns the maximum pixel value of that image. Higher PSNR score indicates better quality of the augmented image.

Fréchet Inception Distance (FID). FID [44] is a metric used to compare the quality between two images by considering their feature distribution $\mathcal{N}(\mu, \Sigma)$. This feature distribution can be obtained from one of the layers of a trained neural network. Equation 16 is used for the FID score calculation,

$$FID(x, \hat{x}) = ||\mu - \hat{\mu}||^2 + tr(\Sigma + \hat{\Sigma} - 2(\Sigma\hat{\Sigma})^1/2) \tag{16}$$

where, x and $\hat{x}$ represent the original and the augmented image, with feature distributions $\mathcal{N}(\mu, \Sigma)$ and $\mathcal{N}(\hat{\mu}, \hat{\Sigma})$, respectively. μ and Σ denote the mean feature vector and the covariance matrix for the features of an input image. The function $tr()$ computes the trace of a matrix, i.e. the summation of it's diagonal elements.

Structural Similarity Index (SSIM). SSIM [6] performs comparison between two images by exploiting their luminance, contrast, correlation and structure. SSIM is computed using Equation 17 in a channel-bases; if more than one channels are present in an image, then the SSIM value is computed as the mean of the SSIM values of every channel. Specifically, the SSIM value is calculated as,

$$SSIM(x, \hat{x}) = \frac{(2\mu_x\mu_{\hat{x}} + c_1)(2\sigma_{x\hat{x}} + c_2)}{(\mu_x^2 + \mu_{\hat{x}}^2 + c_1)(\sigma_x^2 + \sigma_{\hat{x}}^2 + c_2)} \tag{17}$$

where, x and $\hat{x}$ represent the two images being compared, which should be of the same dimensions, μ_i and σ_i^2 are the mean of the pixels for any image i and it's variance, respectively. $\sigma_{x\hat{x}}$ notes the covariance of the two images x and $\hat{x}$. To stabilize the division, c_1 and c_2 are used, where $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$, with L[1] capturing the dynamic range of the pixel values, and $k_1 = 0.01$ and $k_2 = 0.03$.

SSIM values range from -1 to 1, with higher SSIM values indicating higher similarity of the two images.

Learned Perceptual Image Patch Similarity (LPIPS). LPIPS [63] measures perceptual similarity by exploiting a deep neural network, i.e., AlexNet, VGG, and comparing high-level visual representations. Equation 18 summarizes the computation of the LPIPS score between two images,

$$LPIPS(x, \hat{x}) = \sum_{l \in L} \frac{1}{H_l \cdot W_l} \Big(\sum_{h \cdot w} ||w_l \odot (x_{hw}^l - \hat{x}_{hw}^l)||_2^2 \Big) \tag{18}$$

where, x and $\hat{x}$ represent the original and the augmented image, L denotes the number of layers of the neural network, w_l are the weights of the network at layer l.

Visual Information Fidelity (VIF). VIF [52] is derived from two mutual information quantities: the mutual information between the input and the output of the HVS channel of the original image, and the mutual information between the input and the output of the HVS channel for the augmented image. Overall, the VIF score evaluates the amount of information preserved in the augmented image by computing:

$$VIF(x, \hat{x}) = \frac{\sum_{j \in \text{subbands}} I(C^{N,j}; F^{N,j} | s^{N,j})}{\sum_{j \in \text{subbands}} I(C^{N,j}; E^{N,j} | s^{N,j})} \tag{19}$$

where, $I()$ represents the information that could be extracted from an image. $I(C_N; E_N | s_N)$ and $I(C_N; F_N | s_N)$ denote the amount of information that can be extracted from the outputs of the HVS for the original image and augmented image, respectively, for a specific element N of the original image, and for model parameters s_N. VIF score values range from 0 to 1, with values closer to 1 representing an image which is very close to the original image, and values closer to 0 denote a small amount of shared information between the original and the generated image.

[1] L is usually $2^{\#\text{bits per pixel}} - 1$.

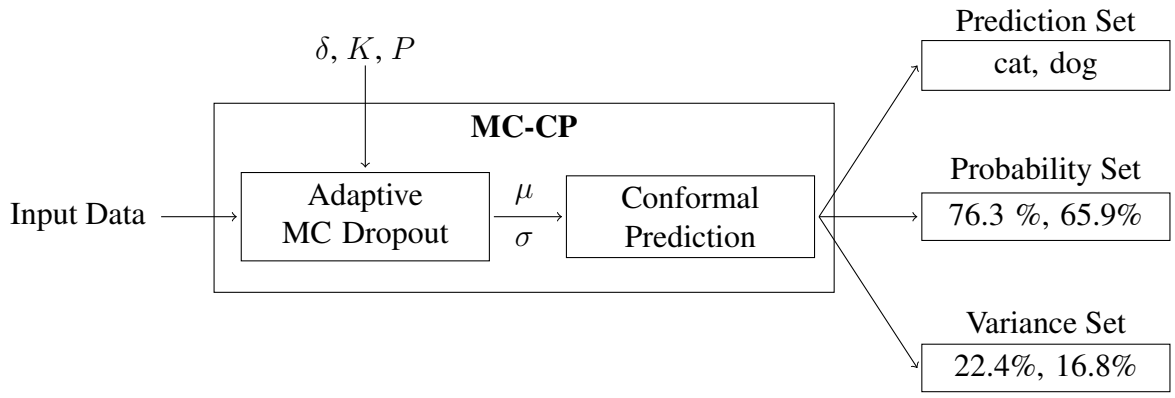


Figure 14: Overview of the MC-CP method, specifically for classification.

Enhancing the Dataset's Semantic Diversity

SEMAUG aims to identify and introduce new semantically meaningful image data to the test set. The semantic diversity of these samples is quantified based on their ability to maximize the selected coverage criteria when added to the test set. Several test adequacy criteria exist for analyzing the inner behavior of a neural network model. SEMAUG's current implementation supports neuron coverage (NC) [45], DeepImportance's IDC [15], likelihood-based surprise adequacy (LSA) [30], and the neuron-level criteria k-Multisection Neuron Coverage (KMNC) and Neuron Boundary Coverage (NBC) from DeepGauge [39].

3.1.3 Uncertainty Estimation of AI Models

Uncertainty estimation methods provide an approximation of an AI model's confidence per prediction by considering the effect of randomness and model misspecification. In this section, we introduce the Monte Carlo Conformal Prediction (MC-CP) method to address the tasks of classification and regression, as well as a Box-area Bin-based Conformal Prediction (BB-CP) method for the task of detection. For both MC-CP and BB-CP, a calibration (C) set and a validation (V) set, are used.

Uncertainty Estimation for Classification and Regression

MC-CP Firstly, MC-CP is a hybrid uncertainty quantification approach that estimates robust prediction intervals with the complete predictive distribution, yielding an uncertainty estimate. The method is based on Monte Carlo (MC) dropout [14] and conformal prediction [59], and a high-level overview of the approach for classification is illustrated in Figure 14.

An adaptive MC dropout approach is employed to enable the reduction of the number of not-needed forward passes when convergence has been granted. Algorithm 4 presents the adaptive MC dropout method used, where f represents the predictive model employed, $x \in \{C, V\}$ the input, K is the number of maximum forward passes, and $\delta \in (0, 1)$ and $P \in \mathbb{Z}_+$ represent the hyperparameter threshold and patience, respectively, for detecting convergence and early termination. δ acts as the maximum difference in variance required to denote that the class/quantile prediction has likely converged. Patience P signifies the number of consecutive forward passes, where all classes/quantiles are below δ to stop the execution early. A predictive model f considers as input x, and K forward passes are performed, to produce the mean of the predictive posterior as the final prediction and the variance of the predictive posterior as the prediction uncertainty.

Specifically, Algorithm 4 uses as an input the predictive model f, the input sample x, the number of maximum forward passes K, the variance threshold δ and patience P. The algorithm runs for a number of times, until either the number of forward passes reaches K or the variance difference between two consecutive predictions is lower than the variance threshold δ for P number of times.

Algorithm 4: Adaptive Monte Carlo Dropout

Input: f, x, K, δ, P
Output: μ, σ

```
Count ← 0
Predictions ← []
while (Count < P)&(size(Predictions) < K) do
    y ← f(x)
    Predictions ← Predictions ∪ y
    σ ← Var(Predictions)
    if (Predictions > 1) then
        diff ← |σ_prev − σ|
        if (∀z : diff_z ⩽ δ) then
            Count ← Count + 1
            else
                Count ← 0
    σ_prev ← σ
return Predictions, σ
```

In each run, the model f produces a prediction using the input x. That prediction is added in the list of predictions, which in the end will aid to compute the mean of the predictive posterior. The variance is computed from the list of predictions. Consequently, the difference between the computed variance and the variance from the previous run is calculated and checked if it exceeds the threshold δ. If that is the case, then a counter is increased by 1. This counter checks if the patience P is reached. Finally, the computed variance is kept for the next run.

Classification For addressing the task of classification, MC-CP is formed by combining the Adaptive MC Dropout method with conformal prediction. Algorithm 5 describes the two stages of MC-CP, where f represents the model employed, K denotes the number of maximum forward passes, and δ and P represent the threshold and patience hyperparameters used for the Adaptive MC Dropout method.

Conformal calibration is the initial stage of MC-CP, where a test set is used and split into a calibration set C and a validation set V. Platt scaling is then performed on the pre-trained model f using the calibration set. For every sample the conformal scores are calculated, which are then used to compute the quantile threshold $\hat{q}$.

The second stage of MC-CP is the conformal prediction stage. During this stage the outputs of the Adaptive MC Dropout method, i.e. the mean prediction and the variance, are used to determine the final prediction set and variance set. More specifically, the cumulative softmax output of all classes is calculated and ranked from most to least likely not exceeding the quantile threshold.

Regression Algorithm 6 presents an extension of the MC-CP method for deep quantile regression. A model f is used, K represents the number of maximum ensembles, and δ and P denote the threshold and patience hyperparameters of the Adaptive MC Dropout method. Firstly, during the conformal calibration stage, the conformal scores are approximated based on the magnitude of error for the desired quantiles, and the threshold is computed in regards to the calibration set.

During the conformal prediction stage, the Adaptive MC Dropout method is employed on the validation. Finally, a prediction interval is computed on unseen data points of the validation set using the calculated threshold from the conformal calibration stage.

Algorithm 5: MC-CP for classification

Input: f, C, V, K, δ, P

Output: Prediction set, variance set

Function `Conformal Calibration()`**:**

  Calibrate: perform Platt scaling on the model using C.

  Calculate conformal score: for each sample in the calibration set, define $E_j = \sum_{i=1}^{k'}(\hat{\pi}_{(i)}(x_j) + \lambda_1[i > k_{reg}])$, where k' is the model's ranking of the true class y_i and $\hat{\pi}_{(i)}(x_j)$ is the i^{th} largest score for the j^{th} sample in C.

  Find the threshold: assign $\hat{T}_{ccal}$ to the $1-\alpha$ quantile of the E_j.

Function `Conformal Prediction()`**:**

  Mean softmax: retrieve softmax and variance from Adaptive Monte Carlo Dropout (f, V, K, δ, P).

  Prediction set: output the k^* highest-scored classes, where $E_{i=1}^{k^*} = \sum_{i=1}^{k'}(\hat{\pi}_{(i)}(x_{n+1}) + \lambda_1[j > k_{reg}) \geqslant \hat{T}_{ccal}$.

Algorithm 6: MC-CP for deep quantile regression

Input: f, C, V, K, δ, P

Output: Prediction interval, variances

Function `Conformal Calibration()`**:**

  Calculate conformal score: for each data point in C, define $E_j = max\{\hat{q}_{\alpha_{lo}}(x_i) - y_i, y_i - \hat{q}_{\alpha_{hi}}(x_i)\}$.

  Find the threshold: compute $Q_{q-\alpha}(E, C)$, the $(1-\alpha)(1+1/|C|)$-th empirical quantile of $\{E_i : i \in C\}$.

Function `Conformal Prediction()`**:**

  Mean softmax: retrieve softmax and variance from Adaptive Monte Carlo Dropout (f, V, K, δ, P).

  Prediction Interval: output the prediction interval $I(V) = [\hat{q}_{\alpha_{lo}}(V) - Q_{a-\alpha}(E, C), \hat{q}_{\alpha_{hi}}(V) + Q_{1-\alpha}(E, C)]$ for unseen validation data V.

Uncertainty Estimation for Detection

BB-CP On the other hand, performing uncertainty estimation for the detection task, entails addressing both the classification, as well as the regression task, referring to the detected class labels and bounding boxes' coordinates, respectively. Figure 15 illustrates the methodology of quantifying uncertainty for detection. Specifically, image data frames $\{x_1, x_2, \cdots, x_N\} \in \{C, V\}$, are used as inputs in a deep learning detection model. If the predictive model contains Dropout layers, then the Adaptive MC Dropout functionality, as described in Algorithm 4, is enabled during the Inference Sampler stage. Otherwise, a single inference step is performed and the outputs of the model are then processed by the Conformal Prediction stage. In the former case, the mean of predictive posterior (μ) is considered as the prediction ($\hat{x}$) to be processed by Conformal Prediction, where detected objects, between K forward passes, are grouped together whilst using Hungarian algorithm, by employing their bounding boxes' overlap. f denotes the Inference Sampler method. Thus, every input x_i, where $i \in [1, N]$, outputs d_i number of detections of possible objects, i.e. $\hat{x}_i = f(x_i)$, where $\hat{x}_i = \{\hat{x}_i^{(1)}, \hat{x}_i^{(2)}, \cdots, \hat{x}_i^{(d_i)}\}$. For every detection $\hat{x}_i^{(j)}$ $(0 \leq j \leq d_i)$, during inference, the uncertainty estimation method provides as set of outputs concerning the class prediction, and another set of outputs for the detected bounding boxes.

