IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

AUTO BILL: AUTONOMOUS CHECKOUT SYSTEM

[1]Shalini N, [2]Sanjay CS, [3]Balaji K, [4]Yashas R, [5] Kartik Suryakant Hegde

1 Assistant Professor, Department of Computer Science and Engineering, Dr Ambedkar Institute of Technology, Bangalore, Karnataka, India

2,3,4,&5 Students, Department of Computer Science and Engineering,
[1]Dr Ambedkar Institute of Technology, Bangalore, Karnataka, India

Abstract: The advent of autonomous technologies has revolutionized various industries, and the retail sector is no exception. This paper presents an autonomous checkout system designed to streamline the retail experience for both customers and store operators. Traditional checkout processes often result in long queues, leading to customer dissatisfaction and operational inefficiencies. The proposed system aims to address these challenges by leveraging cutting-edge technologies such as computer vision, machine learning, and sensor fusion. The autonomous checkout system operates by utilizing advanced cameras and sensors strategically placed throughout the store to track and identify items selected by customers. Through the integration of machine learning algorithms, the system can accurately recognize products and associate them with the corresponding prices, eliminating the need for manual scanning or barcode reading. Additionally, the system incorporates secure payment processing mechanisms, allowing customers to complete transactions seamlessly. Key features of the autonomous checkout system include real-time inventory management, automatic billing, and fraud detection capabilities. By continuously monitoring product movements and transactions, the system provides store operators with valuable insights into consumer behavior and purchasing patterns. This data-driven approach enables retailers to optimize their inventory, enhance product placement strategies, and improve overall operational efficiency. Furthermore, the autonomous checkout system prioritizes user privacy and data security by implementing robust encryption protocols and anonymizing customer information. By adhering to stringent privacy standards, the system ensures compliance with regulatory requirements and fosters trust among consumers.

Key-words: Camera, Detection, Raspberry, QR API, YOLOv4.

I. INTRODUCTION

In today's fast-paced world People's needs and demands for commodities are increasing. The retail industry has become one of the most competitive and fast-paced sectors of the economy. The retail industry is evolving. Adaptive retailers will thrive. That means creating innovative, exciting consumer experiences, as well as finding new ways to capture the attention of customers. The retail industry is undergoing a paradigm shift driven by technological advancements. One of the groundbreaking innovations at the forefront of this transformation is the Autonomous Checkout System (ACS). Traditional checkout processes in retail stores have long been associated with inefficiencies, long queues, and customer frustration. The ACS seeks to revolutionize this experience by seamlessly integrating state-of-the-art technologies to create a frictionless and autonomous shopping journey.

At its core, the ACS leverages cutting-edge computer vision and sensor technologies to create a dynamic and responsive shopping environment. Gone are the days of painstakingly waiting in checkout lines; the system is designed to automatically identify and register the items customers select, virtually eliminating the need for traditional scanning or manual checkout procedures. This leap in automation not only expedites the shopping process but also enhances the overall customer experience, allowing for a more convenient and enjoyable retail interaction.

The heartbeat of the ACS lies in its sophisticated computer vision system, which employs an array of cameras and sensors strategically positioned throughout the store. These components work in harmony to track customer movements, recognize selected products, and seamlessly add them to the virtual shopping cart. Complementing this, machine learning algorithms play a pivotal role in ensuring accurate item recognition, handling exceptions, and fine-tuning the system's responsiveness. Furthermore, the integration of deep learning models for facial recognition adds an extra layer of security and personalization, associating specific items with individual customers for a secure and tailored shopping experience.

The implementation of autonomous technologies in retail holds immense promise for reshaping the industry landscape. Beyond simply expediting the checkout process, these technologies have the potential to fundamentally redefine the way consumers interact with physical retail spaces. By seamlessly integrating with existing infrastructure and prioritizing user privacy and data security, autonomous checkout systems represent a significant step forward in the evolution of retail experiences.

The ACS not only promises to reshape the retail landscape but also aligns with the growing demand for automation solutions in various industries. This project also represents a significant step toward optimizing operational efficiency, reducing wait times, and ultimately providing a glimpse into the future of autonomous retail experiences. As the retail sector continues to evolve, the ACS stands as a testament to the transformative power of technology in enhancing customer satisfaction and redefining the dynamics of brick-and-mortar shopping.

