Journal of Open Aviation Science (2026), Vol.**4**
doi:10.59490/joas.2026.9001

JOAS

EDITORIAL

Reviews and Responses for A Physics-Guided Gradient Boosting Framework for Open-Source Aviation Fuel Estimation

Authors: Yiannis Grigoriou, Eftychios Eftychiou, Christian Vitale, Giorgos Pettemeridis, Nicolas Souli, and Panayiotis Kolios

Reviewers: Richard Alligier, Recep Ayyildiz, and Quinten Goens

Editor: Enrico Spinielli

1. Original paper

The DOI for the original paper is https://doi.org/10.59490/joas.2026.8764

2. Review - round 1

2.1 Reviewer 1

Review of "A Physics-Guided Gradient Boosting Framework for Open-Source Aviation Fuel Estimation"

This article presents the methodology used by the authors to obtain a machine learning model predicting the amount of fuel burnt at certain segment of flight. This work was done to compete in the Performance Review Commission (PRC) Data Challenge 2025.

This article describes the data and methods used to obtain good prediction. An analysis across different factors is performed: phase, aircraft type and category (narrow or widebody), and flight haul. The authors use Gradient Boosting Machines, along with features selection and data augmentation.

This work presents an interesting approach to aircraft fuel consumption modeling particularly regarding the data augmentation process. There are several areas that would benefit from clarification or refinement to improve the significance of the results. These concerns include the potential for data leakage during the cross-validation process, randomness of the training of the Gradient Boosting model, the rationale behind aircraft type encoding and data augmentation. I am confident that these improvements can be done or at least discussed in the manuscript, I recommend an accept with minor revision.

1 Detailed Remarks/Questions

1. Line 371, I am wondering if it is a good idea to ordinally encode the aircraft type. Doing so, you order aircraft types, did you choose a particular ordering, for instance increasing MTOW, or is it somewhat "random"? If it was not "random", maybe clarify this in the text. I wonder if all the important features (identified as such in Figure 3) related to the aircraft type (fuselage Height, the wing mac Flaps Sf/S, Limits OEW) are not proxies of the aircraft type/size?

2. In Figure 3b, which method is used to compute the feature importance? The method used should be named in the article. Ideally, it could be nice to use permutation importance instead of the get score of XGBoost.
3. In order to select the best hyper-parameter of the XGBoost model, from your github code, you use optuna on a cross validation on 80% of the training set, using the already selected features (computed on this same 80% training set). Then you took the top 10 hyper-parameters that have the best score computed by the cross-validation. Using these 10 hyper-parameters, you train 10 models on the 80% training set, and score them once again but on the 20% unseen part of training set (serving as a validation set here). The Section 3.4.4 do not describes this. A simpler and more classical approach would have been to use optuna on a cross-validation on 100% of the training set. Did you gain some benefits from doing it the way you did? If so, it would be nice to have some words on this in the text.
4. Ideally, methodologically speaking, the Forward Selection algorithm should be part of the learning algorithm (using the pipeline concept in scikit-learn), as it might make different choices depending on the fold used, and on the parameter of the XGBoost algorithm.
5. As a side note, from the github code, you performed a cross-validation on the segments, and as a consequence, some segments of the same flights could be in different folders, leading to data leakage. As the segments of the same flight are consecutive in the data set, a simple way to mitigate data leakage is to set shuffle=False. If you correct this, it could be nice to write a text raising awareness on this kind of possible data leakage.
6. From Table 5 to Table 11, the different gradient boosting model compared are built with a random seed, leading to a randomness in the score obtained. To reduce this randomness, it could be nice to have different random seed, and compute the average MAE/RMSE in this Table. Reducing this randomness might be helpful as sometimes results are very close (Table 11 on Test).
7. From the text, it is not clear what is the rationale for using 24996 synthetic examples? Why not more? Or at least as much as necessary to have the same number of segments for narrowbody and widebody?
8. What is the impact of taking more long segment than the general widebody population? Is it necessary to do so?
9. In my opinion, the data augmentation process is the most interesting part of the paper, it could be nice to better justify it through different experiments. Is it required to add noise to the generated example to improve predictive performance? If so, would adding noise to all the examples in the dataset improve the predictive performance? Why not use a fixed α and choose an α maximizing the predictive performance?

2 Text Improvement

2.1 Clarification

1. Lines 426-427 are confusing, I believe it is a reference to a log-scaling of the variable to be predicted. This has already been mentioned line 370, so I would remove the reference to this on lines 426-427.
2. Either a reference for Sequential Forward Selection or an explanation could be nice.
3. Figure 6, the RMSE Loss is around 1, orders of magnitude larger than the 200-ish kg discussed along the paper. I guess that it is the log-transformed loss: the RMSE of the model when predicting the log of the fuel burnt. Please clarify this in the text, or plot the real RMSE loss. Even Figure 6b is not clear in this matter. It is specified "(log scale)" on the legend, but the values on the "y-axis" change linearly. If the log of the RMSE is plotted, then the RMSE ranges from $10^{0.28}$ to $10^{0.40}$ which is not the expected ranges of values for the RMSE.

