ARGO: An Interactive Data Governance System for Machine Learning via Hierarchical Reinforcement Learning

Shuang Hao
Beijing Jiaotong University
Beijing, China
haoshuang@bjtu.edu.cn

Jun Wei
Beijing Jiaotong University
Beijing, China
24125281@bjtu.edu.cn

Yingqi Mu
Beijing Jiaotong University
Beijing, China
25125304@bjtu.edu.cn

Diwen Cheng
Beijing Jiaotong University
Beijing, China
22331156@bjtu.edu.cn

ABSTRACT

Training data quality is a key determinant of machine learning performance, yet real-world datasets often contain noisy labels, missing features, and imbalanced distributions. Existing methods typically address these issues independently and often rely on predefined pipelines, making them difficult to generalize across diverse data modalities and evolving data characteristics. As a result, effectively coordinating multiple data governance operations in a principled and dynamic manner to improve downstream model performance remains an open challenge.

In this paper, we demonstrate ARGO, an interactive system for adaptive data governance via hierarchical reinforcement learning (HRL). ARGO models data governance for machine learning (ML) as a sequential decision-making process, where a high-level agent dynamically selects operations such as feature repairing, relabeling, augmentation, and sample removal considering the evolving state of the dataset, while operation-specific sample selectors identify where and how to apply each data governance action. The system further integrates human feedback at multiple levels, incorporating user interactions as rewards and supervision signals to improve both decision policies and selector models. ARGO features a user-friendly graphical interface that supports real-time visualization of dataset quality, model performance, and decision trajectories, and enables audiences to inspect, validate, and influence system behavior. Through multiple scenarios, VLDB audiences can observe how adaptive decisions over data operations improve dataset quality and model performance over time.

PVLDB Reference Format:
Shuang Hao, Jun Wei, Yingqi Mu, and Diwen Cheng. ARGO: An Interactive Data Governance System for Machine Learning via Hierarchical Reinforcement Learning. PVLDB, 19(12): 4734 - 4737, 2026.
doi:10.14778/3827998.3828109

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/WeiJun0919/ARGO-Demo.

Proceedings of the VLDB Endowment, Vol. 19, No. 12 ISSN 2150-8097.
doi:10.14778/3827998.3828109

1 INTRODUCTION

The quality of training data has emerged as a fundamental bottleneck in modern machine learning systems. Recent studies in data-centric AI have shown that, as model architectures mature, improvements in data quality often contribute more to performance gains than further model refinements [6]. Consequently, many efforts are being made to improve training data through operations such as data cleaning [2], label correction [1], and data augmentation [7].

While effective individually, these methods lack a unified framework to coordinate multiple data governance operations. In practice, preparing training data may require performing more than one actions, such as addressing both label noise and class imbalance concurrently. Moreover, different operations are highly interdependent. For example, relabeling may reduce uncertainty but alter the data distribution, and augmentation may improve coverage while amplifying noise. Therefore, simply applying these techniques in a predetermined order may lead to suboptimal results.

This motivates a key challenge: how to adaptively orchestrate multiple data governance operations to optimize model performance. The problem is inherently sequential and state-dependent, as each operation modifies the dataset and affects subsequent decisions. Moreover, effective data governance requires not only selecting wich operation to apply, but also determining where to apply it, *e.g.,* which samples to relabel or which regions to augment. The challenge is further complicated by the need to incorporate human feedback in interactive settings.

In this demonstration, we present ARGO, an interactive system for adaptive data governance for ML via hierarchical reinforcement learning [4]. ARGO formulates data governance as a sequential decision-making problem, where a high-level reinforcement learning (RL) agent dynamically selects operations, including feature repairing (only for tabular data), relabeling, sample removal, augmentation and leaving the data unchanged, based on the current dataset state, downstream model signals and historical decisions of RL. This high-level policy enables the system to adaptively determine what action to perform based on the evolving condition of the dataset.

