ExperimentLens: Interactive Visual Analytics and Explainability for ML Experiment Management

Stavros Maroulis
stavmars@athenarc.gr
ATHENA Research Center
Athens, Greece

Vassilis Stamatopoulos
bstam@athenarc.gr
ATHENA Research Center &
University of Ioannina
Greece

Panagiotis Gidarakos
pgidar@athenarc.gr
ATHENA Research Center
Athens, Greece

Konstantinos Tsopelas
k.tsopelas@athenarc.gr
ATHENA Research Center
Athens, Greece

Nikolas Masouras
nmasouras@athenarc.gr
ATHENA Research Center
Athens, Greece

Konstantinos Kozanis
kkozanis@athenarc.gr
ATHENA Research Center
Athens, Greece

Nikolas Theologitis
n.theologitis@athenarc.gr
ATHENA Research Center
Athens, Greece

George Papastefanatos
gpapas@athenarc.gr
ATHENA Research Center
Athens, Greece

Giorgos Giannopoulos
giann@athenarc.gr
ATHENA Research Center
Athens, Greece

Erik G. Nilsson
erik.g.nilsson@sintef.no
SINTEF Digital
Oslo, Norway

ABSTRACT

The widespread adoption of experiment tracking and MLOps platforms has streamlined the management of machine learning workflows. Yet, these platforms often fall short in supporting interactive visual analysis that combines experiment results, data exploration, and model explainability within a unified interface. To address this gap, we introduce ExperimentLens, an extensible experiment analytics tool that operates on top of existing tracking infrastructures and supports multiple platforms through a simple adapter interface. ExperimentLens offers a rich, web-based environment for comparing runs, visualizing performance metrics, exploring datasets, and interpreting model outputs. Its modular architecture augments standard tracking systems with flexible, interactive capabilities that support both routine monitoring and in-depth analysis. We illustrate ExperimentLens' functionality through a walkthrough of its architecture and user interface.

CCS CONCEPTS

• **Human-centered computing → Visualization systems and tools**.

KEYWORDS

Machine Learning, MLOps, Visual Analytics, Explainability, Interactive Data Exploration

VLDB Workshop Reference Format:
Stavros Maroulis, Vassilis Stamatopoulos, Panagiotis Gidarakos, Konstantinos Tsopelas, Nikolas Masouras, Konstantinos Kozanis, Nikolas Theologitis, George Papastefanatos, Giorgos Giannopoulos, and Erik G. Nilsson. ExperimentLens: Interactive Visual Analytics and Explainability for ML Experiment Management. VLDB 2025 Workshop: GuideAI - 2nd Governance, Understanding and Integration of Data for Effective and Responsible AI.

VLDB Workshop Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/extremexp-HORIZON/vis-frontend/.

1 INTRODUCTION

The rapid growth of machine learning has driven the adoption of experiment tracking and MLOps platforms, such as MLflow, Neptune, ClearML, and ZenML [3, 5–7]. These tools provide essential infrastructure to log and manage experiments by capturing parameters, metrics, and artifacts, supporting reproducibility and model development. ML experiments typically involve iterative training and evaluation with varying hyperparameters, data, and algorithms.

Consider a common setup where practitioners use Scikit-learn [8] to train ML models and MLflow to track their experiments. They iteratively adjust preprocessing steps, tune hyperparameters, and retrain models, logging each run's metrics, and output artifacts. This setup enables them to manage different configurations and iterations of their experiments, with the goal of comparing performance across runs and refining model performance. Achieving this requires users to assess how preprocessing steps, hyperparameters,

Proceedings of the VLDB Endowment. ISSN 2150-8097.

and data characteristics affect performance. It also calls for rich, interactive visual exploration of metrics, underlying data, and output artifacts. Explainability methods further support this process by helping users interpret model behavior and guide refinement.

While platforms like MLflow streamline experiment tracking, their analysis capabilities are often limited to basic metric visualizations. Most such tools lack a unified interface that combines tracking with interactive exploration of metrics, datasets, model outputs, and explainability results. To perform more advanced visual analysis, users must generate plots manually during experiment execution and log them as static artifacts, limiting interactivity and constraining analysis to what was predefined during experiment setup. As experiment volumes and parameter spaces grow, flexible and insightful interfaces become increasingly important. Visual analytics and interactive exploration help users identify patterns, compare workflows, and better understand model decisions.

