UNDER REVIEW

RESCIENCE C

Data Science Lab3

[Re] Cross-Validated Off-Policy Evaluation

Mete Harun Akcay[1], Masa Cirkovic[1] Alexis Gbeckor-Kove[1]

[1]Abo Akademi University

Reproducibility Summary

Scope of Reproducibility – This report aims to reproduce the primary claims from the original paper [1], which asserts that cross-validation (CV) can be effectively applied to off-policy evaluation (OPE) for selecting estimators and tuning hyperparameters. The original work challenges a common belief that CV is infeasible in OPE settings and demonstrates that CV-based OPE yields accurate and computationally efficient results.

Methodology – We used the authors' original code, following the provided instructions and configuration options to replicate their figures and results. While the code included an option to directly load saved models for faster reproduction, we aimed to replicate the full training process for greater fidelity, leading to substantial runtimes. Experiments were distributed across team members, with total execution times ranging from 6 to 78 hours per run. We utilized local machines and a remote machine provided to us. We managed to successfully run five out of six experiments conducted in the paper after some minor adjustments to the code.

Results – We successfully reproduced the main claims of the original paper that (i) the cross-validation method proposed can effectively select estimators within OPE, (ii) the Off-policy Cross-Validation (OCV) approach consistently chooses well-performing estimators with lower MSE compared to existing baselines, and (iii) the proposed OCV method efficiently tunes hyperparameters, providing robust results across multiple datasets. Our reproduced results further suggest that (iv) OCV offers a practical solution that generalizes across different policies without relying on multiple logging policies or complex neural networks, aligning closely with the original paper's outcomes.

What was easy – The authors provided an organized documentation, including clear instructions for setting up and running the code. The code itself was well-structured, and the paper, though complex, was organized in a way that allowed us to understand and follow the methodology effectively.

What was difficult – The primary challenge was the duration of the program executions. Several runs were extremely time-consuming, and we encountered small technical issues, such as VS Code crashes and minor code errors, which required us to rerun parts of the program. Additionally, we encountered issues reproducing one of the figures due to an error.

Communication with original authors – We did not reach out to the original authors for this reproduction study.

1 Introduction

In the field of machine learning, model selection and hyperparameter tuning are critical steps to improve predictive performance and robustness across different tasks. While cross-validation is a standard approach for these tasks in supervised learning, off-policy evaluation (OPE) has primarily relied on theory-based methods rather than data-driven techniques like cross-validation. This paper aims to bridge this gap by introducing a cross-validation-based method for off-policy evaluation that leverages logged data from a different policy, making it more accessible and practical for broader applications.
The original work by Cief et al. [1] presents a novel approach that challenges the belief that cross-validation cannot be adapted to OPE due to the unknown policy value and limited unbiased evaluation options. By designing a cross-validation framework for OPE and validating it empirically across various real-world datasets, the authors make a compelling case for the feasibility and advantages of this approach. Our reproduction of this study seeks to verify the effectiveness and practicality of this method by replicating their experiments, evaluating both estimator selection and hyperparameter tuning tasks as originally outlined.

2 Scope of Reproducibility

In this reproducibility study, we aim to verify the main claims presented in the original paper [1]. The primary focus of the paper is on improving the estimator selection and hyper-parameter tuning process in OPE using a cross-validation (CV)-based approach. In brackets we refer to figures and tables from the original paper. Specifically, the main claims we investigate are:

1. Cross-validation-based estimator selection (OCV) can reliably choose a suitable estimator among Inverse Provensity Scoring (IPS), Direct Method (DM), and Doubly-Robust (DR), demonstrating better performance in multiple datasets (Figure 1).
2. OCV performs well even when the validation estimator does not directly match the best estimator, adapting to the data and achieving low MSE in scenarios where the validator is suboptimal (Figure 2).
3. OCV serves as a general solution for hyper-parameter tuning and joint estimator selection, achieving comparable or superior performance to theory-based methods across various estimators (Figure 3).

