

mzLearn as a data-driven LC/MS signal detection algorithm that enables pre-trained generative models for untargeted metabolomics

Corresponding Author: Dr Leila pirhaji

This file contains all reviewer reports in order by version, followed by all author rebuttals in order by version.

Version 0:

Reviewer comments:

Reviewer #1

(Remarks to the Author)

The authors have developed a user-friendly peak detection algorithm, mzLearn, which requires no parameter tuning from the user. By applying this algorithm to 20,548 publicly available LC-MS datasets (.mzML format) and constructing VAE models from the resulting data matrices, the authors propose a data processing pipeline for LC-MS-based untargeted metabolomics that yields biologically and clinically informative representations. Considering the diversity of LCMS conditions (mobile phase, gradient, column, polarity, MS type, etc.) employed in this field and inevitable drift of sensitivity and retention time, the establishment of a unified data processing framework remains to be challenging. Therefore, the development of a data processing methodology that can handle diverse LC-MS conditions is an important and welcome contribution to the untargeted metabolomics community. From this perspective, the generation of synthetic QC in mzLearn is also beneficial. However, there are open questions and issues to support the authors' claims.

Major comments:

1. The authors claim that no parameter settings are required on the user side in mzLearn. While the authors mention the use of "generous thresholds" (0.5 for auto-correlation and 0.0 for cross-correlation) in the initial step (as quoted below), the specific logic for iteratively updating these thresholds in relation to the following description remains unclear. (The source code for peak picking and alignment is also unavailable on GitHub.) Even if no user-side parameter input is required, the key internal parameters should be explicitly specified, along with a detailed explanation of how they are initialized and updated in a data-driven manner. If there are parameters that have been heuristically determined through the authors' verification, these should also be presented.

P21. "We filter patches whose autocorrelation scores do not exceed a certain threshold. Initially, a generous autocorrelation threshold of 0.5 is selected, and a stricter threshold is later learned by the algorithm."

P22. "Patch pairs above the cross-correlation threshold are kept as correlated patches.

Initially, a generous cross-correlation threshold of 0 is selected, and this threshold is refined by the algorithm through an iterative learning process."

2. (Related to 1.) To ensure the reproducibility of the extracted peak information, the parameters determined internally in a data-driven manner should be output together with the results when mzLearn finished processing data. In addition, the analysis parameters used in this study (determined by mzLearn algorithm) should be addressed in the manuscript.

3. Although raw data can be obtained from the repository index, comparisons of mzLearn and other methods cannot be evaluated in detail due to no information about the ion peak table. The authors should make the result matrix open in the format of Supplementary Data.

4. The authors compared the number of extracted peaks by mzLearn with conventional methods and claim that mzLearn outperformed the detection numbers in almost all datasets. However, one of the key metrics that researchers and users truly seek or wish to evaluate is the number of ion peaks that are worth for annotation and downstream analysis. Even if mzLearn can now extract weak signals that may lead to false positives in the conventional methods, the effectiveness of this algorithm

is limited when most of these are isotopic peaks. Since it is unclear whether the result ion tables from mzLearn and conventional methods include only monoisotopic peaks or mixture of isotopic peaks, this should be specified. Furthermore, if the current comparison includes isotopic ions, comparisons between monoisotopic peaks are also informative for researchers.

5. In Figure 4, the authors claim that the UMAP topology of the trained VAE model captures biological features compared to raw data. However, visual clustering based solely on category labels is insufficient, as various UMAP topologies can emerge from raw data depending on parameter settings such as `n_neighbors`, `min_dist`, and `metric`. The authors should demonstrate whether the marker molecules highlighted in the original studies of the datasets are actually reflected in the topology and clusters of the disease groups. In addition, to enhance the interpretability of the model, the authors should provide examples of the molecules or ion peaks that contribute to each topology and perform feature importance analysis such as SHAP.

6. In this study, the VAE model was constructed using 20,548 blood samples. Expanding the training dataset will likely be necessary in future applications. Therefore, it would be beneficial to either provide data or at least include a discussion on the scalability of VAE models in untargeted metabolomics, particularly regarding the trade-off relationship between the number of batches incorporated (i.e., the extent of batch effects considered) and the amount of biological information retained in the data.

7. Regarding the prognosis prediction and stratification of ccRCC patients using the VAE model, the authors should discuss the lack of significant predictive performance observed in the mTOR inhibitor group.

Minor comments:

1. The mzLearn source code available on GitHub (mzEmbed) is insufficiently documented. At a minimum, for the methods intended to be called by users, the code should include clear explanations of their functionality, arguments, and expected data types.

