

A Multi-Task Deep Network for Estimating Engine Torque, and Tailpipe NO_x Emissions in Heavy-Duty Trucks

Amirreza Yasami* Mohamadali Tofigh*
Charles Robert Koch* Mahdi Shahbakhti*

** Mechanical Engineering Department, University of Alberta, AB, Canada (e-mail: {ayasami, tofigh, ckoch, mahdi}@ualberta.ca).*

Abstract: This paper presents a multi-task deep neural network for the concurrent prediction of tailpipe nitrogen oxides (NO_x) emissions, engine-out NO_x, and engine torque in a heavy-duty diesel engine of a commercial regional-haul truck. The proposed architecture integrates a one-dimensional causal convolution layer for adaptive delay embedding, followed by Long Short-Term Memory (LSTM) units to capture long-range temporal dependencies, and multi-head self-attention mechanisms to extract task-specific features within a shared latent representation. Trained and validated on real-world measurements collected at 10 Hz from Class 8 Peterbilt trucks under diverse operating conditions, the multi-task framework achieves substantially higher predictive accuracy than conventional LSTM, Temporal Convolutional Network (TCN), and attention-based baselines, yielding average reductions of approximately 32% in MAE and 40% in RMSE, with the maximum RMSE improvement reaching 43%. To further address prediction degradation in high-value emission regimes, an Extreme Value Theory (EVT)-inspired tail-weighted loss function is introduced and evaluated. Relative to the standard objective, the EVT-based variant improves MAE, RMSE, and MAPE in the high-value regime by 5.10%, 6.13%, and 5.76%, respectively, and reduces the maximum prediction error by 7.06%, at a modest cost of 4-6% to overall average performance. The resulting architecture is compatible with supervisory Model Predictive Control (MPC) schemes, enabling the optimization of key engine control variables such as fuel injection quantities, exhaust gas recirculation rates, and intake air mass flow for improved emissions management.

Keywords: Data-driven modeling, Neural network, Time series prediction, Diesel engine control, Extreme Value Theory, real-world truck data.

1. INTRODUCTION

Diesel engines remain the dominant power source for heavy-duty transportation because of their robustness, durability, and high fuel efficiency. However, they are also among the major contributors to nitrogen oxides (NO_x) emissions, which have well-documented adverse impacts on air quality and public health Shahpour et al. (2024). As emission regulations continue to tighten, particularly under Real Driving Emissions (RDE) requirements for on-road heavy-duty vehicles, accurate and computationally efficient estimation of NO_x under transient operating conditions has become increasingly important for modern engine calibration, diagnostics, and emission control.

In heavy-duty diesel engines, torque generation and NO_x formation are governed by highly coupled and nonlinear processes involving fueling, air handling, combustion temperature, exhaust gas recirculation (EGR), and after-treatment dynamics. These interactions become even more complex under real-world driving, where rapid load changes, varying vehicle mass, road grade, and ambient conditions continuously alter the engine operating regime. As a result, developing reliable predictive models for engine torque and NO_x emissions remains a challenging task,

particularly when the objective is to support real-time estimation using signals available from production-level sensors and Electronic Control Units (ECUs).

Conventional emission-estimation approaches rely primarily on physics-based and semi-empirical formulations. These include zero-dimensional and multi-zone thermodynamic models, as well as chemical-kinetics-based descriptions of NO_x formation through thermal NO, prompt NO, and fuel NO pathways Heywood (2019). Although such models provide useful physical interpretation, they often require extensive calibration effort and may not generalize well across different engine platforms and operating conditions. In addition, their computational burden limits their direct applicability in real-time embedded environments Tadros and Boulougouris (2025).

To improve physical fidelity, higher-order simulation tools such as computational fluid dynamics (CFD) and one-dimensional engine simulation platforms including GT-Suite have also been employed to analyze in-cylinder combustion, temperature distributions, and species transport Dahifale and Patil (2017). While these approaches offer detailed insight into combustion and emission mechanisms, they are generally too computationally intensive

for real-time deployment and are therefore more suitable for offline analysis, calibration, or hardware-in-the-loop studies D’Errico et al. (2014). Hybrid approaches that combine simulation outputs with machine learning models have shown strong predictive accuracy under controlled or steady-state conditions Mohammad et al. (2021), but their practical deployment remains constrained by modeling complexity and limited adaptability to highly transient real-world operation.

These limitations have motivated growing interest in data-driven methods that directly map measured engine signals to performance and emission outputs. Using readily available variables such as engine speed, fuel rate, intake air flow, EGR flow, and intake manifold pressure, machine learning models can capture nonlinear relationships without requiring explicit first-principles combustion modeling. For example, Random Forest models have been used with onboard diagnostic (OBD) data to estimate engine-out NO_x , yielding moderate predictive accuracy Selvam et al. (2021). More recently, recurrent neural networks (RNNs) and Long Short-Term Memory (LSTM) architectures have demonstrated promising capability for modeling transient engine behavior and NO_x dynamics under varying conditions Shin et al. (2023); Shahpouri et al. (2024).

Deep learning is particularly attractive for engine applications because it can learn hierarchical spatio-temporal representations from high-frequency multivariate data. Within this context, *multi-task learning* (MTL) offers a compelling alternative to conventional single-target prediction. Rather than learning each output independently, MTL exploits common latent structure across related tasks, allowing the model to improve generalization by sharing informative representations. This distinction is important in engine applications, where quantities such as torque, engine-out NO_x , and tailpipe NO_x are physically coupled through combustion phasing, air-fuel mixing, EGR dynamics, and after-treatment response. In contrast to standard *multi-output* neural networks ÖZENER et al. (2013); Norouzi (2022); Gad et al. (2024), which typically share most layers and only separate at the final output layer, a true MTL framework introduces both shared and task-specific components, enabling the network to learn common dynamics while preserving specialization for each target Ruder (2017); Li et al. (2021).