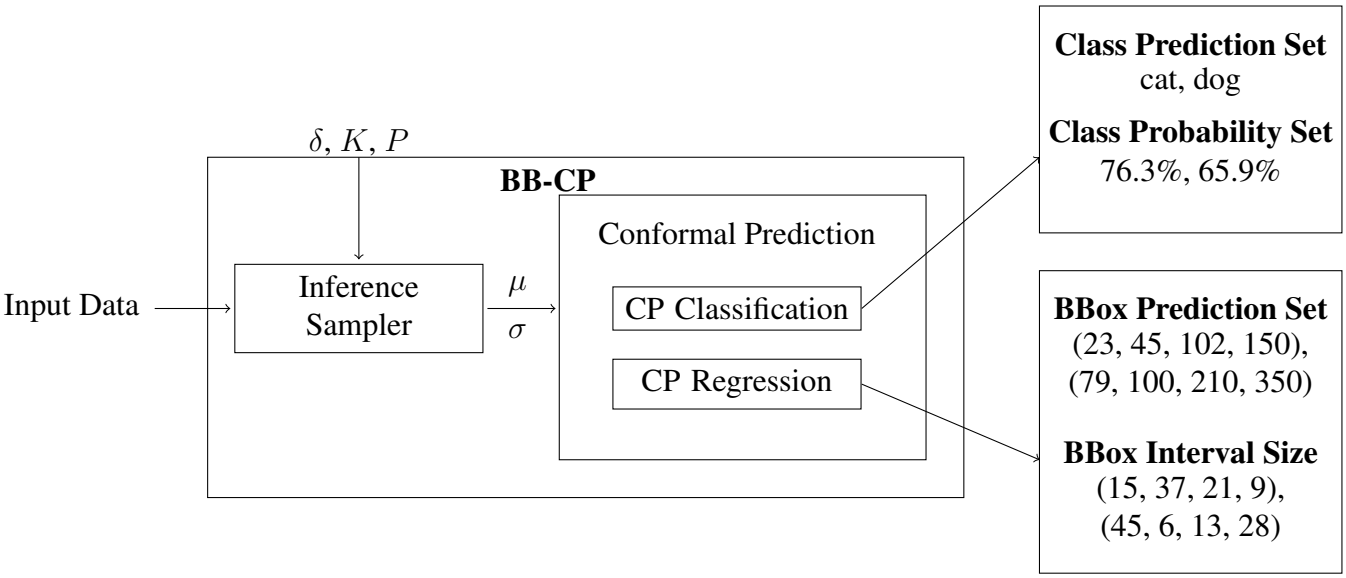


Figure 15: Overview of the BB-CP method for the detection task.

Detection To address the task of detection, we focus on both the class labels, and the bounding box coordinates. For the class labels, we handle the classification task by employing a class thresholding algorithm as described in Algorithm 7. Initially, we define $\hat{X} = f(X)$, where $\hat{X} = \{\hat{x}_1, \hat{x}_2, \cdots\}$ denotes the set of predictions for the X image set. Every detection $j \in d_i$ of a prediction i is a vector of the form: $\hat{x}_i^{(j)} = [\hat{l}_i^{(j)}, p_i^{(j)}, \hat{\pi}_i^{(j)}, \hat{c}_i^{1,(j)}, \hat{c}_i^{2,(j)}, \hat{c}_i^{3,(j)}, \hat{c}_i^{4,(j)}]$, where $\hat{l}$ denotes the predicted label, p the predicted probability, i.e. confidence score of the prediction, $\hat{\pi}$ corresponds to a vector of the softmax values for all the classes ($\hat{\pi}_i^{(j)} \in \hat{\pi}_i = \hat{\pi}(x_i)$), and $(\hat{c}^1, \hat{c}^2, \hat{c}^3, \hat{c}^4)$ is the collection of coordinates of the predicted bounding box. During calibration, a quantile (Q^y) is computed for every class $y \in Y$, where Y is the calibration ground truth set. Whilst, during prediction, the appropriate quantile based on the detected class is employed to define the prediction set. More specifically, the softmax values ($\hat{\pi}$) of the detections are exploided to compute the conformal scores, which are in turn used to define the quantile thresholds during calibration. At inference time, the conformal scores aid in defining the prediction set.

Algorithm 7: Class Thresholding for class detection

Input: C, Y, V

Output: Prediction set, variance set

1 **Function** `Conformal Calibration()`**:**

2 **Calculate conformal scores:** for each sample in the calibration set C, define $E(x) = 1 - \hat{\pi}(x)$, where $\hat{\pi}(x)$ is the softmax values of the input x.

3 **Find the threshold:** for each label y, compute $Q^y_{1-\alpha}$, which is the $1-\alpha$ quantile of the G_y, where $G_y = \{E_l(x) | \forall l : l = y\}$, where $E_l(x)$ represents the conformal score of an input x of class l.

4 **Function** `Conformal Prediction()`**:**

5 **Prediction set:** for each detection j output the classes, where, $E_l(x_i) = 1 - \hat{\pi}_i^{(j)} \geqslant Q^l_{1-\alpha}$, $\forall x \in V$, and l denoting the class.

For the bounding box coordinates, we employ Algorithm 8 to address the regression part of the detection task. The idea is based on standard conformal however, quantiles are computed based on box-area bins. More specifically, detected bounding boxes from the calibration data are grouped into equal-mass bins ($\mathcal{B} \in \{b_1, \cdots, b_B\}$) based on their computed area. Equal-mass bins entails that each bin will comprise $m = \left\lfloor \frac{|C|}{B} \right\rfloor$

Algorithm 8: Standard Conformal for detected coordinates of bounding boxes

Input: $f, C, Y, V, \mathcal{B}$

Output: Prediction set, prediction interval, variance set

Function `Conformal Calibration()`**:**

- **Calculate conformal scores:** for each sample in the calibration set C, define $E = |\hat{x} - y|$, where $\hat{x} = f(x)$, and $x \in C$.
- **Find the threshold:** compute $Q^b_{c^i}(E)$ for each coordinate c^i, which is the $1 - \alpha$ quantile of G_b, where $G_b = \{E_r | \forall r \in b\}$, where $b \in \mathcal{B}$, and $\mathcal{B} =$ BinComputation().

Function `Conformal Prediction()`**:**

- **Prediction Set:** compute the prediction set for each coordinate c^i of every detection $\hat{x}^{(j)}$ of some input x from the validation set V, $P(\hat{x}^{(j)}) = \{[\hat{c}^i - Q^b_{c^i}, \hat{c}^i + Q^b_{c^i}] : \forall i \in \{1, 2, 3, 4\}\}$.
- **Prediction Interval:** output the prediction coordinate intervals, $I(\hat{x}^{(j)}) = \{2 \cdot Q^b_{c^i} : b = \text{FindBin}(\hat{x}^{(j)}), \forall c^i | i = \{1, 2, 3, 4\}\}$.

Algorithm 9: Bounding Box Area-based Bins

Input: f, C, Y, V, m

Output: Prediction set, prediction interval, variance set

Function `Bin Computation()`**:**

- **Calculate box area:** for each sample x in the calibration set C, where $\hat{x} = f(x)$, compute the bounding box area, $A(\hat{x}) = \{|\hat{c}^{3,(j)} - \hat{c}^{1,(j)}| \cdot |\hat{c}^{4,(j)} - \hat{c}^{2,(j)}| : \forall j \in [1, d]\}$, where d represents the total number of detections, and $(\hat{c}^1, \hat{c}^2, \hat{c}^3, \hat{c}^4)$ denote two (x, y) pairs of diagonal coordinates. Consider the sorted set of the box areas from low to high, $A_{sorted}(\hat{x})$.
- **Assign boxes to bins:** create groups of the calibration data based of the box areas and the number of bins, $G_b = \{A^i_{sorted} | i \in [j \cdot m, (j + 1) \cdot m]\}, \forall b_j \in \mathcal{B}$.
- **Define bin thresholds:** create groups of the ground truth calibration data, $G'_b = \{A'^i_{sorted} | i \in [j \cdot m, (j + 1) \cdot m]\}, \forall b_j \in \mathcal{B}$, where A'_{sorted} represent the sorted set of the calibration ground truth box areas from low to high. Then, the bin thresholds are defined, $T(b) = (min(G'_b), max(G'_b)), \forall b \in \mathcal{B}$.

Function `Find Bin()`**:**

- **Calculate box area:** for each sample x in the validation set V, where $\hat{x} = f(x)$, compute the bounding box area of a single detection (j), $A(\hat{x}^{(j)}) = |\hat{c}^{3,(j)} - \hat{c}^{1,(j)}| \cdot |\hat{c}^{4,(j)} - \hat{c}^{2,(j)}|$, where $(\hat{c}^1, \hat{c}^2, \hat{c}^3, \hat{c}^4)$ denotes two (x, y) pairs of diagonal coordinates.
- **Find bin:** search in T to find the bin that, $b(\hat{x}^{(j)}) = \{b \in \mathcal{B} | A(\hat{x}^{(j)}) \in T(b)\}$.

samples. Algorithm 9 shows how bins are defined, how data are assigned to different bins, as well as how bins are selected during inference. During prediction, the data box area is computed and assigned to the appropriate bin, which in turn defines the quantile $Q^b_{c^i}$ to employ for each coordinate $c^{i,(j)}$, where $i \in \{1, 2, 3, 4\}$, of a detection j.

3.2 Preliminary Results

3.2.1 Experimental Setup

We evaluate our methods on 7 widely used publicly available datasets and various models. A detailed description of the datasets and models used is provided in the following sections.

Datasets

- **MNIST** [34] is a widely-used dataset in the machine learning community, consisting of grayscale images of handwritten digits. The dataset contains 60,000 training images and 10,000 testing images, with each image sized 28×28 pixels. Each input belongs to one of ten classes, representing the digits 0 through 9. MNIST is commonly used for benchmarking classification algorithms due to its simplicity and high-quality annotations.
- **FashionMNIST** [61] is a dataset designed as a more challenging alternative to MNIST, containing grayscale images of various fashion items. It consists of 60,000 training samples and 10,000 test samples, with each image being 28×28 pixels. There are ten classes, including items such as t-shirts, trousers, and sneakers. This dataset is frequently used to evaluate algorithms' performance on more complex patterns while retaining MNIST's size and structure.
- **CIFAR-10** [32] is a labeled subset of the 80 million tiny images dataset, comprising 60,000 32×32 color images divided into ten classes, such as plane, car, and dog. It includes 50,000 training samples and 10,000 testing samples. CIFAR-10 is commonly used for image classification tasks to benchmark deep learning models due to its small image size and balanced class distribution.
- **CIFAR-100** [32] is an extension of CIFAR-10, containing 60,000 32×32 color images, split into 50,000 training and 10,000 testing samples. Unlike CIFAR-10, CIFAR-100 has 100 classes, with each class containing 600 images. This dataset is hierarchically structured into 20 superclasses, making it more challenging and suitable for fine-grained image classification tasks.
- **ImageNet** [10] is a large-scale dataset designed for visual recognition research, containing over 14 million annotated images across more than 20,000 categories. A popular subset, used in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), consists of 1.2 million training images, 50,000 validation images, and 100,000 test images categorized into 1,000 classes. Images in ImageNet typically have three channels (RGB), representing color images. The data format is usually stored as $Height \times Width \times 3$ or $3 \times Height \times Width$ depending on the framework used (e.g., TensorFlow or PyTorch), in which they will normally be reshaped into 224×224. ImageNet is widely regarded as a benchmark for evaluating large-scale image classification and object detection models.
- **Boston Housing** [20] is a regression dataset commonly used in statistical and machine learning tasks. It consists of 506 instances, each representing a district in Boston, with 13 numerical and categorical features such as crime rate, number of rooms, and property tax rate. The target variable is the median value of owner-occupied homes in thousands of dollars. This dataset is frequently used for evaluating regression algorithms.
- **Abalone** [42] is a dataset used for regression problems, containing 4,177 instances with eight features that describe physical attributes of abalones, such as length, diameter, and shell weight. The goal is to predict the age of abalones based on the number of rings in their shells. This dataset is widely utilized in predictive modelling and exploratory data analysis.
- **Concrete Compressive Strength** [62] is a dataset for regression tasks, containing 1,030 instances with eight numerical features, such as cement, water, and coarse aggregate content. The target variable is the compressive strength of concrete. This dataset is commonly used to evaluate the performance of regression models in predicting material properties.
- **COCO-2017 test** [37] is a large-scale dataset for addressing detection tasks, specifically the test set containing 5K images of 80 classes, focusing on everyday scenes, with multiple objects, and natural settings. This dataset is commonly used in the literature for tasks as object detection, segmentation, localization, and captioning.

Models

Various off-the-shelf models were employed for the implementation of the augmentation and explainability-driven testing described in Section 3.1.2 and 3.1.1.

Dataset	DL Models	#Params.
MNIST [34]	LeNet [35]	62,006
CIFAR-10 [32]	LeNet	62,006
	ResNet-18 [22]	11,689,512
	VGG-16 [54]	138,357,544
ImageNet [10]	MobileNet-v3-small [24]	2,542,856
	efficientnet-v2-small [57]	21,458,488
	ConvNext-Base [38]	88,591,464
COCO2017 [37]	YOLOv11n [27]	2,624,080

Table 7: Datasets and DL Models in Explainability-driven Testing Evaluations.



(a) Random-based perturbation

(b) Importance-based perturbation

Figure 16: A perturbation example using ImageNet dataset, illustrating the original image (left), along with the perturbed (middle), where (a) shows a sample from U_{R+O} and (b) a sample from U_{I+O}, and the pixel difference between the original and the perturbed (right).

