Benchmarking a novel bio-informed RNN framework on Bulk RNA-seq data

Author: Christian Langridge

Supervisor: Dr. Cheng Zhang

LiDO rotational project 1

Word Count: 5000

Benchmarking a bio-informed RNN framework on Bulk RNA-seq data

Supervisor: Dr. Cheng Zhang

LiDO rotational project 1

Word Count: 5000

Abstract

Deep Learning (DL) models have historically struggled to compete with tree-based model performance on biological tabular datasets. A potential hypothesis for poor DL generalization is a mismatch between model design choices and the biologically relevant data characteristics of these resources. A novel custom recurrent neural network (RNN) architecture, LEMBAS-RNN, attempts to address this mismatch by integrating a transcription factor (TF) motif-based prior generated using TF-DNA mapping before employing a vanilla RNN to learn edge-node weights from tabular expression data to ultimately predict TF-target gene expression values. In this study, I conducted a holistic benchmarking of LEMBAS-RNN through the lenses of prediction accuracy, generalization, explainability and latency using linear and non-linear baseline models. Training fit and hold-out testing revealed that LEMBAS-RNN can identify mechanistic relationships but exhibits performance heterogeneity across prediction tasks and is largely outcompeted by tree-based modeling in prediction accuracy. Unseen data validation using generalization and latency metrics clearly show limitations in the LEMBAS-RNN architecture in handling unordered tabular data like bulk RNA-seq. SHAP values detailed interesting correlations between TF and predicted target values, highlighting differences between classical and DL methods. These findings illustrate this data-architecture mismatch likely persists and would suggest other ML strategies would be better suited for gene expression prediction tasks using tabular data. This research represents a systematic perspective on a current strategy for DL integration in biological research, consolidating past efforts while considering emerging prioritization of model explainability as essential for AI implementation in research pipelines.

Acknowledgements

I would like to thank Dr. Cheng Zhang for his guidance and help over this rotational project. I'm grateful for his patience and mentorship in my transition into bioinformatics. I enjoyed our research discussions very much.

I acknowledge my use of the conceptual, programming and unpublished works conducted by Kejun Li, including but not limited to scripting of the LEMBAS-RNN model.

I acknowledge the use of Perplexity (Perplexity AI, https://www.perplexity.ai/) to assist in reference searching. I acknowledge the use of Claude (Claude AI, https://www.claude.ai/) as script overview tool to challenge my design choices and assumptions during scripting.

After using these tools, the author reviewed and edited all content to take full responsibility for this work.

Table of Contents

Abbreviations

Abbreviation	Meaning
AI	Artificial Intelligence
ABI	Approximate Bayesian Inference
ANN	Artificial Neural Network
CV	Coefficient of Variation
DL	Deep-Learning
GAT	Graph Attention Network
GNN	Graph Neural Network
ML	Machine-Learning
MLP	Multi-layer Perceptron
LEMBAS	Large-scale knowledge EMBedded Artificial Signaling network
MLR	Multiple Linear Regression model
PCC	Pearson's Correlation Coefficient
RNA-seq	RNA sequencing
RNN	Recurrent Neural Network model
Sc-RNA-seq	Single cell RNA sequencing
SHAP	Shapley Additive exPlanations
SRCC	Spearman's Rank Correlation Coefficient
TF	Transcription Factor
XGBRF	XGBRegressor Random Forest model

1. Introduction

Artificial Neural Networks (ANN) offer unique advantages when processing biological datasets[1]. Robust pattern recognition and noise reduction capabilities make ANNs a promising addition to the bioinformatician's toolkit[2]. Tabular compatibility positions ANNs as effective on transcriptomics data, with early findings detailing ANNs accurately capture non-linear biological relationships during gene expression prediction tasks[1,3].

ANN models typically generate predictions by feeding forward input data using full-connected layers[4]. Through iterations across the training data, ANN architectures conclude on a final output by minimizing prediction error against a ground truth[5]. This optimization results in higher final accuracy, but the process of updating learned weights over the course of model training loses the relationships between datapoints[6]. This renders the model's decision path uninterpretable to developers, making the model training a black-box for human understanding[6].

Prioritizing performance over explainability is appropriate for many general AI applications, but this opaque decision-making obscures the reasoning behind how a model navigated data to reach its conclusion[7]. Black-box modeling contradicts how scientific research is practiced and leaves models unaccountable for the answers they generate[8]. Unlike predicting future financial or weather trends, predicting TF-target interaction is much more than a typical Deep-Learning (DL) forecasting task. Speculated relationships carry dependencies about the nature of real, mechanistic interactions, and these themselves hold biological significance about the how and why they have evolved in cells[9]. This lack of transparency is a major limitation of ANN architecture that lowers trust in AI among researchers[8]. Recent efforts now include explainability as a central consideration in novel DL design[10].

One strategy, initially designed by Li *et al*. (unpublished thesis), uses a Vanilla Recurrent Neural Network (RNN) that initializes over a Large-scale knowledge Embedded Artificial Signaling network (LEMBAS) to predict target expression patterns based on TF activity[11]. This model constructs a fixed network in latent space from TF-DNA motif mappings where transcription factors (TF) and target genes are nodes and regulatory relationships are edges, creating a biological topology interpretable for ML models (*Supplementary 3*)[11]. LEMBAS-RNN overlays expression data over the nodes to learn edge-specific regulatory weights using sequential linearity[11].

The network that LEMBAS-RNN is trained on is prior biological knowledge[11]. In this way, edge-specific weights represent TF-target gene regulatory relationships unlike the intermediate connections in classical ANNs[12]. Over training, learned weights become representative of the strength and direction, activation or inhibition, between TF and target gene[12]. The biological explainability of these outputs builds confidence in the architecture, while knowledge-graphing gives researchers a scalable tool to investigate mechanisms of transcriptional regulation[11]. Initial

findings show strong performance, although further development requires a systematic analysis against baseline ML methods[11].

Benchmarking LEMBAS-RNN on gene expression prediction tasks is a bleeding-edge focus in Bioinformatics[13]. Li *et al*. initially reported that aggregate performance metrics poorly capture model performance, particularly in highly heterogeneous datasets[11]. Testing on independent single cell RNA-seq data showed LEMBAS-RNN could partially identify cell-type target expression variation, but more granular evaluation methods would be needed to understand model performance on cell-type specific prediction[11]. Moreover, it is still unclear how well LEMBAS-RNN can generalize over other kinds of datasets (smaller sample sizes, Perturb-seq datasets, etc.)[11]. To address these free-ended questions and expand on how generalizable LEMBAS-RNN is, a systematic analysis against classic baselines is crucial. Classical ML models, often more familiar to bioinformaticians, serve as baselines to determine the added value of more complex DL architectures[13]. This helps assess whether their integration into usual data processing practice is worthwhile[3]. Going beyond ranking their performance, benchmarking and understanding how the trained DL model navigates data when compared current methods allow researchers to weigh up implementation cost.

Here, I explore the LEMBAS-RNN model functioning on TF-target association gene expression tasks relative to a formal evaluation of how DL advances gene expression prediction modeling. Firstly, I aimed to establish a benchmarking configuration to quantify LEMBAS-RNN alongside Multiple Linear Regression (MLR) and Extreme Gradient Boosted Random Forest (XGBRF) models using traditional performance metrics. Next, I aimed to test the data requirements for LEMBAS-RNN's ability to generalize robustly without overfitting using an independent validation dataset. Lastly, I aimed to quantify the LEMBAS-RNN for future biological modeling using model-agnostic prediction explainability techniques for research applicability. This work addresses a critical gap in performance evaluation: whether LEMBAS-RNN's improved explainability justifies its implementation for TF-target prediction in transcriptomics pipelines.