2. It is recommended that the authors provide test data along with a notebook-style example on GitHub to demonstrate how to construct the VAE model with mzLearn pipeline.

Reviewer #2

(Remarks to the Author)

The work by Pirhaji et al. presents mzLearn, a data-driven, non-parametric algorithm for LC-MS signal detection that addresses significant gaps in LC-MS untargeted metabolomics. The authors demonstrate that mzLearn produces high-quality results compared to established algorithms (XCMS, Asari, etc) and, importantly, enables dataset integration across studies for improved generative model pretraining for translational applications. The manuscript is well-written with convincing results.

I have several comments that would hope authors can clarify:

1. Peak Detection Algorithm

XCMS and Asari's key contributions lie in their novel peak detection algorithms from raw spectra, which serve as foundations for subsequent tools. The Methods section begins with detecting peaks from mzML files, but it's unclear which peak detection algorithm mzLearn employs. Please specify the underlying peak detection methodology.

2. Implementation Details

The key contribution of mzLearn appears to be the integration of proven empirical practices for feature detection, QC and retention time drift adjustment within a unified workflow. However, critical implementation details are missing. Is mzLearn implemented in Python (like Asari), R (like XCMS), or another language? What are the key dependencies and runtime requirements?

3. Benchmarking Reproducibility

Benchmarking tests can be inherently biased due to data selection, parameter choices, and other factors. Established algorithms like XCMS and Asari have gained community trust through public availability and third-party validation over time. The provided GitHub repository contains mzEmbed rather than mzLearn itself. Could you clarify how researchers can access mzLearn for independent benchmarking and validation?

4. Pretraining Dataset Selection

The ability to pretrain generative models across multiple studies represents an important advancement enabled by mzLearn. Given the diversity of LC-MS datasets across instruments and column types, improper dataset selection could dilute signals and degrade performance. While Supplementary Table 4 shows the pretraining studies (blood + HILIC + ion mode), what guidelines should researchers follow when selecting appropriate pretraining datasets? Are there plans to develop "mixture of experts" approaches for handling sufficiently different data types?

Reviewer #3

(Remarks to the Author)

1. The abstract should be accessible to a broader scientific audience, not just machine learning specialists. For example, the phrase "improved downstream clinical and survival tasks" is vague, improved compared to what? What specific outcomes and how much improvement?
2. Moderate Claims. Terms like "first-in-class" and "democratizes" are overstated.
3. Benchmarking against well-established tools. Instead of supplementary table 2 (which does not seem to be referenced in the manuscript) and numbers buried in text, consider a bar plot for a figure in the main manuscript to clearly support the claim "mzLearn outperforms existing methods in both signal quantity and quality".
4. Include the reference of the papers corresponding to all the public datasets used. This is important to acknowledge the work of the original authors and appreciate that they made their data public, which fosters publishing data.
5. Improve description of validation approach. In the manuscript "targeted metabolomics features" are mentioned as ground truth but important details are missing. How were targeted features originally identified? What kind of matching and tolerance windows were applied? What makes them "true positives"?
6. Approach seems to be limited to MS1: The manuscript does not explain how the algorithm or pipeline considers MS/MS for metabolite identification or how signals are mapped to metabolite IDs. This represents a significant gap in the methodology description. How were metabolites identified from the untargeted signals? What databases or libraries were used?
7. There seems to be a confusion between the signal detection algorithm and the generative model. As described, mzLearn is not generative. Despite the manuscript's framing, mzLearn appears to be a signal detection and processing algorithm rather than a trained generative model. While the authors describe an "iterative, data-driven process," this seems to refer to parameter optimization and threshold refinement rather than machine learning training in the traditional sense. The manuscript provides no description of training data, loss functions, or model architecture for mzLearn itself. This distinction is critical because it means mzLearn is a deterministic processing pipeline that adapts its parameters based on data characteristics, not a learned model that requires training on labeled examples or that can generate new synthetic data.

Version 1:

Reviewer comments:

Reviewer #1

(Remarks to the Author)

The authors have adequately addressed all major and minor comments. The revised manuscript demonstrates substantial improvement in methodological transparency, reproducibility, and interpretability. The inclusion of parameter documentation, supplementary data matrices, and SHAP-based feature attribution particularly strengthens the work. Although a dedicated comparison of monoisotopic peaks could further validate the conclusions, the current justification and scope are reasonable. Overall, I consider that the authors have satisfactorily addressed all previous concerns, and the revised manuscript is now suitable for publication in Communications Chemistry.

