Enhancing Railway Safety: A Machine Learning Approach for Automated Detection of Missing Track Bolts

Aayush Damai,[1] Hwapyeong Song,[2] Husnu S. Narman, Ph.D.,[3] Alexander Lambert[4], and Ammar Alzarrad, Ph.D.[5]

[1]Marshall University, Department of Computer Sciences and Electrical Engineering, 1 John Marshall Drive, Huntington, WV 25755; e-mail: damai2@marshall.edu
[2]Marshall University, Department of Computer Sciences and Electrical Engineering, 1 John Marshall Drive, Huntington, WV 25755; e-mail: song24@marshall.edu
[3]Marshall University, Department of Computer Sciences and Electrical Engineering, 1 John Marshall Drive, Huntington, WV 25755; e-mail: narman@marshall.edu
[4]Marshall University, Department of Computer Sciences and Electrical Engineering, 1 John Marshall Drive, Huntington, WV 25755; e-mail: lambert232@marshall.edu
[5]Marshall University, Department of Civil Engineering, 1 John Marshall Drive, Huntington, WV 25755; e-mail: alzarrad@marshall.edu

ABSTRACT

Railways are a critical component of transportation infrastructure, enduring significant physical stress daily due to the massive weight of trains and their cargo. This constant use can lead to the wear and tear of railway components, posing safety risks. One essential part of the railway infrastructure is the track bolts, which secure the rails in place, ensuring safe train traversal and preventing derailments that could cause severe injuries to passengers. This study investigates a machine learning algorithm approach to detect missing track bolts from image and video data automatically. The aim is to develop an algorithm that accurately identifies missing bolts on railways, thereby mitigating safety risks associated with poorly maintained tracks and streamlining maintenance processes. Therefore, various machine learning algorithms are evaluated using a specially curated dataset to detect missing track bolts to provide more efficient tool for detecting missing track bolts, contributing to railway operations' reliability and safety.

1. INTRODUCTION

Railways are used everywhere worldwide to transport large amounts of freight and passengers. Increased usage has resulted in constant stress to the railroad tracks caused, making the risk of railway parts breaking down inevitable. Track bolts are at risk of falling out completely. Missing bolts is a common condition that has a considerable impact on safety. Identifying the defects can be strenuous. However, with advancements in object detection and machine learning, potential ways to detect missing track bolts with high accuracy have been developed. Research into the innovative approach is essential to increasing rail transport safety for freight and passengers.

A growing body of literature implements machine learning models to address the missing bolt detection issue. Marino et al. developed an algorithm to detect missing bolts in railway tracks with 95% effectiveness using an exhaustive search method (Marino, et al. 2007). While this approach was impractical for widespread usage, it proved the potential for future research, including this study, which will incorporate more sophisticated detection techniques. Liu et al. developed an algorithm capable of detecting missing fastening bolts with 99.96% accuracy at speeds of roughly 62 mph using the gradient orientation co-occurrence matrix method and a hierarchical detection framework (Liu, Zhou and He 2016). This research also shows that the potential for large-scale detection of railroad flaws is possible. Chandran et al. employed a Convolutional Neural Network (CNN) and ResNet-50 algorithms, achieving an accuracy rate of 98% in identifying missing fasteners (Chandran, et al. 2021). Their findings prove that CNNs can be an effective model for detecting missing parts.

Recently, considerable literature has been written about utilizing the You Only Look Once (YOLO) model, designed for object detection using a CNN. Fu et al. changed the architecture of YOLOv4 by replacing the CSPDarknet53 backbone with MobileNet and developing MobileNet-YOLOv4, which has a false positive rate of approximately 5% (Fu, Chen and Lv 2022). Li et al. developed an algorithm using the YOLOv5 model with an impressive average mean precision of 97.4% (Li, et al. 2023).