2. Methods

2.1 Experimental Set-Up

All scripts were written using Python (version 3.12.12) and all data processing, baseline model initialization, training and evaluation was conducted on a remote CPU cluster server hosted by the Zhang Lab (AMD Ryzen Threadripper PRO 5995WX 64-Cores).

See direction for LEMBAS-RNN hyperparameter specifications in Supplementary 3.

2.2 Dataset selection and pre-processing

Bulk RNA sequencing data (RNA-seq) from human liver samples was used for both model training and testing, sourced from the ARCHS4 project. An external human liver bulk RNA-seq dataset was used for model validation, sourced from Yang *et al* [14].

For all model analysis, TF expression values (x) and target gene expression (y) values were extracted from both RNA-seq datasets as two separate subsets, input features (x_train, x_test, x_validation) and target labels (y_train, y_test, y_validation).

Bulk raw expression counts were converted into Transcripts per Million (TPM) before $\log_{10}$ fold transformation was applied for training and testing. Gene features were overlaid onto the network to ensure only genes with matching node representations were included into training and testing sets.

The external bulk RNA-seq dataset was already prepared via TPM and $\log_{10}$ fold transformation like the first dataset, so the network mapping was repeated to ensure only matching node represented genes were kept. Gene features missing from the external dataset but included in the knowledge graph were included as 0 values for testing consistency.

2.3 Network generation

See direction network generation pipeline in Supplementary 3.

2.4 Selection of Evaluation Metrics

Coefficient of Determination (R^2), Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) were used as metrics of model goodness-of-fit for all 3 architectures to evaluate model training fit, test set prediction accuracy and generalization ability on an independent dataset.

Coefficient of Variation (CV) was used to group individual gene expression prediction tasks into 3 tertile bins (low-variance, medium-variance, high-variance) by relative dispersion around the mean along 33rd and 67th percentiles.

Pearson's Coefficient of Determination (r) (PCC) and Spearman's Rank Correlation Coefficient (ρ) (SRCC) were both used to evaluate the correlation between predictions and ground truth values. Because of Pearson's R optimistic assumption of normal distributed data, unsurety about the nature of the relationship between predicted and ground truth and increased overfitting risk with large datasets, Spearman's R was also included. All 95% confidence intervals were generated with Fisher's Z Transformation given the scale of the datasets to ensure computational reproducibility.

To better evaluate models as holistic tools for Bioinformatics, we also included inference latency to better understand the computational speed and scalability of each model. Shapley Additive exPlanations (SHAP) were also used as model-agnostic method of tracking and comparing feature contributions through each model's predictions.

2.5 Pilot Data handling and basic modeling

Initial model selection and data preprocessing workflows were tested through exploratory analysis in ARCHS4 liver datasets using standard differential expression workflows.

2.6 Selection of Baseline Models

MLR was chosen as a very simple and interpretable model tasked with fitting a linear relationship between a multiple dependent and independent variables, described by Eq. (1):

$$Y = AX + E$$

where Y represents the target gene expression value response matrix, A represents the slope coefficient matrix, X is the TF expression value input matrix and E are the error matrix. Given the high-dimensionality of RNA-seq datasets, MLR was chosen to act as the simplest possible solution to the data and provide a true baseline and performance floor to justify the need for more complex modelling architecture.

XGBRF was chosen as a robust, ensemble method model suited to capturing non-linearity within a tabular dataset. XGBRF is a variant of the original XGBoost library that trains a standalone random forest instead of a sequence of boosted trees like other XGBoost variants. Here, XGBRF relies on parallelized tree estimator models that each predict an output from the input data. These individual sub-outputs are averaged together to give a finalized, model output. XGBRF combines the performance robustness and resistance to overfitting of standard RF with the high accuracy and training efficiency offered by the gradient boosted network of XGBoost[15]. XGBRF training can be expressed as the minimization of a core objective function, defined by the Eq. (2):

$$L(\phi) = \sum_{i} l(\, y_{i'}, y_i) + \sum_{k} \Omega(f_k)$$

where $L(\phi)$ is the total model cost that the algorithm is minimizing over training, $l(\, y_{i'}, y_i)$ is the training loss function, calculating the difference between predicted label $y_{i'}$ and ground truth label y_i and $\Omega(f_k)$ defines the regularization term that penalizes model complexity to limit overfitting.

XGBRF uses the same Gain splitting formula as regular XGBoost, defining the reduction of the training loss function when a node is split in two. This function coordinates the algorithm "branching", and the final tree structure of the trained model. The Gain formula is represented by Eq. (3):

$$Gain = \frac{1}{2}\left[\frac{G^2{}_L}{H_L + \lambda} + \frac{G^2{}_R}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}\right] - \gamma$$

Where $Gain$ is the relative reduction of the node split on total loss value of model training , G the sum of residuals that indicates room for improvement at a given node, H, the Hessian or the sample weight that scales model gradients to prevent overreaction to outliers, λ the L2 regularization factor that manages model complexity to prevent overfitting and γ the complexity cost regularization factor that defines the threshold for a given node's ability to split. As a hybrid model, XGBRF serves as intermediary model between MLR and LEMBAS-RNN that is well-suited for tabular dataset, capable of effectively handling noise while capturing non-linearity within the data.

2.7 Validation Strategy

Holdout validation (80% training and 20% testing with random seed 888) was performed on training data before model initialization. An independent liver bulk RNA-seq dataset to externally validate model generalization. True latency (time needed to process one instance for all input features) and batch throughput (time needed to process all instances of all input features) were inferenced for each model with 20 warm-up runs before measurements begin to minimize the influence of start-up computations are taken, and 100 inference loop runs to minimize the effects of noisy CPU load. Feature contributions involved in albumin (*ALB*) and alpha-fetoprotein (*AFP*) predicted expression, two pleiotropic liver-specific genes in humans, were dissected using SHAP to further explain how models were navigating unseen data.