Moreover, authors conducted additional experiments to more dive into the performance of the proposed CV-based estimator selection and made the following claims:

4. Their improvements make standard cross-validation more stable. (Figure 4)
5. The validator used in cross-validation has to be unbiased to block the optimization objective shifting to prefer the estimators biased in the same direction. (figure 6)
6. Cross-validation is computationally efficient. (Table 3)

3 Methodology

The approach used in this study involved running the original authors' code exactly as specified in their paper, without any re-implementation or modification except a configuration change about the usage of latex in produced figures. This strict adherence to

the specified commands enabled us to replicate the authors' analysis pipeline, estimator selection, and hyper-parameter tuning procedures, minimizing deviations from the intended approach.

Setting Up the Environment: The authors provided a requirements.txt containing libraries needed to run their code successfully. We created a Python virtual environment to run their code.

Implementation Adjustments: To execute the estimator selection and hyper-parameter tuning processes, we adjusted file paths within the code to match our local directory structure. For running evaluation commands, we installed **pyyaml**.

Dataset and Evaluation Scripts: The repository provided by the authors, available at github/cross-validated-ope, contained all necessary scripts to reproduce the evaluation figures from the paper. Specific configurations for each figure were available, and we followed the commands provided in the repository to generate the results as described. Precomputed results for these runs were available under the "results" folder in the repository, which facilitated efficient verification and comparison of outcomes with those presented in the paper.

Due to resource limitations, we could not replicate the exact hardware setup specified by the authors in the paper. Instead, we used laptops and later switched to a GPU-enabled server with NVIDIA RTX 2080 Ti, with 11 GB GDDR6 and 4352 NVIDIA CUDA cores.

3.1 Estimator Descriptions

In this study, we evaluated various estimators to assess their suitability for off-policy evaluation in a cross-validation setting. Below is a description of estimators and estimator selectors used in the main experiments.

- **Inverse Propensity Score (IPS)**: An unbiased estimator that reweights logged samples as if they were collected by the target policy. IPS has high variance, which can be mitigated by tuning a clipping constant to truncate large propensity weights.
- **Direct Method (DM)**: This estimator uses a regression model to directly predict rewards, reducing variance compared to IPS but potentially introducing bias. Key configurations include the choice of regression model, which can influence the estimator's bias-variance trade-off.
- **Doubly Robust (DR)**: A combination of IPS and DM, this estimator aims to reduce variance by leveraging both direct reward predictions and reweighting techniques. DR requires tuning of both the IPS clipping constant and parameters for the reward prediction model.
- **SLOPE**: SLOPE is an estimator selection method that uses a sequence of hyperparameters arranged by decreasing variance. It calculates confidence intervals for each value and selects the point where intervals stop overlapping, aiming to balance bias and variance. However, SLOPE assumes a monotonic increase in bias along the hyperparameter order, which may be difficult to establish for different estimators.
- **PAS-IF**: PAS-IF creates two surrogate policies from the logged dataset, ensuring that their propensity weights mimic those of the true logging and target policies. The logged dataset is split to simulate data from each surrogate policy. A neural network optimizes this objective, enabling effective estimator selection based on these surrogate policies and split data.
- **OCV**: Proposed by the authors, OCV utilizes cross-validation with an unbiased validator to identify the most suitable estimator, adapting dynamically to the data and reducing reliance on theoretical assumptions. By leveraging standard cross-validation techniques, OCV offers a practical, widely applicable solution that enhances estimator selection in real-world scenarios.

3.2 Datasets

This study used nine datasets in total and turned them into contextual bandit problems. The different characteristics of these datasets, given in Table 1, provide a comprehensive evaluation across diverse scenarios. Each dataset is split into two subsets: bandit feedback subset, to generate the logged data and evaluate policy values; and policy learning dataset, for training policies.

Table 1. Characteristics of the datasets used in the experiments.

Dataset	ecoli	glass	letter	optdigits	page-blocks	pendigits	satimage	vehicle	yeast
Classes	8	6	26	10	5	10	6	4	10
Features	7	9	16	64	10	16	36	18	8
Sample size	336	214	20000	5620	5473	10992	6435	846	1484

3.3 Hyperparameters

Hyperparameter tuning in this study was conducted for various estimators using a grid search method across specified ranges. Below, we outline the approach for each estimator and provide the ranges and specific tuning strategies applied.