Similarly, others have highlighted the relevance of identifying small objects. Tang et al. refined the YOLOv5 model to better detect small objects by utilizing higher resolution (Tang, Zhang and Fang 2024), which is particularly relevant for identifying small railway components like bolts and fasteners. Wang et al. enhanced the YOLO algorithm to version 9 (YOLOv9) (Wang, Yeh and Liao 2024) (Wang, Yeh and Mark Liao 2025). They improved object detection accuracy for items like bolts by incorporating programmable gradient information; this helps reduce data loss when using deep learning algorithms. Wang et al. developed further improvements to create YOLOv10 (Wang, et al. 2024). This version preserves the accuracy of YOLO models at significantly reduced parameter counts, thereby increasing overall performance compared to its predecessors. Together, these studies provide important insights into how YOLO models can be an effective alternative for missing bolt detection.

This research goal is to increase the effectiveness of rail maintenance by comparing recent developed object detection models in YOLO model while detecting missing track bolts in joint bars. Different versions of the YOLO model, selected based on previous literature studies, will be explored in more detail. Although extensive research has been conducted on missing bolts, single studies comparing different YOLO models on this issue are limited. This study aims to contribute to this growing area of research by exploring this pioneering model. Implementing this technological advancement creates space for innovative devices to collect rail track data using computer vision and object detection techniques. According to the results of this research, the YOLO model can automatically detect missing bolts with high accuracy, which helps in rapid maintenance to ensure the safety of the railway.

The following sections of this paper are as follows: Section 2describes the used YOLO models used. Section 3 explains the methodology used for this study. Section 4 presents the results and interpretations to evaluate the performance of each model. Finally, Section 5 summarizes and concludes the findings of this study.

2. YOLO MODELS

YOLO is a real-time object detection model that uses a CNN to “look” at an image once and then detect or identify objects in the image. This scanning gives it a massive speed boost over other object detection models and makes it more efficient. This algorithm works by splitting the image into a grid and then tries to predict the likelihood that the object is in those boxes.

YOLO has been evolving through multiple iterations, from YOLOv1 to YOLOv11. Each iteration builds on the previous version. The new model may exhibit enhanced object detection performance. Conversely, in the case of missing track bolts, where the object's size is small, newer versions might not perform as well as older ones as they have not been as optimized for small objects as the previous versions. In addition to the various models, each mode has sub-versions. Multiple sub-versions exist that scale from tiny or nano, designed for speed and to be used with less powerful hardware, to extensive, designed for more powerful hardware. Consequently, this study focuses on four YOLO models: YOLOv5, YOLOv8, YOLOv9, and YOLOv10.

A. YOLOv5

YOLOv5 is the fifth iteration of the YOLO series (Jocher 2020). It added many new changes to its architecture. One of the changes to YOLOv5 was to the architecture's backbone, utilizing an advanced version of the CSP-Darknet53. This version was later updated to the YOLOv5u series, which introduced some features from YOLOv8, such as the anchor-free split head in YOLOv5. All our tests with YOLOv5 were done with the YOLOv5u versions.

B. YOLOv8

The eighth version of the YOLO series, YOLOv8 introduced the anchor-free and objectness-free split-head (Jocher, Chaurasia and Qiu 2023). These allow the predictions to be made with fewer computations, making them quick and efficient while not missing out on accuracy. It also had advanced backbone and neck architectures, which refined the model's feature extraction performance.

C. YOLOv9

YOLOv9 introduced two new significant techniques: Programmable Gradient Information (PGI) PGI and Generalized Efficient Layer Aggregation Network (GELAN) (Wang, Yeh and Liao 2024) (Wang, Yeh and Mark Liao 2025). PGI helps reduce data loss when using deep learning algorithms, which was a previous issue. GELAN helped YOLOv9 to be more computationally efficient.

D. YOLOv10

YOLOv10 changes the architecture using an updated CSPNet version as its backbone (Wang, et al. 2024). This change increases the computation and gradient flow. It also allowed for reduced latency by utilizing a non-maximum suppression-free method. However, it might struggle with small object detection more due to a reduced number of parameters.

3. METHODOLOGY

Quantitative analysis is a well-established approach in research using machine learning techniques. This study uses the approach to identify models that effectively detect missing bolts, focusing on four models of the YOLO algorithm: YOLOv5, YOLOv8, YOLOv9, and YOLOv10.