Specifically, pre-trained models were used for the detection of the region of interest for augmentation, the augmentation of an image caption, and the augmentation of an image sample based on that caption. Mask R-CNN [21] produces regions in the image domain, i.e. instance segmentation of objects, which were considered for augmentation. A pre-trained medium-size language model[2] was employed for performing the caption augmentation step. For the image augmentation step, a pre-trained inpainting stable diffusion model [3] was used [46], which utilized the original image, along with the detected mask of object of interest, and a caption describing the output of the visual augmentation to be performed.

For model testing, described in Section 3.1.1, we utilize standard architectures, including LeNet-5, ResNet18, and VGG16. More advanced models, such as MobileNetV2, EfficientNetV2, and ConvNeXt, are intended for extended evaluations on a larger dataset (ImageNet). Details of the models are summarized in Table 7.

3.2.2 Experimental Results

Explainability-driven Testing

The datasets employed in explainability-driven testing parts are fundamental benchmarks in the field of machine learning and computer vision. Table 7 shows the datasets and models used in our experimental evaluation. All pre-trained models were trained on GPU platforms using specific datasets. Specifically, we utilized NVIDIA 3090, 4090, and A100 GPUs. The CUDA Toolkit version used was 12.0, and the NVIDIA driver version was 525.60.13. For the testing phase, computations were performed on a CPU platform. The CPU utilized was a 20-core Intel Xeon Gold 5218R @ 2.10GHz with 128 GB of RAM.

We try to use the following two experiments (Table and Figure) to showcase WISDOM is able to capture semantic diversity in modified inputs (*diversity*) and WISDOM is sensitive enough to detect adversarial inputs

[2] `https://spacy.io/models/en#en_core_web_sm`

[3] `https://github.com/faraday/runway-stable-diffusion-inpainting`

(*effectiveness*). An additional experiment is also given to show WISDOM is able to shift to a detection use-case (i.e., YOLO model with the COCO2017 dataset) as well.

For the *diversity* experiment, we select 2% of the input pixels to create 2 perturbed variants from the original dataset. Firstly, by applying Gaussian white noise on randomly selected 2% of the input pixels, we obtain a *random* perturbation set of data (U_R). Whereas, by injecting noise on the pixels of the top 2% of important input pixels, we acquire a *semantically important* perturbation set (U_I). Figure 16 illustrates an example of a sample from the U_R dataset (Figure 16(a)) and a sample from the U_I set (Figure 16(b)). In both perturbed examples, the same image from the ImageNet dataset was employed (shown on the left). Moreover, an additional image is visualized, presenting the pixel difference (right) between each perturbation and the original image, where pixel dissimilarity is signified with a lighter gray color.

For evaluating our approach against the state-of-the-art methods, we employ these two perturbed sets by adding them to the original dataset (U_O), thus creating the supersets U_{R+O} and U_{I+O}, from U_R and U_I, respectively. Table 8 summarizes the average coverage results over 5 independent runs, which are normalized into the interval $[0, 1]$. While a similar setup has been conducted for the detection use-case, Table 10 shows that, given a test set, we construct four more perturbed variants:

- U_I^{obj} / U_R^{obj}: Restrict importance-based / random perturbation to object regions (defined by YOLOv11 detected bounding boxes).
- U_I^{bg} / U_R^{bg}: Restrict perturbation to background regions (complement of detected bounding boxes).

Unlike the normal classification DNN model, we compute the coverage across each of the YOLOv11 groups, while each group contains several layers chosen from a heuristic method. Group-level coverages are averaged within each of three stage groups - *early* (layer[0:5]), *middle* (layer[6:12]), and *late* (layer[13:22]), and the overall coverage is the mean across all monitored layers.

From these results, it is evident that WISDOM consistently achieves higher coverage increases for U_{I+O} compared to U_O and U_{R+O}, outperforming both structural metrics, i.e NC, and single-attribution approaches, i.e. DeepImportance. This demonstrates that perturbing semantically meaningful regions leads to more diverse neuron activations than random perturbations. Moreover, the improvements are most notable in deeper architectures, such as ResNet18, suggesting that WISDOM effectively captures complex decision-relevant regions. And in the detection case, $\Delta U_{I+O}/\Delta U_{R+O}$ is consistently greater than 1.0, indicating that WISDOM importance scores identify pixels whose perturbation activates previously unseen neuron cluster-state combinations more effectively than random perturbation, confirming that WISDOM captures structurally meaningful input regions for the YOLO-based object detection.

Additionally, we evaluate the impact of Bayesian Optimization. Table 9 reports an ablation study, where WISDOM runs with and without Bayesian Optimization (BO). For this study, we consider KMeans as our default method, using a fixed number of clusters and random seeds. WISDOM (w/o-BO) exploits the chosen clustering method as in (w/-BO) but without further hyperparameter optimization, i.e. optimal clustering method with default settings. From the reported results, using BO consistently boosts WISDOM 's coverage, achieving up to 20% and on average up to 380% coverage gains. This confirms that optimized clustering produces more representative neuron groups. This highest gain was achieved on VGG16–CIFAR10 using `MeanShift(bandwidth = 0.43)`, which was empirically selected after 30 BO trials. On ImageNet–ResNet18, BO selected `Agglomerative Clustering(linkage='ward', n _clusters=7)`, yielding $2.33\times$ coverage gain. These results confirm BO's ability to adapt the clustering strategy to the network's complexity. We conclude that BO helps scale the semantic clustering for complex DNN testing.

An effective testing criterion should be sensitive to the *attacks*. For the *effectiveness* experiment, we incorporate adversarial examples into the test data and observe the coverage changes. Specifically, we set test sets of size 100, 1000, and 3000 samples, which are drawn from clean (benign) test data. For each test size, we construct a *mixed* test set by replacing a small fraction of the inputs (1%, 2%, 3%, 5%, 7%, and 10%) with adversarial examples generated by three attack methods: PGD [40], FGSM [16], and CW [7]. The coverage for each *mixed* set is computed and compared with the coverage of the *all-clean* set, that is, the set that does not contain

	LeNet (MNIST)			LeNet (CIFAR10)			VGG16 (CIFAR10)			ResNet18 (CIFAR10)			ResNet18 (ImageNet)		
	U_O	U_{R+O}	U_{I+O}	U_O	U_{R+O}	U_{I+O}	U_O	U_{R+O}	U_{I+O}	U_O	U_{R+O}	U_{I+O}	U_O	U_{R+O}	U_{I+O}
WISDOM-6	$62.5^{\pm0}$	$70.31^{\pm0}$	$71.87^{\pm0}$	$98.43^{\pm0}$	$98.44^{\pm0}$	$100^{\pm0}$	$98.43^{\pm0}$	$95.31^{\pm0}$	$96.87^{\pm0}$	$98.43^{\pm0}$	$98.43^{\pm0}$	$99.43^{\pm0}$	$100^{\pm0}$	$100^{\pm0}$	$100^{\pm0}$
WISDOM-8	$42.96^{\pm0.8}$	$47.65^{\pm0.9}$	$48.43^{\pm0.9}$	$85.93^{\pm1.2}$	$87.50^{\pm1.3}$	$88.67^{\pm1.4}$	$70.31^{\pm1.0}$	$70.70^{\pm1.1}$	$72.27^{\pm1.2}$	$75.00^{\pm1.5}$	$78.51^{\pm1.6}$	$78.51^{\pm1.6}$	$96.09^{\pm0.03}$	$96.10^{\pm0.09}$	$96.87^{\pm0.05}$
WISDOM-10	$16.21^{\pm0.5}$	$18.35^{\pm0.6}$	$18.70^{\pm0.6}$	$50.19^{\pm0.9}$	$53.32^{\pm1.0}$	$53.41^{\pm1.0}$	$35.93^{\pm0.9}$	$42.28^{\pm1.1}$	$42.48^{\pm1.1}$	$17.52^{\pm0.7}$	$21.04^{\pm0.9}$	$21.33^{\pm0.9}$	$73.43^{\pm0.6}$	$75.00^{\pm0.7}$	$77.53^{\pm0.8}$
WISDOM-12	$10.03^{\pm0.4}$	$11.35^{\pm0.5}$	$11.62^{\pm0.5}$	$25.31^{\pm0.7}$	$27.85^{\pm0.8}$	$27.97^{\pm0.8}$	$19.97^{\pm0.7}$	$23.53^{\pm0.9}$	$25.33^{\pm0.9}$	$20.33^{\pm0.8}$	$23.65^{\pm0.9}$	$24.09^{\pm0.9}$	$45.38^{\pm0.9}$	$48.02^{\pm1.0}$	$51.51^{\pm1.1}$
NC	$99.57^{\pm0.00}$	$99.57^{\pm0.00}$	$99.57^{\pm0.00}$	$100^{\pm0.00}$	$100^{\pm0.00}$	$100^{\pm0.00}$	$98.51^{\pm0.00}$	$98.65^{\pm0.00}$	$98.63^{\pm0.00}$	$88.96^{\pm0.00}$	$89.12^{\pm0.00}$	$89.08^{\pm0.00}$	$97.48^{\pm0.00}$	$97.51^{\pm0.00}$	$97.53^{\pm0.00}$
KMNC	$77.81^{\pm0.20}$	$82.66^{\pm0.24}$	$82.63^{\pm0.24}$	$56.74^{\pm0.28}$	$62.29^{\pm0.31}$	$62.05^{\pm0.30}$	$68.53^{\pm0.26}$	$74.00^{\pm0.29}$	$73.85^{\pm0.29}$	$58.19^{\pm0.25}$	$61.56^{\pm0.27}$	$62.24^{\pm0.28}$	$54.07^{\pm0.22}$	$58.02^{\pm0.25}$	$58.25^{\pm0.25}$
SNAC	$16.94^{\pm0.10}$	$16.94^{\pm0.10}$	$17.79^{\pm0.11}$	$18.64^{\pm0.11}$	$19.06^{\pm0.12}$	$19.06^{\pm0.12}$	$18.30^{\pm0.12}$	$27.39^{\pm0.16}$	$28.17^{\pm0.16}$	$15.38^{\pm0.10}$	$17.46^{\pm0.12}$	$17.60^{\pm0.12}$	$4.31^{\pm0.05}$	$4.71^{\pm0.06}$	$4.53^{\pm0.06}$
NBC	$17.37^{\pm0.11}$	$17.58^{\pm0.11}$	$18.22^{\pm0.12}$	$15.88^{\pm0.10}$	$16.52^{\pm0.11}$	$16.52^{\pm0.11}$	$19.27^{\pm0.13}$	$29.21^{\pm0.17}$	$31.43^{\pm0.18}$	$15.14^{\pm0.10}$	$16.52^{\pm0.11}$	$16.72^{\pm0.11}$	$4.73^{\pm0.05}$	$5.17^{\pm0.06}$	$5.01^{\pm0.06}$
TKNC	$86.44^{\pm0.00}$	$86.44^{\pm0.00}$	$87.71^{\pm0.00}$	$99.15^{\pm0.00}$	$99.15^{\pm0.00}$	$99.15^{\pm0.00}$	$51.81^{\pm0.00}$	$53.82^{\pm0.00}$	$54.29^{\pm0.00}$	$30.54^{\pm0.00}$	$31.35^{\pm0.00}$	$31.28^{\pm0.00}$	$72.48^{\pm0.00}$	$73.01^{\pm0.00}$	$72.68^{\pm0.00}$
TKNP	$0.16^{\pm0.00}$	$0.32^{\pm0.00}$	$0.32^{\pm0.00}$	$0.16^{\pm0.00}$	$0.32^{\pm0.00}$	$0.32^{\pm0.00}$	$0.16^{\pm0.00}$	$0.32^{\pm0.00}$	$0.32^{\pm0.00}$	$0.16^{\pm0.00}$	$0.32^{\pm0.00}$	$0.32^{\pm0.00}$	$0.16^{\pm0.00}$	$0.32^{\pm0.00}$	$0.32^{\pm0.00}$
LSC	$6.71^{\pm0.06}$	$6.15^{\pm0.06}$	$6.13^{\pm0.06}$	$21.96^{\pm0.12}$	$13.34^{\pm0.09}$	$13.20^{\pm0.09}$	$1.03^{\pm0.02}$	$0.60^{\pm0.02}$	$0.61^{\pm0.02}$	$0.96^{\pm0.02}$	$0.54^{\pm0.02}$	$0.54^{\pm0.02}$	$13.72^{\pm0.10}$	$7.65^{\pm0.07}$	$7.64^{\pm0.07}$
DSC	$0.17^{\pm0.00}$	$0.09^{\pm0.00}$	$0.09^{\pm0.00}$	$0.17^{\pm0.00}$	$0.09^{\pm0.00}$	$0.09^{\pm0.00}$	$0.37^{\pm0.00}$	$0.19^{\pm0.00}$	$0.22^{\pm0.00}$	$0.14^{\pm0.00}$	$0.07^{\pm0.00}$	$0.07^{\pm0.00}$	OOM	OOM	OOM
MDSC	$0.11^{\pm0.00}$	$0.09^{\pm0.00}$	$0.12^{\pm0.00}$	$0.05^{\pm0.00}$	$0.03^{\pm0.00}$	$0.03^{\pm0.00}$	$1.68^{\pm0.03}$	$1.34^{\pm0.03}$	$1.31^{\pm0.03}$	$2.68^{\pm0.04}$	$1.74^{\pm0.03}$	$4.15^{\pm0.06}$	OOM	OOM	OOM
NLC	$100^{\pm0}$	$100^{\pm0}$	$100^{\pm0}$	$100^{\pm0}$	$100^{\pm0}$	$100^{\pm0}$	$100^{\pm0}$	$100^{\pm0}$	$100^{\pm0}$	$100^{\pm0}$	$100^{\pm0}$	$100^{\pm0}$	$100^{\pm0}$	$100^{\pm0}$	$100^{\pm0}$
CC	$100^{\pm1.5}$	$85.33^{\pm0.9}$	$94.58^{\pm0.7}$	$100^{\pm0.0}$	$100^{\pm0.0}$	$100^{\pm0.0}$	$14.63^{\pm0.3}$	$11.08^{\pm0.3}$	$12.43^{\pm0.3}$	$55.62^{\pm2.1}$	$42.06^{\pm1.8}$	$37.98^{\pm1.7}$	$100^{\pm0}$	$100^{\pm0}$	$100^{\pm0}$
DeepImportance	$15.62^{\pm0.12}$	$17.87^{\pm0.15}$	$17.72^{\pm0.14}$	$62.01^{\pm0.35}$	$65.23^{\pm0.42}$	$66.40^{\pm0.45}$	$93.75^{\pm0.40}$	$96.48^{\pm0.35}$	$96.77^{\pm0.30}$	$59.93^{\pm0.90}$	$69.60^{\pm1.10}$	$70.65^{\pm1.20}$	$9.96^{\pm0.09}$	$11.13^{\pm0.08}$	$11.33^{\pm0.02}$

Table 8: Average coverage results for WISDOM ($k \in \{6, 8, 10, 12\}$) compared with other SOTA coverage criteria using LeNet, VGG16, and ResNet18 on the MNIST, CIFAR10 and ImageNet dataset. We ran experiments on the original dataset (U_O), on a superset of the original dataset with some additional randomly perturbed data (U_{R+O}), and a superset of the original dataset with some extra importance-driven perturbed data (U_{I+O}).