Despite increasing interest in learning-based virtual sensing, much of the existing literature remains focused on predicting individual outputs, most commonly engine-out NO_x or torque, in isolation. Several studies have demonstrated that deep neural networks can estimate diesel engine NO_x using production-level engine and after-treatment signals without detailed combustion modeling Samosir et al. (2023); Kim et al. (2021). Likewise, temporal deep learning architectures have been applied to transient NO_x estimation with encouraging results Norouzi et al. (2023). However, these studies generally do not address the simultaneous estimation of multiple physically related outputs within a unified learning framework.

A further limitation of prior work is that many published models are developed using steady-state or quasi-steady datasets, where engine operating conditions occupy

relatively narrow regions of the full operating envelope. Models trained under such restricted conditions often exhibit degraded generalization when exposed to the strong transients and broader variability encountered in real-world driving. Moreover, standard loss functions optimized for average error performance tend to underperform on extreme tail events, i.e., high-value emission spikes that are precisely the conditions most critical for regulatory compliance. Addressing this limitation requires not only a robust architecture, but also a training objective that explicitly accounts for the statistical properties of the tail distribution.

To address these gaps, this study proposes an *explainable multi-task deep learning framework* for the simultaneous prediction of engine torque, engine-out NO_x , and tailpipe NO_x using high-frequency real-world data collected from Class-8 Peterbilt regional-haul trucks. The proposed architecture combines a one-dimensional causal convolution layer, LSTM-based temporal modeling, and multi-head self-attention to capture both short-term transients and long-range temporal dependencies. By jointly learning multiple correlated targets through a shared latent representation with task-specific heads, the model improves predictive accuracy, robustness, and generalization across diverse operating scenarios. Furthermore, an Extreme Value Theory (EVT)-inspired tail-weighted loss function is introduced to specifically improve prediction fidelity in high-value emission regimes, providing a principled mechanism for addressing distributional asymmetry in real-world engine data.

The main contributions of this study are summarized as follows:

- A high-resolution real-world dataset was collected at 10 Hz from instrumented Class-8 Peterbilt trucks operating between Edmonton and Calgary under diverse terrains, loads, and ambient conditions, providing a challenging basis for transient engine modeling.
- A novel multi-task deep learning architecture is developed to jointly estimate engine torque, engine-out NO_x , and tailpipe NO_x by integrating causal convolution, LSTM, and self-attention mechanisms within a shared representation framework.
- An EVT-inspired tail-weighted loss function is formulated and incorporated into the MTL training objective to explicitly penalize errors in the high-value tail of the emission distribution, reducing maximum prediction errors and improving accuracy on extreme operating events.
- The proposed model, in both its standard and modified loss configurations, is benchmarked against conventional LSTM, Temporal Convolutional Network (TCN), attention-based, and multi-output baselines, demonstrating superior predictive accuracy and strong computational efficiency for practical implementation.

The remainder of this paper is organized as follows. Section 2 describes the experimental setup, vehicle specifications, and data acquisition procedure. Section 3 presents the proposed model architecture and learning formulation, including the EVT-inspired loss. Section 4 reports the comparative experimental results, and Section 5 concludes

the paper with final remarks and directions for future work.

2. EXPERIMENTS AND DATA COLLECTION PROCEDURE

This section describes the experimental setup, including the vehicle and engine specifications, after-treatment system configuration, test conditions, and the data acquisition system.

2.1 Vehicle and Engine Specifications

The experiments were conducted on a 2019 Peterbilt 579 trailer tractor, a Class 8 heavy-duty truck featuring a tractor paired with a 53-ft trailer shown in Figure 2.1 (a). The vehicle is equipped with an Eaton automated 12-speed transmission, a Meritor steering axle rated at 13,200 lbm, and combined Meritor rear axles supporting up to 40,000 lbm. It is powered by a Cummins X15-Efficiency, 14.9-liter engine that meets US EPA Tier III emission regulations. Table 1 provides a summary of the engine characteristics.

Table 1. Cummins X15 engine parameters

Parameter	Value
Turbocharger	VGT Turbocharger
Fuel Type	Diesel
Number of Cylinders	6
Displacement Volume	14.9 L
Rated Power	336 kW @ 2100 rpm
Rated Torque	2373 N·m @ 900 rpm
Emission Standard	US EPA Tier III (0.2 g/bhp-hr)

2.2 After-Treatment System Configuration

To meet emission standards, the engine is equipped with a Single-Module™ after-treatment system that comprises a Diesel Oxidation Catalyst (DOC), a Diesel Particulate Filter (DPF), and a Selective Catalytic Reduction (SCR) unit. The DOC facilitates the oxidation of CO and hydrocarbons, the DPF captures particulate matter from the exhaust, and the SCR converts NO_x to nitrogen and water vapor using urea as a reducing agent. Table 2 details the operating temperature ranges for each of these exhaust gas after-treatment components.

Table 2. Operating Temperature of After-Treatment Components

Component	DOC	DPF	SCR
Temperature	260 to 399 °C	482 to 649 °C	> 150 °C

2.3 Test Conditions and Driving Route

On-road tests were conducted on Highway 2 between Calgary and Edmonton in Central Alberta, Canada. The route, primarily a 4-lane divided highway with long straight sections and low grades (less than 6%), presented a terrain with an elevation change from 670 m in Edmonton to 1048 m in Calgary, an elevation difference of 378 m corresponding to an average slope of approximately 1.3 m/km. Ambient temperatures during the tests ranged from −24°C to 12°C. Under commercial operation conditions and with varying truck loads (gross vehicle weight up to 39,287 kg), the truck completed 37 round trips covering over 10,000 km.