Ultralytics provides command-line and programming interfaces for training, validating, and predicting on many versions of YOLO. This open-source framework and its pre-trained models were used for testing. For YOLOv8, pre-trained models on Common Objects in Context

(COCO) (Lin, et al. 2014) and Open Images v7 datasets (Kuznetsova, et al. 2020) were used for testing.

A. Dataset

Developing a good dataset to train a model is crucial for tuning the model and ensuring that the images used are varied in distinctive characteristics, such as the location of the missing bolt or orientation of the image. Eunus et al. created a dataset of 384 images to train models for detecting cracks in railroad tracks and missing bolts (Eunus, et al. 2024). Technofly Solution compiled a dataset of 1,075 images featuring various angles and types of cracks for a computer vision project aimed at railway track crack detection (Technofly Solution 2022). These included some images of rail joint bars with missing bolts. Ranganath assembled a dataset of 380 images showing various kinds and angles of cracks in railroad tracks. (Ranganath 2023) These also included some images of rail joint bars with missing bolts. System Thinking Project released a dataset of 717 images for finding railway defects, including joint bar cracks and missing or broken bolts (System Thinking Project 2024).

After re-labeling and combining proper rail joint bar images from the datasets, 191 images were split into 39 testing and 152 training. The validation images consist of 73 rail joint bar images, bringing the dataset's image count to 264. The dataset also did not include any images with true negatives; therefore, the column was removed from the table as all the results were zero. In future work, we will increase the dataset by collecting data from the field.

B. Training Parameters

The models were trained using 500 max epochs, a patience 100, an image size of 640x480, and batch size 2. Each model was trained on the same dataset with the same parameters. Epochs are complete passes through the training dataset. Patience is a training time optimization metric, allowing one to set how many epochs to go without improvement before stopping the training. Image size is the size of images passed through the dataset in pixels. Batch size is how many image samples the model processes each iteration. A larger batch size uses more GPU video memory, affecting only the training time and not the results.

C. Model Comparison Metrics

Ultralytics generates validation metrics at the end of each training. These metrics include normalized and basic confusion matrices, F1-confidence curves, precision-confidence curves, precision-recall curves, and recall-confidence curves. More metrics are collected at each epoch over the training period: box loss, classification loss, and distribution focal loss are collected for training and validation, and more results can be obtained for mAP50 (mean Average Precision), mAP50-95, precision, and recall. Some training and validation image batches are generated with actual and predicted labels.

Precision, Recall, and F1 Score, the core indicators of the classification model, were used for objective performance evaluation. Precision is the ratio of positive predictions to actual positives (true and false positives), measuring how well an object is anticipated. Higher precision minimizes false positives. On the other hand, Recall is the rate at which an object that is actually positive is predicted as positive, which is closely related to false negative. In this study, a false negative is a prediction that a bolt is predicted as not missing, but it is actually missing. Higher recall minimizes the false negatives. The F1 Score complements the trade-off characteristics of these two indicators.

The researchers aim to minimize false negatives in the context of missing bolts. However, a result with more false negatives may be preferable. For example, if a dataset has 90 total images and model validation results in 45 true positives (TP), 45 false positives (FP), and zero false negatives (FN), a similar result can be gathered by declaring a positive result for every image. Therefore, judging solely based on the number of false negatives is unreasonable.

4. RESULTS

This section shows the efficiency of different YOLOv5, YOLOv8, YOLOv9, and YOLOv10 models. The models' TP, FP, FN, precision, recall, and F1 scores were recorded and compared using the dataset and parameters shown in the methods sections.

Table 1 shows the TP, FP, FN, precision, recall, and F1 for the YOLOv5 model. The best scores for each metric are bolded in the table.

Table 1: Results of the YOLOv5 Model Tests

Model	TP	FP	FN	Precision	Recall	F1
YOLOv5nu	62	17	12	0.7848	0.8378	0.8105
YOLOv5n6u	62	12	12	0.8378	0.8378	0.8378
YOLOv5su	61	11	13	**0.8472**	0.8243	0.8356
YOLOv5s6u	61	14	13	0.8133	0.8243	0.8188
YOLOv5mu	62	18	12	0.7750	0.8378	0.8052
YOLOv5m6u	65	19	9	0.7738	0.8784	0.8228
YOLOv5lu	**67**	14	**7**	0.8272	**0.9054**	**0.8645**
YOLOv5l6u	62	24	12	0.7209	0.8378	0.7750
YOLOv5xu	64	27	10	0.7033	0.8649	0.7758
YOLOv5x6u	61	**11**	13	**0.8472**	0.8243	0.8356