To address these challenges, we introduce ExperimentLens, an *experiment analytics tool* that integrates with experiment tracking infrastructures and augments them with comprehensive visual analytics, dataset exploration, and explainability-driven analysis. By leveraging existing tracking backends for data storage and retrieval, ExperimentLens allows users to retain their preferred experiment management workflows while benefiting from a powerful suite of interactive visualization and analysis tools. Through a simple adapter interface, ExperimentLens supports multiple tracking tools without vendor lock-in, enabling users to compare and monitor experiment runs alongside detailed exploration of input datasets and model predictions within a unified interface.

Outline. The remainder of this paper is structured as follows. Section 2 discusses related work and situates ExperimentLens in the context of existing experiment tracking and MLOps tools. Section 3 presents the overall architecture of ExperimentLens. Section 4 provides a walkthrough of the UI, demonstrating key features for experiment monitoring, comparison, and workflow inspection. Finally, we conclude with a summary of our contributions.

2 RELATED WORK

MLOps tools vary in scope and capabilities, offering features such as experiment orchestration, tracking, model versioning, data management, and deployment [1]. Platforms like W&B, ClearML, and ZenML [3, 5, 10] target the full ML lifecycle, while others focus on specific aspects like experiment tracking and monitoring [2, 6, 7].

Such tracking tools typically store metadata, metrics, and artifacts (e.g. trained models), to support reproducibility, comparison, and analysis. While most tools provide basic visualizations, these are often limited to scalar metrics and artifact listings. Further, they generally do not support interactive exploration of input and output data. To enable richer visual analysis, users must generate plots manually during experiment setup and log them as static artifacts.

In parallel, tools for visual model explainability have been proposed [4, 9], offering rich interfaces for inspecting predictions and explanations. However, these tools are not integrated with experiment tracking pipelines, making it difficult to analyze model behavior within the broader experimentation workflow. While users can manually add explainability methods to run during the experiment and store the results as plot artifacts, this still limits the analysis to what was predefined during setup. The results cannot be dynamically generated or adapted once the experiment has completed.

These limitations highlight the need for tools that bridge experiment tracking with interactive, explainability-driven analysis. ExperimentLens is designed to address this gap.

3 ARCHITECTURE

ExperimentLens uses a modular architecture with three main layers (Figure 1): the *Experiment Infrastructure*, the *Experiment & Data Abstraction Layer*, and the *User Interface*. This separation enables integration with existing tracking systems, efficient backend processing, and interactive exploration of experiment results.

The **Experiment Infrastructure** layer includes external components such as experiment trackers (e.g., MLflow), which manage run-level metadata (e.g., parameters, metrics, workflow structure). It also includes artifact storage systems that hold input datasets, intermediate results, experiment outputs (e.g., predictions), and trained models. The system is designed to be extensible: new experiment trackers and storage sources can be integrated by implementing standardized adapter interfaces.

At the core of the system, the **Experiment & Data Abstraction Layer** acts as the backend engine. The *Experiment Metadata Service* connects to external trackers through these adapters and translates their metadata into a unified internal format, exposing consistent access to parameters, metrics, and workflow information. Currently, an adapter has been implemented for MLflow, enabling integration with this tracking backend. The *Experiment Data Service* retrieves input/output datasets and supports query execution, filtering, and formatting of data for visualization. To ensure responsiveness, the *Data Access Optimization* module caches metadata, stores remote files locally, and maintains in-memory indexes and caches over them for efficient querying and rapid visualization rendering.

Finally, the *ML Evaluation & Explainability* component integrates model artifacts and predictions to compute performance metrics and post-hoc explanations, supporting both global and local interpretability methods. To enable this, certain data must be available at analysis time—such as the trained model, prediction outputs, ground truth labels, and test datasets. These resources must be explicitly stored during experiment execution to ensure compatibility with downstream explainability modules.