3. Results

Model fitting on Training set

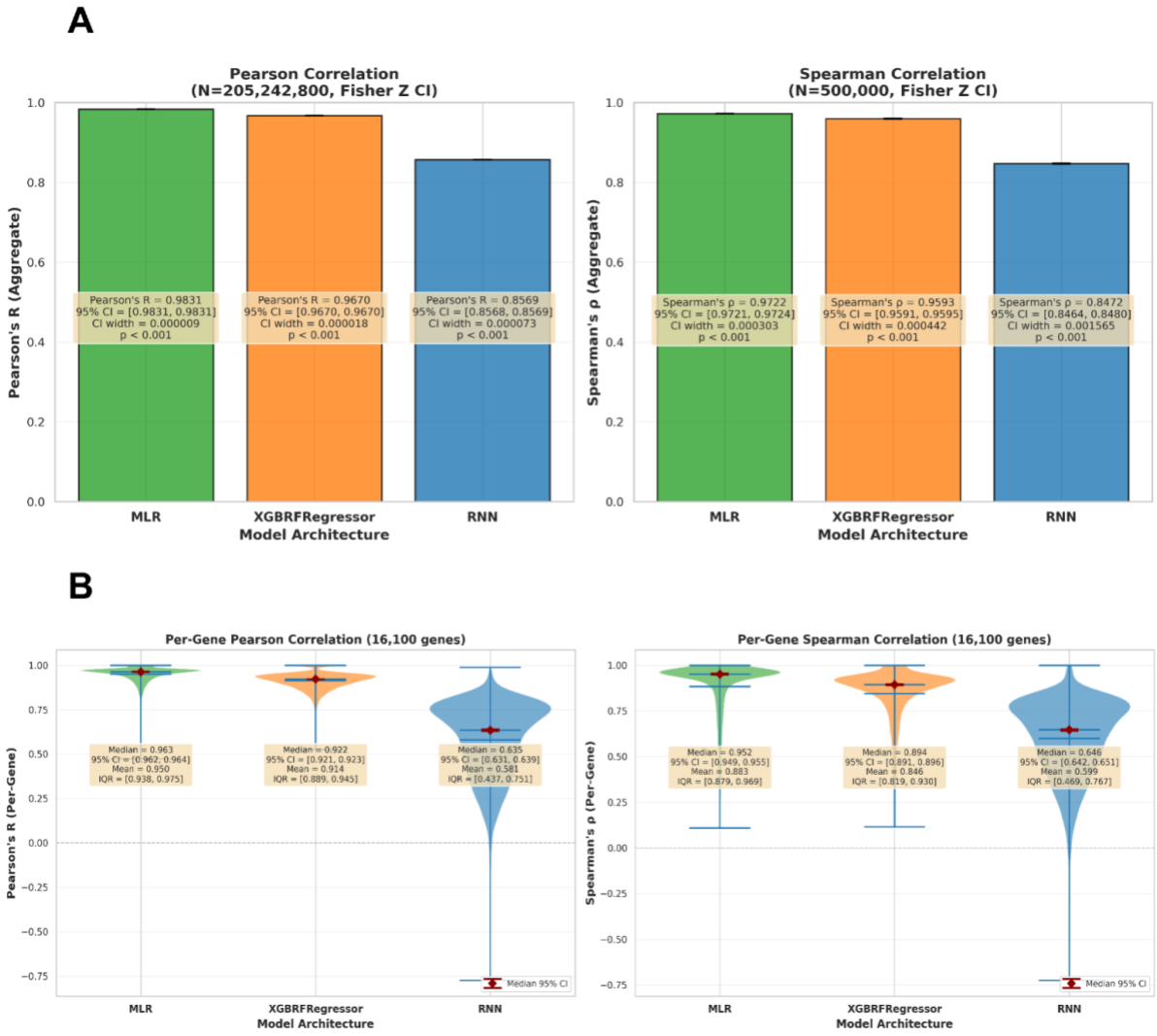

Figure 1. Model fitting on training set with 95% confidence intervals generated with Fisher's Z transformation. **A)** Histogram comparison of aggregate PCC and SRCC across a flattened representation of all prediction tasks. **B)** Violin plot comparison of per-gene PCC and SRCC distributions for each model.

Performance on the training set was evaluated using Pearson's Correlation Coefficient (PCC) and Spearman's Rank Correlation Coefficient (SRCC) with 95% confidence internals generated via Fisher's Z Transformation and results displayed in (***Fig 1***). All models presented strong fitting to

the data at the aggregate level (***Fig 1.A***), although LEMBAS-RNN presented the lowest of the three (RNN: [PCC = 0.8569, SRCC = 0.8472]). Looking further at the distribution of PCC and SRCC values respectively for each prediction task (***Fig 1.B***), MLR and XGBRF models presented comparable performance distributions with similar median performance (MLR: [PCC = 0.963, SRCC = 0.952]) (XGBRF: [PCC = 0.922, SRCC = 0.894]) and interquartile ranges. Conversely, LEMBAS-RNN performance is more heterogeneous at the granular level, with both a lower median performance (RNN: [PCC = 0.635, SRCC = 0.646]), wider interquartile range and long negative correlation tail.

The disjunction between LEMBAS-RNN respectable aggregate performance and heterogeneous per-gene performance likely because the inductive bias and training process of its architecture are worse suited for capturing the regression in the training set. Compared to MLR and XGBRF, which both look for gene expression linearity and output stable performances from unordered features, LEMBAS-RNN introduces an iterative message-passing style computation constrained over the latent network. This inductive bias facilitates mechanistic interpretability but imposes a rigid convergence threshold that misaligns with the biological noise of bulk-RNA-seq.

This results in a mismatch between the characteristics of the tabular RNA-seq dataset and the data expectations that the RNN is making. This underlying per-gene instability flags limitations to LEMBAS-RNN's convergence logic, highlighting a conflict between training stability and stochasticity in the biological data. Furthermore, I moved over to evaluating model testing performance (***Fig 2***).

Model testing on Held-Out set

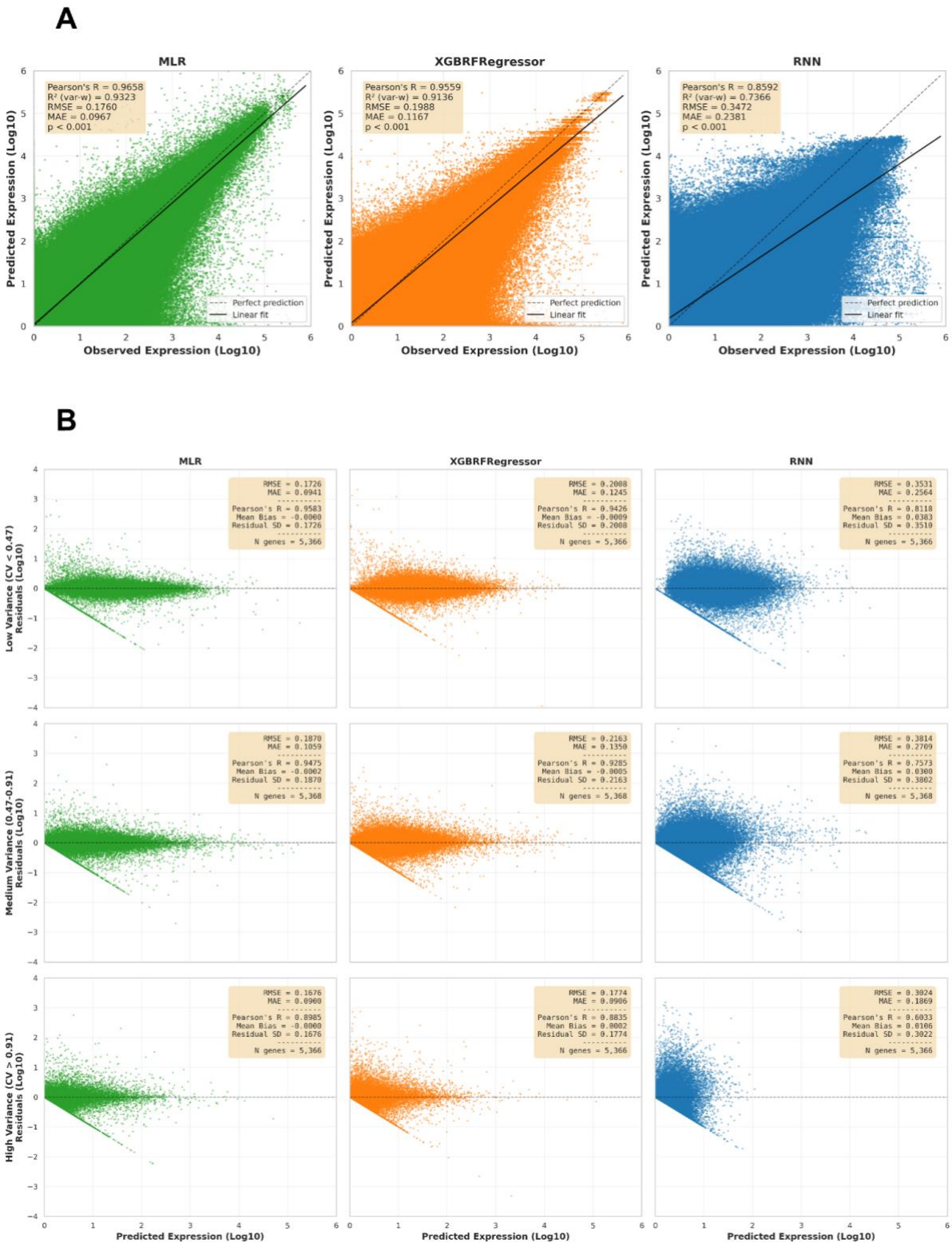

Figure 2. Model testing on testing set. **A)** Scatter plot comparison of $\log_{10}$ observed vs. $\log_{10}$ predicted expression for each model with aggregate metrics. **B)** Scatter plot comparison of $\log_{10}$ predicted vs. $\log_{10}$ residuals for 3 CV tertile binned expression prediction pools (low-variance, medium-variance, high-variance) along 33rd and 67th percentiles.