Method	LeNet (MNIST)	LeNet (CIFAR10)	VGG16 (CIFAR10)	ResNet18 (ImageNet)
Wisdom (default)	1	1	1	1
Wisdom (w/o BO)	1.1x	0.98x	1.1x	1.2x
Wisdom (w/ BO)	**1.67x**	**1.2x**	**3.8x**	**2.33x**

Table 9: Coverage improvement between random (U_{R+O}) and semantic (U_{I+O}) perturbations with and without Bayesian Optimization. For this comparison, we select $k = 12$, i.e. using top-12 neurons.

any adversarial data. From Eq. 20, an effective coverage metric should be able to show a noticeable difference when even a few adversarial samples are present in the set, thus a change in normalized coverage indicates sensitivity. This facilitates evaluating if the coverage criteria can serve as an *error detector*, i.e. detect whether they respond differently to test suites containing faults versus only benign inputs.

$$\Delta Cov(x, x') = \frac{Cov(x) - Cov(x')}{\max(Cov(x) - Cov(x')) - \min(Cov(x) - Cov(x'))} \tag{20}$$

In Figure 17, WISDOM shows the largest coverage changes across all attack types and error ratios, e.g., +42% versus NC, TKNC, TKNP, LSC, DSC and CC and +4% versus DeepImportance, on VGG16–CIFAR10 at PGD attack with 3,000 samples at 10% errors. This indicates that WISDOM reacts more strongly to subtle adversarial perturbations, revealing its advantage in capturing semantically relevant neuron activations affected by adversarial inputs. In contrast, activation-only metrics, such as NC and KMNC, show muted responses, underscoring their insensitivity to fine-grained adversarial effects. Similarly, across all the attack methods and samples, WISDOM consistently shows near-linear coverage shifts with larger samples or larger error rate, while other SOTA methods lack that ability, confirming its ability to capture subtle adversarial neuron activations. Similar results were also shown in Table 11. We conduct higher attack error rates on the images (i.e., 5%, 10%, and 20%) and evaluate the coverage change (C_{clean} vs C_{mixed}) of the YOLOv11n model. Again, we observe consistently positive ΔC across all stage groups (*early*, *middle*, *late*, and overall), with the magnitude increasing alongside the error rate. In particular, the feature attack at a 20% error rate with $N = 200$ yields the strongest signal ($\Delta_{\text{overall}} = 0.0269$). This trend suggests that WISDOM-identified neurons are highly sensitive to distributional shifts introduced by adversarial examples.

Semantic Data Augmentation

The evaluation of the Data Augmentation component was performed on the CIFAR-10 dataset. We visualize in Figure 21, Figure 22, Figure 23, and Figure 24 some qualitative results of the augmentation methods: *inpaint*, *noise*, and *erase*, whilst using as an input the image illustrated in Figure 19 and Figure 20, respectively. Table 12 and Table 13 denote the quantitative results for these experiments, using the fidelity scores described

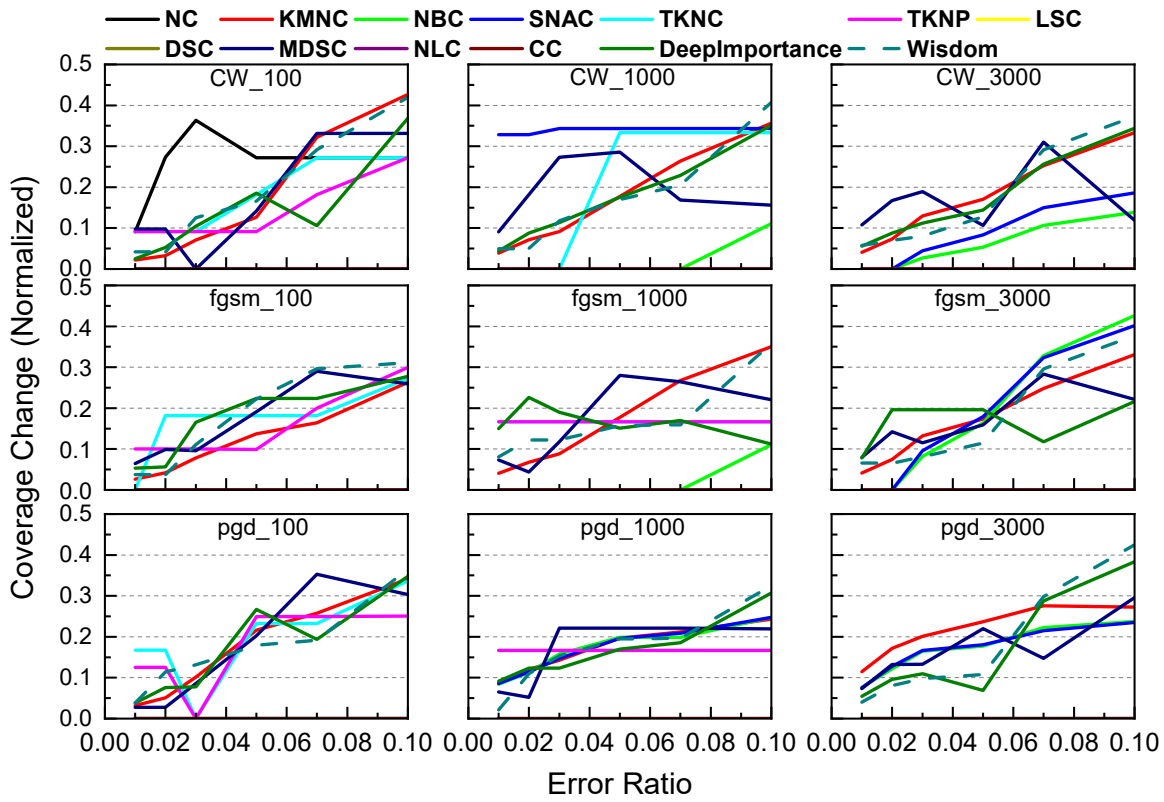


Figure 17: Normalized coverage changes for VGG16-CIFAR10 adversarial subsets under PGD, FGSM, and CW attacks across error ratios (1–10%). Higher values reflect greater sensitivity to adversarial examples.

in Section 3.1.2. Specifically, the LPIPS fidelity measure is computed based on the AlexNet [31], VGG [28], and SqueezeNet [25] models.

In Table 12 and Table 13, for each fidelity score we have noted, i.e. making bold, the best value for the inpainting augmentation method, for the different augmentation captions. It is evident, that in both tables the fidelity scores which consider pixel-wise correlations, i.e. MSE, PSNR, point to the same augmentation class, i.e. *deer* in Table 12 and *bird* in Table 13. Considering all the fidelity scores, apart from MSE and PSNR, we look into any kind of matches between the fidelity scores with the highest values. In Table 12, the augmentation classes *ship* and *deer*, seem to have the most of the best fidelity score values, i.e. LPIPS and VIF. In Table 13, the augmentation classes *horse* and *automobile* are presented to have the most best fidelity score values, which comprise of the values from FID, SSIM, LPIPS, and VIF scores.

Uncertainty Estimation

For the evaluation of the Uncertainty Estimation component, we performed several experiments for the classification, regression, and detection tasks on the validation set. Each dataset is split into a training and a test set. The test set is further split into a calibration set, which is employed to calibrate the conformal prediction step, and a validation set, which is used to the evaluation of the method.

For the classification task, a convolutional neural network was employed, comprising of 2 convolutional layer and a fully connected layer. This network was trained on the CIFAR-10, CIFAR-100, MNIST, and FashionMNIST datasets. For these experiments the P was set to 10, δ was 0.005, K to 1000, n to 2500, and α to 0.05. Table 14 summarizes the error for each dataset, by computing $1 - accuracy$.

For the regression task, a shallow fully connected neural network was used. For the Adaptive Monte Carlo Dropout step, P was set to 10, δ to 0.005, and K to 1000. Moreover, the parameter n was set to 10, α to 0.1, and the quantiles considered where 0.05 and 0.95. The results from the Boston Housing, Abalone, and

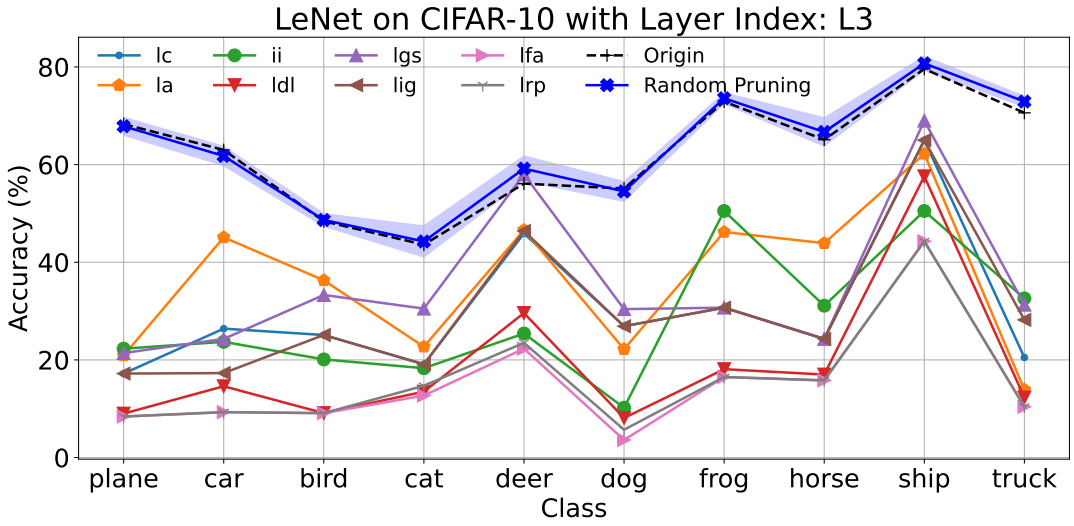


Figure 18: The Pruning performance for LeNet FC layer-1.

Stage	U_O	U_{I+O}	U_{R+O}	U^{obj}_{I+O}	U^{obj}_{R+O}	U^{bg}_{I+O}	U^{bg}_{R+O}
Stage	U_O	U_{I+O}	U_{R+O}	U^{obj}_{I+O}	U^{obj}_{R+O}	U^{bg}_{I+O}	U^{bg}_{R+O}
Early	0.8364	0.8580 ± 0.00	0.8642 ± 0.00	0.8549 ± 0.00	0.8549 ± 0.0031	0.8519 ± 0.0031	0.8580 ± 0.00
Middle	0.9053	0.9218 ± 0.0024	0.9177 ± 0.0014	0.9154 ± 0.0008	0.9154 ± 0.0008	0.9209 ± 0.0021	0.9204 ± 0.0024
Late	0.8906	0.9158 ± 0.0017	0.9147 ± 0.0026	0.9097 ± 0.0010	0.9074 ± 0.0045	0.9097 ± 0.0010	0.9063 ± 0.0042
Overall	0.8865	0.9071 ± 0.0016	0.9061 ± 0.0007	0.9014 ± 0.0007	0.9006 ± 0.0015	0.9033 ± 0.0013	0.9031 ± 0.0020
Δ Early	/	0.0216 ± 0.00	0.0278 ± 0.00	0.0185 ± 0.00	0.0185 ± 0.0031	0.0154 ± 0.0031	0.0216 ± 0.00
Δ Middle	/	0.0165 ± 0.0024	0.0123 ± 0.0014	0.0101 ± 0.0008	0.0101 ± 0.0008	0.0155 ± 0.0021	0.0151 ± 0.0024
Δ Late	/	0.0253 ± 0.0017	0.0241 ± 0.0026	0.0191 ± 0.0010	0.0168 ± 0.0045	0.0191 ± 0.0010	0.0157 ± 0.0042
Δ Overall	/	0.0206 ± 0.0016	0.0196 ± 0.0007	0.0150 ± 0.0007	0.0142 ± 0.0015	0.0168 ± 0.0013	0.0166 ± 0.0020
$\Delta U_{I+O}/\Delta U_{R+O}$	/	1.052 ± 0.063		1.065 ± 0.101		1.024 ± 0.135	

Table 10: Stage-wise coverage (mean $\pm$ std, 3 iterations) for YOLOv11n with COCO2017 dataset. Using WISDOM per-group coverage approach with #cluster=3, $k = 3$ and 5% pixels perturbation. $U^{\text{obj}}_{I/R+O}$ represents the perturbation is conducted to the object, similar for the $U^{\text{bg}}_{I/R+O}$.