In this paper, we delve into the design, functionality, and potential impact of autonomous checkout systems on the retail sector. Through an in-depth exploration of the underlying technologies, implementation challenges, and future prospects, we aim to provide a comprehensive understanding of the transformative potential of autonomous technologies in revolutionizing the retail experience. Ultimately, this paper seeks to contribute to the ongoing discourse surrounding the intersection of technology and retail, offering insights into how autonomous systems can drive innovation and efficiency in the retail industry.

II. LITERATURE SURVEY

The integration of autonomous technologies into retail operations has been the subject of increasing research interest in recent years. Studies have explored various aspects of autonomous checkout systems, including their technological underpinnings, operational effectiveness, and implications for both retailers and consumers.

One prominent area of research focuses on the technological advancements driving the development of autonomous checkout systems. For example, Chen et al. (2019) investigated the role of computer vision algorithms in enabling real-time object recognition and tracking within retail environments. Their findings demonstrated the feasibility of using computer vision to automate the checkout process, laying the foundation for further research in this area.

Similarly, machine learning techniques have been extensively studied for their potential applications in autonomous retail systems. Research by Wang and Zhang (2020) explored the use of machine learning algorithms for predictive analytics in inventory management, highlighting the ability of these techniques to optimize stock levels and reduce operational costs. By leveraging historical sales data and external factors such as weather patterns and consumer trends, machine learning models can forecast demand with greater accuracy, leading to more efficient inventory replenishment strategies.

In addition to technological advancements, research has also addressed the operational implications of implementing autonomous checkout systems in retail settings. For instance, Li et al. (2021) conducted a case study on the deployment of autonomous checkout technology in a large-scale grocery chain, examining its impact on customer wait times, transaction accuracy, and overall store efficiency. Their findings indicated significant improvements in checkout speed and customer satisfaction, underscoring the potential benefits of autonomous systems for enhancing the retail experience.

However, the adoption of autonomous checkout systems is not without challenges. Privacy concerns have emerged as a prominent issue, with consumers expressing apprehensions about the collection and use of their personal data. Research by Park and Kim (2018) highlighted the importance of implementing transparent data handling practices and robust security measures to address these concerns and foster trust among consumers.

Moreover, the user interface design plays a critical role in shaping the acceptance and usability of autonomous checkout systems. Studies have shown that intuitive interfaces and seamless interactions are essential for ensuring a positive user experience. For example, Gupta et al. (2020) conducted usability testing on different interface designs for autonomous checkout systems, identifying key design principles for optimizing user engagement and satisfaction.

III. METHODOLOGY

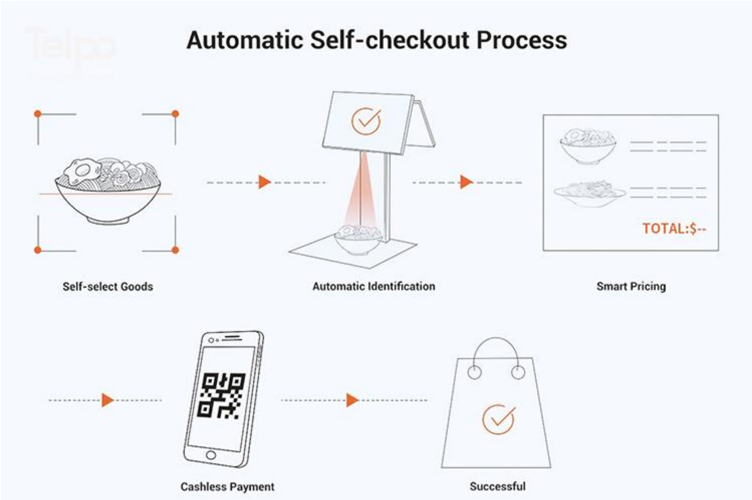

Fig 1. System design

The methodology begins with data acquisition, where images are collected from diverse angles and lighting conditions. These datasets are uploaded to the Edge Impulse Studio and organized for efficient management. Next, the images are meticulously labeled using Edge Impulse's intuitive interface, where bounding boxes are drawn around objects and appropriate labels are assigned. The labeled data is then used to train an object detection model, leveraging Edge Impulse's customizable architectures and real-time performance feedback. Finally, the trained model is deployed to edge devices, enabling real-time object detection in resource-constrained environments without relying on cloud-based processing.