The **User Interface** layer provides interactive tools for exploring experiments and analyzing results. It includes views for workflow inspection, metric and parameter visualizations, model performance summaries, explainability outputs, and dataset-level exploration. Users can also rate individual runs, enabling the capture of subjective feedback alongside tracked metrics.

Implementation Details. ExperimentLens features a Java 17 backend built with Spring Boot and a React-Redux frontend. Explainability components are implemented in Python and integrated into the backend via gRPC services.

4 EXPERIMENTLENS USER INTERFACE

The User Interface (UI) of our tool comprises several pages designed to facilitate detailed analysis and intuitive navigation:

Experiment Monitoring Page. The landing page of our tool; shown in Figure 2. The main component of the page is a table

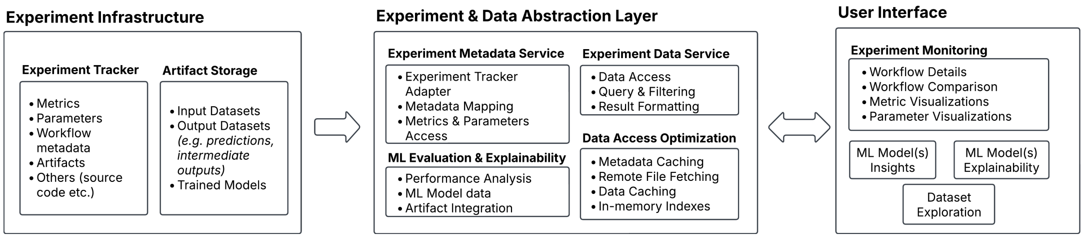

Figure 1: System architecture of ExperimentLens.

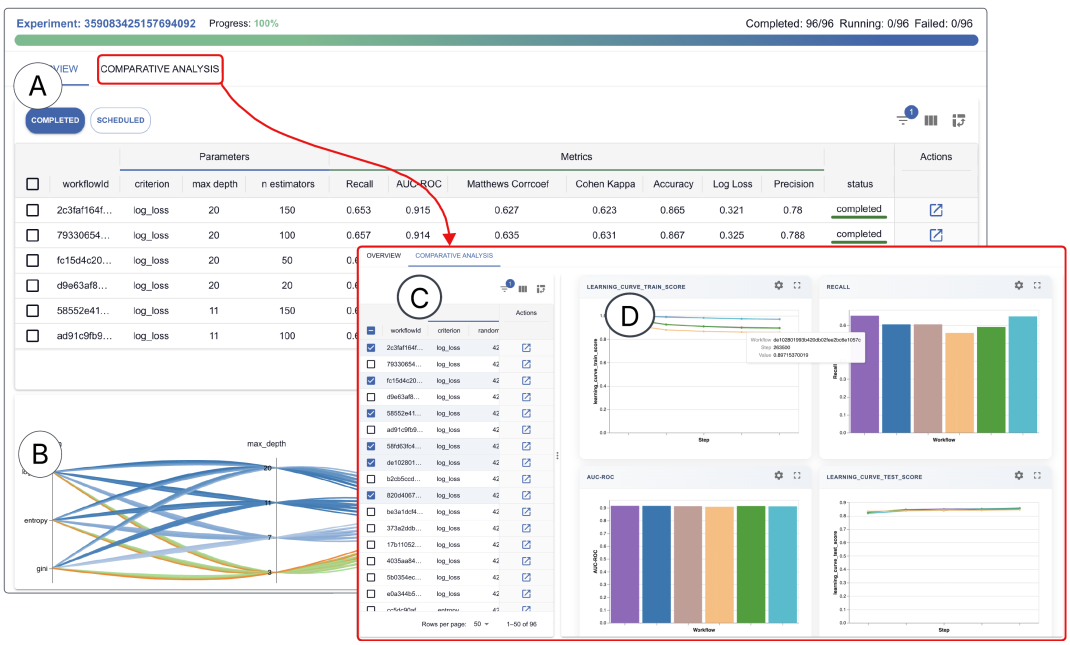

Figure 2: Experiment Monitoring and Comparative Analysis Pages in ExperimentLens.

displaying the workflows comprising the experiment (Ⓐ), with each column corresponding to a recorded metric or parameter. Users can interact with a single row to navigate to a detailed *Workflow Page* or select multiple rows for side-by-side comparison. A parallel coordinates visualization (Ⓑ) beneath the table depicts the same set of workflows, where each vertical axis represents a parameter, and the lines can be colored based on any selected metric to support multi-dimensional visual filtering.

