

CSC 591 (145) - Homework 1

Priyanshu Malaviya - pmalavi

February 2025

GitHub Code Repository: <https://github.com/Priyanshu9898/GenAI-Homework1>

Section 1 - Conceptual Questions

Question 1

The authors mention two approaches to improve cache performance (hit rate). Which approach does the paper focus on?

Answer:

The paper focuses on improving cache performance by optimizing **cache replacement policies**. While prefetching (loading data before it is requested) is mentioned as another method, the authors prioritize replacement policies because existing heuristic-based strategies (e.g., LRU) perform poorly on complex memory access patterns. Their solution, Parrot, uses imitation learning to mimic Belady's optimal replacement policy.

Question 2

What are the previous state-of-the-art approaches mentioned in the paper?

Answer:

The paper discusses two main categories of previous approaches. The first includes traditional heuristic policies such as LRU (Least Recently Used) and MRU (Most Recently Used), which base eviction decisions on recent access history. The second category covers recent learning-based methods like Hawkeye and Glider, which employ machine learning techniques to predict whether a cache line is likely to be reused (i.e., cache-friendly versus cache-averse) and thereby guide the eviction process.

Question 3

Explain the cache hierarchy used for experimentation by the authors. This means you have to give the size of L1/L2/L3 caches including the number of sets and ways.

Answer:

The experiments are conducted on a three-level cache hierarchy:

- **L1 Cache:** A 32 KB cache that is 4-way set-associative. With a cache line size of 64 bytes, the number of sets is:

$$\frac{32 \text{ KB}}{4 \times 64 \text{ bytes}} = 128 \text{ sets}.$$

- **L2 Cache:** A 256 KB cache organized as an 8-way set-associative cache. This yields:

$$\frac{256 \text{ KB}}{8 \times 64 \text{ bytes}} = 512 \text{ sets}.$$

- **L3 Cache:** A 2 MB last-level cache with 16-way set-associativity. The number of sets is:

$$\frac{2 \text{ MB}}{16 \times 64 \text{ bytes}} = 2048 \text{ sets}.$$

Question 4

What is the normalized cache hit rate? Why have the authors used it as a metric?

Answer:

The normalized cache hit rate is defined by the formula:

$$NormalizedCacheHitRate = \frac{r - r_{LRU}}{r_{Belady} - r_{LRU}},$$

where r is the hit rate of the policy under evaluation, r_{LRU} is the hit rate achieved by the LRU policy, and r_{Belady} is the hit rate of Belady's optimal policy. This metric is used to standardize performance comparisons across different workloads by anchoring LRU at 0 and the optimal policy at 1, thereby clearly indicating how closely any given policy approaches the ideal performance.

Question 5

Interpret Figure 2 from the paper. Explain what you understand by looking at the figure and the caption.

Answer:

Figure 2 plots the normalized cache hit rate as a function of the lookahead window size—the number of future accesses considered by Belady's algorithm. The x-axis represents the window size, and the y-axis shows the normalized hit rate. The figure demonstrates that to achieve approximately 80% of Belady's optimal performance, the algorithm must consider around 2600 future accesses. This underscores the challenge of approximating the optimal replacement decision since even a large amount of future information only partially bridges the gap to the ideal.

Question 6

What formula might be used to calculate the values in the last column?

Answer:

The paper uses the formula:

$$CacheMisses = TotalAccesses - (CacheHitRate \times TotalAccesses)$$

to estimate the total number of cache misses. For instance, if there are 5000 memory accesses and the cache hit rate is 90%, then the number of misses would be calculated as:

$$5000 - (0.90 \times 5000) = 5000 - 4500 = 500.$$

This formula subtracts the number of hits (the product of the hit rate and total accesses) from the total number of accesses to derive the number of misses.