Following the test-train split, trained models were evaluated on the held-out testing data using PCC, variance-weighted R^2, RMSE and MAE. All models presented strong testing performance with PCC and R^2 (**Fig 2.A**). MLR performed the highest overall [PCC = 0.9658; R^2 = 0.9323], followed by XGBRF [PCC = 0.9559; R^2 = 0.9136]. LEMBAS-RNN demonstrated the lowest global performance [PCC = 0.8592; R^2 = 0.7366], but still strongly captures the linear relationship with observed values. Despite these results, RMSE and MAE across all models, MLR [RMSE: 0.1760; MAE: 0.0967], XGBRF [RMSE: 0.1988; MAE: 0.1167] and RNN [RMSE: 0.3472; MAE: 0.2381], point to the existence of substantial outliers and general error across all prediction tasks. LEMBAS-RNN presented the highest RMSE:MAE ratio (~1.46), which would indicate that the error distribution has a heavier tail with likely more outliers compared to baseline models.

Increasing the dimensions of the plot reveals critical differences between models on predicted values for low-expression genes, where $\log_{10}$ expression becomes close to zero (***Supplementary Figure 1***). XGBRF grouped all datapoints within the [0, ∞] space. All models were trained on $\log_{10}$ expression values which are negative for inputs below 1, making this behavior interesting. In this large sparse-adjacency matrix, the minimum expression value is zero and these make up most of the values in the test set. As a bagged-ensemble method, XGBRF is non-extrapolative and incapable of values lesser than the minimum observed value, allowing the tree-based model to be more "biologically compliant". XGBRF reduces variance by outputting an averaged final prediction across all leaf nodes, so when most of them are zero, the final average is likely to be zero as well. This clipping mechanism would explain the hard floor observed in (***Supplementary Fig. 1***). In contrast, LEMBAS-RNN failed to constrain datapoints to (0,0) boundary even with L1/L2 regularization. Observing the dense clustering of predictions about the origin, LEMBAS-RNN is trying to respect the same hard floor as XGBRF but struggles to fit as the model relies on a continuous function. This results in a noisier boundary compared to XGBRF but much tighter than MLR. MLR, lacking regularization and pruning, was obligated to minimize the sum of square residuals, making 'illegal' predictions into the negative which are not biologically plausible. This achieves a high PCC but heightens RMSE as these outliers are heavily penalized.

All models presented vertical streaking along the zero-observed expression axis, which suggests aleatory uncertainty where technical noise likely disrupt the biological signal of measured value more than for higher variance prediction tasks. To investigate this, all prediction tasks were evenly binned using CV tertiles into low-variance, medium-variance and high-variance groupings along 33rd and 67th percentiles, before being evaluated using PCC, RMSE and MAE (***Fig 2.B***). Despite high PCC scores for all models in all binned tasks, the data dispersion across models and bins indicated strong performance heterogeneity and overall error as indicated by large RMSE:MAE

ratios. MLR and XGBRF presented similar performance and error metric consistency, whereas LEMBAS-RNN displayed both lower performance scores and more error across all binned tasks. MLR and XGBRF also presented more similar cone shaped data patterns across tasks bins, which differed significantly from the diffuse cloud patterns exhibited by LEMBAS-RNN.

All models presented a diagonal lower-bound artefact across task bins as expression trends towards zero, which likely represented model over-prediction on low-expression predictions. Of the 3 bins, all models struggled most on high-variance expression predictions, in particular LEMBAS-RNN that presented a performance collapse transitioning from medium to high-variance predictions (medium-variance PCC = 0.7573; high-variance PCC = 0.6033). LEMBAS-RNN presents a consistent regression-to-the mean for high-variance target predictions compared to the other task bins. As the network optimizes overall loss through training, this would suggest easier prediction tasks (high-expression genes with low-variance potentially associated with housekeeping functions) are prioritized over more complex predictions (low-expression genes with high-variance potentially associated with differential expression), explaining the clear performance heterogeneity.

These testing results showcase the interplay between precision and reliability for model performance. Strong performance scores across models overall masked significant underlying error, which were further confirmed by breaking down prediction tasks and visualizing datapoint dispersion. Generally, high $\log_{10}$ residuals suggest that models capture the general scope of biological expression variability but lacked the resolution required for more individual gene inference (***Fig 2.B***).

Model validation on unseen external data

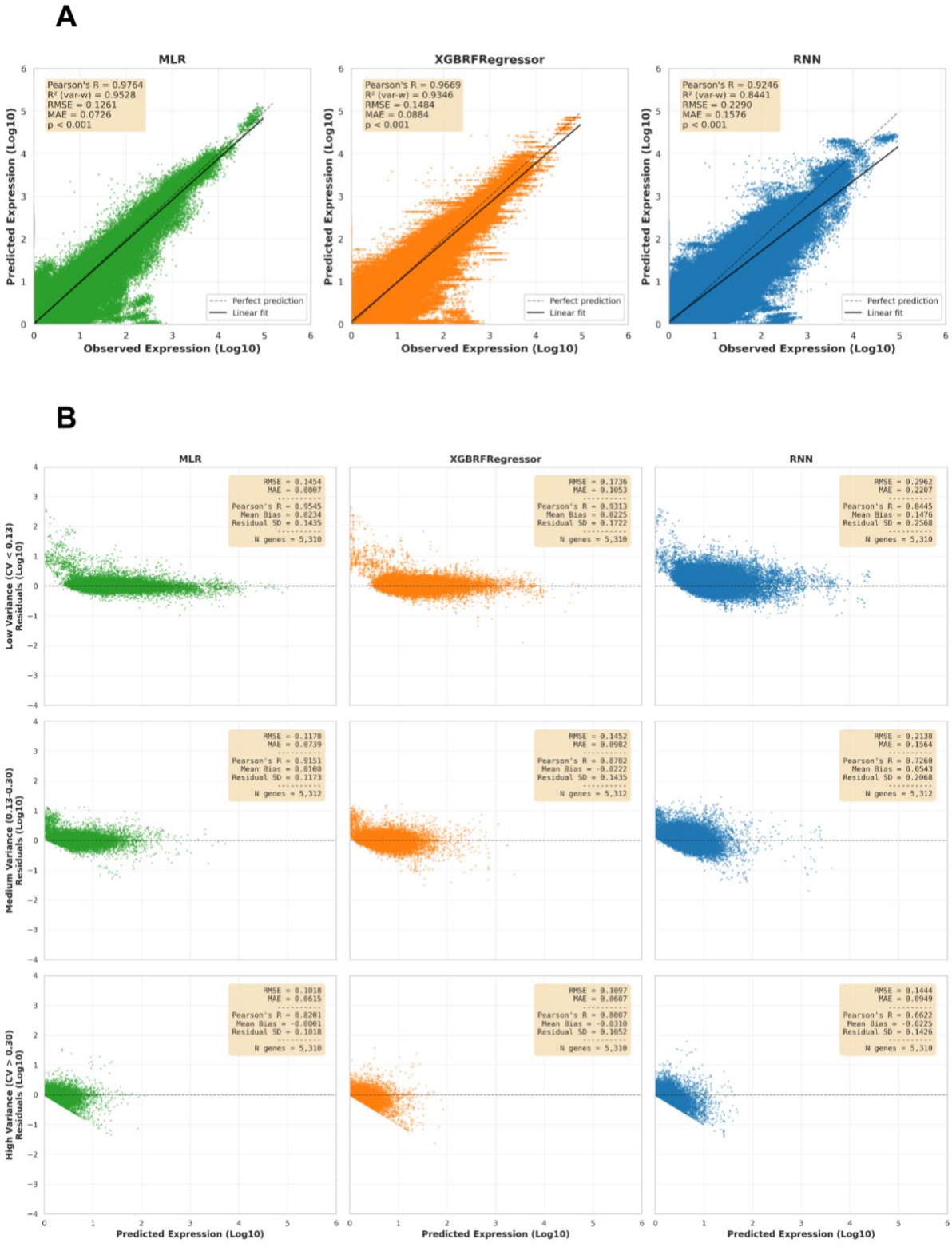

Figure 3. Model validation of an unseen bulk RNA-seq data. Evaluation strategy from testing was repeated. **A)** Scatter plot comparison of $\log_{10}$ observed vs. $\log_{10}$ predicted expression for each model with aggregate metrics. **B)** Scatter plot comparison of $\log_{10}$ predicted vs. $\log_{10}$ residuals for 3 even CV tertile binned expression prediction pools (low-variance, medium-variance, high-variance) along 33^{rd} and 67^{th} percentiles.