Concrete datasets are denoted in Table 15. For each experiment, the mean and the standard deviation of the MAE error, the coverage, and the calibrated coverage are reported, over the course of 10 independent runs of the algorithm.

For the detection task, Yolov5 was employed on the COCO-2017 test set. For the class labels, α was set to 0.01, whereas for the bounding box coordinates the α was set to 0.1, whilst using the Bonferroni correction [60]. Moreover, the total number of bins is set to $m = 10$. The results for the regression part are summarized in Table 17, where the intersection-over-union (IoU) ($\uparrow$) of the large (L) and small (S) bounding boxes, mean prediction interval width (MPIW) ($\downarrow$), interval size (IS) ($\downarrow$), and coverage (Cov) ($\uparrow$) are presented. The large and small bounding boxes are considered based on the prediction sets of the coordinates. More specifically, each prediction set of a predicted coordinate $\hat{c}^i$, comprise of the pair: $(c^i_{-q}, c^i_{+q}) = [\hat{c}^i - Q^b_{c^i}, \hat{c}^i + Q^b_{c^i}]$, where $c^i_{-q} < c^i_{+q}$. Since the coordinates are in the format of *xyxy*, the large bounding box comprise of the coordinates: $(c^1_{-q}, c^2_{-q}, c^3_{+q}, c^4_{+q})$, whereas the small bounding box considers the coordinates: $(c^1_{+q}, c^2_{+q}, c^3_{-q}, c^4_{-q})$. Also, we performed a comparison with two other baseline approaches: a class-based quantile approach (Timans et al.) [58], where coordinate quantiles are computed per class group, and a bounding-box-size-based quantile approach (BS-CP), where coordinate quantiles are computed per box-size group. In the latter baseline, samples are grouped based on the three COCO groups, i.e. small, medium, large, according to their computed normalized area. It is evident that BB-CP achieves the tightest intervals, which leads to lowest interval sizes

Attack	ϵ_{err}	N	C_{clean}	C_{mixed}	Δ_{early}	Δ_{middle}	Δ_{late}	Δ_{overall}
FGSM	5%	50	0.6147	0.6170	0.0011	0.0063	0.0017	0.0023
	5%	100	0.7189	0.7264	0.0092	0.0093	0.0046	0.0075
	5%	200	0.7937	0.8020	0.0058	0.0079	0.0102	0.0084
	10%	50	0.6147	0.6226	0.0006	0.0018	0.0192	0.0078
	10%	100	0.7189	0.7240	0.0019	0.0181	0.0080	0.0051
	10%	200	0.7937	0.7962	0.0031	0.0146	0.0080	0.0026
	20%	50	0.6147	0.6219	0.0065	0.0066	0.0152	0.0072
	20%	100	0.7189	0.7187	0.0153	0.0092	0.0026	0.0002
	20%	200	0.7937	0.7867	0.0067	0.0113	0.0021	0.0070
PGD	5%	50	0.6147	0.6189	0.0090	0.0047	0.0010	0.0042
	5%	100	0.7189	0.7198	0.0006	0.0026	0.0000	0.0010
	5%	200	0.7937	0.8028	0.0063	0.0121	0.0071	0.0091
	10%	50	0.6147	0.6246	0.0074	0.0139	0.0065	0.0098
	10%	100	0.7189	0.7142	0.0062	0.0026	0.0061	0.0047
	10%	200	0.7937	0.8018	0.0096	0.0155	0.0011	0.0081
	20%	50	0.6147	0.6210	0.0132	0.0225	0.0017	0.0062
	20%	100	0.7189	0.7124	0.0158	0.0054	0.0199	0.0065
	20%	200	0.7937	0.8001	0.0088	0.0125	0.0016	0.0065
Feature	5%	50	0.6147	0.6193	0.0091	0.0150	0.0098	0.0046
	5%	100	0.7189	0.7282	0.0166	0.0179	0.0045	0.0093
	5%	200	0.7937	0.8012	0.0099	0.0115	0.0019	0.0076
	10%	50	0.6147	0.6356	0.0215	0.0320	0.0078	0.0208
	10%	100	0.7189	0.7366	0.0102	0.0297	0.0082	0.0177
	10%	200	0.7937	0.8079	0.0151	0.0145	0.0135	0.0143
	20%	50	0.6147	0.6301	0.0146	0.0109	0.0209	0.0154
	20%	100	0.7189	0.7325	0.0023	0.0228	0.0120	0.0137
	20%	200	0.7937	0.8205	0.0132	0.0339	0.0264	0.0269

Table 11: Adversarial sensitivity for YOLOv11n: coverage shift ΔC when mixing clean and adversarial images at varying error rates. Configurations: #cluster=3, $k = 3$ with 3 groups. Higher Δ at higher ϵ_{err} indicates WISDOM neurons are sensitive to adversarial perturbation.

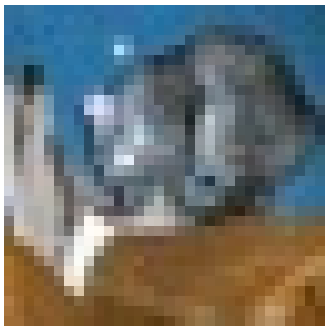


Figure 19: Image of a cat used as an input for augmentation.

Augmentation	category	MSE (↓)	PSNR (↑)	FID (↓)	SSIM (↑)	LPIPS (AlexNet) (↓)	LPIPS (VGG) (↓)	LPIPS (SqueezeNet) (↓)	VIF (↑)
Inpaint	airplane	85.827799	28.794524	**0.000767**	0.493527	0.000008	0.004562	0.000766	0.143539
Erase	airplane	104.481771	27.940398	0.006583	-0.004414	0.000114	0.032792	0.00427	0.031422
Noise	airplane	75.45638	29.28545	0.006994	0.901453	0.000002	0.000627	0.00016	0.962326
Inpaint	automobile	86.511068	28.760087	0.00134	0.509327	0.000008	0.003681	0.000709	0.136252
Erase	automobile	104.892578	27.923356	0.004066	0.000287	0.000116	0.033487	0.004287	0.03142
Noise	automobile	75.192058	29.266291	0.006001	0.901196	0.000002	0.000644	0.000158	0.959395
Inpaint	bird	84.927409	28.737532	0.000936	0.51134	0.000008	0.005991	0.000651	0.154076
Erase	bird	102.958984	28.004161	0.00345	0.054927	0.000009	0.030024	0.004607	0.043436
Noise	bird	75.832357	29.229465	0.004972	0.90114	0.000002	0.000637	0.000158	0.950347
Inpaint	deer	**84.41276**	**28.866723**	0.004316	0.5507	0.000009	0.005937	**0.000634**	**0.163515**
Erase	deer	104.515951	27.938978	0.007072	0.051675	0.00001	0.030131	0.004943	0.040468
Noise	deer	75.534505	29.31522	0.0064	0.901188	0.000002	0.000642	0.000159	0.963171
Inpaint	dog	86.231445	28.740018	0.003569	0.509486	0.000008	0.0075145	0.00081	0.130931
Erase	dog	104.254883	27.94984	0.00714	0.01987	0.000119	0.039486	0.00521	0.033479
Noise	dog	75.253255	29.26275	0.006135	0.901307	0.000002	0.0006445	0.000158	0.962922
Inpaint	frog	83.101237	28.831936	0.002389	0.477942	0.000008	0.005855	0.001023	0.132656
Erase	frog	105.512044	27.897783	0.003131	0.004236	0.000118	0.047772	0.003777	0.033647
Noise	frog	76.676758	29.215773	0.004606	0.901344	0.000002	0.000622	0.00016	0.958797
Inpaint	horse	85.709635	28.800507	0.003433	**0.567078**	0.000009	0.008457	0.000655	0.158448
Erase	horse	107.062174	27.834443	0.005209	-0.01173	0.000118	0.051205	0.003549	0.031932
Noise	horse	74.761393	29.325637	0.005607	0.901293	0.000002	0.000638	0.000157	0.962182
Inpaint	ship	85.594076	28.806367	0.001212	0.544427	**0.000007**	**0.003351**	0.000833	0.144954
Erase	ship	103.259766	27.991492	0.016461	0.048811	0.000106	0.030086	0.004773	0.041227
Noise	ship	75.160482	29.302515	0.006323	0.901328	0.000002	0.000623	0.00016	0.964913
Inpaint	truck	84.754557	28.849173	0.0012335	0.527616	0.000117	0.007743	0.00103	0.12035
Erase	truck	103.576823	27.978178	0.008487	0.056338	0.000009	0.029116	0.004915	0.040222
Noise	truck	75.8343	29.263753	0.00563	0.901331	0.000002	0.000632	0.00016	0.967042

Table 12: Results of fidelity scores for the various augmentation methods and the different augmentation captions. The bold values indicate, fidelity-score-wise, which augmentation caption has produced the best synthetic image, in terms of its similarity to the original one.

Augmentation	category	MSE (↓)	PSNR (↑)	FID (↓)	SSIM (↑)	LPIPS (AlexNet) (↓)	LPIPS (VGG) (↓)	LPIPS (SqueezeNet) (↓)	VIF (↑)
Inpaint	airplane	83.586589	28.806645	0.016322	0.70487	0.000003	0.00217	0.00015	0.234343
Erase	airplane	106.387044	27.861916	0.004815	0.119993	0.000007	0.036236	0.003617	0.052268
Noise	airplane	54.049479	30.700096	0.001815	0.978372	0.0000004	0.000009	0.000002	0.958398
Inpaint	automobile	83.716146	28.765382	0.010551	**0.74237**	**0.000002**	0.001842	**0.0001048**	0.254795
Erase	automobile	106.901042	27.840984	0.001897	0.063427	0.000107	0.052461	0.002948	0.03686
Noise	automobile	54.572591	30.658265	0.001816	0.978175	0.0000004	0.000009	0.000002	0.959407
Inpaint	bird	**82.01888**	**28.923272**	0.011807	0.656522	**0.000002**	0.0015664	0.000168	0.197338
Erase	bird	107.132487	27.831592	0.004291	0.06623	0.000116	0.065359	0.004348	0.038035
Noise	bird	53.532878	30.707269	0.000893	0.978201	0.0000004	0.000009	0.000002	0.953858
Inpaint	deer	84.944987	28.771033	0.015664	0.629434	**0.000002**	0.006026	0.000248	0.16279
Erase	deer	109.223958	27.747624	0.008718	0.026057	0.000136	0.074059	0.004279	0.03488
Noise	deer	54.609375	30.655339	0.001855	0.978163	0.0000004	0.000009	0.000002	0.964034
Inpaint	dog	84.238607	28.772899	0.024436	0.662703	**0.000002**	0.005525	0.000133	0.185794
Erase	dog	108.457357	27.778213	0.012544	0.02311	0.000142	0.074227	0.004241	0.034879
Noise	dog	56.053711	30.541967	0.001254	0.978277	0.0000004	0.000009	0.000002	0.96044
Inpaint	frog	83.341471	28.819399	0.02514	0.634166	**0.000002**	0.003323	0.000261	0.159312
Erase	frog	108.658529	27.770165	0.012849	0.025634	0.000126	0.072439	0.00462	0.035742
Noise	frog	54.557943	30.693831	0.000968	0.978365	0.0000004	0.000009	0.000002	0.963839
Inpaint	horse	87.104818	28.55838	**0.004832**	0.697535	0.000005	**0.001369**	0.000109	**0.265502**
Erase	horse	108.521159	27.775659	0.011035	0.01155	0.000129	0.071448	0.004694	0.032246
Noise	horse	54.779622	30.710484	0.000958	0.978321	0.0000004	0.000009	0.000002	0.965753
Inpaint	cat	87.491862	28.50431	0.014181	0.702714	**0.000002**	0.00142	0.000109	0.227067
Erase	cat	108.882813	27.76121	0.003682	0.026798	0.000116	0.074779	0.004108	0.035118
Noise	cat	54.16276	30.656467	0.002717	0.978164	0.0000004	0.000008	0.000002	0.963226
Inpaint	truck	82.577799	28.824841	0.016323	0.729247	**0.000002**	0.002104	0.00019	0.262019
Erase	truck	108.69043	27.768891	0.0137122	0.010915	0.00016	0.085545	0.005072	0.033226
Noise	truck	54.493164	30.630054	0.001123	0.97841	0.0000004	0.000009	0.000002	0.97121

Table 13: Results of fidelity scores for the various augmentation methods and the different augmentation captions. The bold values indicate, fidelity-score-wise, which augmentation caption has produced the best synthetic image, in terms of its similarity to the original one.



Figure 20: Image of a ship used as an input for augmentation.

Dataset	Error
CIFAR-10	0.035467
CIFAR-100	0.049733
MNIST	0.000533
FashionMNIST	0.010533

Table 14: MCCP results for classification.

(IS) across the coordinates, and lower MPIW. Moreover, tighter interval with similar coverage scores to the SOTA, produce higher IoU values for both the large and small bounding boxes.

Moreover, Table 16 denotes the quantitative results for the classification part, by reporting the classification coverage (Class Cov) (↑), top-1 and top-5 accuracy (↑), as well as mean set size (MSS) (↓). There is not significant difference between the various methods considering the classification part, since all of them employ the same approach.

Additionally, we present Figure 25 that illustrates some qualitative results of Timans et al. and the BB-CP method, in two sample images from the validation set. It is evident that BB-CP captures better the visualized objects in the scene and provides tighter large and small bounding boxes.

3.3 Integration

3.3.1 Uncertainty Estimation

We performed integration of the uncertainty estimation module on the E-Terry use case, using OBP/HMO detective models for humans and boxes in strawberry fields, in an offline setup. E-Terry provided us with image and groundtruth data depicting humans and boxes in a field, and NTUA granted us the fine-tuned Yolov5-based detector model. We evaluated our uncertainty estimation approach for the task of detection, and the results are summarized in Table 18 and Table 19.