2.2 Better organization

1. I personally find that the introduction is lengthy with repeated references with the "Related Work" part (references 6, 7 and 8). Maybe introduction could be more concise.
2. Lines 431-432 is in Section "Hyperparameter Tuning and Final ensemble" and it refers to the final generalization assessment, which is separate from the hyperparameter tuning and to the OpenAP physics-based model which has no hyperparameter tuning. Maybe move the lines 428-432, starting from "Final generalization", to the beginning of Section 4 "Performance Evaluation".
3. It could be nice to better introduce the "OpenAP physics model" in line 435, followed by its coverage limitation.

2.2 Reviewer 2

In this paper, the authors are presenting an open-source approach using XGBoost, a gradient boosting ensemble for aircraft fuel consumption prediction part of the PRC Data Challenge 2025. The approach consists of pipelines which incorporates data ingestion, physics guided feature engineering, and feature selection on ADS-B trajectories. The proposed model is compared to baselines models and shows promising results.

Major Points

1. Title - the title is a bit of misleading. Physics guided gradient boosting sounds like a new approach in which you incorporate physics model into gradient boosting such as PINNs - but in reality, its physics guided feature engineering/selection and using a gradient boosting model, and also the approach is more like a pipeline than a framework no?
2. Images and tables caption need more text. Table 8 comes a bit late - its mentioned before figure 5 but comes afterwards
3. Introduction and Related Works needs a bit of revision. Both are repeating the motivation more than once. Be careful to not incorporate related works into introduction and be careful of the red line for following - think about the research question here generally.
4. On line 232 and 362 you state similar things, the second is a bit misleading "The model is based on an XGBoost gradient-boosted ensemble." First its repetitive and this sounds like you modified the model or changed it and the underlying model is XGBoost, which is not the case. And also saying gradient-boosted ensemble after XGBoost is strange because that's what XGBoost is, which is used. So, something like the model used is XGBoost which is a gradient-boosting ensemble for example. Be consistent with the wording also for gradient-boosting.
5. Statement on line 442 is strange since LightGBM is also a gradient boosting framework.
6. Line 552: Isn't the training curve still declining? And why is the overfitting here acceptable? I thought transferability is important?

Overall, the presented approach is well presented and explained. However, there are some statements and unclear wordings which needs to be addressed. I recommend acceptance subject to revisions. Addressing the points raised above would significantly enhance the overall quality of the paper.

2.3 Reviewer 3

Review of A Physics-Guided Gradient Boosting Framework for Open-Source Aviation Fuel Estimation

1 Introduction

This document is a review of the submission A Physics-Guided Gradient Boosting Framework for Open-Source Aviation Fuel Estimation by Grigoriou et al., submitted to the Journal of Open Avia-

tion Science (JOAS) as part of the EUROCONTROL PRC 2025 Data Challenge. The paper presents a modular, three-phase pipeline for segment-level fuel consumption prediction that integrates ADS-B trajectories with METAR observations, airport infrastructure, aircraft performance coefficients, and regional passenger load factors. Physics-informed features are derived from the OpenAP FuelFlow model with dynamic mass tracking, a synthetic widebody augmentation strategy is applied to address class imbalance, and an XGBoost ensemble is tuned via Bayesian optimisation following Sequential Forward Selection.

The review is organised as follows. Section 2 summarises the strengths of the submission. Section 3 lists potential issues identified in the manuscript. Section 4 raises questions and suggestions for further research. Section 5 lists minor textual issues. Line and section references correspond to the reviewed pre-print (v1).

2 Strengths

- The introduction and related work are well structured. Section 2 organises prior work around three macro-themes (physics-based modelling, data-driven prediction from proprietary data, and open-data approaches), and Section 2.5 makes the methodological gap explicit, motivating the contributions clearly.
- The synthetic widebody augmentation strategy (Section 3.3.6) is well motivated and addresses the narrowbody/widebody class imbalance directly, raising the widebody training share from 18.7% to 34.1%. The ablation in Tables 10 and 11 quantifies its effect cleanly.

3 Potential issues

- Source of the METAR data not cited in the methodology. METAR is introduced as a key input, but its source (Iowa Environmental Mesonet [42]) is only cited in the Open Data Statement on page 20. We suggest citing it at first use in Section 3.2.1.
- OpenAP FuelFlow used both as a feature and as the baseline. The OpenAP fuel estimate is an input to XGBoost (Table 1) and also the physics baseline in Table 6. Beating OpenAP when the model already sees its prediction is expected, so the comparison overstates the gain over the physics baseline. Reporting Table 6 with the OpenAP feature removed, or isolating its contribution in the ablation, would clarify the actual improvement.

4 Questions and suggestions for further research

- Quantify the contribution of the OpenAP feature. Adding a Table 11 row with the OpenAP feature group removed would isolate how much of the gain comes from the learned residual correction over the physics estimate.
- Narrowbody/widebody split as an intermediate model. Section 6 already proposes per-type models, but given that fuselage height alone accounts for 52.9% of the XGBoost gain (Section 4), a simpler narrowbody/widebody two-model split may capture much of that benefit at a fraction of the complexity. Reporting this configuration would clarify whether finer per-type stratification is worth the added cost.

5 Typographical and editorial remarks

The following terminology and capitalisation choices are used inconsistently across the manuscript:

- “Wing Mac” vs. “Wing MAC”, both are used.
- “narrowbody/widebody” vs “narrow-body/wide-body”, both are used.