Trained models were evaluated on an unseen validation set using the same metrics as held-out data testing (***Fig 3***). All models presented higher aggregate performance scores and lower error scores **(*Fig 3.A*)**. LEMBAS-RNN saw a significant boost in performance (RNN: [PCC = 0.9246, R^2 = 0.8441]), making it more competitive with baseline models here than with testing data. MLR and XGBRF presented smaller error scores and smaller RMSE:MAE ratios compared to performance on testing data. LEMBAS-RNN also had smaller magnitude error scores but a similar RMSE:MAE ratio to that of the testing data (~1.45).

Reduced RMSE:MAE ratio in the unseen data suggests fewer outliers compared to the testing set, implying reduced aleatoric noise in the validation set. The fact that LEMBAS-RNN presents a consistent ratio, but smaller error magnitude would support the hypothesis that even on cleaner data, model performance retains the same proportion of outliers. Looking across binned tasks, LEMBAS-RNN presents the shaped pattern and performance collapse as with the testing set (***Fig 3.B***). On low-variance binned expression predictions, LEMBAS-RNN displays a significant positive mean bias unseen in baseline models, which shifts considerably compared to the high-variance binned tasks (RNN: [low-variance mean bias = 0.1476, high-variance mean bias = -0.0225].

This overprediction of low-variance genes and under-prediction of high-variance genes represents a systematic limitation of the LEMBAS-RNN. The model's significant inductive bias as first observed in Figure 1.B makes the model unsuccessful in capturing non-linear relationships and variability from biological expression data. Considering the RMSE:MAE ratio stability, this case confirms the error distribution of the model is due to an architectural mismatch between LEMBAS-RNN and RNA-seq data.

Model latency benchmarking on unseen external data

Model	Latency (ms / sample)	Throughput (samples / sec)
MLR	4.801	587.8
XGBRF	4564.857	40.5
LEMBAS-RNN	9415.526	7.8

Table 1. Latency and throughput metrics for each of the model architectures.

A latency test was conducted to understand how computationally efficient each model. Mean latency, the time needed to produce a single prediction and throughput, how many predictions can be generated in one second of server time, were both calculated.

MLR presented the lowest values for both latency and throughput, which was expected given the model compute a single dot product with no computational overhead and can scale linearly with available CPUs. XGBRF and LEMBAS-RNN both showed much higher computational costs. The trained XGBRF model loops through 17 sub-models for each instance to generate a prediction, meaning latency becomes the sum of all iterations. Tree-based models also are reputed for being more intensive than simpler matrix math showcased in MLR[16]. LEMBAS-RNN is the most computationally costly model by far. LEMBAS-RNN's central signaling network block needs iterate through data repeatedly to settle on a final prediction, severely increasing wait times.

Another consideration is the latency test capturing the computational time of each model, but also data configuration cost specific to the architectural decisions. XGBRF is looping over a list of models in a Python loop, which needs to be interpreted back into the native C++ language of the XGBoost Library[17]. This creates computational overhead that likely adds considerable processing time to each prediction[17].Similarly, LEMBAS-RNN is slowed down by overhead. The *Tolerance* training hyperparameter is set very low (1e-20) to maximize model precision, which forces the signaling network's logic to continue iterating long after the model's representation of a biological relationship has converged, increasing latency (*see 6. Code and Data availability*) [18]. Constrained by RNN's logic, LEMBAS-RNN performs matrix multiplication, an activation function and a state update one instance at a time, unable to rely on CPU parallelizing strategies to recover lost efficiency[18].

These results showcase how increasing model complexity comes with increased computational cost and implementation overhead that. This serves as a primary consideration for DL implementation observed in current deployments into research.

Model feature contribution using SHAP on unseen external data

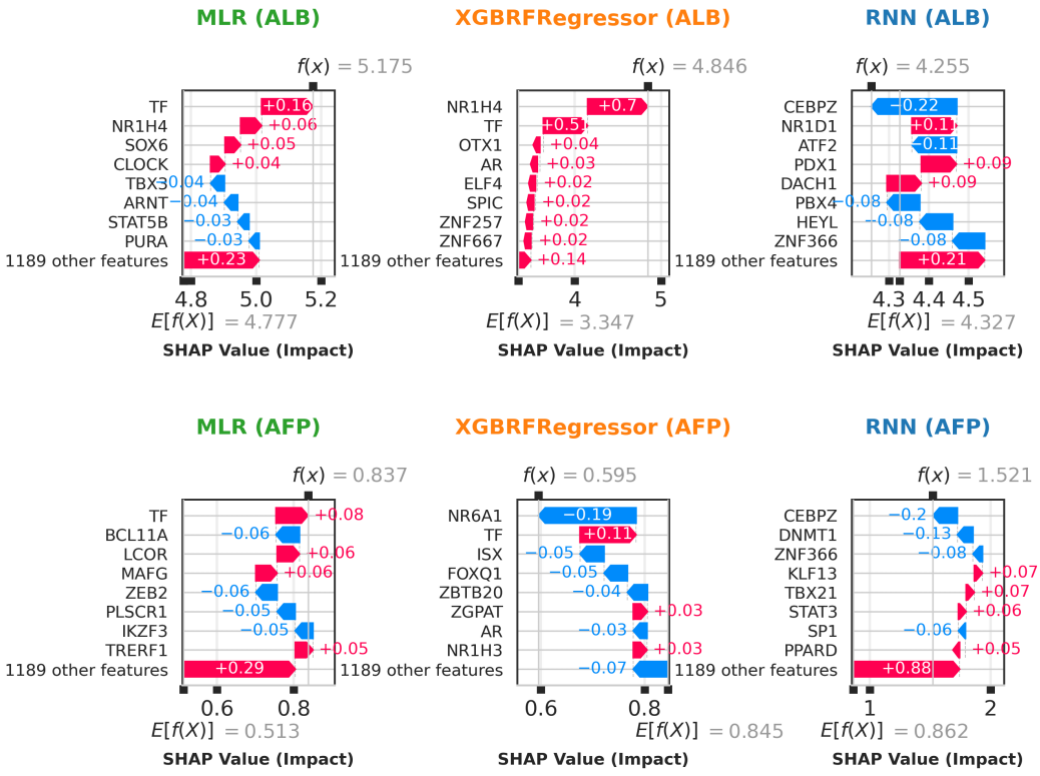

Figure 5. SHAP waterfall plot of feature contributions for liver-specific *ALB* and *AFP* expression value predictions on unseen data using model-agnostic SHAP explainability.

Shapley values for *ALB* and *AFP* singular predictions across MLR, XGBRF and LEMBAS-RNN were generated with LinearExplainer, TreeExplainer, GradientExplainer classes from the SHAP library. $E[f(x)]$ represents the global average $\log_{10}$ predicted expression value for that specific gene across all samples before looking at any specific TF features in the unseen dataset. $f(x)$ represents the final $\log_{10}$ predicted expression value after all TFs have influenced the prediction. The delta between these two values is statistical net push applied by all TFs.