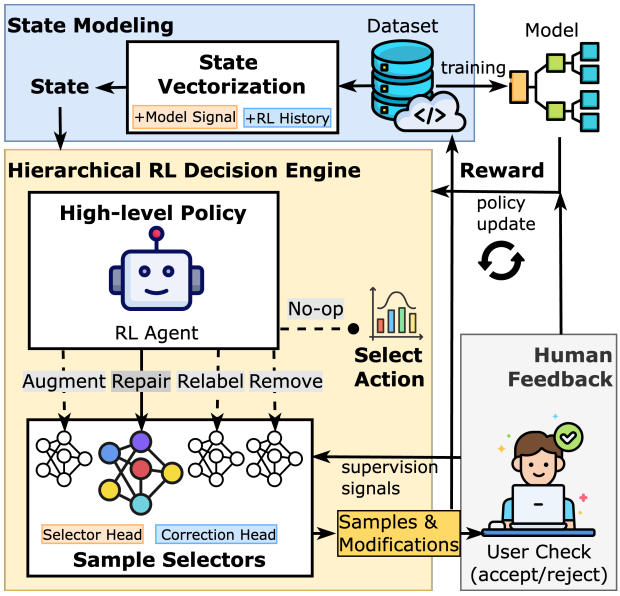

Figure 1: An Overview of ARGO

To enable fine-grained control, ARGO further decomposes each operation into a lower-level decision problem through a set of operation-specific conditional sample selectors. Given the operation chosen by the high-level agent, the corresponding selector identifies where to apply the action and exactly how it should be performed, *e.g.,* which samples should be relabeled and how should they be relabeled. This two-level hierarchical design decouples global decision-making from local data modification, enabling scalable operation on large datasets while preserving flexibility in addressing heterogeneous data quality issues. Moreover, it naturally captures the sequential and interdependent nature of data governance, forming a closed-loop process in which the agent continuously updates its state representation and optimizes long-term performance.

In addition, the hierarchical structure of ARGO naturally accommodates human feedback. Aggregated feedback is incorporated as rewards to guide high-level decisions, while instance-level feedback refines sample selectors at the low level. This integration provides corrective signals to improve the policies, enhancing robustness and interpretability in practice. To operationalize this human-in-the-loop capability, ARGO provides a graphical user interface (GUI) that enables the VLDB audience to have hands-on experience in understanding, interacting with, and ultimately controlling the data governance process, gaining insights into how adaptive decisions over data operations lead to improved model performance.

2 SYSTEM ARCHITECTURE

ARGO is an end-to-end system for adaptive data governance that iteratively improves training data quality through coordinated operations. As shown in Figure 1, ARGO consists of four main components: (i) dataset encoder, which summarizes the current dataset and model into a compact state representation; (ii) operation selector, which selects data operations, and together with (iii) sample selector which identify where and how to apply each operation, forms a hierarchical decision engine; and (iv) interactive user interface, which exposes system decisions and enables human feedback. At a high level, ARGO executes an iterative loop: it encodes the current dataset state, selects an operation, applies it to selected samples, updates the dataset and model, and incorporates evaluation signals and user feedback to improve subsequent decisions.

Data Representation and State Modeling. ARGO represents the evolving dataset using a set of dataset- and model-aware signals: (1) Data quality estimation signals, such as the ratio of feature and label noise, outlier scores and feature consistency checks, produced by lightweight, pluggable estimators; (2) Data distribution features, such as class proportions and density statistics, to capture imbalance and representativeness; (3) Model-derived signals, such as accuracy, confidence and loss statistics, to reflect prediction accuracy and reliability; (4) The historical decisions of RL, *i.e.,* the actions taken in the most recent few steps. These signals are aggregated into a fixed-length **state** vector that supports fast updates after each action.

Hierarchical Decision Engine. ARGO adopts a two-level decision design, where the high-level policy focuses on operation selection based on the state, and the low-level selectors specialize in instance selection conditioned on the chosen operation. The hierarchical design significantly reduces the effective action space, improving learning efficiency and stability, and enables each selector to be tailored to the semantics of its corresponding operation.

High-level Policy (Operation Selection). Given the current state, the high-level policy selects an operation from a small and interpretable **action** set, including no-op, remove, augment, relabel and feature repair (for tabular data), where no-op allows the system to skip unnecessary interventions. We implement it using the Proximal Policy Optimization (PPO) [5]. PPO is chosen for its robustness and sample efficiency in environments with delayed and noisy rewards, which are common in data governance scenarios. By restricting the action space and leveraging a stable policy optimization method, the high-level policy can effectively capture global dataset needs while remaining robust to noise and variability across different datasets.