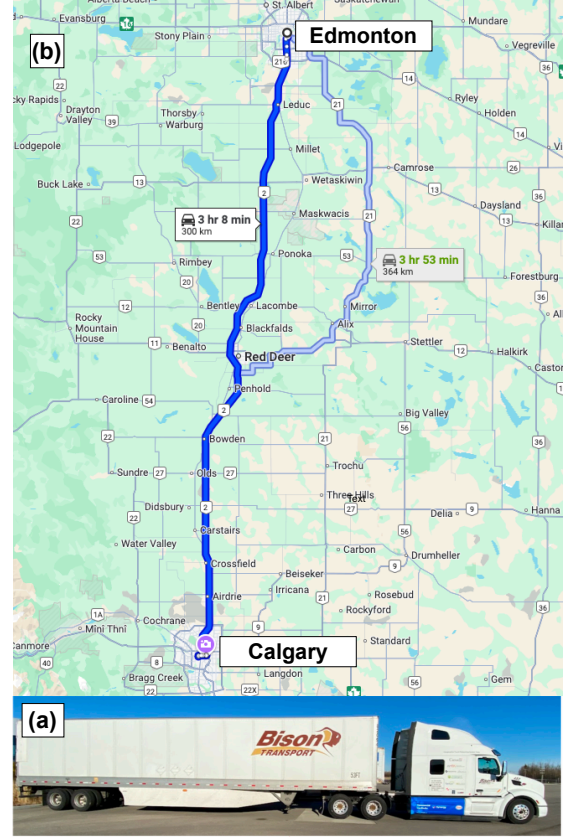


Fig. 1. Experiment Setup: a) the tested Peterbilt 579 trailer vehicle used to collect data for this study, b) the tested route in this study.

2.4 Data acquisition and measured parameters

A custom-designed Data Acquisition (DAQ) system was implemented to collect, synchronize, and safeguard data from various sensors at a uniform sampling frequency of 10 Hz. This system recorded an extensive array of engine and vehicle parameters, including ambient temperature (ranging from −24°C to 12°C), vehicle speed (1 to 115.5 km/h), engine speed (486.9 to 2036.8 rpm), engine torque (0 to 2601.1 N·m), fuel consumption (0 to 71.4 L/h), Exhaust Gas Recirculation (EGR) flow rate (0 to 570 kg/h), intake air mass flow rate (83.9 to 2101.6 kg/h), SCR intake temperature (79 to 374°C), and SCR intake NO_x concentration (0.2 to 3012 ppm). Overall, the dataset comprises over 4 million data points gathered during 37 trips between Edmonton and Calgary, as shown in Figure 2.1 (b), with data from 31 trips allocated for model training and 3 trips each reserved for validation and testing. More details about the experimental setup and data collection are found in Jiang et al. (2024).

In summary, the comprehensive experimental framework, comprising a well-instrumented heavy-duty truck, a state-of-the-art after-treatment system, rigorous on-road testing under diverse environmental conditions, and high-frequency data acquisition, establishes a robust foundation for the development and validation of deep learning models for NO_x emissions.

3. PROPOSED MODEL ARCHITECTURE

3.1 Problem definition

The system of this study is a multi-variate time series for multi-task predictions. We define vector $\mathbf{u}_t = [\text{EngSp}(t), \dot{m}_{\text{EGR}}(t), \dot{m}_{\text{EIA}}(t), \dot{m}_{\text{fuel}}(t), \text{EIMP}(t)]^\top \in \mathbb{R}^5$ containing 5 input variables where $\text{EngSp}(t)$ is the engine speed, $\dot{m}_{\text{EGR}}(t)$ is the EGR mass flow rate, $\dot{m}_{\text{EIA}}(t)$ is the engine intake air mass flow rate, $\dot{m}_{\text{fuel}}(t)$ is the engine fuel rate, and $\text{EIMP}(t)$ is the engine intake manifold pressure and vector $\mathbf{y} \in \mathbb{R}^3$ as three outputs $\mathbf{y} = [y^{(1)}, y^{(2)}, y^{(3)}]^\top = [\text{Torq}_{\text{eng}}, \text{NOx}_{\text{eng}}, \text{NOx}_{\text{tail}}]^\top \in \mathbb{R}^{3 \times 1}$, as discussed in the previous section. There are a length- T sequence of observations of outputs $\mathbf{y} = [\mathbf{y}_1, \dots, \mathbf{y}_T] \in \mathbb{R}^{3 \times T}$ and inputs $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_T] \in \mathbb{R}^{5 \times T}$. The task is to train a learning algorithm in a supervised setting to predict all three output variables one step ahead, given a lookback (L) of previous observations:

$$\hat{\mathbf{y}}_{t+1} = f([\mathbf{u}_{t-L} \dots \mathbf{u}_{t-1}, \mathbf{u}_t] | \theta) \quad (1)$$

where f is a mapping with learnable parameters θ that are optimized over the training set to minimize the training loss.

3.2 Standard Loss Function

The standard training objective minimizes the weighted sum of mean absolute errors across all three tasks:

$$\mathcal{L}_{\text{std}} = \frac{1}{n} \sum_{i=1}^n [\alpha |e_i^{(1)}| + \beta |e_i^{(2)}| + \gamma |e_i^{(3)}|] \quad (2)$$

where $e_i^{(k)} = y_i^{(k)} - \hat{y}_i^{(k)}$ denotes the prediction error for task k , and $\alpha, \beta, \gamma \in (0, 1]$ are fixed task-weighting coefficients that regulate the relative importance of each output during training. In this study, all three coefficients are set to unity, treating each task with equal weight. While this formulation performs well on average prediction accuracy, it assigns equal penalty to errors regardless of the magnitude of the target variable, which may limit the model's ability to capture rare but safety-critical high-emission events in the tail of the distribution.