This pushing mechanism is leveraged differently by all three architectures. While $E[f(x)]$ and $f(x)$ values and their deltas show no obvious trends across architecture, the ten most impactful feature contributions present interesting patterns that partition classical ML versus novel DL strategies. For *ALB*, MLR and XGBRF both captured transcription factors *TF* and *NR1H4* having front-running positive associations with prediction expression, which are missing from the top 10 for LEMBAS-RNN. Instead, CEBPZ leads and shows significant negative association in the final LEMBAS-RNN prediction. For *AFP*, the leading TFs contributions were scattered for MLR and XGBRF, whereas LEMBAS-RNN still had CEBPZ as the most impactful singular TF.

These results indicate that while models may have similar predictive performance, architectures often navigate data in very different ways. Classical ML models seem to identity more similar individual features compared to DL approaches, with LEMBAS-RNN consistently identifying *CEBPZ* as important for both final predictions.

4. Discussion

SHAP feature contributions

While SHAP is a powerful post-hoc tool in discerning how models work, it is limited in reflecting the logic of the models more than biological ground truth from the data itself. Using SHAP waterfall plots, a TF can positively or negatively push the prediction towards its final value, turning statistical association into biological correlation, which can inform hypotheses about co-regulatory mechanisms for target genes.

The strong negative weighting CEBPZ identified by LEMBAS-RNN is interesting given its identity within the CCAAT/enhancer-binding protein (C/EBP) family and its role as a strong up regulator of undifferentiated fate in myeloid cells as demonstrated by Barbieri et al[19]. As ALB and AFP are both well-characterized markers for hepatocyte differentiation in the liver, LEMBAS-RNN may have identified CEPBZ as high-impact feature in the transition from progenitor to hepatocyte cell fate[19,20]. CEBPZ might in competition for binding sites and clashing with similar TFs like C/EBP-α, which are strongly associated with hepatocyte differentiation and upregulation of ALB and AFP during liver development[20]. Conversely, MLR will only see linear trends within the data, so any subsequent SHAP values will also be biased and only pick up on strong correlations. Defining Transferrin (TF) as its most impactful TF is justified as this targeted transcriptional regulator is highly expression in human liver, but rather as a driver of hepatocyte functionality through iron homeostasis than an upregulator of ALB or AFP directly[21]. This presents MLR as accurately capturing statistical coincidence more so than actual biology. The identification of CEBPZ by LEMBAS-RNN as a mechanistic inhibitor identified by simpler correlation methods illustrates that despite lower accuracy, DL models offer interesting biological insights for research discovery when used alongside robust explainability tools. This interestingly suggests that while tree-based models are “better” on MSE and statistical metrics, LEMBAS-RNN may be “better” when considering biological discovery by identifying subtle relationships overlooked by classical models.

However, SHAP is biased by model-specific noise. For AFP expression prediction, LEMBAS-RNN identified the remaining 1889 features as hugely influential whereas XGBRF completely overlooked this trend and showcased higher contribution magnitudes among its top 10 ranked features. If another trio of models were introduced with slightly different training runs on the same unseen data, the ranked most impactful features would likely have been different. This disjunction points to the explainability being more a product of algorithmic design than biological insight from the data. SHAP findings showcase how this technique cannot capture biological ground truth but instead model logic. Rather than a clear limitation, using SHAP with different model designs in an “ensemble filtering” method could prove useful for isolating higher confidence features that can be further validated.

Data-architecture mismatch with RNN

The data-LEMBAS-RNN architecture mismatch is interesting and points to foundational expectations that RNN architectures have of training data. The ability to discern biological meaning from tabular data of is not inherent in LEMBAS-RNN, rather a facet of the fixed knowledge graphs the DL architecture sits on. This prior strategy, first popularized in 2020, provides biologically relevant railings that assist in distinguishing between real values and error[22].

While RNN seem to not be the best choice as the engine behind this framework, replacing it with a different DL architecture might prove more effective with some additional benefits. Graph DL architectures, particularly Graph Neural Networks (GNNs) and their subgroup Graph Attention Networks (GATs) might have better suitability for this task, capable of handling heterogenous representation whilst offering better prediction explainability.

Graph Variational Autoencoders like scGAC combine the probabilistic strength of variational autoencoders, embedding cell expression information in latent space, before learning the representation of single-cell expression profiles and relationships between them with a graph attentional autoencoder[23]. Using attention mechanisms would also have the added benefit of leaving a mathematic link between input data and embedding via attention weights, a form of basic explainability that can be plotted to look at correlations between nodes (genes, cells, other biological units, etc.) that share a latent space[24].

Heterogenous GNNs like scHetG work differently, allowing connected nodes to represent cell, genes, proteins and even pathways in latent space[25]. By pairing a knowledge graph with scHetG, a model could reconstruct the target gene expression matrix, distinguishing cell and gene embeddings whilst respecting the structural information from biological priors[25].

This transition from sequential linearity to biological topology would align more natively with the "network" structure of transcriptomics data. Pairing anchored relationships with flexible node representations and ad-hoc explainability mechanism would provide a robust engine for discovery across biological scales.

Model applicability on bulk-RNA-seq vs sc.RNA-seq

All models were trained on bulk-RNA-seq tabular datasets, but their broader applicability to novel tabular biological datasets like single-cell RNA-seq (sc-RNA-seq) is still underexplored.

Preliminary LEMBAS-RNN testing by Li *et al.* pointed to reduced model validation performance on sc-RNA-seq, but this training run never included this datatype and no further diagnostic analysis was conducted[11]. In this benchmarking assessment, only bulk-RNA-seq was considered for training-test and validation datasets.

Bulk-RNAseq and sc-RNA-seq differ fundamentally in the biological scale they represent. Unlike bulk-RNAseq, sc-RNA-seq takes in individual cells as instances not samples. Where bulk-RNA-seq is representative of the averaged expression profile of a sample, sc-RNA-seq represents a granular snapshot of high variable gene expression (HVGs) for a cell in a specific tissue, with a specific fate at a specific metabolic stage. These datasets prioritize richer biological relationships and cell variability but amplify noise from minor distraction to unavoidable challenge of data processing.

LEMBAS-RNN already showed performance collapse on heterogeneous sampled gene expression predictions in bulk-RNA-seq, so it is unlikely that the model could handle nosier data. Moreover, bulk-RNA-seq are not as zero-inflated as what would be expected from sc-RNA-seq where technical dropouts become more significant, potentially representing up to 90% to the total matrix[26]. This blurs the meaning of zero expression values, creating uncertainty about whether they represent truly functionally-off genes or failures of data acquisition[27].

LEMBAS-RNN is based on a prior knowledge network, which does smooth over some uncertainty, but this is not without limitations. Using these networks biases models on known biology, making discovery tasks more difficult[28]. Furthermore, if LEMBAS-RNN encounters a situation where both the node and its neighborhood of connecting nodes are zero-inflated, then a whole patch of the dataset is obscured from model training[28].

These limitations expose LEMBAS-RNN a preliminary step towards the development of models that natively handle sc-RNA-seq data, a necessary step in the effort to integrate different transcriptomics datasets together for richer ML interpretation across biological resolutions.

Tree-based models vs NN models on tabular data

Across model fit, testing and validation, XGBRF can back consistently with the highest prediction accuracy and model stability of the three architectures. This raises an important debate about whether DL frameworks are solutions worthwhile for tasks on tabular datasets like bulk-RNA-seq when tree-based models are performing.