- “Modeling” vs “Modelling”, both are used.
- “high-fidelity” vs “high fidelity”, both are used.
- “hyper-parameter(s)” vs “hyperparameter(s)”, both are used.
- “takeoff” vs “take-off” vs “take off”, both are used.

3. Response - round 1

3.1 Response to reviewer 1

3.1.1 Comment 1

Line 371, I am wondering if it is a good idea to ordinally encode the aircraft type. Doing so, you order aircraft types, did you choose a particular ordering, for instance increasing MTOW, or is it somewhat “random”? If it was not “random”, maybe clarify this in the text. I wonder if all the important features (identified as such in Figure 3) related to the aircraft type (fuselage Height, the wing mac Flaps Sf/S, Limits OEW) are not proxies of the aircraft type/size?

Response

We thank the reviewer for this observation. The aircraft types were ordinally encoded using integer labels assigned alphabetically. Therefore, the encoding does not follow a physical ordering, such as increasing MTOW. We have clarified this point in Section 3.4.2.

We also agree that several of the important features shown in Figure 3(b), including fuselage height, wing MAC, flap-area ratio, and OEW-related variables, are correlated with aircraft type and size and may therefore partly act as proxies for the aircraft category. However, these variables describe distinct physical and operational characteristics of the aircraft. Retaining them alongside the aircraft-type identifier allows the feature-selection procedure to evaluate these characteristics individually and select those that are most relevant to the prediction task. We have clarified this rationale in Section 3.4.3, where the retained feature groups are introduced.

Specifically, the text in Line 365 was revised as follows:

“Categorical features (aircraft type, origin, destination) are converted into integer labels, with aircraft-type labels assigned alphabetically rather than according to a physical ordering such as increasing MTOW. All numerical features are standardized using a scaler fitted exclusively on the training fold to prevent data leakage.”

Additionally, the following text was added at Line 383:

“Although several aircraft performance parameters are correlated with aircraft type and size and may therefore partly act as proxies for the aircraft category, they describe distinct physical and operational characteristics. Retaining these variables alongside the aircraft-type identifier allows the feature-selection procedure to evaluate them individually and select the aircraft properties that are most relevant to the prediction task.”

3.1.2 Comment 2

In Figure 3b, which method is used to compute the feature importance? The method used should be named in the article. Ideally, it could be nice to use permutation importance instead of the get score of XGBoost.

Response

The feature importance shown in Figure 3b is computed using the XGBoost get_score method with the importance type set to gain.

Regarding the suggestion of permutation importance, we chose the model-intrinsic gain-based metric because our feature set contains several correlated aircraft physical specifications (such as OEW, wingspan, and wing MAC). Permuting correlated features individually makes it difficult to assess each feature's contribution to model performance. The gain-based metric avoids this issue by directly representing the splitting decisions made by the algorithm.

Specifically, the text in Line 405 was revised as follows:

"The gain-based importance score is retrieved using the XGBoost get_score method with the importance type set to gain, which measures the average fractional contribution of each feature to the reduction of the mean squared error (MSE) across all splits in the ensemble of trees."

3.1.3 Comment 3

In order to select the best hyper-parameter of the XGBoost model, from your github code, you use optuna on a cross validation on 80% of the training set, using the already selected features (computed on this same 80% training set). Then you took the top 10 hyper-parameters that have the best score computed by the cross-validation. Using these 10 hyper-parameters, you train 10 models on the 80% training set, and score them once again but on the 20% unseen part of training set (serving as a validation set here). The Section 3.4.4 do not describes this. A simpler and more classical approach would have been to use optuna on a cross-validation on 100% of the training set. Did you gain some benefits from doing it the way you did? If so, it would be nice to have some words on this in the text.

Response

We thank the reviewer for this detailed observation regarding the hyperparameter tuning workflow.

The two-stage tuning strategy was designed to prevent overfitting and hyperparameter optimization leakage. In a standard cross-validation run on the full dataset, the search algorithm (Optuna) evaluates hundreds of trials and selects the best configuration. This process can lead to overfitting to the cross-validation partitions.

By partitioning the data into an 80% tuning split and a 20% validation holdout, the optimization process is constrained to search only on the 80% split. The evaluation of the top 10 configurations on the 20% validation holdout serves as a selection stage. This structure helps mitigate optimization leakage that could occur if we selected the best parameter set from hundreds of Optuna trials using only the cross-validation folds. Specifically, the text in Line 436 was revised as follows:

"To prevent overfitting to the cross-validation folds during hyperparameter optimization, a two-stage tuning strategy was implemented. First, Optuna was executed using 3-fold cross-validation on an 80% training split of the data to identify configuration candidates. Second, the top 10 hyperparameter configurations from this search were retrained on the full 80% training split and evaluated on the remaining unseen 20% validation holdout split. The single parameter configuration achieving the lowest validation RMSE on this holdout set was selected for the final model. This two-stage workflow ensures that the final parameters generalize well to unseen data by mitigating optimization leakage that can arise from evaluating many trials on the same cross-validation folds."

3.1.4 Comment 4

Ideally, methodologicaly speaking, the Forward Selection algorithm should be part of the learning algorithm (using the pipeline concept in scikit-learn), as it might make different choices depending on the fold used, and on the parameter of the XGBoost algorithm.

Response

We agree that nesting feature selection within the cross-validation loop is methodologically ideal to capture fold-specific variations. However, running Sequential Forward Selection (SFS) inside the outer cross-validation or hyperparameter optimization loop is computationally prohibitive for this dataset.