We evaluated the proposed BB-CP method, with the baseline BS-CP, which was introduced in Section 3.2.2, as well as an additional baseline, C-CP, which employs class-based quantiles per coordinate for the regression part.

Figure 26 illustrates some qualitative results of the C-CP baseline and the BB-CP method. It is evident, as the results summarized in Table 19 suggest, that the bounding boxes produced by BB-CP are tighter, i.e.the

Dataset	MAE (mean)	MAE (std)	Cov. (mean)	Cov. (std)	Cal. Cov. (mean)	Cal. Cov. (std)
Boston Housing	0.458445	0.130649	0.922826	0.007609	0.923913	0.062251
Abalone	0.730013	0.197472	0.892606	0.001644	0.894667	0.074672
Concrete	0.699635	0.400379	0.872959	0.006228	0.925	0.070624

Table 15: MCCP results for regression.

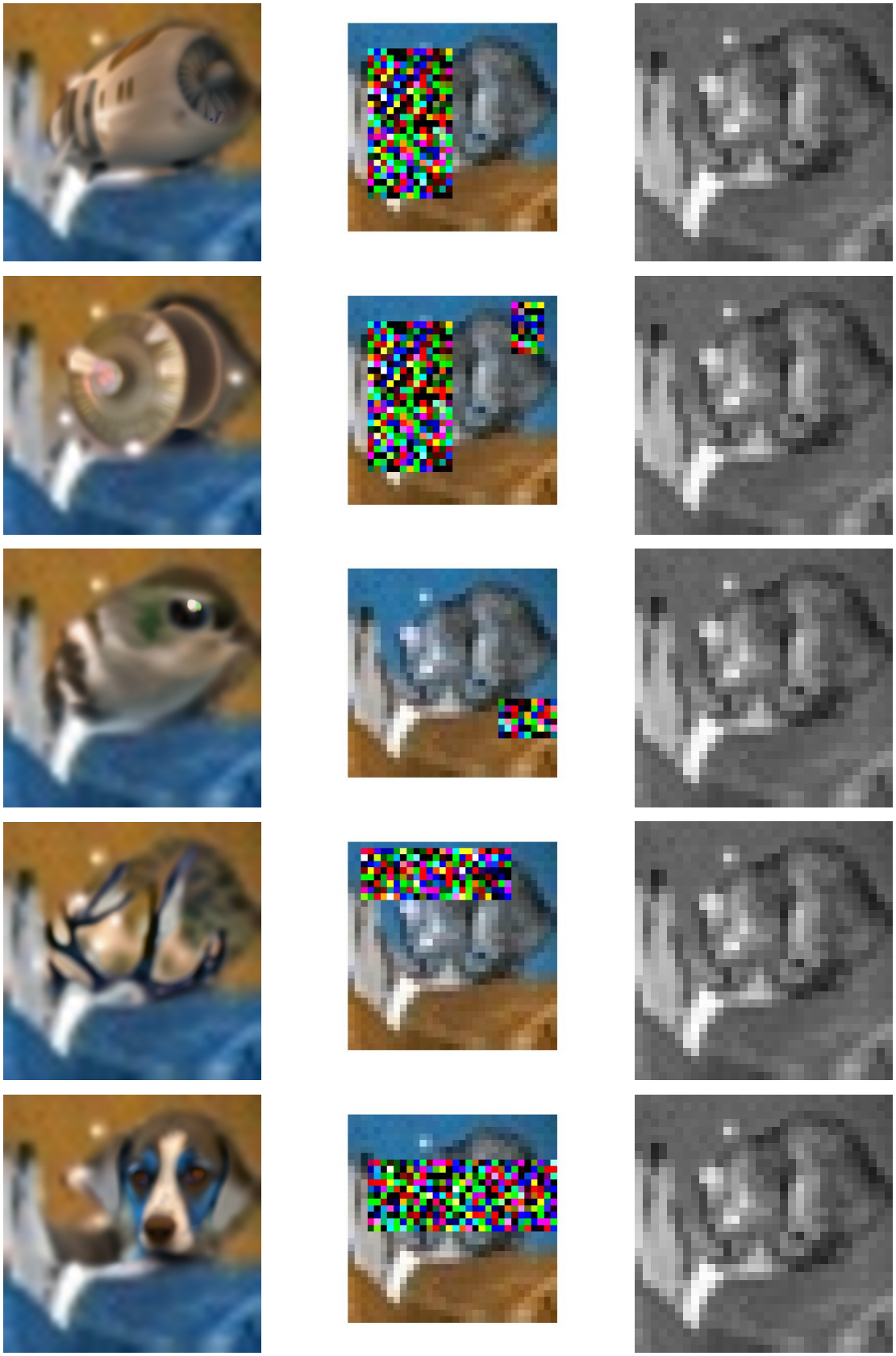


Figure 21: Qualitative results of the augmentations: *inpaint* (left), *noise* (middle), *erase* (right), for the augmentation captions (top to bottom): *high-fidelity image of airplane*, *high-fidelity image of automobile*, *high-fidelity image of bird*, *high-fidelity image of deer*, *high-fidelity image of dog*.

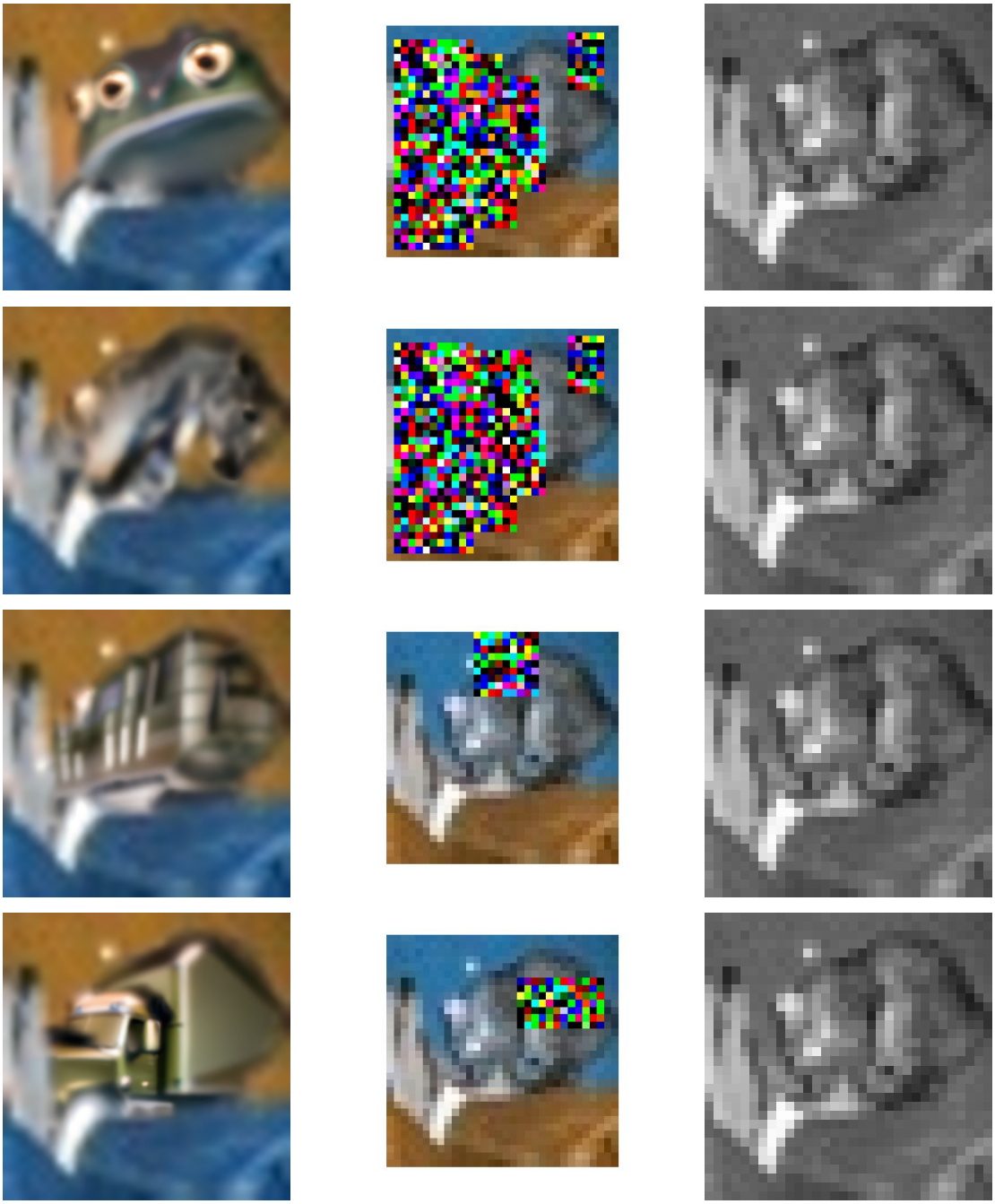


Figure 22: Qualitative results of the augmentations: *inpaint* (left), *noise* (middle), *erase* (right), for the augmentation captions (top to bottom): *high-fidelity image of frog*, *high-fidelity image of horse*, *high-fidelity image of ship*, *high-fidelity image of truck*.

Method	Class Cov	Top-1 Acc	Top-5 Acc	MSS
Timans et al.	0.9808 ± 0.0030	0.9268 ± 0.0027	0.9793 ± 0.0030	3.2558 ± 0.2194
BS-CP	0.9801 ± 0.0016	0.9278 ± 0.0017	0.9788 ± 0.0017	3.3521 ± 0.4085
BB-CP	0.9805 ± 0.0008	0.9260 ± 0.0023	0.9790 ± 0.0008	3.2494 ± 0.3816

Table 16: Quantitative results for the classification part in uncertainty estimation for detection.

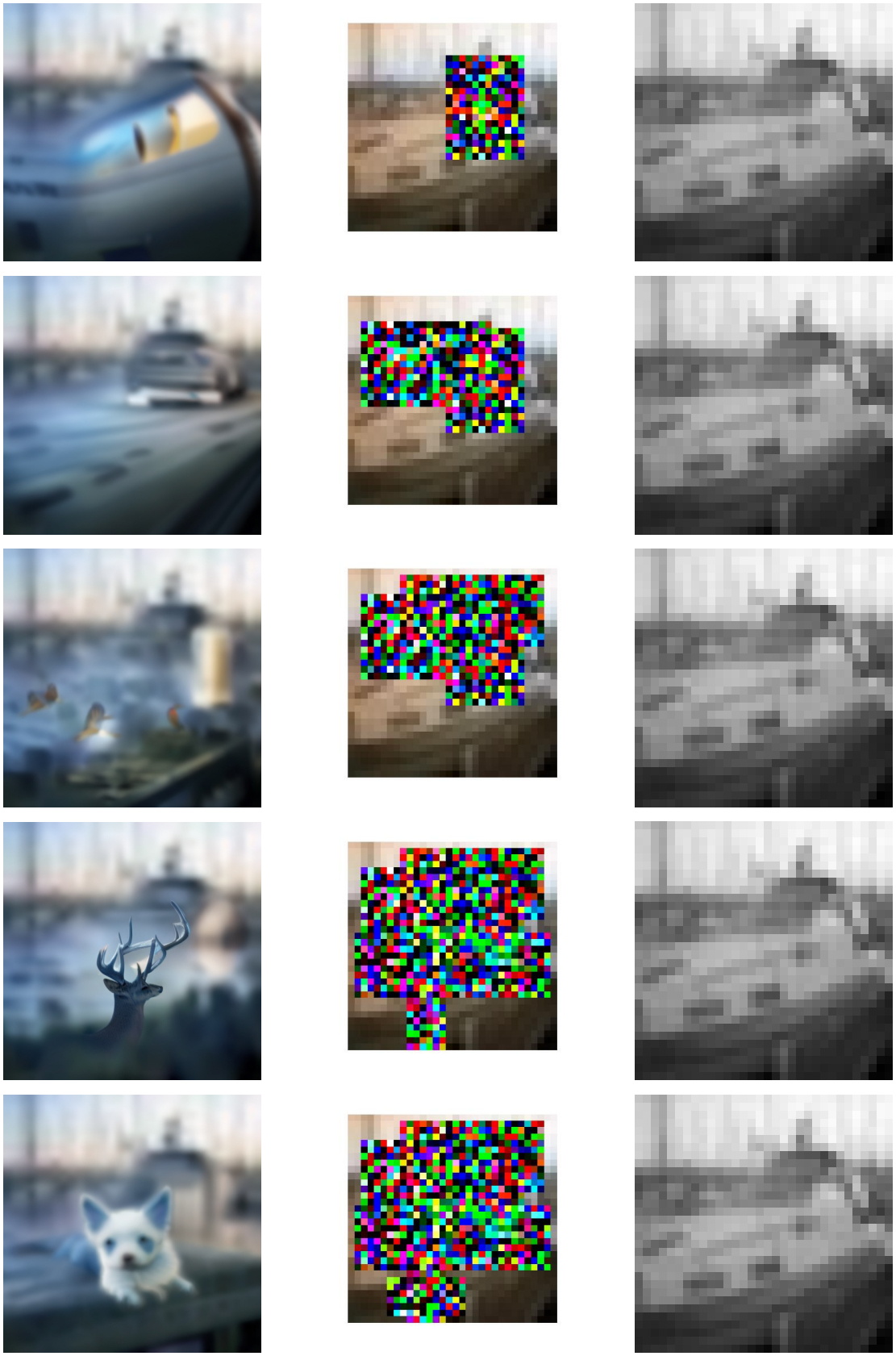


Figure 23: Qualitative results of the augmentations: *inpaint* (left), *noise* (middle), *erase* (right), for the augmentation captions (top to bottom): *high-fidelity image of airplane*, *high-fidelity image of automobile*, *high-fidelity image of bird*, *high-fidelity image of deer*, *high-fidelity image of dog*.