According to our test results, among the variants of the YOLOv5 model, YOLOv5lu showed the best performance. It has the highest F1 score, the highest recall score, and the fourth-highest precision score. This study prioritized recall scores in the context of missing bolts because it desired the highest number of true positives possible. The confusion matrix (see Figure 1) shows a high number of true positives (67) alongside a slightly high number of false negatives (14). This brings its score down slightly, but the small number of false negatives helps increase its recall score, 0.9054.

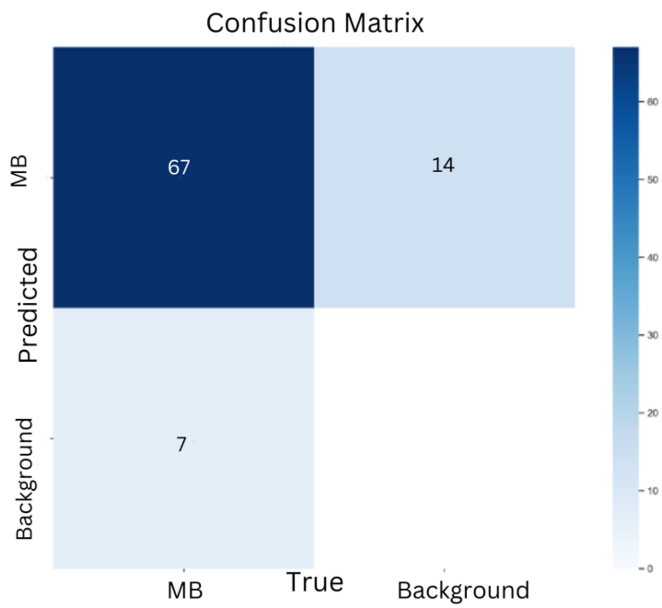

Figure 1. The Confusion Matrix for YOLOv5lu

Table 2 shows the results for YOLOv8. The best model would be either YOLOv8x or YOLOv8m-oiv7. Both models perform well, with YOLOv8m-oiv7 outperforming YOLOv8x in TP, FN, and recall. As stated before, this study prioritized the highest amount of TP and a high recall score. In these tests, YOLOv8m-oiv7 has a high TP rate at 66, only one less than the highest recorded value, and has the highest recall score, 0.8919, which is only slightly lower than the highest recorded value, 0.9054. Those scores give YOLOv8m-oiv7 a slight edge over YOLOv8x.

Table 2: Results of the YOLOv8 Model Tests

Model	TP	FP	FN	Precision	Recall	F1
YOLOv8n	64	17	10	0.7901	0.8649	0.8258
YOLOv8n-oiv7	59	17	15	0.7763	0.7973	0.7867
YOLOv8s	66	14	**8**	0.8250	**0.8919**	0.8571
YOLOv8s-oiv7	57	8	17	0.8769	0.7703	0.8201
YOLOv8m	64	20	10	0.7619	0.8649	0.8101
YOLOv8m-oiv7	**66**	12	**8**	0.8462	**0.8919**	0.8684
YOLOv8l	**66**	17	**8**	0.7952	**0.8919**	0.8408
YOLOv8l-oiv7	62	12	12	0.8378	0.8378	0.8378
YOLOv8x	65	**7**	9	**0.9028**	0.8784	**0.8904**
YOLOv8x-oiv7	62	25	12	0.7126	0.8378	0.7702

Table 3 shows the performances of the best YOLOv9 models. It is split between YOLOv9t and YOLOv9e. Both models share identical TP, FN, and recall scores. YOLOv9e has a slightly lower FP rate at 20 FP compared to 23 for YOLOv9t. It has a higher precision of 0.7701, which is slightly higher than YOLOv9t's precision of 0.7444. Finally, YOLOv9e also has a higher F1 score, which shows a better balance between precision and recall. Both models have the highest recall scores at 0.9054 and TP at 67, so choosing either would be good for the detection of missing track bolts.