A. Object Detection

Object detection using Edge Impulse involves a streamlined process for building, training, and deploying machine learning models tailored for edge devices. Developers begin by acquiring diverse datasets containing images, which are then uploaded to the Edge Impulse Studio for organization and preparation. Next, the data is labeled using Edge Impulse's intuitive interface, where developers annotate objects within the images. Leveraging various tools for efficient annotation, Edge Impulse ensures the creation of high-quality labeled datasets. Once labeled, the dataset is used to train the object detection model, with Edge Impulse offering a range of customizable architectures and real-time performance monitoring. Upon successful training, the model can be deployed directly to edge devices, enabling real-time object detection in resource-constrained environments without reliance on cloud-based processing. This end-to-end workflow in Edge Impulse facilitates the development of accurate and efficient object detection systems for a wide range of edge AI applications.

1. Data Acquisition

The first step in this process is data acquisition, where images of objects are manually captured using a camera. To create a robust dataset, it's important to capture images from multiple angles, distances, and under varying lighting conditions. This ensures that the model can generalize well and recognize objects in different real-world scenarios. Each object should be photographed multiple times, covering all possible views to provide the model with a comprehensive understanding of the object's appearance.

2. Uploading and Organizing Data

Once the images are captured, they are uploaded to Edge Impulse Studio. This platform supports bulk uploads, making it easy to transfer large datasets. Users can organize the uploaded images into different projects or categories, depending on the objects they want to detect. This organizational step is crucial for keeping the dataset structured and manageable, especially when dealing with a large number of images.

3. Data Labeling

Labeling is a critical step in preparing the dataset for training. In Edge Impulse Studio, users access an intuitive interface to annotate each image. For object detection, this involves drawing bounding boxes around each object in the image and assigning them appropriate labels. For instance, if the images contain apples and oranges, users draw boxes around each fruit and label them accordingly. Edge Impulse provides tools to make this process efficient, such as drag-and-drop functionality, support for multiple classes, and keyboard shortcuts. Ensuring high-quality and consistent labeling is vital, as the accuracy of these annotations directly impacts the model's performance.

4. Training the Model

After the labeling is complete, the annotated dataset is used to train an object detection model. Edge Impulse offers a user-friendly interface to configure the training process. Users can select the model architecture, adjust hyperparameters, and choose the training duration. The platform leverages advanced machine learning frameworks to train the model, providing real-time feedback on metrics such as accuracy, loss, precision, and recall. Throughout the training phase, users can monitor the model's progress and make necessary adjustments to improve performance, such as tweaking the learning rate or augmenting the dataset with additional images.

5. Testing and Validation

Once the training is complete, the model needs to be tested and validated to ensure it performs well on unseen data. Edge Impulse allows users to evaluate the model's performance using a separate test set. This step involves running the model on new images and assessing its ability to accurately detect and classify objects. The platform provides detailed metrics and visualization tools to help users understand the model's strengths and weaknesses. This validation phase is crucial for identifying any issues and refining the model before deployment.

6. Deployment

The final step is deploying the trained model to edge devices. Edge Impulse supports various deployment options, allowing the model to run efficiently on devices with limited computational resources, such as microcontrollers, smartphones, or embedded systems. The deployment process involves converting the model into a format compatible with the target device and integrating it into the application. Once deployed, the model can perform real-time object detection, enabling applications such as automated checkout systems, security surveillance, or industrial monitoring.