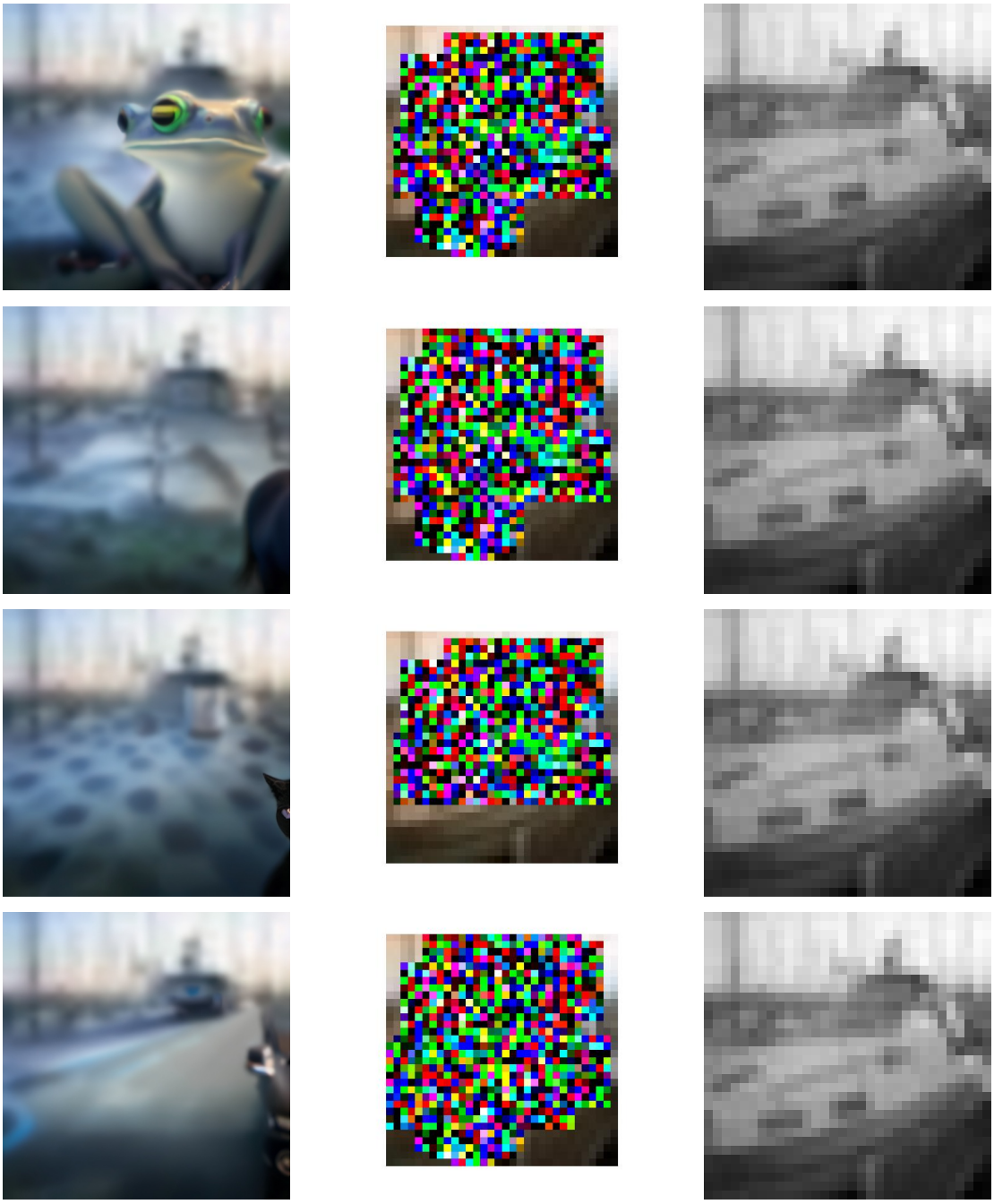


Figure 24: Qualitative results of the augmentations: *inpaint* (left), *noise* (middle), *erase* (right), for the augmentation captions (top to bottom): *high-fidelity image of frog*, *high-fidelity image of horse*, *high-fidelity image of cat*, *high-fidelity image of truck*.

Method	MPIW	IoU (L)	IoU (S)	IS c1	Cov c1	IS c2	Cov c2	IS c3	Cov c3	IS c4	Cov c4
Timans et al.	0.2443 ± 0.0095	0.2401 ± 0.0097	0.0702 ± 0.0070	0.2304 ± 0.0082	0.9759 ± 0.0018	0.2064 ± 0.0106	0.9754 ± 0.0018	0.2555 ± 0.0106	0.9788 ± 0.0017	0.2849 ± 0.0204	0.9786 ± 0.0041
BS-CP	0.1807 ± 0.0019	0.2930 ± 0.0039	0.0446 ± 0.0024	0.1639 ± 0.0060	0.9744 ± 0.0012	0.1725 ± 0.0033	0.9772 ± 0.0021	0.1722 ± 0.0067	0.9749 ± 0.0022	0.2126 ± 0.0050	0.9761 ± 0.0019
BB-CP	0.1277 ± 0.0030	0.3890 ± 0.0060	0.1310 ± 0.0114	0.1083 ± 0.0049	0.9481 ± 0.0058	0.1209 ± 0.0051	0.9537 ± 0.0061	0.1134 ± 0.0034	0.9509 ± 0.0014	0.1676 ± 0.0054	0.9608 ± 0.0032

Table 17: Quantitative results for uncertainty quantification on detection, specifically for the bounding box coordinates.

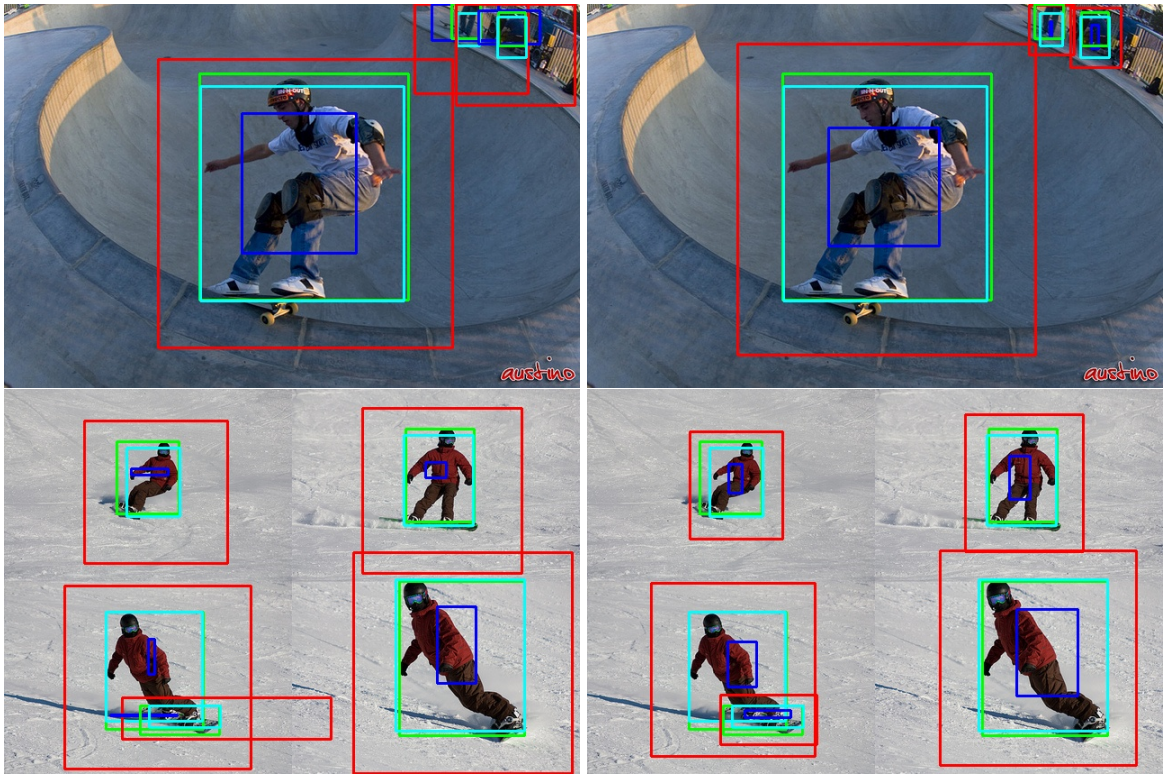


Figure 25: Qualitative results for comparison Timans et al. (left), and BB-CP (right). The red and blue bounding boxes represent the large and small predicted bounding boxes, whereas green denotes the ground truth, and the cyan the predicted bounding box.

Method	Class Cov	Top-1 AAcc	Top-5 Acc	MSS
C-CP	0.9789 ± 0.0041	0.9695 ± 0.0011	0.9789 ± 0.0041	1.4054 ± 0.3290
BS-CP	0.9843 ± 0.0043	0.9730 ± 0.0048	0.9843 ± 0.0043	1.4236 ± 0.3084
BB-CP	0.9838 ± 0.0033	0.9738 ± 0.0039	0.9838 ± 0.0033	1.5381 ± 0.2558

Table 18: Quantitative results for the classification part in uncertainty estimation for the E-Terry use case of humans and box detection.

Method	MPIW	IoU (L)	IoU (S)	IS c1	Cov c1	IS c2	Cov c2	IS c3	Cov c3	IS c4	Cov c4
C-CP	0.2018 ± 0.0626	0.3726 ± 0.0414	0.1353 ± 0.0373	0.0565 ± 0.0185	0.9617 ± 0.0096	0.3379 ± 0.1874	0.9769 ± 0.0066	0.1049 ± 0.0196	0.9818 ± 0.0063	0.3079 ± 0.0723	0.9892 ± 0.0073
BS-CP	0.1627 ± 0.0466	0.4228 ± 0.0467	0.1808 ± 0.0412	0.0807 ± 0.0078	0.9817 ± 0.0111	0.2954 ± 0.2007	0.9656 ± 0.0184	0.0784 ± 0.0076	0.9754 ± 0.0176	0.1963 ± 0.0283	0.9833 ± 0.0055
BB-CP	0.0966 ± 0.0198	0.6207 ± 0.0502	0.4617 ± 0.0836	0.0454 ± 0.0141	0.9353 ± 0.0233	0.2051 ± 0.0573	0.8909 ± 0.0161	0.0425 ± 0.0050	0.9226 ± 0.0258	0.0934 ± 0.0220	0.9060 ± 0.0240

Table 19: Quantitative results for uncertainty quantification on detection, specifically for the bounding box coordinates, for the E-Terry use case.



Figure 26: Qualitative results for comparison C-CP (left), and BB-CP (right). The red and blue bounding boxes represent the large and small predicted bounding boxes, whereas green denotes the ground truth and the cyan the predicted bounding box.

difference between the large (red) and the small (blue) bounding boxes is smaller, than the respective bounding boxes by C-CP.

3.4 Discussion and Next Steps

The work so far acts as a baseline for the complete trustworthy AI tools and further refinements are noted to be completed for the next steps. We aim to focus on strengthening the scalability of the components to real-world scenarios, by integrating them on the Pilot's technologies and investigating and testing corner-cases specific to the use cases. Moreover, we aim to extend the experimental evaluation with additional datasets, metrics, and methods. The next steps will build on top of the work included in this report, and will try to address issues and limitations and align solutions from the different project partners.

More specifically, the next steps for the Explainability-driven Testing component involves its extension to robustly select the most appropriate attribution method and. The next steps for the data augmentation component will focus its integration with the industrial Pilot partners, to create specific corner cases regarding each use case. Concerning the next steps on the uncertainty estimation component, we aim to perform full integration of the component with the industrial Pilot partners' technologies, i.e. E-Terry and CRF, to assess the trustworthiness of the neural networks used for the tasks of object detection, e.g. screw detection, pest detection, in real-world scenarios.

4 Conclusion

This deliverable document constitutes the updated version of the responsible AI (RAI) and trustworthy AI (AIT) components that will contribute towards achieving safe AI technologies for the SOPRANO project. More specifically, we have investigated and described the steps for attaining responsible AI models through the RAI component by evaluating the technologies from a legal, privacy, and ethical perspective. Furthermore, through the AIT component, we provide modules for testing neural networks by considering the internal architectures of the AI models employed, augmenting datasets to improve generalization, and estimating the uncertainty of the predictions made by the AI models.

We have described in detail the methods used behind each of these components as well as the metrics employed for their evaluation. Moreover, we have presented results from our implementation, in a lab environment. For the evaluation, we experimented using state-of-the-art AI models, and various open-source datasets from different domains. Future work will focus on further validation in industrial environments and increasing the maturity of the developed components

Our next steps will build on these efforts, working closely with the industrial stakeholders to align the solutions with stakeholder needs and the complex scenarios of MH-MR systems in the industrial environments targeted by the SOPRANO pilots and beyond.

A Progress Since D3.2

A.1 Ethics Readiness Levels for industrial AI robots

At M15, the methodology for RAI was already well advanced, since the LPERL evaluation tool was initially developed in EU projects MultiRate and TechEthos. The description of the tool itself is developed in D3.2 and the main progress are along the following lines

A.1.1 Block of indicators for human/robot interaction

The methodology developed in RAI is based on a modular evaluation tool. There are several blocks of indicators that correspond to EU referentials (GDPR, AI Act, etc.). In SOPRANO, the technologies that are developed are not limited solely to AI, but also around robotics since all the use cases are intelligent robots designed to assist humans in various industrial settings. Thus, we developed another block of indicators that focuses on robotics and more specifically on the ethics of human-robot interaction (HRI), as well as multi-HRI.