- **TruncatedIPS**: The Truncated Inverse Propensity Scores (IPS) estimator includes a clipping constant M to control the influence of large propensity weights, balancing bias and variance. A grid of 30 geometrically spaced values between the 0.05 and 0.95 quantiles of the propensity weights was searched. The theory-suggested value $M = O(\sqrt{n})$ was also included.
- **SWITCH-DR**: The Switch Doubly-Robust (SWITCH-DR) estimator uses a threshold parameter τ to switch between DM and DR, activating DR for low-propensity weights and DM otherwise. The search grid for τ similarly spanned 30 values between the 0.05 and 0.95 quantiles of the propensity weights, following a conservative bias upper bound strategy.
- **CAB**: The Continuous Adaptive Blending (CAB) estimator uses a blending parameter M to adaptively weight the IPS and DM estimators based on propensity weights. A grid of 30 values was tested, ranging from $w_{0.05}$ to $w_{0.95}$.
- **DRos and DRps**: These Doubly-Robust estimators with optimistic (DRos) and pessimistic (DRps) shrinkages adjust weights with a parameter λ to balance variance and bias. The tuning grid for λ spanned from $0.01 \times (w_{0.05})^2$ to $100 \times (w_{0.95})^2$ for DRos, while DRps used $w_{0.05}$ to $w_{0.95}$.
- **IPS-λ**: The IPS-λ estimator modifies IPS by introducing a regularization parameter λ to achieve subgaussian concentration of propensity weights. The authors proposed a differentiable tuning objective, and we used a grid of 30 values in the range $(1 + \exp(-x))^{-1}$, with x linearly spaced from -10 to 10.
- **GroupIPS**: The GroupIPS estimator clusters actions based on their reward predictions to reduce variance in large action spaces. The number of clusters M was selected from {2, 4, 8, 16, 32}, as per the authors' guidance.

Overall, the tuning was conducted on nine datasets, under 90 different conditions combining target and logging policies, with five repetitions per condition to obtain robust MSE estimates. This process provided a comprehensive evaluation across various settings to identify the best hyperparameters and validate the cross-validated estimator selection approach used in this study.

3.4 Experimental Setup and Code

The experiments in this study were set up according to the cross-validation procedures and configurations detailed in the original paper. We followed a structured environment setup to ensure reproducibility and consistency. Below is a comprehensive description of the experimental setup:

- **Environment Setup**: Experiments were conducted on a machine with NVIDIA RTX 2080 Ti, with 11 GB GDDR6 and 4352 NVIDIA CUDA cores. The Python version used was 3.12. From the original paper it is stated that the code was tested with python version 3.10+
- **Cloning the Repository and Setting up the Environment**:
  1. Clone the repository
  2. Create a Python virtual environment
  3. Activate the virtual environment
  4. Install the required libraries from the `requirements.txt` file
- **Code Execution**: To reproduce the figures in the paper, we used specific commands associated with individual configuration files, as detailed below:
  - **Figure 1**: `python src/run.py -config configs/dr_strong.yaml`
  - **Figure 2**: `python src/run.py -config configs/dr_weak.yaml`
  - **Figure 3**: `python src/run.py -config configs/tuning.yaml`
  - **Figure 4**: `python src/run.py -config configs/ablation`
  - **Figure 5**: `python src/run.py -config configs/k_splits.yaml`
  - **Figure 6**: `python src/run.py -config configs/ocv_dm.yaml`

  Precomputed results were also provided in the "results" folder within the repository, enabling consistent comparison with the outcomes in the paper.
- **Evaluation Metrics**: The experiments were evaluated using Mean Squared Error (MSE) as the primary metric to assess estimator performance.

3.5 Hardware and Computational Requirements

The experiments were conducted on a machine with the following hardware specifications:

- **CPU**: 8-Core Intel Core i9-9900K
- **GPU**: NVIDIA GeForce RTX 2080 Ti
- **RAM**: 32GB

The overall runtime for the full set of experiments, including hyperparameter tuning, cross-validation runs, figure generation, and post-processing, ranged from 6 hours to 78 hours.