Selecting 74 features from 133 candidates using 5-fold cross-validation inside SFS requires training 7,142 XGBoost models. Nesting this inside the outer cross-validation folds would require training over 35,000 XGBoost models on a dataset of over 130,000 rows. To ensure computational efficiency, SFS was executed once using a reference model with fixed hyperparameters. This offline selection identifies a stable and physically consistent feature set while remaining computationally feasible. Specifically, the text in Line 376 was revised as follows:

"To ensure computational efficiency, feature selection is performed once offline rather than nested inside the hyperparameter optimization loop, as nesting SFS within the cross-validation partitions would require training over 35,000 XGBoost models on a dataset of over 130,000 rows."

3.1.5 Comment 5

As a side note, from the github code, you performed a cross-validation on the segments, and as a consequence, some segments of the same flights could be in different folders, leading to data leakage. As the segments of the same flight are consecutive in the data set, a simple way to mitigate data leakage is to set shuffle=False. If you correct this, it could be nice to write a text raising awareness on this kind of possible data leakage.

Response

We thank the reviewer for this observation. We tested the suggested configuration by setting shuffle=False during cross-validation and train-validation splitting. However, when evaluating on the hidden test set, which was never used at any point for model training or hyperparameter selection, the configuration with shuffle=False performed worse, increasing the test RMSE from 214.92 kg to 219.38 kg. Consequently, we selected the original configuration for the final model.

3.1.6 Comment 6

From Table 5 to Table 11, the different gradient boosting model compared are built with a random seed, leading to a randomness in the score obtained. To reduce this randomness, it could be nice to have different random seed, and compute the average MAE/RMSE in this Table. Reducing this randomness might be helpful as sometimes results are very close (Table 11 on Test).

Response

We thank the reviewer for this constructive suggestion. To address this concern, the validation metrics in Table 5 have been re-evaluated across 5 seeds (42, 43, 44, 45, and 46). The table has been updated to report the mean and standard deviation for each model, and corresponding discussions have been added to Section 4.1:

As shown in the updated Table 5, the standard deviations of the non-linear models are small (under 1.1 kg for MAE and under 15 kg for RMSE), which confirms the statistical stability of the results. The

relative ranking of the models remains consistent across all random seeds, with the proposed SFS-tuned model consistently outperforming Ridge Regression, Random Forest, and LightGBM.

For Table 11, the large size of the test set (N = 61,745) ensures high statistical stability. Therefore, we did not repeat the computationally expensive ablation experiments across multiple seeds, as the low standard deviations in Table 5 confirm that seed randomness has a negligible effect on our findings. Specifically, the text in Line 459 was revised as follows:

"On the validation set, to account for statistical variations from the random seed split, metrics are reported as the mean and standard deviation across 5 random seeds (42, 43, 44, 45, and 46). Ridge Regression achieves a validation RMSE of 903.6 ± 81.1 kg and R^2 = 0.2393 ± 0.1485, confirming the strongly non-linear nature of the task. Random Forest and LightGBM reduce validation RMSE to 229.5 ± 12.6 kg and 235.6 ± 14.6 kg, respectively, with R^2 values of 0.9513 ± 0.0044 and 0.9486 ± 0.0054. The XGBoost reference reaches an RMSE of 200.3 ± 13.4 kg and R^2 = 0.9628 ± 0.0043, demonstrating that the gradient boosting inductive bias is well suited to this tabular regression task. The proposed model, which incorporates SFS feature selection, synthetic widebody augmentation, and physics-informed features, achieves a validation RMSE of 203.1 ± 14.1 kg and R^2 = 0.9618 ± 0.0046. While the full-feature reference model achieves a marginally lower mean RMSE on validation, the proposed SFS-tuned version demonstrates significantly better generalization on the hidden test set (RMSE 214.9 kg versus 223.2 kg), confirming that the compact feature subset effectively mitigates overfitting."

3.1.7 Comment 7

From the text, it is not clear what is the rationale for using 24,996 synthetic examples? Why not more? Or at least as much as necessary to have the same number of segments for narrowbody and widebody?

Response

The choice of 24,996 synthetic widebody samples was determined through validation sweeps during the development phase. Increasing the synthetic dataset size beyond this threshold did not provide statistically significant improvements in validation RMSE, but it increased training time and memory footprint during hyperparameter tuning. This size was selected as the optimal trade-off between model generalization and computational cost.

Specifically, the text in Line 338 was revised as follows:

"The choice of 24,996 synthetic widebody samples was determined through validation sweeps during the development phase. Increasing the synthetic dataset size beyond this threshold did not provide statistically significant improvements in validation RMSE, but it increased training time and memory footprint during hyperparameter tuning. This size was selected as the optimal trade-off between model generalization and computational cost."

3.1.8 Comment 8

What is the impact of taking more long segment than the general widebody population? Is it necessary to do so?

Response

Oversampling long segments is necessary for two operational reasons.

First, widebody aircraft are designed and primarily operated on long-haul routes, but standard ADS-B datasets are dominated by short regional flights. Oversampling ensures the synthetic data has a strong

representation of steady-state cruise phases.

Second, because we optimize Root Mean Squared Error (RMSE) in absolute kilograms, errors on long-duration high-burn flights are penalized disproportionately and dominate the loss function. Oversampling these segments ensures the model learns high-fuel states accurately, preventing large prediction outliers.