Comparative Analysis Page. This page offers a focused view for comparing selected workflows. A compact version of the workflow table is displayed on the left side of the page (Ⓒ), enabling filtering and grouping. Grouping is based on parameters selected by the user, and results within each group are aggregated, for example by averaging metric values. The main panel presents comparative plots, which can be bar charts for single-value metrics or line plots for series metrics, such as accuracy across epochs (Ⓓ).

Workflow Page. Each workflow has a dedicated page that visualizes its metric and parameter data, metadata, and artifacts, such as models and datasets. At the top of the page, users can rate the workflow, allowing personal feedback to be captured alongside tracked metrics. A tree view on the left occupies one-fourth of the page, with two sections (Ⓔ). The *Workflow Details* section that shows the datasets, metrics, and parameters of this particular workflow, and the *Model Insights* section, which provides performance and explainability insights for the trained models within the workflow.

Workflow Details. In this view, users can explore individual metrics or parameters by selecting them from the tree. For example, selecting a single parameter presents a visualization of its value distribution across workflows, along with an option to navigate to a comparison view filtered to the workflows where this parameter appears (F_1). Selecting a dataset activates the data exploration component, which initially presents a tabular view of the data (F_2). From this interface, users can generate configurable visualizations such as scatter plots, line charts, or bar charts, with relevant axes and groupings suggested automatically based on the data types and columns selected. If the dataset contains geolocated information, a map view is automatically enabled, offering additional visualization options including point maps, heatmaps, or trajectory maps when temporal information is available. This allows for flexible and

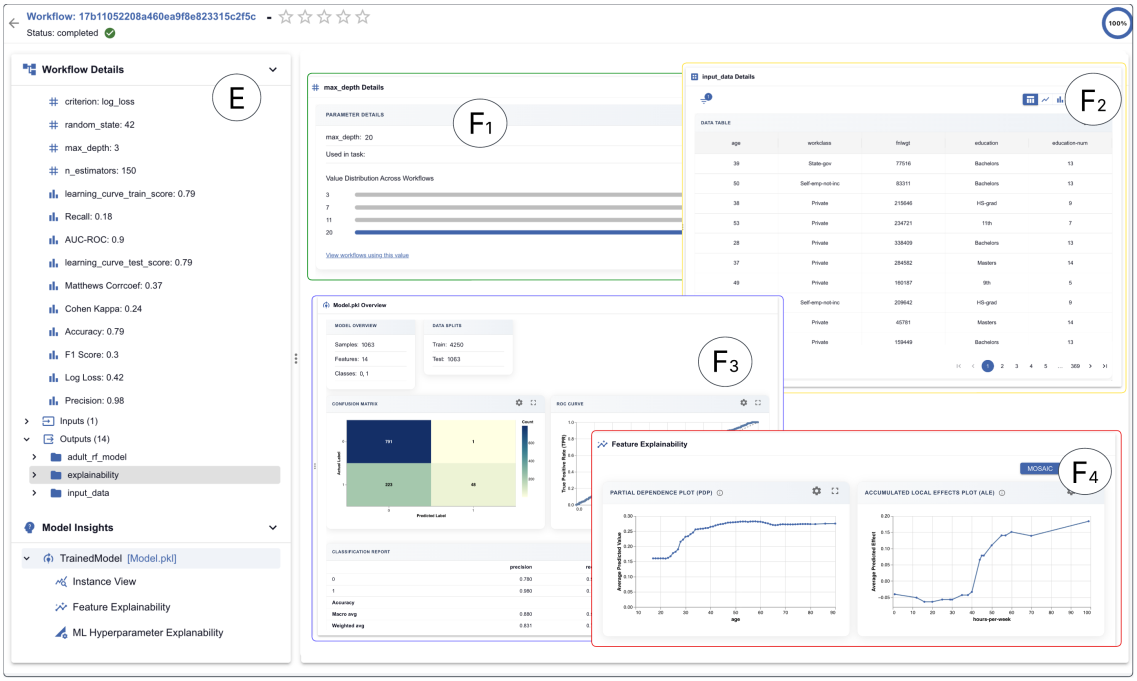

Figure 3: Workflow Page showing workflow metadata, dataset exploration, and model insights with explainability.

intuitive exploration of both standard and more complex datasets directly within the interface, something that distinguishes ExperimentLens from similar offerings.