Reviewer #2

(Remarks to the Author)

I am happy with the responses to my questions. During the evaluation (Oct 12, 2025), I notice the website (<https://mzlearn.com/>) is not accessible (500 server error).

I have no other comments.

Reviewer #3

(Remarks to the Author)

The authors have addressed this reviewer's concerns and suggestions.

Open Access This Peer Review File is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

In cases where reviewers are anonymous, credit should be given to 'Anonymous Referee' and the source.

The images or other third party material in this Peer Review File are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this license, visit <https://creativecommons.org/licenses/by/4.0/>

Reviewers' comments:

Reviewer #1 (Remarks to the Author):

The authors have developed a user-friendly peak detection algorithm, mzLearn, which requires no parameter tuning from the user. By applying this algorithm to 20,548 publicly available LC-MS datasets (.mzML format) and constructing VAE models from the resulting data matrices, the authors propose a data processing pipeline for LC-MS-based untargeted metabolomics that yields biologically and clinically informative representations. Considering the diversity of LCMS conditions (mobile phase, gradient, column, polarity, MS type, etc.) employed in this field and inevitable drift of sensitivity and retention time, the establishment of a unified data processing framework remains to be challenging. Therefore, the development of a data processing methodology that can handle diverse LC-MS conditions is an important and welcome contribution to the untargeted metabolomics community. From this perspective, the generation of synthetic QC in mzLearn is also beneficial. However, there are open questions and issues to support the authors' claims.

We thank the reviewer for the thoughtful, constructive comments and for recognizing the need for robust, parameter-free processing across diverse LC-MS settings. Below we respond point-by-point and describe the analyses, clarifications, and additions we have made in the revised manuscript and Supplement.

Major comments:

1. The authors claim that no parameter settings are required on the user side in mzLearn. While the authors mention the use of “generous thresholds” (0.5 for auto-correlation and 0.0 for cross-correlation) in the initial step (as quoted below), the specific logic for iteratively updating these thresholds in relation to the following description remains unclear. (The source code for peak picking and alignment is also unavailable on GitHub.) Even if no user-side parameter input is required, the key internal parameters should be explicitly specified, along with a detailed explanation of how they are initialized and updated in a data-driven manner. If there are parameters that have been heuristically determined through the authors' verification, these should also be presented.

P21. ” We filter patches whose autocorrelation scores do not exceed a certain threshold. Initially, a generous autocorrelation threshold of 0.5 is selected, and a stricter threshold is later learned by the algorithm.”

P22. “Patch pairs above the cross-correlation threshold are kept as correlated patches. Initially, a generous cross-correlation threshold of 0 is selected, and this threshold is refined by the algorithm through an iterative learning process.

We appreciate the request for greater transparency. We clarify that mzLearn requires no user-defined parameters, but it does use a small set of internal seeds that are subsequently refined from the data through an iterative procedure. For example, at the initiation stage (Stage 0), autocorrelation seed $T_{AC}^{(0)} = 0.5$ to do a loose noise screen and cross-correlation seed $T_{XC}^{(0)} = 0$ to retain all non-negative pairs. These are not tuned by users and are not reused after the first pass.

To address this and further clarifying the other learned parameters, we added a new subsection in Methods (“**Data-driven threshold learning and update schedule**”) that:

- (i) enumerates all internal parameters,
- (ii) specifies how they are initialized and updated, and
- (iii) lists the fixed heuristic constants used by policy.

Code availability:

As noted by the reviewer, the public GitHub hosts the downstream embedding/fine-tuning code (mzEmbed), not the detector/alignment engine. mzLearn itself is publicly usable for academic/non-profit users via a hosted, versioned Docker pipeline at mzlearn.com: each job authenticates, pulls a pinned image (e.g., `mzlearn/public:v4.5`), runs with a unique `JOB_CODE`, user-set CPU count, and a read-only mount of the mzML directory. We have updated the manuscript’s Code Availability statement to clarify this.

2. (Related to 1.) To ensure the reproducibility of the extracted peak information, the parameters determined internally in a data-driven manner should be output together with the results when mzLearn finished processing data. In addition, the analysis parameters used in this study (determined by mzLearn algorithm) should be addressed in the manuscript.