3.1.9 Comment 9

In my opinion, the data augmentation process is the most interesting part of the paper, it could be nice to better justify it through different experiments. Is it required to add noise to the generated example to improve predictive performance? If so, would adding noise to all the examples in the dataset improve the predictive performance? Why not use a fixed α and choose an α maximizing the predictive performance?

Response

We thank the reviewer for this insightful comment. To address these questions, an experimental evaluation was conducted comparing the proposed Gaussian-perturbed widebody augmentation against a baseline duplication scheme with 0% noise perturbation.

Specifically, the text in Line 527 was revised as follows:

"To assess the specific impact of the Gaussian perturbation used during data augmentation, an ablation test was conducted comparing the proposed perturbed augmentation against a simple sample duplication scheme with 0% noise perturbation. Training the model with duplicate synthetic widebody samples instead of perturbed ones increases the overall validation MAE from 66.03 kg to 66.34 kg and the RMSE from 165.16 kg to 165.38 kg. For widebody segments, using simple duplication increases the MAE from 115.64 kg to 116.35 kg. For narrowbody segments, the MAE increases from 40.57 kg to 40.68 kg, and the RMSE increases from 86.58 kg to 87.52 kg. While the performance difference is small, adding stochastic perturbation to the augmented samples provides a regularizing effect that helps the model generalize beyond the exact observed trajectories."

Regarding the second question, adding noise to real training examples would corrupt high-fidelity measurements and degrade predictive accuracy. Noise perturbation is only applied to synthetic widebody samples to fill sparse regions in the feature space and address class imbalance without distorting the real observations.

Finally, the randomized noise scale $\alpha \sim \mathcal{U}[0.05, 0.15]$ is a percentage representing 5% to 15% of each feature's standard deviation σ_j. This ratio ensures scale-invariant noise and prevents larger features from dominating the perturbation. Varying α dynamically acts as a regularizer during training to improve robustness.

Specifically, the text in Line 350 was revised as follows:

"This scale-invariant ratio prevents larger features from dominating the perturbation. Varying α dynamically during training acts as a regularizer to improve model robustness."

3.1.10 Comment 10 (Clarifications and Text Improvements)

The reviewer lists several points for clarification and organization:

1. Lines 426-427 are confusing, I believe it is a reference to a log-scaling of the variable to be predicted. This has already been mentioned line 370, so I would remove the reference to this on lines 426-427.

Response

We would like to thank the reviewer for this comment. To address the comment, we removed the duplicate reference to log-scaling in lines 426-427. The log transformation and inverse transformation are already described in Section 3.4.2, where fuel targets are log-transformed during training and exponentiated at inference time to recover fuel values in kilograms. The revised sentence in Section 3.4.4 now avoids repeating this explanation.
Specifically, the text in Line 433 was revised as follows: "The final architecture consists of a vertically stacked set of gradient-boosted trees in which each subsequent tree minimizes the residual error of the preceding ensemble. During inference, the predictions are combined to generate the final segment-level fuel-consumption estimate."

2. Either a reference for Sequential Forward Selection or an explanation could be nice.

Response

We have added a reference for Sequential Forward Selection in Section 3.4.3 of the revised manuscript.

3. Figure 6, the RMSE Loss is around 1, orders of magnitude larger than the 200-ish kg discussed along the paper. I guess that it is the log-transformed loss: the RMSE of the model when predicting the log of the fuel burnt. Please clarify this in the text, or plot the real RMSE loss. Even Figure 6b is not clear in this matter. It is specified "(log scale)" on the legend, but the values on the "y-axis" change linearly. If the log of the RMSE is plotted, then the RMSE ranges from $10^{0.28}$ to $10^{0.40}$ which is not the expected ranges of values for the RMSE.

Response

Thank you for pointing this out. We agree that the previous Figure 6 was unclear because it showed the RMSE in the log-transformed target space, while the rest of the paper reports RMSE in kilograms. We have revised Figure 6 so that the plotted RMSE values are now shown in the original fuel-burn scale, in kg. Therefore, the figure is now consistent with the RMSE values discussed throughout the paper. The y-axis label and caption have also been updated to remove the ambiguity about "log scale."Figure 6 has been updated to report RMSE in the original fuel-burn scale, in kg.

4. I personally find that the introduction is lengthy with repeated references with the "Related Work" part (references 6, 7 and 8). Maybe introduction could be more concise.

Response

The reviewer is correct in pointing out that the Introduction included repetitive references and overlapped with material discussed in the Related Work section. We therefore revised the manuscript to reduce this overlap. Specifically, the discussion in lines 28–44 on parametric and physics-based fuel-flow estimation is condensed, while the details of these studies are included in the Related Work section (in lines 102–125) where physics-based and phase-specific fuel-modeling approaches are presented in greater depth. We also summarized the discussion in lines 45–65 on machine learning fuel prediction approaches by removing detailed descriptions of individual model architectures, flight phases, aircraft types, and numerical performance values. This material is presented in more depth in the Related Work section (in lines 126–152). As a result, the revised Introduction now focuses on motivating the proposed work, the limitations of parametric and data-based approaches, as well as the need for a transferable open-data framework.
Specifically, the text in Line 28 was revised as follows: "Accurate fuel-flow estimation is challenging because it depends on aircraft performance, operating procedures, aircraft condition, and uncertain environmental factors. Traditional methods employ parametric models, such as the Base of