4 Results

Cross-validation consistently chooses a good estimator - To verify Claim 1, we ran the code given in the Github repository for 500 iterations across nine datasets (ecoli, glass, letter, optdigits, page-blocks, pendigits, satimage, vehicle, and yeast). For each dataset, we ran the program with a configuration that enabled comparison of several estimators, including IPS, DM, DR, and the OCV-based estimators, as well as baseline methods such as SLOPE and PAS-IF.
The reproduced results aligned perfectly with those reported in the original paper, with no deviations observed in the mean validation accuracy across all estimators. This supports the claim that OCV-based estimators effectively enhance model selection and performance across various dataset characteristics, particularly favoring the DR estimator where appropriate. The reproduced results, shown in Figure 1, confirm that OCV consistently outperformed other baseline methods, providing reliable estimator selection.

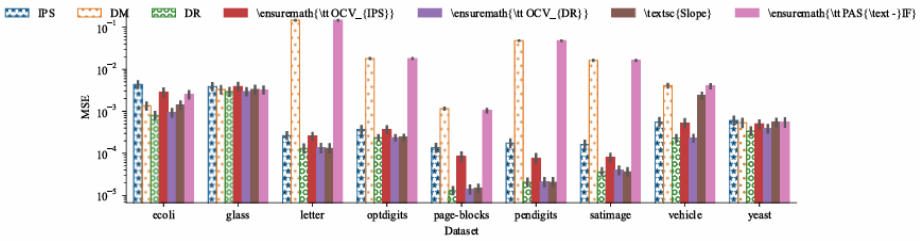

Figure 1. MSE of the proposed estimator selection methods, OCV_{IPS} and OCV_{DR}, compared against two other estimator selection baselines, SLOPE and PAS-IF

Cross-validation with DR performs well even when DR performs poorly - To verify Claim 2, we configured an experiment to evaluate OCV's adaptability in scenarios where the validation estimator does not necessarily align with the best estimator. This experiment aimed to test if OCV can still effectively minimize mean squared error (MSE) under these conditions, validating its flexibility and robustness in estimator selection.
The configuration for this experiment used a negative target policy temperature to create conditions where the optimal estimator might differ from the validator, specifically challenging OCV to adapt dynamically. Temperature set to -10, favoring lower-value actions. The results, presented in Figure 2, demonstrate that OCV effectively minimized MSE, selecting estimators that performed well despite the validator not directly aligning with the best-performing estimator. Again, the reproduced figure was exactly the same as the one given in the original paper.

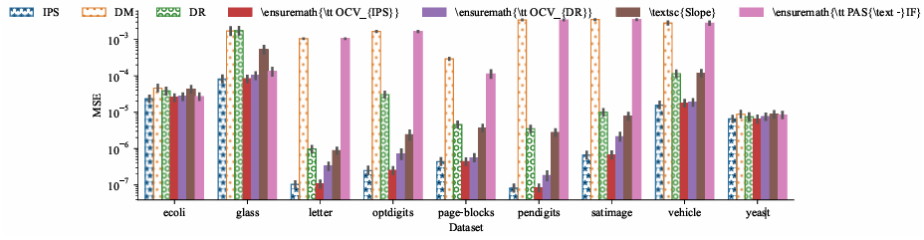

Figure 2. MSE of the the estimator selection methods for temperatures $\beta_0 = 1$ and $\beta_1 = -10$

OCV provides a robust solution for hyper-parameter tuning and estimator selection - To verify Claim 3, we ran experiments on several estimator models with different tuning methods, including TruncatedIPS, SwitchDR, CAB, DR_{OS}, DR_{PS}, IPS-λ, and GroupIPS. The experiments were conducted by varying key hyperparameters, such as temperature

values for the logging and target policies (ranging from -10 to 10), and tuning each estimator using four different methods: OCV with IPS as a validator, OCV with DR as a validator, SLOPE, and PAS-IF.
Given that the code encountered an error when attempting to run from scratch, the experiment was instead conducted using the pre-saved data file. Using these results, we were able to validate the third claim of the paper regarding the effectiveness of OCV in hyperparameter tuning. The findings, displayed in Figure 3, show that OCV achieved lower Mean Squared Error (MSE) across multiple estimators compared to baseline tuning methods, supporting the claim that OCV provides a robust and practical solution for tuning various models across different policy setups.