For reproducibility, mzLearn already writes a machine-readable `params_overall.csv` with each run. This file records the dataset- and instrument-specific settings that mzLearn infers before learning thresholds—e.g., m/z precision from mzML files, the initialized coarse bins (MZ_{coarse} , RT_{coarse}), and global $m/z/rt$ bounds/dispersion used for tiling and search. These fields correspond directly to the initialization rules and primitives described in Methods (coarse patching; seed thresholds for auto- and cross-correlation; pin definition) and the subsequent data-driven learning. We now deposit `params_overall.csv` alongside the feature matrices to document the run-time configuration.

3. Although raw data can be obtained from the repository index, comparisons of mzLearn and other methods cannot be evaluated in detail due to no information about the ion peak table. The authors should make the result matrix open in the format of Supplementary Data.

We now provide the full result matrices as a **Supplementary Data** in a single Excel workbook (.xlsx). Each sheet corresponds to one study and contains the mzLearn output, a normalized sample $\times$ feature intensity matrix with feature-level metadata (detection frequency, median m/z, median RT). This update is explicitly stated in the manuscript.

4. The authors compared the number of extracted peaks by mzLearn with conventional methods and claim that mzLearn outperformed the detection numbers in almost all datasets. However, one of the key metrics that researchers and users truly seek or wish to evaluate is the number of ion peaks that are worth for annotation and downstream analysis. Even if mzLearn can now extract weak signals that may lead to false positives in the conventional methods, the effectiveness of this algorithm is limited when most of these are isotopic peaks. Since it is unclear whether the result ion tables from mzLearn and conventional methods include only monoisotopic peaks or mixture of isotopic peaks, this should be specified. Furthermore, if the current comparison includes isotopic ions, comparisons between monoisotopic peaks are also informative for researchers.

Thank you for raising this important point. In this work, mzLearn outputs ion-level features and we do not perform de-isotoping or metabolite annotation. Therefore, the result tables (and our peak-count benchmark) include a mixture of monoisotopic and isotopic/adduct ions. We compared methods at the same detection stage (pre-de-isotoping) to avoid introducing downstream, tool-specific heuristics and parameters, which are beyond the scope of this paper. Our aim is to maximize high-quality ion coverage; this larger, more consistent feature set provides richer evidence for downstream grouping of isotopes/adducts and compound identification using established post-processing tools. We have clarified this explicitly in Methods.

5. In Figure 4, the authors claim that the UMAP topology of the trained VAE model captures biological features compared to raw data. However, visual clustering based solely on category labels is insufficient, as various UMAP topologies can emerge from raw data depending on parameter settings such as n_neighbors, min_dist, and metric. The authors should demonstrate whether the marker molecules highlighted in the original studies of the datasets are actually reflected in the topology and clusters of the disease groups. In addition, to enhance the interpretability of the model, the authors should provide examples of the molecules or ion peaks that contribute to each topology and perform feature importance analysis such as SHAP.

We agree that UMAP visuals alone are insufficient and that topology can vary with UMAP hyperparameters. In our analysis, UMAP settings were already fixed and applied identically to the raw ion matrix and to the VAE embedding; labels were used only for coloring.

To provide quantitative interpretability, we computed SHAP values on the supervised fine-tuned models (a prognostic VAE across all patients and an ICI-predictive VAE within the Nivolumab cohort), rather than on the unsupervised VAE/UMAP, because SHAP requires a prediction function and yields signed, additive attributions tied to clinical endpoints.

We ran SHAP on held-out baseline samples using a training-set background (GradientExplainer), ranked ions by mean(|SHAP|) (global importance), and used cohort-mean signed SHAP to indicate direction (protective vs harmful). We report Top-20 ions per model (shown in the updated **Figure 6**).

The Top-20 overlap was 3/20 (15%): FT8656/FT8665 (a +1.003 Da isotopic pair at identical RT, i.e., the same underlying lipid signal) and FT4553 (C6 carnitine). The prognostic list emphasizes membrane/redox lipid state (e.g., PE plasmalogens C38:6, C38:7, C36:3; PUFA LPE C18:2), whereas the Nivolumab-predictive list emphasizes mitochondrial/FAO readiness (e.g., acyl-carnitines C6, C8, C12:1, C18:1; LPC C16:1).

We do not perform de-isotoping or annotation in this paper; analyses are at the ion level. The presence of an isotopic pair among top SHAP features indicates a coherent underlying signal rather than random noise.

Collectively, these changes (fixed UMAP + SHAP on supervised models) address the concern by moving beyond label visualization to quantified, biologically coherent feature contributions, now documented in the manuscript and **Figure 6**.