3.3 EVT-Inspired Tail-Weighted Loss Function

To address the degraded accuracy observed at extreme emission levels, a tail-weighted loss function is introduced, inspired by Extreme Value Theory (EVT). EVT provides a principled statistical framework for characterizing the behavior of rare events in the upper tail of a distribution, most commonly through the Generalized Pareto Distribution (GPD), which describes exceedances above a high threshold μ ?

The key idea is to augment the standard loss with a tail penalty that grows as a function of the target value's deviation above a prescribed threshold. Specifically, for each sample i and task k , a tail weight $w_i^{(k)}$ is defined as:

$$w_i^{(k)} = 1 + \lambda \left[\max \left(0, \frac{y_i^{(k)} - \mu^{(k)}}{\sigma^{(k)}} \right) \right]^\xi \quad (3)$$

where $\mu^{(k)}$ is a task-specific threshold (set to the 90th percentile of the training distribution for each output), $\sigma^{(k)} > 0$ is a scale parameter estimated from the empirical tail of the training data, $\xi \geq 0$ is a shape parameter controlling the rate of tail growth, and $\lambda > 0$ is a penalty coefficient governing the overall strength of the tail emphasis. The modified multi-task loss then becomes:

$$\mathcal{L}_{\text{mod}} = \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^3 w_i^{(k)} |e_i^{(k)}| \quad (4)$$

By construction, samples with target values below the threshold $\mu^{(k)}$ receive unit weight, preserving standard loss behavior in the bulk of the distribution. For samples exceeding the threshold, the weight increases monotonically according to the GPD-shaped penalty, directing the optimizer to allocate greater capacity toward accurately reproducing high-value events. The EVT parameters ($\sigma^{(k)}, \xi$) are estimated by fitting a GPD to the empirical tail exceedances of the training set using maximum likelihood estimation, while λ is treated as a tunable hyperparameter selected via validation performance on the high-value subset.

It is important to note that this modification introduces an explicit performance trade-off: by allocating more loss weight to tail samples, the optimizer sacrifices a fraction of its capacity for average-regime accuracy. The extent of this trade-off is quantified in the experimental results of Section 4.

3.4 Model description and equations

Figure 2 shows the schematic of the proposed neural network to fulfill the multi-task defined throughout Equations (1)–(4). One of the major difficulties in dealing with multivariate time series prediction is selecting an optimal set of lags from historical observations to create lookback windows. Delay embedding is an established technique in time series analysis to reconstruct the state space of a dynamical system from a sequence of observations as $[u(t), u(t - \tau), u(t - 2\tau), \dots, u(t - (d - 1)\tau)]$, in which d , the embedding dimension, and τ , the time delay, are two critical parameters Tan et al. (2023). This embedding allows models to capture the underlying dynamics of the system by providing a richer, higher-dimensional view of its evolution, making it particularly useful for system identification tasks.

Mutual information and False Nearest Neighbors (FNN) are two widely used methods to determine optimal embedding parameters, τ and d . The first minimum of the average mutual information between $x(t)$ and $x(t + \tau)$ is often used to select an appropriate τ , as it reflects the time when the delayed values add the most new information. The FNN method helps determine the embedding dimension d by identifying when increasing the dimension no longer significantly reduces the number of falsely close neighbors, indicating that the reconstructed state space is sufficiently unfolded. A major challenge when working with multiple time series is that each series often requires its own distinct set of optimal embedding parameters. Using a single, sub-optimal parameter set for all series can lead to degraded model performance.