Table 3: Results of the YOLOv9 Model Tests

Model	TP	FP	FN	Precision	Recall	F1
YOLOv9t	**67**	23	**7**	0.7444	**0.9054**	0.8171
YOLOv9s	65	**16**	9	**0.8025**	0.8784	**0.8387**
YOLOv9m	65	28	9	0.6989	0.8784	0.7784
YOLOv9c	66	18	8	0.7857	0.8919	0.8354
YOLOv9e	**67**	20	**7**	0.7701	**0.9054**	0.8323

Another metric to consider is the model's architecture. YOLOv9t or YOLOv9 tiny is a lightweight model with less power and speed. It has 2 million parameters. YOLOv9e or YOLOv9 extensive is the opposite, made to handle larger datasets and work on more powerful computers. It boasts higher precision and has 58.1 million parameters (Yaseen 2024). Despite the vast differences in these models' architecture, they perform similarly in looking for missing track bolts. If the focus is on enhancing performance speed using less power, YOLOv9t is the better option, while yielding lower precision and higher false positives. In cases where precision is a priority, YOLOv9t may be a more reasonable alternative.

Table 4 shows all of the YOLOv10 model results. Out of these models, YOLOv10l showed the best performance among the variants of YOLOv10. While YOLOv10m, YOLOv10b, and YOLOv10l all share the same recall score, 0.8378, YOLOv10l has the highest scores in each category. Its TP rate is 62, which is tied for the highest rate at 62 with two other models, YOLOv10m and YOLOv10b. Next, it has a remarkably low FP rate at 6 FP, which is the lowest recorded out of every YOLO model. YOLOv10l is also tied for the lowest FN rate at 12 with the previously mentioned two models. Due to its low FP number, the models' precision is exceptionally high at 0.9118, the highest out of all the YOLO models. The model's recall score is the highest out of the YOLOv10 models but is tied with the previous two other models. Finally, it has the highest F1 score at 0.8732, indicating a good balance between precision and recall.

Table 4: Results of the YOLOv10 Model Tests

Model	TP	FP	FN	Precision	Recall	F1
YOLOv10n	56	11	18	0.8358	0.7568	0.7943
YOLOv10s	55	10	19	0.8462	0.7432	0.7914

YOLOv10m	**62**	23	**12**	0.7294	**0.8378**	0.7799
YOLOv10b	**62**	26	**12**	0.7045	**0.8378**	0.7654
YOLOv10l	**62**	**6**	**12**	**0.9118**	**0.8378**	**0.8732**
YOLOv10x	61	12	13	0.8356	0.8243	0.8299

Looking for missing track bolts, YOLOv5lu, YOLOv9t, and YOLOv9e all display the highest values for TP (67) and recall scores (0.9054). So, all these models perform similarly in predicting TP. To decide which is better, we need to look at their precision and F1 curves. In precision, YOLOv5lu scores higher at 0.8272 compared to YOLOv9t and YOLOv9e, which scores 0.7444 and 0.7701, respectively. Lastly, with an F1 score, YOLOv5lu triumphs again with its score at 0.8645, performing better than YOLOv9t and YOLOv9e at 0.8171 and 0.8323, respectively, signifying that YOLOv5lu has a better balance of precision and recall. This shows that YOLOv5lu is the best model, displaying high TP and recall, alongside having a high precision and F1 score.

5. CONCLUSION

Our research shows the potential of employing machine learning algorithms to improve the safety of railways by detecting missing track bolts more efficiently, reliably, and securely. This research evaluated four YOLO models, YOLOv5, YOLOv8, YOLOv9, and YOLOv10, which exhibited incredible performance in detecting missing track bolts. YOLOv5lu had a recall value of 0.9054, showing the best performance in missing bolt detection among the tested models. Our findings show that machine learning models can effectively be utilized in railway defection detection, specifically for missing track bolts.

Future research will focus on assessing the model’s ability to detect when a missing bolt is absent to see how well these models’ accurate negative scores are. Also, testing out with models outside of the four. YOLOv11 was recently released and shown to have greater accuracy and better feature extraction. Other models outside the YOLO series, such as ResNet or Faster R-CNN, should also be considered as they have been used extensively in object detection. Explainable AI can also provide insights into the models’ behaviors and see how future research tunes the model for missing track bolt detection.