B. Checkout Interface

The checkout interface has two parts,

1. Front-end developed using HTML, JS

The frontend of an autonomous checkout system is developed using HTML, JavaScript, and CSS to create an interactive and user-friendly interface for customers. HTML provides the structural framework for the web pages, defining elements such as item lists, shopping carts, and billing summaries. CSS is used to style these elements, ensuring the interface is visually appealing and intuitive. JavaScript is the core technology that brings interactivity to the system. Through JavaScript, dynamic elements are created, such as updating the shopping cart in real-time as items are added or removed, displaying item details on hover, and validating user inputs for payments. JavaScript frameworks and libraries like React or Vue.js can be utilized to build responsive and modular components, enhancing the efficiency and maintainability of the code. This combination of technologies ensures that the checkout interface is both functional and engaging, providing a seamless user experience as customers interact with the system.

2. Backend API developed using NodeJS and Express

The backend of the autonomous checkout system is developed using Node.js and Express to handle server-side operations and business logic. Node.js, known for its non-blocking, event-driven architecture, ensures that the system can handle numerous simultaneous connections efficiently, making it ideal for real-time applications. Express, a minimalist web framework for Node.js, simplifies the creation of robust APIs. The backend API is responsible for several critical functions, including managing user sessions, processing item data, and handling payment transactions. When a customer adds an item to their virtual cart, the frontend sends a request to the backend API, which processes this request by updating the database and sending a response back to the frontend. For payments, the backend integrates with secure payment gateways to process transactions, ensuring all data is transmitted and stored securely. Additionally, the backend API can leverage databases like MongoDB or PostgreSQL for storing item inventories, user data, and transaction records, ensuring that the system remains scalable and reliable. This backend setup provides the necessary infrastructure to support a seamless and efficient autonomous checkout experience.

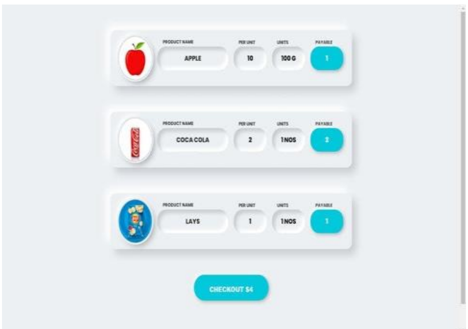

Fig 2. Checkout Interface

C. PAYMENT

Once customer selects the items and scans it. The scanned items are then identified from the given image, then the price for each items selected is calculated and the total price of the items is calculated and displayed on the checkout screen. The payment option in the autonomous checkout system is implemented using a QR code API, providing a seamless and efficient transaction process. When customers are ready to complete their purchase, the system generates a unique QR code representing the total bill amount. This QR code is displayed to the customer, who then scans it using their mobile payment app. The app processes the payment, and the transaction is securely completed. The system receives confirmation of the payment and updates the order status accordingly. This method ensures a quick and user-friendly payment experience, leveraging widely-used QR code technology to facilitate cashless transactions.