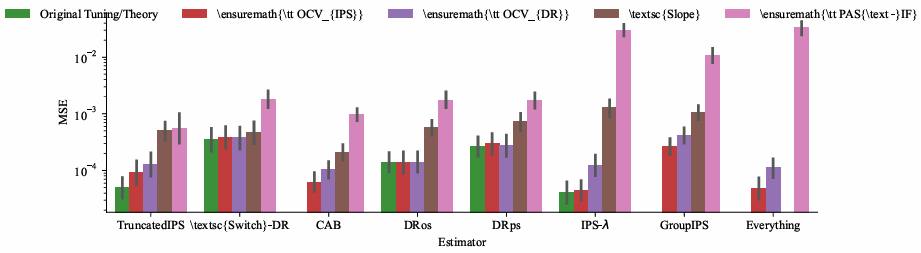

Figure 3. MSE of the the estimator selection methods applied to hyper-parameter tuning of various estimators

Improvements make standard cross-validation more stable - To verify Claim 4, we conducted ablation experiments on key components of the OCV method: the theory-driven training/validation split ratio and the one standard error rule. Starting with a standard 90/10 split and gradually adding these components, we found that both the one standard error rule and adaptive split ratio improved the stability and performance of the method (Figure 4).

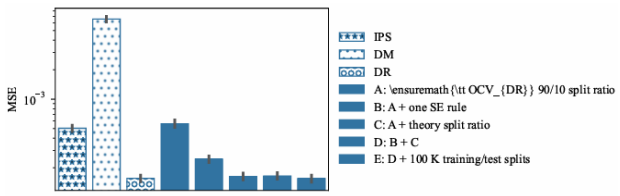

Figure 4. Ablation on proposed improvements with OCV_{DR}

However, increasing the number of splits did not always lead to diminishing returns. While Figure 5 from the original paper shows a consistent decrease in MSE with additional splits, our reproduced results (shown in Figure 6 below) exhibited fluctuations, particularly between 20 and 100 splits. Unlike the previous experiments, this time the reproduced results did not fully match the original paper's findings exactly. However, it can still be claimed that increasing the number of splits in general results in more robust and stable CV performance.

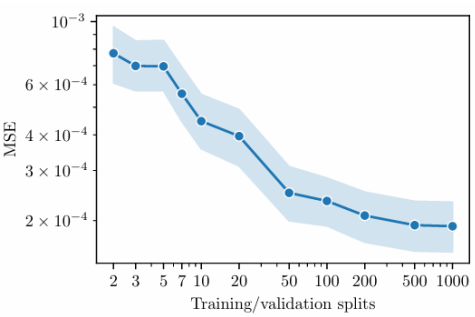

Figure 5. Ablation of the number of repeated training/validation splits with OCV_{DR} on the vehicle dataset over 500 runs (Original)

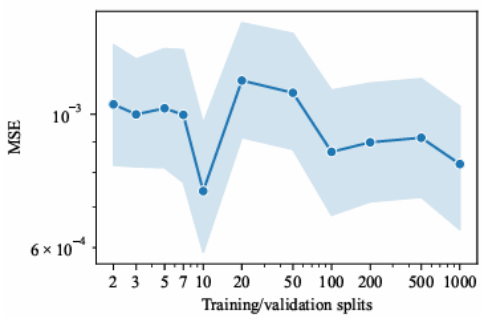

Figure 6. Ablation of the number of repeated training/validation splits with OCV_{DR} on the vehicle dataset over 500 runs (Reproduced)

The validator used in cross validation has to be unbiased - To verify Claim 5, we examined the effect of using a biased validator in cross-validation by testing with the Direct Method (DM), a known biased estimator. Using DM as the validator led the selection process to consistently favor DM itself due to alignment in bias, even when DM performed poorly. In contrast, using an unbiased validator like DR in OCV_{DR} avoided this issue, maintaining more reliable performance across datasets. This result confirms the importance of an unbiased validator for effective estimator selection in OPE. Running the code eventually gave us exactly the same Figure 6 in the original paper.