REFERENCES

Chandran, Praneeth, Johnny Asber, Florian Thiery, Johan Odelius, and Matti Rantatalo. (2021). "An investigation of railway fastener detection using image processing and augmented deep learning." *Sustainability* (MDPI) 13: 12051.

Eunus, Salman Ibne, Shahriar Hossain, A. E. M. Ridwan, Ashik Adnan, Md Saiful Islam, Dewan Ziaul Karim, Golam Rabiul Alam, and Jia Uddin. (2024). "ECARRNet: An Efficient LSTM-Based Ensembled Deep Neural Network Architecture for Railway Fault Detection." *AI* (MDPI) 5: 482–503.

Fu, Junpeng, Xingjie Chen, and Zhaomin Lv. (2022). "Rail fastener status detection based on MobileNet-YOLOv4." *Electronics* (MDPI) 11: 3677.

Jocher, Glenn. (2020). "Ultralytics/yolov5." *GitHub.* <https://github.com/ultralytics/yolov5> (Jan. 2, 2025).

Jocher, Glenn, A. Chaurasia, and J. Qiu. (2023). "YOLOv8 - Ultralytics YOLO Docs." *Ultralytics YOLO Docs.* <https://docs.ultralytics.com/models/yolov8/> (Jan. 2, 2025).

Kuznetsova, Alina, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, et al. (2020). "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale." *International journal of computer vision* (Springer) 128: 1956–1981.

Li, Xue, Quan Wang, Xinwen Yang, Kaiyun Wang, and Hongbing Zhang. (2023). "Track fastener defect detection model based on improved YOLOv5s." *Sensors* (MDPI) 23: 6457.

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. (2014). "Microsoft COCO: Common objects in context." *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13.* 740–755.

Liu, Liu, Fuqiang Zhou, and Yuzhu He. (2016). "Automated status inspection of fastening bolts on freight trains using a machine vision approach." *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit* (SAGE Publications Sage UK: London, England) 230: 1629–1641.

Marino, Francescomaria, Arcangelo Distante, Pier Luigi Mazzeo, and Ettore Stella. (2007). "A Real-Time Visual Inspection System for Railway Maintenance: Automatic Hexagonal-Headed Bolts Detection." *IEEE Trans. Syst. Man Cybern. Part C* 37: 418–428. doi:10.1109/TSMCC.2007.893278.

Ranganath, Chinmay. (2023). "Railway Track Fault Detection Classification dataset." *Roboflow Dataset.* <https://universe.roboflow.com/chinmay-ranganath-ohlji/railway-track-fault-detection> (Dec. 20, 2024).

System Thinking Project. (2024). "Damage detection in tracks object detection dataset and pre-trained model by System Thinking Project." *Roboflow Dataset* https://universe.roboflow.com/system-thinking-project-4rfww/damage-detection-in-tracks.

Tang, Shiyi, Shu Zhang, and Yini Fang. (2024). "HIC-YOLOv5: Improved YOLOv5 for small object detection." *2024 IEEE International Conference on Robotics and Automation (ICRA).* 6614–6619.

Technofly Solution . (2022). "RailwayTrackCrackDetection object detection dataset." *Roboflow Dataset* <https://universe.roboflow.com/technofly-solution/railwaytrackcrackdetection> (Dec.20, 2024)

Wang, Ao, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. (2024). "Yolov10: Real-time end-to-end object detection." *arXiv preprint arXiv:2405.14458.*

Wang, Chien-Yao, I.-Hau Yeh, and Hong-Yuan Mark Liao. (2025). "Yolov9: Learning what you want to learn using programmable gradient information." 1–21.

Wang, Chien-Yao, I.-Hau Yeh, and Hong-Yuan Mark Liao. (2024). "YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information." *Arxiv* <https://arxiv.org/abs/2402.13616> (Dec. 20, 2024).

Yaseen, Muhammad. (2024). "What is YOLOv9: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector." *arXiv preprint arXiv:2409.07813.*