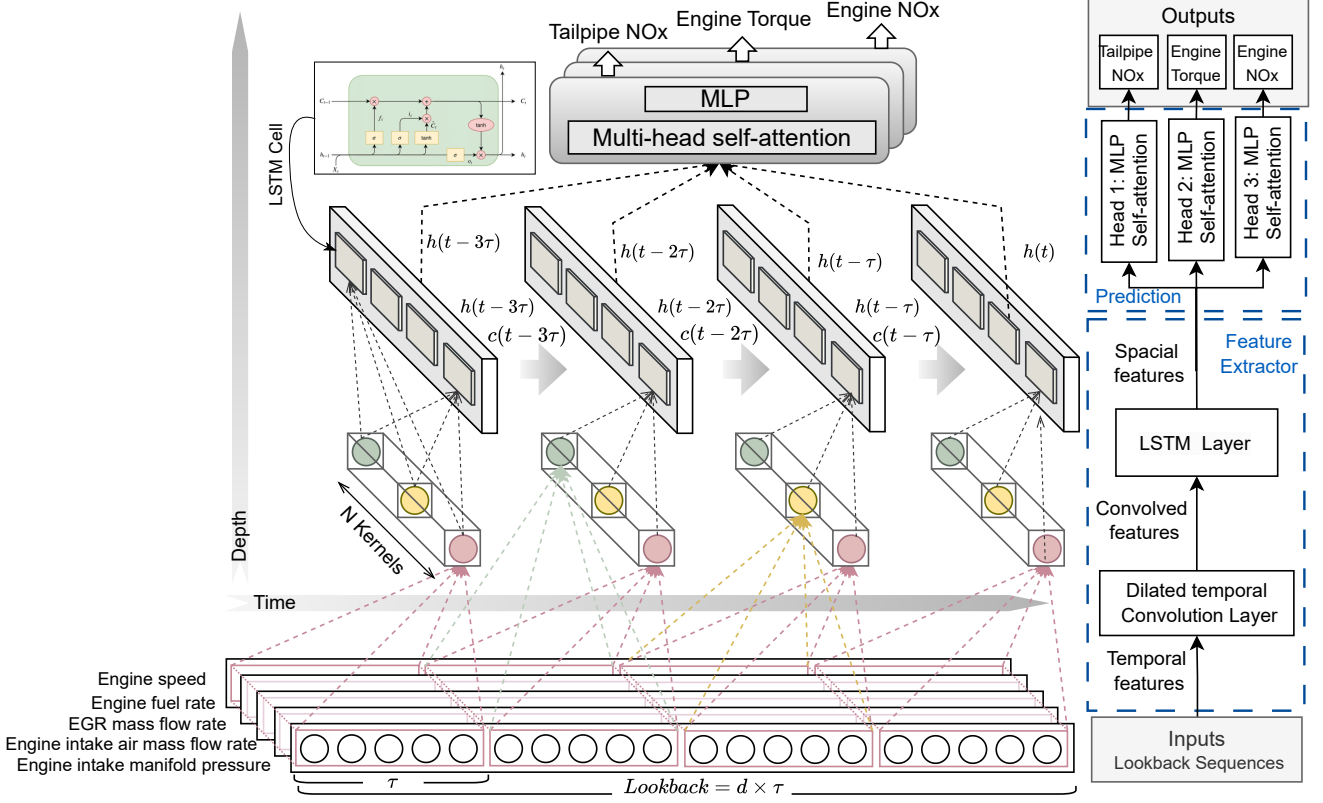


Fig. 2. Explainable neural network schematic for multi-task learning in multi-variate time series. A bank of learnable convolution filters with N channels (kernels) of size τ applies across a Lookback of sequences with a stride of τ , where dimension d and delay τ are the maximum delay-embedding parameters among time series inputs. Each convolution channel can learn the most actionable and uncorrelated lags within each input sequence. An LSTM processes the resulting compressed sequences (length d) to capture temporal dependencies. An attention-based subnet finally employs the spatial/temporal features with inherent positional ordering to fulfill the multiple prediction tasks. Three self-attention heads, each followed by a fully connected MLP, independently predict Tailpipe NO_x, Engine Torque, and Engine-out NO_x.

In this paper, we design a one-dimensional causal convolution layer to automatically perform delay embedding on all of the input sequences Tofigh et al. (2024a). First, the embedding parameters for all input sequences are computed, and the maximum values are selected as τ (time delay) and d (embedding dimension). A one-dimensional convolution is then applied over a lookback window of size τ by d as follows:

$$\mathbf{Z}^{[L \times d]} = \mathbf{W}^{[L \times \tau \times N]} \otimes \mathbf{U}^{[N \times (d \cdot \tau)]} \quad (5)$$

where Z is the convolution output, L is the output channels (number of filters), N is the input channels (number of input sequences), and W is the learnable tensor weight. For each output channel, the convolution operator $\otimes$ yields:

$$Z(t) = \text{ReLU}\left(\sum_{n=1}^N \sum_{k=0}^{\tau-1} W_{n,k} \cdot U_n(t \cdot S + k)\right) \quad (6)$$

in which S is the convolution stride, which we set equal to τ . This structure maps the input tensor of shape $[N \times (d \cdot \tau)]$, with $(d \cdot \tau)$ being the lookback size, into a convolved output of shape $[L \times d]$. We set the number of output channels equal to the number of input channels. This design enables the convolutional filters to learn the most relevant features within each segment of the sequence by

convolving over the d segments, each of length τ , through adjusting their learnable weights during training. Unlike statistical delay embedding methods, our approach allows the model to learn lagged features from multivariate time series adaptively. Specifically, the convolutional filters can discover the most informative lags for each input sequence within each segment. On a higher level, they generate a more expressive representation of the lagged features, effectively transforming each segment of τ time steps and N features into a compact vector of N uncorrelated elements.

A recurrent layer with r LSTM units, comprising hidden states (h) and cell states (c), is applied to the output of the convolutional layer to model temporal dependencies within the lookback window. Specifically, it processes a sequence of d convolved feature vectors in a sequential manner:

$$\mathbf{H}^{[r \times d]} = \text{LSTM}(\mathbf{Z}^{[L \times d]}) \quad (7)$$

where H denotes the output tensor of the LSTM layer. A detailed description of standard LSTM computations can be found in Tofigh et al. (2025). Although LSTMs are powerful models for capturing temporal patterns, they often face challenges related to training stability when processing long sequences. In our framework, the convolutional layer acts as a compression mechanism that reduces

the dimensionality of the original lookback window before it is passed to the LSTM. This design not only preserves relevant temporal features but also facilitates more stable and efficient training through backpropagation through time (BPTT), as discussed in Tofigh et al. (2024b).

To enhance predictive performance and support multi-task predictions, we introduce a multi-head attention layer with three parallel heads, each dedicated to predicting one of the target variables. Self-attention mechanisms compute pairwise interactions between all time steps in a sequence, allowing the model to selectively focus on the most relevant temporal features, regardless of their position. This is achieved by generating query, key, and value vectors, and computing weighted combinations of the values based on the similarity between queries and keys Vaswani et al. (2017). In our design, the attention heads operate on the informative temporal representations produced by the LSTM layer $H \in \mathbb{R}^{r \times d}$. By stacking attention on top of the CNN-LSTM pipeline, we enable the model to capture long-range dependencies and task-specific patterns more effectively. Each attention head is followed by a dedicated MLP that maps the context-aware features to one specific target variable, enabling accurate and decoupled multi-output predictions. Each head independently computes a context vector using a self-attention operation:

$$Q^{(i)} = W_Q^{(i)} H, \quad K^{(i)} = W_K^{(i)} H, \quad V^{(i)} = W_V^{(i)} H \quad (8)$$

$$A^{(i)} = \text{softmax} \left(\frac{Q^{(i)} K^{(i)\top}}{\sqrt{r}} \right) V^{(i)} \quad (9)$$

for $i = 1, 2, 3$. In Eq. (8), W_Q, W_K, W_V are query, key, and value learnable projection matrices, r is the dimension of the LSTM output vector, and $A^{(i)}$ is the attention output of the i -th head. Each attention output is aggregated over time to form a context vector, which is then passed through a separate MLP to produce the final scalar prediction for each task:

$$\hat{y}^{(i)} = W_2^{(i)} \cdot \text{ReLU}(W_1^{(i)} A^{(i)} + b_1^{(i)}) + b_2^{(i)} \quad (10)$$

$$\hat{\mathbf{y}} = [\hat{y}^{(1)}, \hat{y}^{(2)}, \hat{y}^{(3)}]^\top \quad (11)$$

4. EXPERIMENTAL RESULTS

This section evaluates the effectiveness of the proposed explainable framework using real-world data described in Section 2. Two variants of the proposed model are assessed: MTL (Std), trained with the standard uniform loss of Eq. (2), and MTL (Mod), trained with the EVT-inspired tail-weighted loss of Eq. (4). Both variants are benchmarked against three single-task baselines, a standalone LSTM network, a Temporal Convolutional Network (TCN), and an attention-based network, as well as two conventional multi-output architectures, MO-DNN and MO-LSTM.

All models are trained using the Adam optimizer with hyperparameters selected through Bayesian optimization. Prediction performance is assessed using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and Maximum Error (Max Error). The latter two metrics are particularly informative for assessing performance in the high-value tail of the emission distribution. Computational efficiency

is evaluated through training time and inference latency. To ensure a fair comparison: (i) all models are trained, validated, and tested on the same dataset with a split ratio of 70:15:15; (ii) hyperparameters are optimized via Bayesian optimization; (iii) implementation is consistent across models using PyTorch on a Linux system with an NVIDIA GeForce RTX 4090 GPU; and (iv) identical evaluation metrics are applied across all experiments.

Figure 4 illustrates the performance of MTL (Std) on unseen test data. The results demonstrate that the model effectively captures the underlying dynamics of all three outputs, accurately tracking both transient responses and steady-state behavior. The close alignment between predicted and measured values across a broad range of real-world operating conditions highlights the model's strong generalization capability and its potential for practical applications including torque control and NO_x emission management.

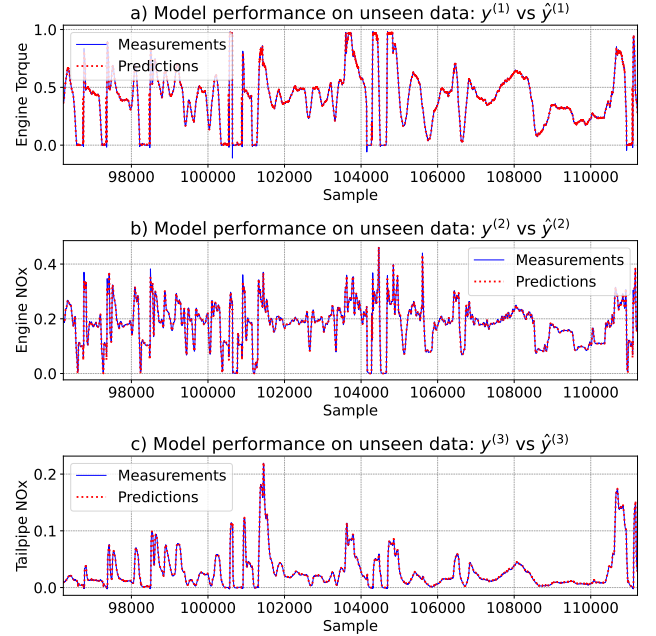


Fig. 3. Performance of the proposed model in the simultaneous prediction of diesel engine outputs: torque, engine-out NO_x, and tailpipe NO_x values.

Table 3 presents the comparative performance of all benchmarked models under both overall and high-value prediction regimes. High-value predictions are defined as samples in which at least one target variable exceeds its 90th-percentile threshold in the test set, capturing the tail region of primary regulatory concern.

Under overall predictions, MTL (Std) achieves the lowest MAE, RMSE, and MAPE across all models, yielding average reductions of approximately 32% in MAE and 40% in RMSE relative to the single-task baselines (LSTM, TCN, attention), with the maximum RMSE improvement reaching 43% (vs TCN). Improvements relative to the multi-output baselines (MO-DNN, MO-LSTM) are larger still, exceeding 50% in RMSE, confirming that treating the three targets as independent outputs without explicit task coupling degrades generalization. These results validate

the benefit of shared latent representations in capturing the inter-task dependencies inherent in diesel engine dynamics.

In the high-value prediction regime, MTL (Std) continues to outperform all baselines, though the improvement margin narrows relative to the overall regime. This narrowing is expected: the standard loss function is optimized for average performance and therefore does not explicitly direct model capacity toward accurate tail predictions. As a consequence, absolute errors and maximum errors increase across all models for extreme emission events, reflecting the inherent difficulty of accurately predicting rare transient peaks.

MTL (Mod), trained with the EVT-inspired tail-weighted loss, addresses this limitation directly. As shown in Table 3, MTL (Mod) achieves the lowest Max Error across both the overall and high-value regimes, reducing the maximum prediction error by 7.06% relative to MTL (Std). In the high-value regime specifically, MTL (Mod) outperforms MTL (Std) across all metrics, with MAE, RMSE, and MAPE improvements of 5.10%, 6.13%, and 5.76%, respectively. These gains are consistent with the intended effect of the tail-weighted loss: by assigning greater penalty to high-value samples during training, the optimizer reallocates model capacity toward the tail distribution, improving fidelity precisely where conventional loss functions are weakest.