The **reward** signal is derived from improvements in validation performance after applying operations, combined with incentives based on the amount of processed noise and action types (discouraging excessive deletion or trivial no-op decisions), as well as user satisfaction. This formulation promotes long-term performance gains by providing a delayed, dataset-level signal that reflects the cumulative effect of sequential operations, while the auxiliary terms serve as regularization to stabilize learning and encourage meaningful data transformations.

Low-level Execution (Sample Selectors). For each operation, ARGO invokes an operation-specific conditional selector that identifies target samples. Each selector takes as input a set of per-sample signals, including data encoding, model predictions, loss values, and data quality indicators, and outputs a ranked list of samples to be processed. In practice, selectors can be implemented as lightweight neural models, enabling efficient deployment on large-scale datasets.

Selectors are trained using a combination of reward and supervision signals derived from user feedback. In particular, the impact of

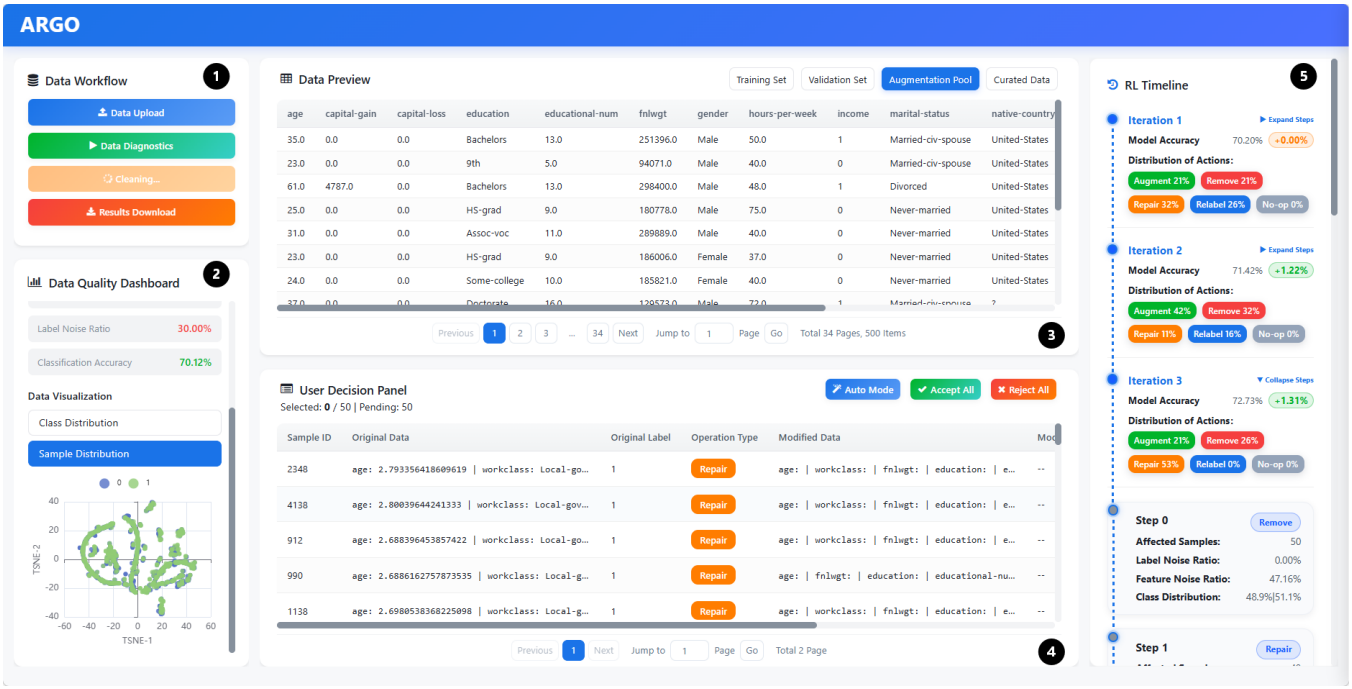

Figure 2: A Runing Example of ARGO

selected samples on downstream validation performance provides a global reinforcement signal, while instance-level feedback from the sample inspector serves as direct supervision to refine selection decisions. For operations involving label or feature repair, we adopt a dual-head design that jointly models both *whether* to operate on a sample and *how* to modify it. Specifically, one head predicts the selection probability for each sample, while the other outputs the corresponding correction, *i.e.,* the target label or repaired feature values. This joint modeling avoids decoupled pipelines and enables coherent decision-making at the instance level. This modular design allows heterogeneous criteria across operations while keeping each selector simple and scalable.