Aircraft Data (BADA), which provide useful physics-based fuel-flow baselines under nominal conditions; however, these techniques can not fully capture aircraft-specific variability, aging effects, or deviations from real-world operations. To address these limitations, recent work has proposed data-driven and enhanced physics-based approaches that learn from operational or high-fidelity simulator data while using BADA as a reference baseline."
Specifically, the text in Line 35 was revised as follows: "More recently, machine-learning methods, including ensemble models trained on operational recorder data, have shown strong potential for aviation fuel prediction across multiple flight phases and aircraft types. However, many high-performing approaches are still based on proprietary Quick Access Recorder (QAR) or flight-recorder datasets from specific airlines, aircraft types, or operating conditions, which limits their reproducibility and transferability to heterogeneous open-data settings. This motivates the need for an open-data framework that can combine noisy ADS-B trajectories with external information, such as meteorological observations, aircraft characteristics, airport metadata, and passenger load factors."

5. Lines 431-432 is in Section "Hyperparameter Tuning and Final ensemble" and it refers to the final generalization assessment, which is separate from the hyperparameter tuning and to the OpenAP physics-based model which has no hyperparameter tuning. Maybe move the lines 428-432, starting from "Final generalization", to the begining of Section 4 "Performance Evaluation".

Response

We agree that the description of the final test set and baseline coverage is better suited for the performance evaluation section. Specifically, the text in Line 447 was revised as follows:
"Final generalization is assessed on the hidden test set, which consists of the rank and final datasets provided with the challenge and contains 61,745 segments. The OpenAP physics-based model covers approximately 54.5% of validation and 73% of test segments and serves as the baseline."

6. It could be nice to better introduce the "OpenAP physics model" in line 435, followed by its coverage limitation.

Response

We thank the reviewer for this suggestion. Specifically, the text in Line 448 was revised as follows:
"The OpenAP physics-based model, which leverages open-source aircraft performance parameters [1] to calculate fuel flow from kinematic state variables, serves as the baseline. However, its coverage is limited because OpenAP only supports a subset of the aircraft types present in its dataset. Specifically, the physics baseline covers approximately 54.5% of validation and 73% of test segments, whereas the proposed machine learning model achieves full coverage."

3.2 Response to reviewer 2

3.2.1 Comment 1

Title, the title is a bit of misleading. Physics guided gradient boosting sounds like a new approach in which you incorporate physics model into gradient boosting such as PINNs, but in reality, its physics guided feature engineering and selection and using a gradient boosting model. Also, the approach is more like a pipeline than a framework.

Response

We thank the reviewer for this comment. We agree that the title could be interpreted as suggesting that physics is incorporated directly into the gradient-boosting architecture, similarly to physics-informed neural networks. However, we respectfully prefer to retain the current title because physics-based information directly guides the model through OpenAP-derived fuel and mass-state features, aircraft performance parameters, and physical constraints applied during mass estimation and data augmentation. The XGBoost architecture itself remains unchanged.

We also retain the term framework because the proposed approach comprises the complete end-to-end methodology, including data integration, preprocessing, physics-guided feature generation, feature selection, hyperparameter optimization, and prediction. The term pipeline is used to describe the modular implementation of this framework. To avoid ambiguity, we have clarified these points in Sections 2.5 and 3.1.

Specifically, the text in Line 189 was revised as follows:

"It combines heterogeneous open data sources with OpenAP-derived fuel and mass-state features, aircraft performance parameters, and physical constraints applied during mass estimation and data augmentation, which together guide a standard gradient-boosting model for direct segment-level fuel estimation."

Additionally, the text in Line 199 was revised as follows:

"The predictive framework developed in this study comprises the complete end-to-end methodology and is implemented as a modular, three-phase pipeline designed to handle the high-dimensional, noisy, and heterogeneous nature of the aviation data."

3.2.2 Comment 2

Images and tables caption need more text. Table 8 comes a bit late, as it is mentioned before Figure 5 but comes after it.

Response

We would like to thank the reviewer for this comment. We therefore revised the figure and table captions to make them more descriptive. For the ordering issue, we moved Table 8 so that it now appears before Figure 5, matching the order in which the two items are discussed in the text.

Revised Figure captions:

- **Figure 1**
  Previous caption: System Overview
  Revised caption: Overview of the proposed physics-guided fuel-estimation pipeline, from data ingestion and preprocessing to OpenAP feature generation, feature selection, hyperparameter tuning, and XGBoost prediction.
- **Figure 2**
  Previous caption: Global flight route coverage of the PRC 2025 dataset
  Revised caption: Global route coverage of the PRC 2025 dataset, showing origin destination arcs colored by route frequency.
- **Figure 3**
  Previous caption: Feature analysis for fuel consumption prediction
  Revised caption: Feature analysis for the selected fuel-prediction variables: (a) Pearson correlations with fuel consumption and (b) XGBoost gain-based feature importance.
- **Figure 4**

Previous caption: Phase-wise fuel flow distribution (kg/s)
Revised caption: Phase-wise fuel-flow distribution, showing differences in fuel-flow rates across climb, cruise, and descent segments.