6. In this study, the VAE model was constructed using 20,548 blood samples. Expanding the training dataset will likely be necessary in future applications. Therefore, it would be beneficial to either provide data or at least include a discussion on the scalability of VAE models in untargeted metabolomics, particularly regarding the trade-off relationship between the number of batches incorporated (i.e., the extent of batch effects considered) and the amount of biological information retained in the data

We agree that scalability and batch heterogeneity are central for untargeted metabolomics. We added a short Discussion paragraph clarifying that our model is trained on the mzLearn MS¹ feature matrix (not raw chromatograms) using mini-batch stochastic gradient descent (SGD)—i.e., parameter updates computed on small batches of samples—so per-epoch compute scales approximately linearly with the number of samples, and memory is governed by batch size $\times$ feature dimension. Upstream, mzLearn applies physics-guided RT-drift alignment and light run-order normalization and avoids aggressive batch removal to reduce the risk of erasing biological signal. As more batches are integrated, there is an inherent trade-off, stronger correction suppresses site effects but may attenuate true biology, whereas weaker correction preserves biology at the cost of residual batch structure. Our default targets only technically plausible artifacts and leaves remaining variation to the generative model; future large-scale deployments may add batch covariates or domain-invariant regularization if needed.

7. Regarding the prognosis prediction and stratification of ccRCC patients using the VAE model, the authors should discuss the lack of significant predictive performance observed in the mTOR inhibitor group.

Thank you for the opportunity to clarify. The result is expected and by design. Our predictive VAE is trained with an adversarial objective to isolate nivolumab (ICI)–specific survival signal while explicitly suppressing treatment-agnostic patterns that also appear in the everolimus (mTOR) arm. Concretely, the primary model learns to predict OS in the ICI cohort, while a concurrent adversarial head on the mTOR cohort penalizes any features that carry over; thus, a non-significant separation in the mTOR arm validates the model’s drug specificity, rather than indicating poor performance. This complements our separate prognostic VAE (treatment-agnostic), which captures outcome signal common to both arms. We now state this explicitly in the revision.

Minor comments:

1. The mzLearn source code available on GitHub (mzEmbed) is insufficiently documented. At a minimum, for the methods intended to be called by users, the code should include clear explanations of their functionality, arguments, and expected data types.

We’ve added comprehensive, NumPy-style docstrings to all user-facing functions and a new API Overview section in the README. These document each method’s purpose, inputs (names, types, shapes), outputs/return types, expected file formats (CSV), and example usage. We also added minimal runnable examples to help users reproduce the figures and train/fine-tune models end-to-end.

2. It is recommended that the authors provide test data along with a notebook-style example on GitHub to demonstrate how to construct the VAE model with mzLearn pipeline.

We added a lightweight synthetic test dataset (CSV floats; shape $[n, 2736]$) at `mzembed_demo/data/synthetic_small/` and four notebook-style examples: `01_quickstart_pretrain.ipynb`, `02_finetune_and_evaluate.ipynb`, `03_adversarial_finetune.ipynb`, and `04_synergistic_learning.ipynb` (all CPU-friendly). A brief guide is included in `mzembed_demo/README_DEMO.md`. We use synthetic data to avoid size/privacy constraints while demonstrating the exact pipeline and file formats end-to-end.

We hope these additions address the reviewer’s concerns.

Reviewer #2 (Remarks to the Author):

The work by Pirhaji et al. presents mzLearn, a data-driven, non-parametric algorithm for LC-MS signal detection that addresses significant gaps in LC-MS untargeted metabolomics. The authors demonstrate that mzLearn produces high-quality results compared to established algorithms (XCMS, Asari ,etc) and, importantly, enables dataset integration across studies for improved generative model pretraining for translational applications. The manuscript is well-written with convincing results.

We thank the reviewer for the careful reading and thoughtful suggestions. We respond point-by-point below and indicate the concrete manuscript changes we have made.

I have several comments that would hope authors can clarify:

1. Peak Detection Algorithm

XCMS and Asari's key contributions lie in their novel peak detection algorithms from raw spectra, which serve as foundations for subsequent tools. The Methods section begins with detecting peaks from mzML files, but it's unclear which peak detection algorithm mzLearn employs. Please specify the underlying peak detection methodology.