Learning and Optimization. ARGO adopts a three-stage training strategy to ensure stable and efficient learning. The core idea is to first make the low-level selectors reasonably effective and then proceed to joint training with the high-level agent. Otherwise, it will be very difficult for the policy to learn which operations are actually beneficial.

Stage I: Warm-up. To mitigate cold-start issues, ARGO initializes selectors using a noise-aware bootstrapping strategy. We first apply existing noise detection methods [1, 3] to identify a subset of high-confidence clean samples. We then inject synthetic noise into these clean samples (*e.g.,* corrupting labels or features) to construct pseudo training pairs with known ground-truth corrections. This process provides explicit supervision for training the selectors to distinguish clean versus corrupted instances and to learn appropriate modification behaviors. As a result, the selectors acquire a reasonable initialization before being exposed to reinforcement learning signals.

Stage II: Selector Refinement with Fixed High-level Policy. In the second stage, the high-level policy is frozen and actions are executed in a controlled manner such as cycling through different operation types. This isolates the learning of operation-specific selectors, which are refined using reward derived from model performance improvements. By decoupling operation selection from sample selection, this stage stabilizes training and allows selectors to specialize in identifying effective samples for each operation.

Stage III: End-to-end Hierarchical Optimization. Finally, ARGO conducts joint optimization of the full hierarchical framework. The high-level policy is updated using PPO, while the low-level selectors are trained concurrently using a combination of reward signals and supervised feedback. This end-to-end training enables coordinated decision-making across levels, allowing the system to learn adaptive strategies that balance different data governance operations over time.

To further improve stability and efficiency, ARGO employs techniques such as batched updates, proxy reward, and incremental model retraining, ensuring robust performance in practical, interactive settings. In fact, ARGO does not depend on a specific RL algorithm. The framework is algorithm-agnostic as long as it supports sequential decision optimization over a compact action space. ARGO also does not rely on any specific algorithm to detect noise from features and labels.

Human-in-the-loop Integration. ARGO is designed to seamlessly incorporate human feedback into the data governance process, enabling a tight interaction loop between automated decision-making and user supervision. Through the interactive interface, users can inspect individual samples, review system-recommended operations, and explicitly decide whether to accept or reject each action.

This feedback can be incorporated at multiple levels. At the high level, aggregated user responses, *e.g.,* accepting or rejecting system decisions, are integrated as additional reward signals, encouraging the policy to favor actions aligned with human judgment. At the low level, sample-level feedback could serve as supervision to refine the corresponding selectors. In particular, user decisions on whether to apply an operation to a sample are treated as explicit labels for the selection component, while any confirmed corrections (e.g., revised labels or feature values) are used to supervise the modification component. This design enables a human-in-the-loop feedback cycle, where user interactions not only provide immediate corrections but also continuously improve both the decision policy and the underlying data quality estimation components, allowing the system to balance human preferences with validation-based performance signals. This design prevents overfitting to sparse or noisy feedback while still benefiting from human expertise.

3 DEMONSTRATION SCENARIOS

We design a set of demonstration scenarios to showcase ARGO from multiple perspectives. Together, these scenarios highlight the flexibility and effectiveness of the proposed hierarchical framework in practical settings.

Scenarios I: Human-in-the-loop Data Inspection. We begin with an interactive scenario that highlights ARGO 's ability to incorporate human feedback into the data governance process.

1) Data Upload. The user uploads the training and validation set (tables, images or text) by clicking the *Data Upload* button (Figure 2-①). ARGO allows the user to review the samples in the *Data Preview* panel (Figure 2-③). If no validation set is provided, the system automatically constructs one by selecting high-confidence clean samples from the training data.