This improvement, however, comes with a trade-off. Under overall predictions, MTL (Mod) incurs a slight degradation relative to MTL (Std), with MAE, RMSE, and MAPE increasing by approximately 4.95%, 5.74%, and 4.27%, respectively (Table 4). This behavior is a direct consequence of the EVT-based reweighting: the modified objective prioritizes tail accuracy at the expense of marginally higher errors in the bulk of the distribution. Despite this trade-off, MTL (Mod) still substantially outperforms all baselines under overall predictions, and delivers the best performance of any model under high-value and maximum-error criteria.

In terms of computational cost, MTL (Mod) introduces negligible overhead relative to MTL (Std), training time increases by less than 1.3%, since the EVT weight computation is performed offline prior to training. Both variants maintain inference latency comparable to LSTM and TCN, and significantly lower than the attention-based baseline, underscoring their suitability for real-time embedded deployment.

5. CONCLUSION

This work presented a multi-task deep neural network for the joint prediction of tailpipe NO_x , engine-out NO_x , and engine torque using high-frequency multivariate data from heavy-duty diesel engines. The proposed architecture integrates a one-dimensional causal convolution layer for adaptive temporal alignment, LSTM units for capturing sequential dependencies, and a multi-head self-attention mechanism to extract task-relevant features within a shared latent space. Validation on an extensive Class 8 Peterbilt truck dataset covering more than 10,000 km of real-world operation demonstrated that the multi-task for-

Table 3. Performance comparison demonstrating the trade-off of the proposed multi-task learning model with a modified tail-weighted cost function (MTL Mod) against the standard cost function (MTL Std) and baseline networks.

Metrics	MTL (Mod)	MTL (Std)	LSTM	TCN	Attention	MO-DNN	MO-LSTM
Overall Predictions							
MAE	0.0106	0.0101	0.0157	0.0162	0.0131	0.0214	0.0179
RMSE	0.0129	0.0122	0.0205	0.0214	0.0193	0.0285	0.0247
MAPE	0.0122	0.0117	0.0183	0.0198	0.0151	0.0249	0.0212
Max Error	0.0790	0.0850	0.0980	0.1020	0.0950	0.1050	0.0990
High-Value Predictions							
MAE	0.0242	0.0255	0.0295	0.0300	0.0285	0.0305	0.0292
RMSE	0.0291	0.0310	0.0360	0.0365	0.0348	0.0370	0.0355
MAPE	0.0278	0.0295	0.0342	0.0348	0.0330	0.0352	0.0338
Max Error	0.0790	0.0850	0.0980	0.1020	0.0950	0.1050	0.0990
Computational Complexity							
Train time [Sec]	6.012e ³	5.934e³	6.870e ³	5.915e ³	9.361e ³	3.512e ³	7.243e ³
Runtime [Sec]	0.131e⁻³	0.131e⁻³	0.155e ⁻³	0.129e ⁻³	0.288e ⁻³	0.098e ⁻³	0.167e ⁻³

Table 4. Percentage improvement of the modified cost function model (MTL Mod) over the standard model (MTL Std) and baselines. Negative values indicate where the modified model sacrifices average accuracy to achieve improved tail prediction performance.

Metric Improvement [%]	MTL (Std)	LSTM	TCN	Attention	MO-DNN	MO-LSTM
Overall Predictions						
MAE Improvement	-4.95	32.48	34.57	19.08	50.47	40.78
RMSE Improvement	-5.74	37.07	39.72	33.16	54.74	47.77
MAPE Improvement	-4.27	33.33	38.38	19.21	51.00	42.45
Max Error Improvement	7.06	19.39	22.55	16.84	24.76	20.20
High-Value Predictions						
MAE Improvement	5.10	17.97	19.33	15.09	20.66	17.12
RMSE Improvement	6.13	19.17	20.27	16.38	21.35	18.03
MAPE Improvement	5.76	18.71	20.11	15.76	21.02	17.75
Max Error Improvement	7.06	19.39	22.55	16.84	24.76	20.20

mulation achieves substantially higher accuracy than conventional single-task and multi-output baselines. Specifically, relative to single-task baselines (LSTM, TCN, attention), MTL (Std) yields average reductions of approximately 32% in MAE, 40% in RMSE, and 33% in MAPE, with the maximum RMSE improvement reaching 43% (vs TCN); improvements relative to the multi-output baselines (MO-DNN, MO-LSTM) are larger still, exceeding 50% in RMSE. The model also maintains computational efficiency suitable for real-time deployment.

To address the limitation of standard loss functions in handling extreme tail events, an EVT-inspired tail-weighted loss function was formulated and evaluated. Relative to MTL (Std), the modified variant MTL (Mod) improved high-value-regime MAE, RMSE, and MAPE by 5.10%, 6.13%, and 5.76%, respectively, and reduced the maximum prediction error by 7.06%. As expected, this improvement entails a modest sacrifice in average prediction accuracy (4.95% MAE, 5.74% RMSE, and 4.27% MAPE in the overall regime), a well-defined trade-off governed by the EVT penalty coefficient, which can be tuned according to the stringency of the target regulatory threshold.

These results collectively highlight the effectiveness of shared representation learning for capturing coupled engine dynamics, and the utility of distribution-aware training objectives for addressing rare but safety-critical emission events. Future work will explore coupling the proposed model with physics-informed constraints and online adaptation mechanisms, as well as extending the EVT-based formulation to jointly optimize threshold selection and penalty strength in an end-to-end differentiable manner,

targeting fully deployable closed-loop emission control in production engine environments.