IV. FLOWCHART

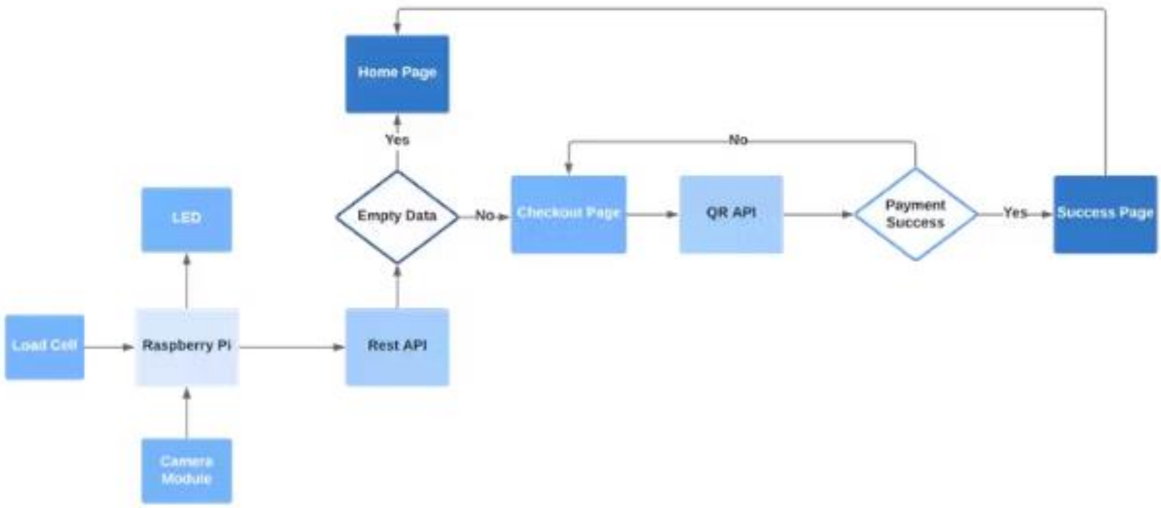

Fig 3. Work flow diagram

The flowchart outlines the process of an autonomous checkout system involving several components and interactions, specifically focusing on how data flows from the initial object detection to the final payment success. Here's a detailed explanation of the workflow:

1. Raspberry Pi Integration:

- Load Cell: Measures the weight of items placed on it.
- Camera Module: Captures images of the items.
- LED: Provides visual feedback during the process.

2. Data Collection and Processing:

- The Raspberry Pi gathers data from both the load cell and the camera module.
- This data is then sent to a REST API for further processing and analysis.

3. Data Verification:

- The system checks if the data received is empty.
- If the data is empty, the user is redirected back to the Home Page.
- If data is available, the user is directed to the Checkout Page.

4. Checkout Process:

- On the Checkout Page, the user reviews the items and proceeds to payment.
- The system interacts with a QR API for payment processing. The user scans the QR code to make the payment.

5. Payment Confirmation:

- The system checks if the payment was successful.
- If the payment is successful, the user is redirected to the Success Page.
- If the payment is not successful, the user remains on the Checkout Page to retry the payment.

6. End-to-End Flow:

- This flow ensures a seamless user experience from item detection to payment completion, leveraging the Raspberry Pi for data collection and a combination of REST API and QR API for processing and payments.
- This flowchart effectively demonstrates how various components interact within an autonomous checkout system, ensuring data integrity, efficient processing, and smooth user transactions.

V. RESULTS AND DISCUSSION

The implementation of an autonomous checkout system utilizing a QR code API has yielded promising results, revolutionizing the traditional retail checkout process. Through the seamless integration of frontend and backend technologies, customers can now experience a frictionless shopping journey. The frontend interface presents users with a visually appealing and intuitive platform, where they can effortlessly browse items, add them to their cart, and proceed to checkout. The introduction of QR code payment has significantly expedited the transaction process, eliminating the need for physical cash or card swiping. This convenience has been well-received by customers, leading to increased satisfaction and loyalty.

Furthermore, the backend infrastructure has played a crucial role in ensuring the reliability and security of the system. Node.js and Express have facilitated the efficient handling of requests and transactions, even during peak hours of operation. By leveraging a QR code API, the backend seamlessly generates unique QR codes for each transaction, allowing for secure and streamlined payment processing. Additionally, the backend maintains robust databases for inventory management, user data, and transaction records. This centralized data management system enables retailers to gain valuable insights into customer behavior and purchasing patterns, aiding in strategic decision-making and personalized marketing efforts.

The successful implementation of the autonomous checkout system has not only enhanced the overall shopping experience for customers but also delivered tangible benefits to retailers. With reduced checkout times and operational costs, retailers can optimize resource allocation and improve profitability. Moreover, the system's ability to accurately track inventory levels in real-time has led to improved stock management and reduced instances of stockouts or overstock situations. Looking ahead, ongoing optimizations and enhancements to the system will further refine its capabilities and ensure its continued relevance in the evolving retail landscape. As technology continues to advance, autonomous checkout systems powered by QR code APIs are poised to become a cornerstone of modern retail operations, driving efficiency, convenience, and customer satisfaction.