A.1.2 Validation of the evaluation tool

While in the former version of the LPERL tool, a global score was calculated, based on weights attributed to the different indicators, two things have been slightly changed. Firstly, while the weights were chosen to reflect the importance of the risk that is discussed, those weights, once chosen, were not changed in case other indicators were added or removed. This could lead to inconsistent assessments, as, for instance adding more indicators would naturally lead to the removal of more points and even negative scoring. Indicators' weights are now normalized so that the resulting score ranges between 0 and 4.

However, and this is the second change, the normalization is achieved, not globally, but at block's level. Thus, each of the now four blocks, which correspond to ALTAI, GDPR, AI Act and Robotics, produces a score ranging from 0 to 4. One benefit of this is that it permits us to identify more clearly an area of improvement, in case one of the four scores is significantly lower than the others. The global ERL is simply taken to be the minimum of the block scores and can be raised not only by working out all of the ethical dimensions at once, but by focusing on the more problematic block. This validation approach supports benchmarking across use cases and enables the monitoring of progress over time against defined evaluation criteria.

A.1.3 Ethics evaluation of three use cases and conception of interview method

Once the above elements were implemented, interviews started M20-21 with E-Terry, CFR and Kuka. Those interviews were preceded by visits to E-Terry and Kuka at M19 to meet with teams, see the systems in action and check that the LPERL tool was well adjusted to the use cases. By M21, ERLs were thus known for all three use cases. At first, the methodology relied entirely on the evaluative tool. The ethics expert from CEA was reading the questionnaire and taking notes of the answers, while points were updated in real time.

However, after this first series of interviews, we found it necessary to perfect the methodology for several reasons. Firstly, it was important to clarify what was expected in case of a question about the implementation of something. A question of the sort "Have you implemented X", while seemingly easy to answer, proved to be more ambiguous than expected, because it happened many times that X hadn't been strictly implemented, but the team of developers had already well identified the need for X and they had even planned to do it. We decided to encourage respondents to give a literal answer so as to obtain an accurate picture of the current situation and to have a better sense of progression in between two evaluations. This validation approach enables the monitoring of progress over time against defined evaluation criteria.

Another evolution of the methodology was to take more distance from our evaluative tool. The LPERL tool described in section 2.1 is key to this task, as it permits us to structure the interview, it serves as a motivational ingredient, it yields an indicator that says something of the ethical preparation and it helps track progress for a given project. But it also requires prudence and subtlety for two reasons. The first one is that the "grading" aspects of the evaluation, while motivating, is also stressing for some of the teams who may avoid saying certain things – speaking about some doubts that they have for instance – by fear of getting a lower note. The second one is that, while ethical readiness levels are both useful and meaningful to get a sense of ethical progression in time, for a given project, the comparison between ERLs of two different projects is not relevant. It suggests a false sense of one of two projects being somewhat "more ethical" than the other because of a greater ERL, when in fact, two projects are simply not comparable with our methodology. After several team members asked about how other projects performed, we understood that it was important to downplay the grading system and to emphasize ethical evaluations as a chance to identify ethical issues. This change of focus on the way we talk about ERLs and we introduce them to the teams has then become an integral part of every evaluation session since and it has helped developers to feel more free and more secure in the process.

A.1.4 ERL publication and GitHub repository

M22-26 were dedicated to the elaboration of a paper describing ERLs and evaluative tools, as well as methodology. The paper has been submitted and is currently in revision. It is accessible at `https://arxiv.org/abs/2512.09729`. Indicators and weights have been made available on a GitHub repository: `https://github.com/LA-NS/ethics-readiness-levels`.

A.1.5 Ethics evaluation of ten new SOPRANO use cases

After the Open Call, ten new use cases were included in SOPRANO, each of which was subject to a proper ethical check at M28. For these interviews, we benefited from the experience gained from the first three use cases to better define our approach, our presentation of the method and objectives to the teams and the interview itself.

At M30, we have then fine-tuned the methodology, namely what we say to the team before the interview, how we help them during the interview, and how the scores are calculated and interpreted.

We have completed a first round of interviews for all use cases in SOPRANO, obtaining their scores for each block of indicators and their global ERL.

A.2 AI Trustworthiness

Since D3.2, the *Explainability-driven Testing* and *Uncertainty Estimation* modules of the AI Trustworthiness component have advanced to include more features, and employ more sophisticated approaches. We describe how these modules have changed in the following subsections.

A.2.1 Explainability-driven Testing for DNNs

While in the former version of the Testing module, WISDOM used *accuracy drops* to choose the optimal attribution method, now it has been changed to *loss difference* (see Equation 1). The core implementation for the Attribution Methods Selector was still based on consensus voting. We further elaborate on the consensus voting mechanism by showing the visual walk-through in Figure 27. In Figure 27, pruning was used for the first stage to obtain $\Delta\mathcal{L}_m$. Then, a fine-grained selection weighted fusion of all attribution method is conducted for choosing the exact important neurons.

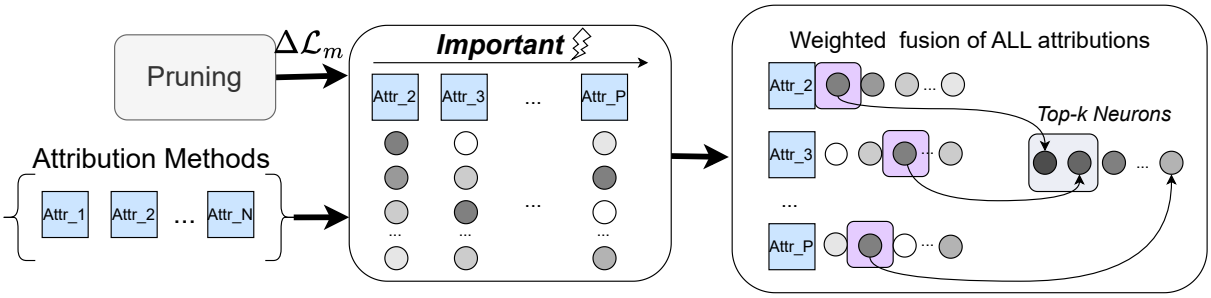


Figure 27: Consensus voting in WISDOM. We conduct the mask pruning to the neurons that cause less loss ($\Delta\mathcal{L}_n$ in Equation 1). WISDOM aggregates rankings from multiple attribution methods via weighted voting to produce a single list of important neurons (Equation 2, Equation 3, Equation 4).

In addition, a Bayesian Optimization (Algorithm 3) was also added to the Testing module for auto hyperparameter tuning. To better adapt our pilot parners' needs, Testing was also evaluated on a detection use-case for the YOLOv11 model with the COCO2017 dataset, while two experimental results were given to show the sensitivity and effectiveness of our module.

A.2.2 Uncertainty Estimation

Since, some of our pilot partners address the detection task in several use cases, we have enhanced the *Uncertainty Estimation* module, by incorporating an uncertainty estimation pipeline for detection. More specifically, we have proposed a pipeline that handles the class and bounding box detections, and provides uncertainty estimates for both, as illustrated in Figure 15. Since, some state-of-the-art detection neural networks, do not include any Dropout layers, we have altered slightly, the *Inference* stage of predictions. Different from the classification and regression pipelines, in detection, depending on the predictive model's architecture, we either deploy the *Adaptive MC Dropout* method (Algorithm 4) or process predictions of the predictive model independently.

Moreover, we have evaluated our approach using Yolov5 predictive model on COCO2017 test set, and compared against 3 baselines. Additionally, we have integrated, in an offline setup, our *Uncertainty Estimation* module, with the OBP/HMO detection component, using real-world data provided by E-Terry, summarized in Table 18 and Table 19.

References

[1] Laurynas Adomaitis, Alexei Grinbaum, and Dominic Lenzi. TechEthos D2.2: Identification and specification of potential ethical issues and impacts and analysis of ethical issues of digital extended reality,neurotechnologies, and climate engineering. Research report, CEA Paris Saclay, June 2022.

[2] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2):49–60, 1999.

[3] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[4] Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. Botorch: A framework for efficient monte-carlo bayesian optimization. *Advances in neural information processing systems*, 33:21524–21538, 2020.

[5] Viacheslav Borovitskiy, Iskander Azangulov, Alexander Terenin, Peter Mostowsky, Marc Deisenroth, and Nicolas Durrande. Matérn gaussian processes on graphs. In *International Conference on Artificial Intelligence and Statistics*, pages 2593–2601. PMLR, 2021.

[6] Dominique Brunet, Edward R Vrscay, and Zhou Wang. On the mathematical properties of the structural similarity index. *IEEE Transactions on Image Processing*, 21(4):1488–1499, 2011.

[7] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

[8] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.

[9] European Commission. *The Assessment List for Trustworthy Artificial Intelligence (ALTAI)*. European Commission, July 2020. Google-Books-ID: cu8dEAAAQBAJ.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[11] Kedar Dhamdhere, Mukund Sundararajan, and Qiqi Yan. How important is a neuron? *arXiv preprint arXiv:1805.12233*, 2018.

[12] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[13] Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007.

[14] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[15] Simos Gerasimou, Hasan Ferit Eniser, Alper Sen, and Alper Cakan. Importance-driven deep learning system testing. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, pages 702–713, 2020.

[16] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[17] Alexei Grinbaum. *Responsible Research and Innovation*, pages 472–476. J.B. Metzler, 08 2021.

[18] Alexei Grinbaum. Responsible research and innovation. In A. Grunwald, editor, *Handbook of Technology Assessment*, page 494. Edward Elgar Publishing, 2024.

[19] David H Guston. Understanding 'anticipatory governance'. *Social Studies of Science*, 44(2):218–242, April 2014. Publisher: SAGE Publications Ltd.

[20] David Harrison Jr and Daniel L Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of environmental economics and management*, 5(1):81–102, 1978.

[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[24] Andrew G Howard. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[25] Forrest N Iandola. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[26] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[27] Glenn Jocher, Jing Qiu, and Ayush Chaurasia. Ultralytics YOLO, January 2023.

[28] Simonyan Karen. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv: 1409.1556*, 2014.

[29] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009.

[30] Jinhan Kim, Robert Feldt, and Shin Yoo. Guiding deep learning system testing using surprise adequacy. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pages 1039–1049. IEEE, 2019.

[31] Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014.

[32] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical Report TR-2009*, 2009.

[33] Alexander Lavin, Ciarán M. Gilligan-Lee, Alessya Visnjic, Siddha Ganju, Dava Newman, Sujoy Ganguly, Danny Lange, Atílím Güneş Baydin, Amit Sharma, Adam Gibson, Stephan Zheng, Eric P. Xing, Chris Mattmann, James Parr, and Yarin Gal. Technology readiness levels for machine learning systems. *Nature Communications*, 13(1):6039, October 2022. Number: 1 Publisher: Nature Publishing Group.

[34] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[35] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[36] Klas Leino, Shayak Sen, Anupam Datta, Matt Fredrikson, and Linyi Li. Influence-directed explanations for deep convolutional networks. In *2018 IEEE international test conference (ITC)*, pages 1–8. IEEE, 2018.

[37] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[38] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

[39] Lei Ma, Felix Juefei-Xu, Fuyuan Zhang, Jiyuan Sun, Minhui Xue, Bo Li, Chunyang Chen, Ting Su, Li Li, Yang Liu, et al. Deepgauge: Multi-granularity testing criteria for deep learning systems. In *Proceedings of the 33rd ACM/IEEE international conference on automated software engineering*, pages 120–131, 2018.

[40] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[41] Daniel Müllner. Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378*, 2011.

[42] Warwick Nash, Tracy Sellers, Simon Talbot, Andrew Cawthorn, and Wes Ford. Abalone. UCI Machine Learning Repository, 1994. DOI: https://doi.org/10.24432/C55C7W.

[43] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001.

[44] Artem Obukhov and Mikhail Krasnyanskiy. Quality assessment method for gan based on modified metrics inception score and fréchet inception distance. In *Software Engineering Perspectives in Intelligent Systems: Proceedings of 4th Computational Methods in Systems and Software 2020, Vol. 1 4*, pages 102–114. Springer, 2020.

[45] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. Deepxplore: Automated whitebox testing of deep learning systems. In *proceedings of the 26th Symposium on Operating Systems Principles*, pages 1–18, 2017.

[46] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[47] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[48] Brian Sauser, Dinesh Verma, Jose Ramirez-Marquez, and Ryan Gove. From TRL to SRL: The concept of systems readiness levels. In *Conference on Systems Engineering Research, Los Angeles, CA*, pages 1–10. Citeseer, 2006.

[49] M Scott, Lee Su-In, et al. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30:4765–4774, 2017.

[50] David Sculley. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178, 2010.

[51] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

[52] Hamid R Sheikh and Alan C Bovik. Image information and visual quality. *IEEE Transactions on image processing*, 15(2):430–444, 2006.

[53] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMlR, 2017.

[54] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[55] Bhargav Srinivasa-Desikan. *Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim, spaCy, and Keras*. Packt Publishing Ltd, 2018.

[56] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[57] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[58] Alexander Timans, Christoph-Nikolas Straehle, Kaspar Sakmann, and Eric Nalisnick. Adaptive bounding box uncertainties via two-step conformal prediction. In *European Conference on Computer Vision*, pages 363–398. Springer, 2024.

[59] Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*, volume 29. Springer, 2005.

[60] Vladimir Vovk and Ruodu Wang. Combining p-values via averaging. *Biometrika*, 107(4):791–808, 2020.

[61] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[62] I-Cheng Yeh. Concrete Compressive Strength. UCI Machine Learning Repository, 1998. DOI: https://doi.org/10.24432/C5PK67.

[63] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[64] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: an efficient data clustering method for very large databases. *ACM sigmod record*, 25(2):103–114, 1996.