ACKNOWLEDGMENT

The experimental data used in this study were collected as part of the Transport Canada and Alberta Motor Transport Association (AMTA) sponsored project (Project No. T8009–190376/001/SC). The authors gratefully acknowledge Luo Jiang from the University of Alberta for his contributions to the Peterbilt truck data collection.

REFERENCES

- Dahifale, B.S. and Patil, A.S. (2017). Diesel engine performance improvement for constant speed application using cfd. In *ASME International Mechanical Engineering Congress and Exposition*, 70012. doi:10.1115/imece2017-70012.
- D'Errico, G., Lucchini, T., Onorati, A., and Hardy, G. (2014). Computational fluid dynamics modeling of combustion in heavy-duty diesel engines. *International Journal of Engine Research*, 16(1), 112–124. doi: 10.1177/1468087414561276.
- Gad, M.S. et al. (2024). Artificial neural network based forecasting of diesel engine performance and emissions for biodiesel blends at varying engine speed. *Scientific Reports*, 14, ?
- Heywood, J.B. (2019). *Internal Combustion Engine Fundamentals*. McGraw-Hill, New York, 2nd edition.
- Jiang, L., Kheyrollahi, J., Koch, C.R., and Shahbakhti, M. (2024). Cooperative truck platooning trial on canadian public highway under commercial operation in winter driving conditions. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 239(7), 2272–2284. doi:10.1177/09544070241245477.
- Kim, J. et al. (2021). Prediction of engine no for virtual sensor using deep neural network. *Oil Gas Science and Technology – Revue d'IFP Energies nouvelles*, 76. doi: 10.2516/ogst/2020087.
- Li, J. et al. (2021). Empirical evaluation of multi-task learning in deep neural networks. *Neural Computing and Applications*. doi:10.1007/s00521-020-05268-w. Shows empirical gains in performance and generalization from MTL.
- Mohammad, A., Rezaei, R., Hayduk, C., Delebinski, T.O., Shahpouri, S., and Shahbakhti, M. (2021). Hybrid physical and machine learning-oriented modeling approach to predict emissions in a diesel compression ignition engine. In *SAE Technical Paper*, 2021-01-0496. doi: 10.4271/2021-01-0496.
- Norouzi, A. (2022). *Machine Learning and Deep Learning for Modeling and Control of Internal Combustion Engines*. Ph.D. thesis, University of Alberta. URL https://sites.ualberta.ca/~ckoch/Thesis/Armin_thesis.pdf.
- Norouzi, A., Heidarifar, H., Borhan, H., Shahbakhti, M., and Koch, C.R. (2023). Integrating machine learning and model predictive control for automotive applications: A review and future directions. *Engineering Applications of Artificial Intelligence*, 120, 105878. URL <https://doi.org/10.1016/j.engappai.2023.105878>.
- Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.
- Samosir, B.F., Quach, N.Y., and Lim, O.K.C. (2023). Nox emissions prediction in diesel engines: a deep neural network approach. *Environmental Science and Pollution Research*. doi:10.1007/s11356-023-30937-3.
- Selvam, H., Shekhar, S., and Northrop, W.F. (2021). Prediction of nox emissions from compression ignition engines using ensemble learning-based models with physical interpretability. In *SAE Technical Paper*, 2021-24-0082. doi:10.4271/2021-24-0082.
- Shahpouri, S., Gordon, D., Shahbakhti, M., and Koch, C.R. (2024). Transient nox emission modeling of a hydrogen-diesel engine using hybrid machine learning methods. *International Journal of Engine Research*, 25(12), 2249–2266.
- Shin, S., Kim, M., Park, J., Lee, S., and Min, K. (2023). Task transfer learning for prediction of transient nitrogen oxides, soot, and total hydrocarbon emissions of a diesel engine. *IEEE Access*, 11, 72462–72476.
- Tadros, M. and Boulougouris, E. (2025). Calibration procedures for no_x emissions model of a high-speed marine diesel engine using optimization procedures. *Journal of Marine Science and Engineering*, 13. doi: 10.3390/jmse13081585.
- Tan, E., Algar, S., Corrêa, D., Small, M., Stemler, T., and Walker, D. (2023). Selecting embedding delays: An overview of embedding techniques and a new method using persistent homology. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 33(3).
- Tofigh, M., Fakouri Hasanabadi, M., Smith, D., Kharazmi, A., Hanifi, A.R., Koch, C.R., and Shahbakhti, M. (2025). Control-oriented modeling of a solid oxide fuel cell affected by redox cycling using a novel deep learning approach. *Journal of Dynamic Systems, Measurement, and Control*, 147(2), 021006.
- Tofigh, M., Kharazmi, A., Smith, D.J., Koch, C.R., and Shahbakhti, M. (2024a). Temporal dilated convolution and nonlinear autoregressive network for predicting solid oxide fuel cell performance. *Engineering Applications of Artificial Intelligence*, 136, 108994.
- Tofigh, M., Salehi, Z., Smith, D., Kharazmi, A., Hanifi, A., Koch, C., and Shahbakhti, M. (2024b). Developing a time-efficient model for solid oxide fuel cells using self-supervised convolutional autoencoder and stateful lstm network. In *2024 American Control Conference (ACC)*, 86–91. IEEE.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
- ÖZENER, O., YÜKSEK, L., and ÖZKAN, M. (2013). Artificial neural network approach to predicting engine-out emissions and performance parameters of a turbo charged diesel engine. *Thermal Science*, 17(1), 153–166. doi:10.2298/TSCI1203212200.