Within the past four years, Grinsztajn *et al* showed that both bagging and boosting tree models excelled on tabular data and beat popular DL strategies of the time like multi-layer perceptrons

(MLP)[29]. Tabular data commonly contains many uninformative features, which have been shown to negatively impact MLP performance and widen the gap with tree-based models when kept[29]. MLPs are invariant, meaning they get "distracted" by filler data, requiring more computation and data to make a confident prediction[29]. Tree-based models operate differently, splitting data using decision thresholds that respect the dataset's original format while filtering uninformative features[29]. XGBRF's branching logic relies on both gain-based pruning and aggressive regularization (λ and γ), allowing the model to respect the format of the data while silencing irrelevant TFs for a given prediction. This evidence supports tree-based models generating strong predictions with much less computational cost, disparaging the need for more complex architectures[29].

Developments in DL and new architecture have reopened this debate. The advent of tabular foundation models like TabPFN show promising generalization ability on smaller tabular cohort[30]. TabPFN handles informative features and data heterogeneity typical of tabular data (missing features, categorical variables, etc.) natively through a pre-training step on synthetic data, generating a prior of what relationships should look like on real data[31]. Combining both row-wise and feature-wise attention mechanisms, TabPFN runs a single forward pass over the whole dataset using Approximate Bayesian Inference (ABI), learning how both samples and columns relate to each other[30]. This produces a permutation invariant model like an MLP but is capable can parse data for feature usefulness like a tree-based model[32].

This juxtaposition of approaches illustrates the speed at which AI research is evolving. XGBRF endures as a utile and reliable model for many biological contexts, however the evolution of new DL architectures points to the horizon of highly specialized architectures that perform more accuracy while handling both the format and heterogeneity of biological data.

In summary, I established a benchmarking strategy for LEMBAS-RNN using meaningful benchmarks and holistic performance consideration. In agreement with previous benchmarking assessments, LEMBAS-RNN as a current DL method is unable to compete with tree-based models on biological regression tasks on tabular data. In a burgeoning field, AI in biological research requires the development of robust ante-hoc methods for explainable predictions. Models like LEMBAS-RNN show potential in building trust in situational fields like research were getting an understanding of how a model makes a prediction is essential to meet safety regulations and scientific rigor.

5. Bibliography

1. Dradjat K, Hamidi M, Bartet P, Hanczar B. Self-supervised representation learning on gene expression data. Cantini L, editor. Bioinformatics. 2025 Nov 1;41(11):btaf533.

2. Eraslan G, Simon LM, Mircea M, Mueller NS, Theis FJ. Single-cell RNA-seq denoising using a deep count autoencoder. Nat Commun. 2019 Jan 23;10(1):390.

3. Yi C, Cheng J, Chen J, Liu W, Liu J, Li Y. Benchmarking deep learning methods for biologically conserved single-cell integration. Genome Biol. 2025 Nov 20;26(1):398.

4. Schmidhuber J. Deep learning in neural networks: An overview. Neural Netw. 2015 Jan;61:85–117.

5. Razavi S. Deep learning, explained: Fundamentals, explainability, and bridgeability to process-based modelling. Environ Model Softw. 2021 Oct;144:105159.

6. Garouani M, Mothe J, Barhrhouj A, Aligon J. Investigating the Duality of Interpretability and Explainability in Machine Learning. In: 2024 IEEE 36th International Conference on Tools with Artificial Intelligence (ICTAI) [Internet]. 2024 [cited 2026 Jan 19]. p. 861–7. Available from: http://arxiv.org/abs/2503.21356

7. Rudin C, Radin J. Why Are We Using Black Box Models in AI When We Don't Need To? A Lesson From An Explainable AI Competition. Harv Data Sci Rev [Internet]. 2019 Nov 1 [cited 2026 Jan 16];1(2). Available from: https://hdsr.mitpress.mit.edu/pub/f9kuryi8

8. Durán JM, Jongsma KR. Who is afraid of black box algorithms? On the epistemological and ethical basis of trust in medical AI. J Med Ethics. 2021 Mar 18;medethics-2020-106820.

9. Erbe R, Gore J, Gemmill K, Gaykalova DA, Fertig EJ. The use of machine learning to discover regulatory networks controlling biological systems. Mol Cell. 2022 Jan;82(2):260–73.

10. Van Hilten A, Katz S, Saccenti E, Niessen WJ, Roshchupkin GV. Designing interpretable deep learning applications for functional genomics: a quantitative analysis. Brief Bioinform. 2024 Jul 25;25(5):bbae449.

11. Li K. MODELING TRANSCRIPTION FACTOR REGULATION USING DEEP LEARNING ON BULK AND SINGLE-CELL RNA-SEQ DATA. 2025 Jun 3;

12. Fortelny N, Bock C. Knowledge-primed neural networks enable biologically interpretable deep learning on single-cell sequencing data. Genome Biol. 2020 Dec;21(1):190.

13. Reynolds J, Pan C. Benchmarking interpretability of deep learning for predictive genomics: Recall, precision, and variability of feature attribution. Domaratzki M, editor. PLOS Comput Biol. 2025 Dec 5;21(12):e1013784.

14. Yang H, Atak D, Yuan M, Li M, Altay O, Demirtas E, et al. Integrative proteo-transcriptomic characterization of advanced fibrosis in chronic liver disease across etiologies. Cell Rep Med. 2025 Feb;6(2):101935.

15. Divya SV, Venkadesh P, Pavithra RJ, M.R A, G L, A L. An Integrated XGBRF Framework for Early Identification of Parkinson's Disease. In: 2024 International Conference on Emerging Research in Computational Science (ICERCS) [Internet]. Coimbatore, India: IEEE; 2024 [cited 2026 Jan 26]. p. 1–9. Available from: https://ieeexplore.ieee.org/document/10894797/

16. Yang B, Gül M, Chen Y. Comparative analysis of deep learning and tree-based models in power demand prediction: Accuracy, interpretability, and computational efficiency. J Build Phys. 2025 Jul;49(1):127–69.

17. Chen T, Moreau T, Jiang Z, Zheng L, Yan E, Cowan M, et al. TVM: An Automated End-to-End Optimizing Compiler for Deep Learning [Internet]. arXiv; 2018 [cited 2026 Feb 6]. Available from: https://arxiv.org/abs/1802.04799

18. Leonardis E, Nagamori A, Thanawalla A, Yang Y, Park J, Saunders H, et al. Massively Parallel Imitation Learning of Mouse Forelimb Musculoskeletal Reaching Dynamics [Internet]. arXiv; 2025 [cited 2026 Feb 6]. Available from: http://arxiv.org/abs/2511.21848

19. Barbieri I, Tzelepis K, Pandolfini L, Shi J, Millán-Zambrano G, Robson SC, et al. Promoter-bound METTL3 maintains myeloid leukaemia by m6A-dependent translation control. Nature. 2017 Dec;552(7683):126–31.

20. Li HM, Ikeda H, Nakabayashi H, Nishi S, Sakai M. Identification of CCAAT enhancer binding protein α binding sites on the human α-fetoprotein gene. Gene. 2007 Mar;389(2):128–35.

21. Yu Y, Jiang L, Wang H, Shen Z, Cheng Q, Zhang P, et al. Hepatic transferrin plays a role in systemic iron homeostasis and liver ferroptosis. Blood. 2020 Aug 6;136(6):726–39.

22. Gema AP, Grabarczyk D, De Wulf W, Borole P, Alfaro JA, Minervini P, et al. Knowledge graph embeddings in the biomedical domain: are they useful? A look at link prediction, rule learning, and downstream polypharmacy tasks. Ma L, editor. Bioinforma Adv. 2024 Jan 5;4(1):vbae097.

23. Cheng Y, Ma X. scGAC: a graph attentional architecture for clustering single-cell RNA-seq data. Mathelier A, editor. Bioinformatics. 2022 Apr 12;38(8):2187–93.

24. Lei L, Han K, Wang Z, Shi C, Wang Z, Dai R, et al. Attention-guided variational graph autoencoders reveal heterogeneity in spatial transcriptomics. Brief Bioinform. 2024 Mar 27;25(3):bbae173.