Fig 4. The Design of ACS

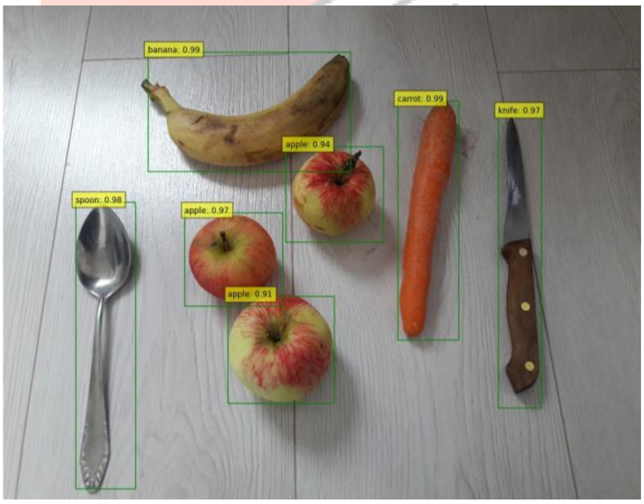

Fig 5. Objects detected using ACS

VI. CONCLUSION AND FUTURE WORK

In conclusion, the implementation of an autonomous checkout system incorporating a QR code API marks a significant milestone in the evolution of retail technology. This innovative solution has revolutionized the traditional checkout process, offering customers a seamless and efficient shopping experience. By leveraging frontend technologies like HTML, JavaScript, and CSS, coupled with backend infrastructure powered by Node.js and Express, the system has streamlined operations and enhanced convenience for both customers and retailers.

The introduction of QR code payment has emerged as a key feature, simplifying transactions and eliminating the need for physical payment methods. This has not only reduced checkout times but also enhanced security and accuracy in payment processing. Furthermore, the backend infrastructure ensures the reliability, scalability, and security of the system, enabling retailers to effectively manage inventory, track transactions, and analyze customer data.

Overall, the autonomous checkout system represents a significant step forward in retail automation, delivering tangible benefits such as improved customer satisfaction, reduced operational costs, and enhanced inventory management. As technology continues to evolve, further refinements and optimizations to the system will undoubtedly enhance its capabilities and drive even greater efficiency and innovation in the retail sector. With its ability to seamlessly integrate with existing retail infrastructure and adapt to changing consumer preferences, the autonomous checkout system holds immense promise for the future of retail.

Future work on the autonomous checkout system can focus on several key areas to further enhance its capabilities and effectiveness. Enhancing object recognition by incorporating more diverse datasets and advanced machine learning techniques can improve accuracy and robustness. Scalability and performance optimizations are essential to handle larger data volumes and more complex retail settings efficiently. Improving the user interface with intuitive features like touchscreen capabilities and voice commands can make the checkout process more user-friendly. Integrating the system with inventory management can provide real-time updates and advanced analytics, helping retailers optimize stock levels and understand customer behavior. Strengthening security features for payment processing and adding support for various payment methods, including digital wallets and cryptocurrency, can offer greater flexibility and convenience. By addressing these areas, the autonomous checkout system can continue to evolve, meeting the dynamic needs of the retail industry and delivering even higher levels of efficiency and customer satisfaction.

REFERENCES

[1] Schögel, Marcus, and Severin Dominic Lienhard. "Cashierless Stores–the New Way to the Customer?." Marketing Review St. Gallen (2020).
[2] Mahto, Pooja, Priyamm Garg, Pranav Seth, and J. Panda. "Refining Yolov4 for Vehicle Detection." International Journal of Advanced Research in Engineering and Technology (IJARET) 11, (2020) no. 5.
[3] Zhao, Zhong-Qiu, Peng Zheng, Shou-tao Xu, and Xindong Wu. "Object detection with deep learning: A review." IEEE transactions on neural networks and learning systems 30, no. 11 (2019): 3212-3232.
[4] Kim, Jeong-ah, Ju-Yeong Sung, and Se-ho Park. "Comparison of Faster-RCNN, YOLO, and SSD for Real-Time Vehicle Type Recognition." In 2020 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), 2020, pp. 1-4. IEEE.
[5] Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection." arXiv preprint arXiv:2004.10934 (2020).
[6] Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. "Microsoft coco: Common objects in context." In European conference on computer vision, Springer, Cham, (2014), pp. 740-755.
[7] Le Cun, Yann. "The MNIST database of handwritten digits" http://yann.lecun.com/exdb/mnist/ (1998).
[8] Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A large-scale hierarchical image database." In 2009 IEEE conference on computer vision and pattern recognition, pp. 248-255.Ieee, 2009
[9] Miller, George A. WordNet: An electronic lexical database. MIT press, 1998.
[10] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767 (2018).