Thank you for the opportunity to clarify. mzLearn does not call XCMS or Asari; it implements a correlation-based, multi-sample peak detector that operates directly on centroided MS¹ mzML. In brief:

1. Coarse tiling & adaptive patches. We tile the RT–m/z plane using instrument precision and split/merge patches based on local point spacing.
2. Within-sample screen. Each patch is tested by autocorrelation to retain chromatographic-like signals (one-time seeds: ACF=0.5, CC=0).
3. Cross-sample confirmation under drift. We compute cross-correlation across samples while constraining RT shifts with a learned RT-drift map anchored by high-confidence “pins.”
4. Data-driven cutoffs (no user tuning). The initial seeds are replaced by learned thresholds computed from the cohort (lower-tail quantiles over pin scores; reapplied over up to three passes).
5. Peak boundaries & quantification. Final patch limits are “scissored” at local minima around the apex; we report apex RT/m/z and integrated intensity.

The output is an ion-level feature table; because we do not perform de-isotoping or identification in this paper, tables may include both monoisotopic and isotopic/adduct ions.

2. Implementation Details

The key contribution of mzLearn appears to be the integration of proven empirical practices for feature detection, QC and retention time drift adjustment within a unified workflow. However, critical implementation details are missing. Is mzLearn implemented in Python (like Asari), R (like XCMS), or another language? What are the key dependencies and runtime requirements?

Thank you for the suggestion. Implementation. mzLearn is implemented end-to-end in Python, with the detector/alignment built on the standard scientific stack (NumPy/SciPy/pandas) and the learning components in PyTorch (fine-tune VAEs; optional GPU). Dimensionality reduction and simple probes use umap-learn and scikit-learn. mzLearn reads centroided MS1 from mzML via a Python parser and is distributed with a Docker image/requirements file that pins versions to ensure reproducibility.

Runtime requirements. Runtime is dominated by the cross-sample correlation and iterative re-alignment steps and therefore grows with the number of mzML files in the cohort. In our largest single-machine run (no cluster), we processed 2,075 files in ≈ 2.5 days; memory usage is modest because processing is tiled and batched. For larger studies, multi-core parallelism (and, if available, multi-node execution) is recommended. These practical considerations are now summarized in the manuscript, and added a section named mzLearn implementation and runtime to the Method.

3. Benchmarking Reproducibility

Benchmarking tests can be inherently biased due to data selection, parameter choices, and other factors. Established algorithms like XCMS and Asari have gained community trust through public availability and third-party validation over time. The provided GitHub repository contains mzEmbed rather than mzLearn itself. Could you clarify how researchers can access mzLearn for independent benchmarking and validation?

Response. We appreciate this request. The public GitHub hosts mzEmbed (embedding/fine-tuning). The mzLearn detector/alignment engine is provided as a version-pinned Docker image that researchers run locally: authenticate, pull the pinned image (e.g., `mzlearn/public:v4.5`), mount the local mzML directory read-only, and set the number of CPU cores to use. The pipeline executes on the user's machine (no GPU required) and returns the ion-level feature table plus **params_overall.csv** (run metadata) together with the exact image tag for reproducibility.

Details are stated in Code Availability.

To facilitate independent evaluation, we also provide all benchmarking outputs as Supplementary Data in a single Excel workbook (.xlsx) covering 15 benchmarking datasets: each sheet corresponds to one study and contains the mzLearn normalized sample $\times$ feature intensity matrix with feature-level metadata (detection frequency, median m/z, median RT). These matrices can be used directly for future benchmarking and comparison.

4. Pretraining Dataset Selection

The ability to pretrain generative models across multiple studies represents an important advancement enabled by mzLearn. Given the diversity of LC-MS datasets across instruments and column types, improper dataset selection could dilute signals and degrade performance. While Supplementary Table 4 shows the pretraining studies (blood + HILIC + ion mode), what guidelines should researchers follow when selecting appropriate pretraining datasets? Are there plans to develop "mixture of experts" approaches for handling sufficiently different data types?

We agree that different LC-MS experiments constitute distinct data modalities (matrix, LC chemistry, polarity, instrument family/resolution, and RT window), and indiscriminate mixing can dilute signal. Accordingly, our current pretraining corpus is restricted to blood, HILIC-positive studies (Supplementary Table 4). For assembling a corpus, we recommend matching modality first—same matrix, LC mode, and polarity—while keeping instrument characteristics and m/z range comparable and using centroided MS¹. Chromatographic conditions should have overlapping gradient length/RT span. Heterogeneity is useful for robustness, but we filter unstable runs by basic QC and prevent any single study from dominating; we also hold out a few studies to detect negative transfer and exclude domains that degrade validation performance. We have clarified these details in the manuscript method section.

Separately, we are developing modality-aware pretraining using a mixture-of-experts: domain-specialized experts (e.g., blood/HILIC+, blood/RP-) combined by a lightweight gating function driven by run-level metadata (matrix, mode, polarity, instrument, gradient length) or a brief calibration pass, preserving modality-specific structure while enabling reuse across datasets. We have clarified these details in the manuscript discussion section.