- **Figure 6**
  Previous caption: Learning dynamics of the XGBoost model.
  Revised caption: Learning dynamics of the final XGBoost model, showing RMSE convergence across boosting rounds and cross-validated RMSE as training-set size increases.

Revised Table captions:

- **Table 1**
  Previous caption: Integrated data sources and feature fusion logic for the fuel prediction pipeline
  Revised caption: Integrated data sources used in the fuel-prediction pipeline, including join keys, extracted features, and functional roles.
- **Table 2**
  Previous caption: Hierarchical Rule-Based Flight Phase Classification Criteria
  Revised caption: Hierarchical rule-based criteria used to classify trajectory points into operational flight phases.
- **Table 3**
  Previous caption: Training set composition before and after synthetic widebody augmentation.
  Revised caption: Training-set composition before and after synthetic widebody augmentation, reported by aircraft category.
- **Table 4**
  Previous caption: XGBoost Hyperparameter Search Space and Optimal Values
  Revised caption: Hyperparameter search space and selected optimal values for the final XGBoost model.
- **Table 5**
  Previous caption: Model performance on the validation and hidden test sets.
  Revised caption: Overall model performance on the held-out validation set and hidden PRC 2025 test set, reported using MAE, RMSE, and R^2.
- **Table 7**
  Previous caption: Per-phase performance on validation and hidden test sets.
  Revised caption: Per-phase performance of the proposed model on the validation and hidden test sets, reported for climb, cruise, and descent segments.
- **Table 10**
  Previous caption: Ablation study: effect of synthetic widebody augmentation on validation performance.
  Revised caption: Validation-set ablation showing the effect of synthetic widebody augmentation on overall, narrowbody, and widebody performance.
- **Table 11**
  Previous caption: Component ablation results for the proposed XGBoost fuel consumption model
  Revised caption: Component ablation results showing the contribution of augmentation, METAR features, load factors, dynamic mass tracking, and elapsed-time features.

3.2.3 Comment 3

Introduction and Related Works needs a bit of revision. Both are repeating the motivation more than once. Be careful to not incorporate related works into introduction and be careful of the red line for following. Think about the research question here generally.

Response

We would like to thank the reviewer for his positive criticism. This comment is addressed along with Reviewer's 1 request to condense the Introduction and reduce repetition with the Related Work section. In the original manuscript, the Introduction included a detailed discussion of BADA and physics-based fuel-flow models, which overlapped with the more systematic review of physics-based and phase-specific fuel modeling in the Related Work section. We therefore condensed the Introduction and maintained the detailed literature discussion in Related Work. Similarly, the original Introduction section discussed machine-learning fuel-prediction studies in detail, including specific architectures, aircraft types, flight phases, and performance metrics. This also overlapped with the Related Work discussion of data-driven prediction using QAR/FDR data. The revised Introduction now provides only the high-level motivation for machine-learning-based fuel prediction, while the detailed review remains in Section 2.2. To make the distinction clearer, the Introduction now focuses on the general motivation, the limitations of parametric and data-driven approaches, and the need for a transferable open-data framework. The Related Work section remains organized around the main literature themes: physics-based modeling, data-driven learning, open-data trajectory approaches, and transferability. Finally, we added a concise objective statement immediately before the contribution list to define the focus of this study.

Specifically, the text in Line 53 was revised as follows: "Accordingly, this work focuses on developing and evaluating a reproducible, physics-guided machine-learning framework for direct segment-level fuel-consumption estimation using heterogeneous open aviation data."

3.2.4 Comment 4

On line 232 and 362 you state similar things, and the second is a bit misleading "The model is based on an XGBoost gradient-boosted ensemble." First its repetitive and this sounds like you modified the model or changed it and the underlying model is XGBoost, which is not the case. And also saying gradient-boosted ensemble after XGBoost is strange because that is what XGBoost is, which is used. So, something like the model used is XGBoost which is a gradient-boosting ensemble for example. Be consistent with the wording also for gradient-boosting.

Response

We would like to thank the reviewer for pointing out this issue. We agree that the phrase "XGBoost gradient-boosted ensemble" is redundant and may imply that the standard XGBoost architecture was modified. Therefore, we revised the terminology throughout the manuscript. In Section 3.1, the model is now introduced as an XGBoost model based on gradient-boosted decision trees, while in Section 3.4.1, the repeated phrase was removed and the model is referred to consistently as the XGBoost model.

Specifically, the text in Line 215 was revised as follows: "Finally, the core inference engine is an XGBoost model based on gradient-boosted decision trees."

Specifically, the text in Line 357 was revised as follows: "The XGBoost model development process proceeds through three sequential stages."

3.2.5 Comment 5

Statement on line 442 is strange since LightGBM is also a gradient boosting framework.

Response

Thank you for pointing this out. We agree that the wording was imprecise, since LightGBM is also a gradient boosting framework.

We have revised the sentence to avoid implying that gradient boosting is specific to XGBoost. The text now distinguishes the proposed XGBoost-based implementation from the LightGBM baseline while acknowledging that both are gradient-boosted tree models.

"On the validation set, Random Forest and LightGBM reduce validation RMSE to 230.2 kg and 236.1 kg, respectively, with R^2 values of 0.951 and 0.948. The XGBoost reference reaches an RMSE of 201.4 kg and R^2 = 0.963, indicating that this XGBoost configuration provides the strongest validation performance among the evaluated tree-based ensemble baselines."