Cross-validation is computationally efficient - To verify Claim 6, we compared the computational efficiency of cross-validation methods, specifically OCV_{IPS}, OCV_{DR}, SLOPE, and PAS-IF. The results given in Table 3 of the original paper were reproduced and given in the table below, Both OCV_{IPS} and OCV_{DR} demonstrated significantly lower computational costs, with PAS-IF being over 100 times more costly due to its complex optimization process involving a neural network. This result confirms that cross-validation-based methods like OCV are highly efficient alternatives to PAS-IF for policy evaluation.

Method	OCV_{IPS}	OCV_{DR}	SLOPE	PAS-IF
Time	0.06s	0.13s	0.005s	13.91s

Table 2. Average computational cost of a single policy evaluation from Figure 1 when doing $K = 10$ training/validation splits with OCV_{DR}, OCV_{IPS}, SLOPE, and PAS-IF. Computed on XXX.

5 Discussion

Our experimental results provide strong support for the main claims of the original paper. Each reproduced claim confirmed the effectiveness of cross-validation (OCV) as a reliable approach for estimator selection and hyper-parameter tuning in off-policy evaluation (OPE) tasks. The reproduced results consistently aligned with the original findings with a minor exception, in overall validating that OCV can adaptively select high-performing estimators and maintain robust performance across diverse datasets and experimental conditions.

Moreover, our results reinforce the claim that unbiased validators are essential in ensuring fair estimator selection within the OCV framework. The computational efficiency of OCV, particularly when compared to more resource-intensive methods like PAS-IF, was clearly demonstrated in our reproduced results. This positions OCV as a practical and adaptable choice for a variety of OPE scenarios, showing it can perform reliably without excessive computational costs.

5.1 What was easy

The reproduction process benefited greatly from the well-structured and informative GitHub repository provided by the authors. The codebase was organized logically, with clear documentation and separate scripts dedicated to each experiment, making it straightforward to navigate and execute specific tasks. This structure allowed us to reproduce each of the key experiments systematically, with minimal adjustments. In fact, only modification made was changing the style of the texts in reproduced figures.
Additionally, the repository included detailed instructions on how to run the code, and each experiment was supported by a corresponding configuration file. This setup made it relatively easy to verify the claims of the paper, as each experiment could be reproduced independently.

5.2 What was difficult

One of the main challenges was understanding the theorems presented in Section 5 of the paper. The theoretical components required a strong mathematical background to comprehend. While they didn't impact the replication process directly, they were challenging to fully understand.
Another difficulty arose with one of the experiments, which could not be reproduced from scratch due to an error in the provided code. To address this, we relied on the saved output file provided in the repository to replicate the results. While this workaround allowed us to obtain the correct figures, it limited our ability to fully re-run and verify the experiment independently.
The most significant challenge, however, was the high computational cost. Some experiments took up to 78 hours to complete due to the intensive cross-validation and tuning processes. To manage these requirements, we utilized computational nodes provided by our university, which involved additional setup and coordination. This involved us manually installing new nvidia drivers, along with compatible CUDA and cuDNN versions. Interestingly enough, estimated running time (ERN) for cpu-based run of one task was 56 hours, while with gpu support it was reduced to 52 hours, which we found to be a very small difference given the fact the machine we were using is powerful. This can perhaps suggest a lack of gpu usage, or the lack of parallel execution on the cores available.

5.3 Communication with original authors

We did not communicate with the original authors during this replication study. Main reason is that the instructions provided in the paper and GitHub repository were clear and comprehensive, allowing us to reproduce most results without needing additional guidance. We encountered issues with reproducing the third experiment, and a different result with reproducing the fifth experiment which could have been reasons to reach out, but we chose not to due to time constraints for submitting this assignment.

Acknowledgements

The structure and tone of this report were inspired by [2], the winner of the 2022 ML Reproducibility Challenge. However, the content is entirely original.

References

1. M. Cief, B. Kveton, and M. Kompan. **Cross-Validated Off-Policy Evaluation**. 2024. arXiv: 2405.15332 [cs.LG].
2. S. B. L. Seungjae Ryan Lee. **[Re] Pure Noise to the Rescue of Insufficient Data**. 2022.