We hope these revisions address the reviewer's questions.

Reviewer #3 (Remarks to the Author):

We thank the reviewer for the careful reading and helpful suggestions. Below we respond point-by-point and indicate the concrete changes made in the revised manuscript and Supplement.

1. The abstract should be accessible to a broader scientific audience, not just machine learning specialists. For example, the phrase “improved downstream clinical and survival tasks” is vague, improved compared to what? What specific outcomes and how much improvement?

We agree. We have rewritten the Abstract to state the specific comparators and effect sizes, and to minimize jargon. The abstract now explicitly stating that mzLearn is an MS¹ signal-detection/alignment method with no user-set parameters, adding concrete benchmarks vs XCMS and Asari across 15 datasets (feature counts and TP/FP rates) and noting drift correction without experimental QC. We specified the pretraining corpus (22 studies; 20,548 blood samples) and that, when fine-tuned on renal cell carcinoma, the model improves risk stratification and overall survival relative to a no-pretraining baseline and the IMDC risk score (effect sizes provided in the Results). We also highlighted interpretability via SHAP, which surfaced biologically plausible lipid/carnitine signals, and we moderated language to avoid overstatement while making all comparators explicit.

2. Moderate Claims. Terms like "first-in-class" and "democratizes" are overstated.

We thank the reviewer for pointing this out. We have tempered the language throughout. In the Abstract and Discussion, phrases such as “first-in-class” and “democratizes” have been removed or qualified to avoid overstatement.

3. Benchmarking against well-established tools. Instead of supplementary table 2 (which does not seem to be referenced in the manuscript) and numbers buried in text, consider a bar plot for a figure in the main manuscript to clearly support the claim “mzLearn outperforms existing methods in both signal quantity and quality”.

Thank you for the suggestion. We already include the requested **bar plot** in the main text (**Figure 2f**), which summarizes quality via true-positive and false-positive rates across methods (mean $\pm$ s.e.m.). We have made this more prominent by explicitly citing Fig. 2f in the Results and clarifying the caption (“benchmark vs. XCMS/Asari; TP/FP rates”).

For quantity, absolute feature counts differ by orders of magnitude across datasets (instruments, cohort sizes, RT windows). A single averaged bar would be misleading and would require

additional normalization choices. Instead, we report per-dataset counts in **Supplementary Table S2** and now reference it directly from the Results.

To facilitate independent checks, we also provide a **Supplementary Data** Excel workbook (.xlsx) containing the normalized sample and feature matrices (with detection frequency, median m/z, median RT) for all 15 benchmarking datasets—one sheet per study—so readers can recompute counts and quality metrics. These clarifications are added to the manuscript.

4. Include the reference of the papers corresponding to all the public datasets used. This is important to acknowledge the work of the original authors and appreciate that they made their data public, which fosters publishing data.

We agree. We added Supplementary Table S2 listing every public dataset used, with repository accession, source, and full citation

5. Improve description of validation approach. In the manuscript "targeted metabolomics features" are mentioned as ground truth but important details are missing. How were targeted features originally identified? What kind of matching and tolerance windows were applied? What makes them "true positives"?

Thank you, we have clarified the validation protocol.

- **Ground truth.** For benchmarking we used the targeted panels provided by each public study (as deposited to Metabolomics Workbench/MetaboLights), relying on the authors' reported m/z and retention time (RT). Dataset accessions are listed in **Supplementary Table 1**.
- **Matching.** After each method ran, we kept features present in $\geq 20\%$ of samples within a dataset, then counted a target as "detected" if any feature matched within $|\Delta m/z| \leq 0.005$ Da and $|\Delta RT| \leq 2\%$ of the dataset's total RT span. Exactly the same windows were applied to mzLearn, XCMS, and Asari.
- **True positives (TPs).** For a given method/dataset, a TP is a targeted metabolite whose repository m/z and RT are matched by a detected feature within those windows. Separately, we estimated false-positive (FP) rates by manual QC of 100 random detections per dataset (peak shape, SNR, alignment); this FP check is independent of the targeted list.

These details are now stated in Methods under **Benchmarking with targeted references**.

6. Approach seems to be limited to MS¹: The manuscript does not explain how the algorithm or pipeline considers MS/MS for metabolite identification or how signals are mapped to metabolite IDs. This represents a significant gap in the methodology description. How were metabolites identified from the untargeted signals? What databases or libraries were used?