2) Data Diagnostics. Upon clicking the *Data Diagnostics* button, ARGO conducts feature-level (only for tabular data) and label noise detection, analyzes the data distribution, and generates candidate samples for augmentation. It then summarizes data quality statistics and presents visualizations in the *Dataset Quality Dashboard* (Figure 2-②).

3) Data Governance. When the user clicks the *Data Governance* button, ARGO begins the three-phase training of the HRL framework. During end-to-end optimization, the user can examine candidate samples selected for each action through the *User Decision Panel* (Figure 2-④) and explicitly decide whether to apply the suggested modifications. These interactions are integrated into the system as additional reward components or supervision signals.

4) Download Results. The user can inspect the processed dataset in the *Data Preview* panel and download the final results. Additionally, ARGO provides updated statistics and visualizations to summarize the quality and characteristics of the processed data.

Scenarios II: Fully Automated Data Governance. We next demonstrate ARGO in a fully automated setting, where the system operates without human intervention. Upon clicking the *Auto Mode* button in the *User Decision Panel* (Figure 2-④), ARGO will automatically select operations and applies them to the data based on its learned policies. At each iteration, the high-level policy determines the most beneficial operations, while the corresponding selector identifies target samples and performs necessary modifications. The system then retrains the downstream model and receives feedback through validation performance, forming a closed-loop optimization process. We could visualize how the dataset evolves over time, including changes in class distribution and noise level, as well as the corresponding improvements in model performance. This scenario demonstrates ARGO 's ability to autonomously discover effective data governance strategies.

Scenarios III: Adaptive Strategy and Behavior Analysis. Finally, we present a scenario that exposes the internal dynamics of ARGO. The system visualizes the evolution of the high-level policy in the *RL Timeline* (Figure 2-⑤), showing current state and selected operation in each step, and how the distribution of selected operations changes across iterations. In addition, the user can inspect operation-level behaviors, including which samples are selected, how they are modified, and how these changes affect dataset statistics such as class distribution, noise ratio, and model accuracy. By combining temporal dynamics with operation-level insights, this scenario provides a comprehensive view of ARGO 's decision-making process, demonstrating that the system is both adaptive and interpretable rather than a black-box pipeline.

4 CONCLUSION

We propose ARGO, a unified system that integrates feature repair, relabeling, augmentation, and data removal within a hierarchical decision framework. By formulating data governance as a sequential decision-making problem, ARGO leverages reinforcement learning to learn adaptive strategies that evolve with the dataset and consistently improve model performance. The system further supports both fully automated execution and human-in-the-loop interaction, enabling users to inspect, control, and refine the governance process. Through our demonstration scenarios, we show that ARGO not only improves data quality but also provides transparent and interpretable insights into operation-level behaviors. These capabilities make ARGO a practical and flexible solution for real-world data-centric AI systems.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (62372034).

REFERENCES

[1] Yuhao Deng, Chengliang Chai, Lei Cao, Nan Tang, Jiayi Wang, Ju Fan, Ye Yuan, and Guoren Wang. 2024. MisDetect: Iterative Mislabel Detection using Early Loss. *Proceedings of the VLDB Endowment* 17, 6 (2024), 1159–1172.

[2] Sanjay Krishnan, Jiannan Wang, Eugene Wu, Michael J Franklin, and Ken Goldberg. 2016. Activeclean: Interactive data cleaning for statistical modeling. *Proceedings of the VLDB Endowment* 9, 12 (2016), 948–959.

[3] Felix Neutatz, Mohammad Mahdavi, and Ziawasch Abedjan. 2019. ED2: Two-stage Active Learning for Error Detection–Technical Report. *arXiv preprint arXiv:1908.06309* (2019).

[4] Shubham Pateria, Budhitama Subagdja, Ah-hwee Tan, and Chai Quek. 2021. Hierarchical reinforcement learning: A comprehensive survey. *ACM Computing Surveys (CSUR)* 54, 5 (2021), 1–35.

[5] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[6] Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, Zhimeng Jiang, Shaochen Zhong, and Xia Hu. 2025. Data-centric artificial intelligence: A survey. *Comput. Surveys* 57, 5 (2025), 1–42.

[7] Zixuan Zhao and Raul Castro Fernandez. 2022. Leva: Boosting machine learning performance with relational embedding data augmentation. In *Proceedings of the 2022 international conference on management of data*. 1504–1517.