**Optimizations to improve iTracker Performance.**

To optimize the iTracker model based on the challenges outlined, particularly the lack of computing power and restricted training duration, we can implement the following strategies:

1. **Increase Number of Epochs:** Addressing the restricted training duration, we can increase the number of epochs used during model training. By extending the training duration, the model will have more opportunities to learn from the dataset and capture complex data patterns. This extended training period can help mitigate the impact of limited computing resources and improve the model's performance metrics, such as reducing the Mean Square Error (MSE).
2. **Batch Processing:** Implementing batch processing techniques can help manage the training process more efficiently, especially with limited computing resources. By dividing the dataset into smaller batches, the model can be trained incrementally, reducing the strain on computational resources and enabling more extensive training sessions without overwhelming system capacity.
3. **Optimized Hyperparameters:** Fine-tuning the model's hyperparameters can enhance its efficiency and effectiveness, particularly when dealing with large datasets. By systematically adjusting parameters such as learning rate, batch size, and regularization techniques, we can optimize the model's performance and accelerate convergence, thereby mitigating the impact of limited computing power on training duration and overall effectiveness.
4. **Parallel Processing and Distributed Computing:** Leveraging parallel processing and distributed computing frameworks can significantly enhance the scalability and efficiency of model training, particularly for large datasets. By distributing computational tasks across multiple processors or nodes, we can expedite the training process and overcome limitations imposed by individual computing resources. This approach can enable more extensive training sessions and improve the model's ability to capture complex data patterns effectively.

By implementing these optimizations, we can enhance the iTracker model's performance and mitigate the impact of challenges related to computing power and restricted training duration. These strategies aim to maximize the efficiency and effectiveness of model training, ultimately improving the model's ability to capture complex data patterns and generate more accurate predictions.