25. Zhang X, Qian K, Li H. Structure-preserved integration of scRNA-seq data using heterogeneous graph neural network. Brief Bioinform. 2024 Sep 23;25(6):bbae538.

26. Phung T, Lee JJR, Oladapo-Shittu O, Klein EY, Gurses AP, Hannum SM, et al. Zero Inflation as a Missing Data Problem: a Proxy-based Approach [Internet]. arXiv; 2024 [cited 2026 Feb 5]. Available from: http://arxiv.org/abs/2406.00549

27. Yu X, Abbas-Aghababazadeh F, Chen YA, Fridley BL. Statistical and Bioinformatics Analysis of Data from Bulk and Single-Cell RNA Sequencing Experiments. In: Markowitz J, editor. Translational Bioinformatics for Therapeutic Development [Internet]. New York, NY: Springer US; 2021 [cited 2026 Feb 6]. p. 143–75. (Methods in Molecular Biology; vol. 2194). Available from: http://link.springer.com/10.1007/978-1-0716-0849-4_9

28. Kim YJ, Chi M. Temporal Belief Memory: Imputing Missing Data during RNN Training. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence [Internet]. Stockholm, Sweden: International Joint Conferences on Artificial Intelligence Organization; 2018 [cited 2026 Feb 6]. p. 2326–32. Available from: https://www.ijcai.org/proceedings/2018/322

29. Grinsztajn L, Oyallon E, Varoquaux G. Why do tree-based models still outperform deep learning on tabular data? [Internet]. arXiv; 2022 [cited 2026 Jan 27]. Available from: http://arxiv.org/abs/2207.08815

30. Hollmann N, Müller S, Purucker L, Krishnakumar A, Körfer M, Hoo SB, et al. Accurate predictions on small data with a tabular foundation model. Nature. 2025 Jan 9;637(8045):319–26.

31. Zabërgja G, Kadra A, Frey CMM, Grabocka J. Tabular Data: Is Deep Learning all you need? [Internet]. arXiv; 2025 [cited 2026 Feb 2]. Available from: http://arxiv.org/abs/2402.03970

32. Raieli S. Tabular Deep Learning: A Survey from Small Neural Networks to Large Language Models [Internet]. Preprints; 2025 [cited 2026 Feb 6]. Available from: https://www.techrxiv.org/users/961472/articles/1332693-tabular-deep-learning-a-survey-from-small-neural-networks-to-large-language-models?commit=3457feeb2fb0233c766d89f500e0e98503dd9900

6. Code and Data availability

All scripts and data files to reproduce results can be found in a public GIT repository (https://github.com/ChristianLangridge/LEMBAS-RNN-benchmark/tree/main)

7. Supplementary

7.1 Supplementary figure S1

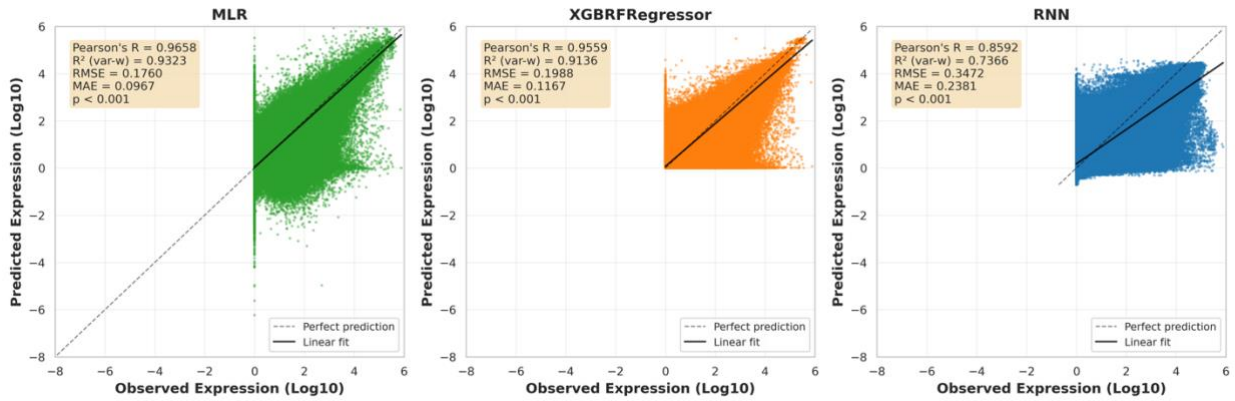

Figure S1. Larger view of (Fig 2.A). Distribution of points outside of (0,0) for predicted versus observed scatter plots was evidence for systematic bias where model express negative expression values for low-expression genes, where actual expression levels were either zero or positive. This phenomenon is common for regression modeling tasks where data is log transformed. MLR and RNN both showed a consistent pattern of values beyond this threshold, evidence that these models are heteroscedastic or that the testing set contained substantial aleatoric noise. XGBRF clearly presented datapoints within (0,0) given the aggressive Gain-based pruning and L1/L2 regularization (shared with RNN's architecture), that are missing in the MLR's architecture. To ensure biological validity, experimenting with LEMBAS-RNN using other activation functions with thresholding behavior like Softplus so that functionally-off predictions can be further explored.

7.2 Supplementary figure S2

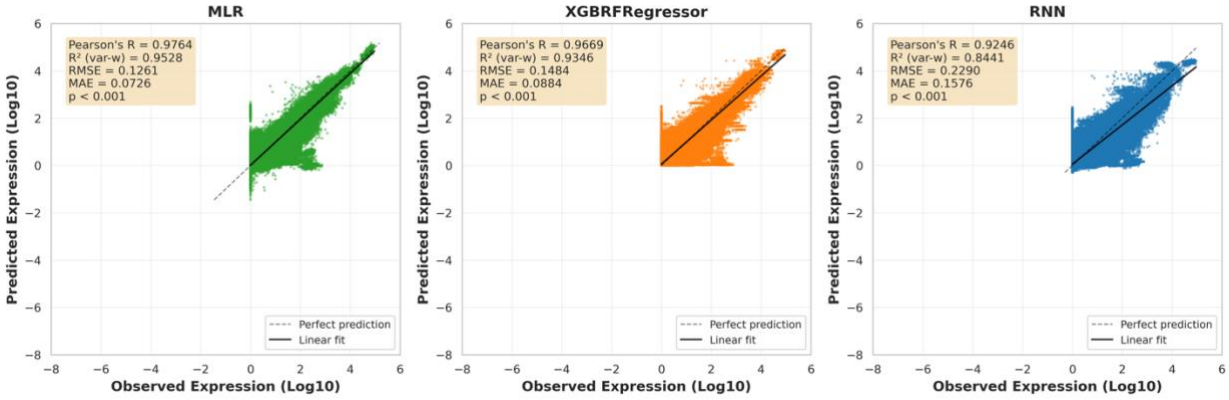

Figure S2. Larger view of (Fig 3.A). Distribution of points outside of (0,0) for predicted versus observed scatter plots were significantly less than for Figure S, where most datapoints are either zero or positive. This phenomenon shows the relationships learned during model training are a better representation of those in the unseen data than in the testing data. This is evidence that the model training contained substantial stochastic noise and/or batch effects that the models could not

fit. Despite strong metrics and good fit, all models still show evidence of heteroscedasticity. Low-level predictions are fanned out, supporting the large variance and residuals from **Figure 2.B**. MLR and RNN still show a consistent pattern, whereas LEMBAS-RNN looks more dispersed with a higher RMSE score.

7.3 Supplementary work S3

This work is the continuation of a previous research thesis conducted by Kejun Li in 2025 under Dr. Cheng Zhang at the KTH Royal Institute of Technology.

See public Github repository (**Code and Data Availability**, ***doc*** folder) for more information about LEMBAS-RNN design and training run.