Thank you for raising this point. In this manuscript, mzLearn is MS¹-only: it performs ion-level detection, alignment, and representation learning. We do not perform de-isotoping, adduct grouping, or MS/MS-based identification in the reported pipeline.

We agree that MS/MS identification is valuable, but high-quality MS¹ is already crucial for discovery: it provides the cohort-wide quantitative layer used for integration, hypothesis generation, and prioritization. By delivering a clean, reproducible MS¹ feature matrix without user tuning, mzLearn enables robust statistics and reliable feature ranking that guides downstream de-isotoping and MS/MS confirmation. Consistent with this, our SHAP analysis on the supervised fine-tuned models highlights biologically coherent ion features (e.g., plasmalogens/PUFA lyso-PLs for prognosis; acyl-carnitines/LPC for ICI response; shared C6 carnitine), showing that MS¹ alone can nominate signals worthy of validation.

Regarding molecule names in the text: these are not new IDs produced by mzLearn. They derive from (i) the targeted panels provided by the original studies (used as reference lists in our benchmarks) and/or (ii) previously reported markers that we overlay for context. No library search was run and no external MS/MS database was used for untargeted identification here. How identification would proceed (outside this study's scope): mzLearn outputs an ion-level table (apex m/z, RT, per-sample intensities). For datasets with MS/MS, these features can be linked downstream to fragmentation spectra by precursor m/z and RT proximity and then identified with standard tools/libraries; we do not prescribe a specific database since identification is beyond the scope of this work.

We have added explicit statements in the Methods/Discussion clarifying that our results are MS¹ features and that MS/MS identification is not performed in this study.

7. There seems to be a confusion between the signal detection algorithm and the generative model. As described, mzLearn is not generative. Despite the manuscript's framing, mzLearn appears to be a signal detection and processing algorithm rather than a trained generative model. While the authors describe an "iterative, data-driven process," this seems to refer to parameter optimization and threshold refinement rather than machine learning training in the traditional sense. The manuscript provides no description of training data, loss functions, or model architecture for mzLearn itself. This distinction is critical because it means mzLearn is a deterministic processing pipeline that adapts its parameters based on data characteristics, not a learned model that requires training on labeled examples or that can generate new synthetic data.

Thank you, this is an important clarification. We have revised the framing to separate mzLearn from the downstream generative model.

mzLearn is a deterministic, non-parametric algorithm that iteratively learns internal thresholds and drift maps from the data but does not train a statistical model (no learned weights and no loss function). The algorithmic details are described in Methods and Figure 2. data-adaptive MS¹ pipeline that tiles the RT–m/z plane, applies autocorrelation/cross-correlation screens, learns drift maps and thresholds, and outputs an ion-level feature table. It has no learned weights, no training data, and does not generate synthetic data.

The generative model in our study is a pre-trained VAE trained after mzLearn on the aligned, normalized MS¹ matrix aggregated across 22 studies. Training minimizes the reconstruction and KL divergence losses with mini-batch updates; the training data and architecture are described in Results/Methods. The VAE is the only learned model in this work.

We have adjusted language in manuscript to clarify mzLearn is a deterministic algorithm, that enables pre-trained generative modeling rather than being generative itself.

We appreciate the reviewer’s feedback, and we hope these revisions address the reviewer’s questions.

Response to Reviewers

We thank all reviewers for their time and constructive feedback. We are pleased that the revised manuscript was found clear, reproducible, and suitable for publication. In this round, we did **not** make additional changes to the manuscript.

Reviewer #1

Comment:

“The authors have adequately addressed all major and minor comments... The inclusion of parameter documentation, supplementary data matrices, and SHAP-based feature attribution particularly strengthens the work. Although a dedicated comparison of monoisotopic peaks could further validate the conclusions, the current justification and scope are reasonable. Overall... suitable for publication.”

Response:

Thank you. We appreciate the suggestion. A dedicated monoisotopic/isotopologue comparison is beyond the scope of this study and not necessary to support our conclusions; therefore, no changes were made.

Reviewer #2

Comment:

“I am happy with the responses... During the evaluation (Oct 12, 2025), I notice the website (<https://mzlearn.com/>) is not accessible (500 server error). I have no other comments.”

Response:

We Thank the reviewer for flagging this. The error was due to a temporary spike in concurrent traffic (capacity limit). We have increased server capacity to handle peak load, and the site is now accessible. No changes to the manuscript were required.

Reviewer #3

Comment:

“The authors have addressed this reviewer’s concerns and suggestions.”

Response:

We appreciate the concise and positive assessment.