Model Insights. This view is accessible from the workflow tree and is enabled for experiments that provide essential resources such as test datasets, ground truth labels, and trained models. These resources allow ExperimentLens to dynamically compute detailed model evaluation metrics and explainability analyses after the experiment execution. The view includes comprehensive summaries of the trained models within the workflow, presenting performance visualizations such as ROC curves and confusion matrices ($\mathbf{F_3}$).

Beyond traditional ML analysis, the *Model Insights* view offers explainability tools to guide users through model behavior analysis. This includes interactive Partial Dependence Plots (PDP) that illustrate the marginal effect of features on model predictions, Accumulated Local Effects (ALE) plots providing localized feature influence, and counterfactual explanations that help understand decision boundaries and model sensitivity by showing minimal input changes required to alter predictions. These explainability features utilize the test data and trained models and enable users to better interpret and trust model outcomes ($\mathbf{F_4}$).

Availability. A demo of the tool can be accessed online at *https://experimentlens.imsi.athenarc.gr/demo.* The code is open source at *https://github.com/extremexp-HORIZON/vis-frontend/.*

5 CONCLUSION

We presented ExperimentLens, an exeriment analytics tool that enhances experiment tracking platforms with interactive visualizations, dataset exploration, and model explainability. Built on standard MLOps infrastructures, it supports richer analysis of complex workflows through a modular architecture and adaptable interface for both routine monitoring and in-depth exploration.

ACKNOWLEDGMENTS

This work was supported by the ExtremeXP project (EU Horizon program, GA 101093164).

REFERENCES

[1] Lisana Berberi, Valentin Kozlov, Giang Nguyen, Judith Sáinz-Pardo Díaz, Amanda Calatrava, Germán Moltó, Viet Tran, and Álvaro López García. 2025. Machine learning operations landscape: platforms and tools. *Artificial Intelligence Review* 58, 6 (March 2025), 167. https://doi.org/10.1007/s10462-025-11164-3

[2] Tim Budras, Maximilian Blanck, Tilman Berger, and Andreas Schmidt. 2022. An In-depth Comparison of Experiment Tracking Tools for Machine Learning Applications. *International Journal on Advances in Software Volume 15, Number 3 & 4, 2022* (2022).

[3] ClearML. 2024. ClearML - Your entire MLOps stack in one open-source tool. https://clear.ml/ Software available from http://github.com/clearml/clearml.

[4] Panagiotis Gidarakos, Nikolas Theologitis, Stavros Maroulis, Loukas Kavouras, Giorgos Giannopoulos, and George Papastefanatos. 2025. GLOVES: Global Counterfactual-based Visual Explanations. https://doi.org/10.48786/EDBT.2025.92

[5] ZenML IO. 2025. ZenML IO. https://zenml.io/. Accessed: 2025-06-05.

[6] Neptune Labs. 2025. Neptune: Experiment Tracking Tool for ML Teams. https://neptune.ai/. Accessed: 2025-06-05.

[7] MLFlow. 2025. MLflow: A Tool for Managing the Machine Learning Lifecycle. https://mlflow.org/. Accessed: 2025-06-05.

[8] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* 12, null (Nov. 2011), 2825–2830.

[9] Ian Tenney, James Wexler, Jasmijn Bastings, Tolga Bolukbasi, Andy Coenen, Sebastian Gehrmann, Ellen Jiang, Mahima Pushkarna, Carey Radebaugh, Emily Reif, et al. 2020. The language interpretability tool: Extensible, interactive visualizations and analysis for NLP models. *arXiv preprint arXiv:2008.05122* (2020).

[10] Inc. Weights & Biases. 2025. Weights & Biases Documentation. https://wandb.ai/. Accessed: 2025-06-05.