3.2.6 Comment 6

Line 552: Isn't the training curve still declining? And why is the overfitting here acceptable? I thought transferability is important?

Response

We thank the reviewer for this correction. The training curve continues to decline while the validation curve plateaus.

Although a gap remains between the curves, the stable validation error indicates that the model is not overfitting to the training data. This generalization is further supported by the model's robust performance on the unseen hidden test sets (combined RMSE of 214.92 kg).

Specifically, the text in Line 564 was revised as follows:

"The training RMSE continues to decline while the validation RMSE plateaus after approximately 1,000 rounds. Although a gap remains between the two curves, the stable validation error indicates that the model is not overfitting to the training data. This generalization is further supported by the model's robust performance on the unseen hidden test sets."

3.3 Response to reviewer 3

3.3.1 Comment 1

Source of the METAR data not cited in the methodology. METAR is introduced as a key input, but its source (Iowa Environmental Mesonet [42]) is only cited in the Open Data Statement on page 20. We suggest citing it at first use in Section 3.2.1.

Response

We thank the reviewer for pointing this out. We have added the citation to the Iowa Environmental Mesonet (IEM) weather archive at the first mention of the METAR data integration in Section 3.2.3 of the revised manuscript.

3.3.2 Comment 2

OpenAP FuelFlow used both as a feature and as the baseline. The OpenAP fuel estimate is an input to XGBoost (Table 1) and also the physics baseline in Table 6. Beating OpenAP when the model already sees its prediction is expected, so the comparison overstates the gain over the physics baseline. Reporting Table 6 with the OpenAP feature removed, or isolating its contribution in the ablation, would clarify the actual improvement.

Response

We agree that comparing the model to a baseline whose prediction is visible as a feature could overstate the performance gain. We updated Table 6 to report the performance of the model trained without the OpenAP feature openap_fuel_kg. Even without this feature, the data-driven framework achieves an RMSE of 115.30 kg on validation and 126.80 kg on test, outperforming the OpenAP baseline by a significant margin.

Specifically, the text in Line 475 was revised as follows:

"To provide a fair and direct comparison against the physics-based baseline, Table 6 evaluates the proposed XGBoost model, the model with the physics baseline feature removed, and the OpenAP FuelFlow model exclusively on the flight segments where OpenAP successfully computed a prediction (N = 17,073 for validation and N = 45,324 for test). On these subsets, the proposed model achieves an RMSE of 117.62 kg and 126.18 kg, respectively, while the model trained without the OpenAP feature achieves 115.30 kg and 126.80 kg, both outperforming the OpenAP baseline estimates of 218.90 kg and 193.58 kg. Although the model without the physics feature achieves a slightly lower validation error, including OpenAP as an input feature improves prediction accuracy on the unseen test dataset. Overall, these results demonstrate that even without direct access to the physics estimate, the data-driven framework recovers the physical relations and outperforms the baseline by a large margin."

3.3.3 Comment 3

Quantify the contribution of the OpenAP feature. Adding a Table 11 row with the OpenAP feature group removed would isolate how much of the gain comes from the learned residual correction over the physics estimate.

Response

To isolate and quantify the exact contribution of the physics-guided feature on overall generalization, we have added a new row to Table 11 representing the model trained without the openap_fuel_kg feature.

The results show that removing the dynamic physics feature increases the overall test RMSE to 222.2 kg on the Final test set. This confirms that the physics-guided feature improves generalization on unseen test flights. We have updated Table 11 and the corresponding discussion in Section 4.5.

3.3.4 Comment 4

Narrowbody and widebody split as an intermediate model. Section 6 already proposes per-type models, but given that fuselage height alone accounts for 52.9% of the XGBoost gain (Section 4), a simpler narrowbody and widebody two-model split may capture much of that benefit at a fraction of the complexity. Reporting this configuration would clarify whether finer per-type stratification is worth the added cost.

Response

We thank the reviewer for this suggestion.

While a split model simplifies the feature space, a single model is simpler to deploy and maintain because it avoids managing multiple training pipelines and hyperparameter sets. A single model also allows shared physical relationships to be learned across all aircraft categories. This is particularly beneficial for widebodies, which represent only 18.7% of the training data.

3.3.5 Comment 5

The following terminology and capitalization choices are used inconsistently across the manuscript:

- "Wing Mac" versus "Wing MAC", both are used.
- "narrowbody/widebody" versus "narrow-body/wide-body", both are used.
- "Modeling" versus "Modelling", both are used.
- "high-fidelity" versus "high fidelity", both are used.
- "hyper-parameter(s)" versus "hyperparameter(s)", both are used.
- "takeoff" versus "take-off" versus "take off", both are used.

Response

We thank the reviewer for identifying these inconsistencies. We have carefully reviewed the manuscript and standardized the terminology throughout. Specifically, we now consistently use "Wing MAC," "narrowbody/widebody," "modeling," "high-fidelity," "hyperparameter(s)," and "takeoff."

References

[1] Junzi Sun, Jacco M. Hoekstra, and Joost Ellerbroek. "OpenAP: An Open-Source Aircraft Performance Model for Air Transportation Studies and Simulations". In: *Aerospace* 7.8 (2020). ISSN: 2226-4310. DOI: 10.3390/aerospace7080104. URL: https://www.mdpi.com/2226-4310/7/8/104.