Section 2 - Belady's Policy and LRU

Task 1 - LRU

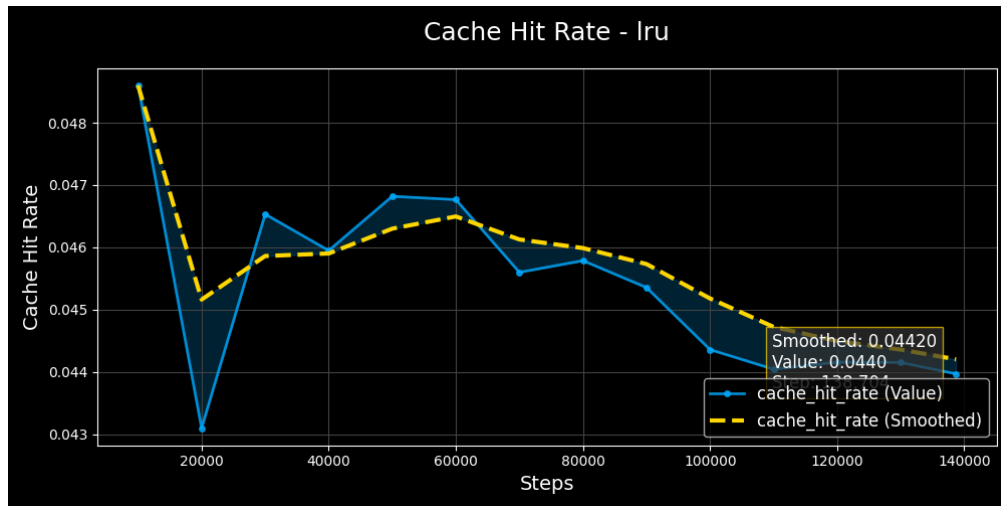


Figure 1: Cache hit rate analysis for LRU algorithm

Task 2 - Belady

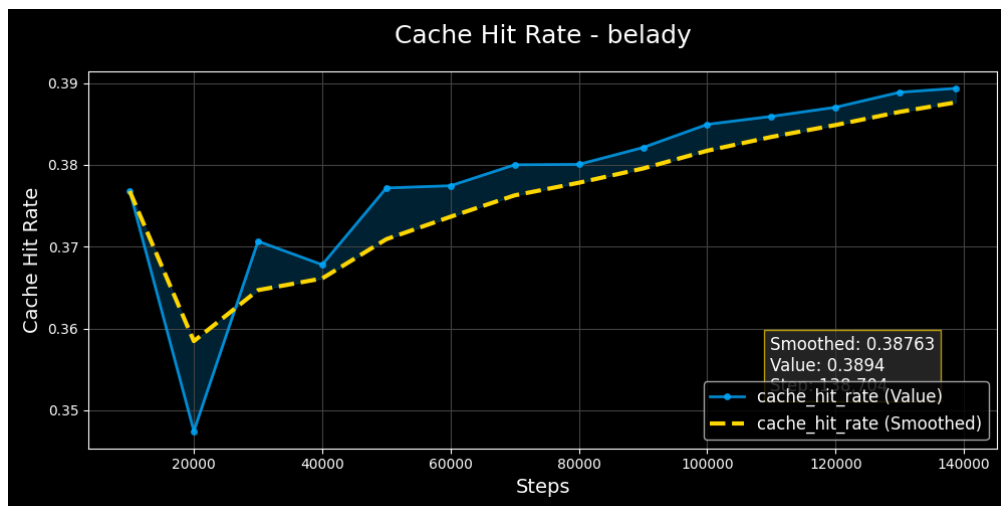
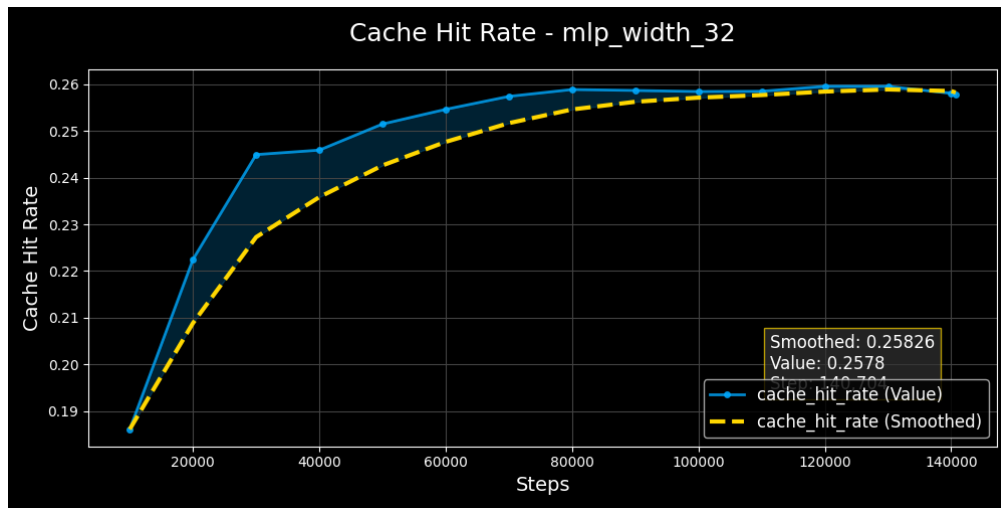


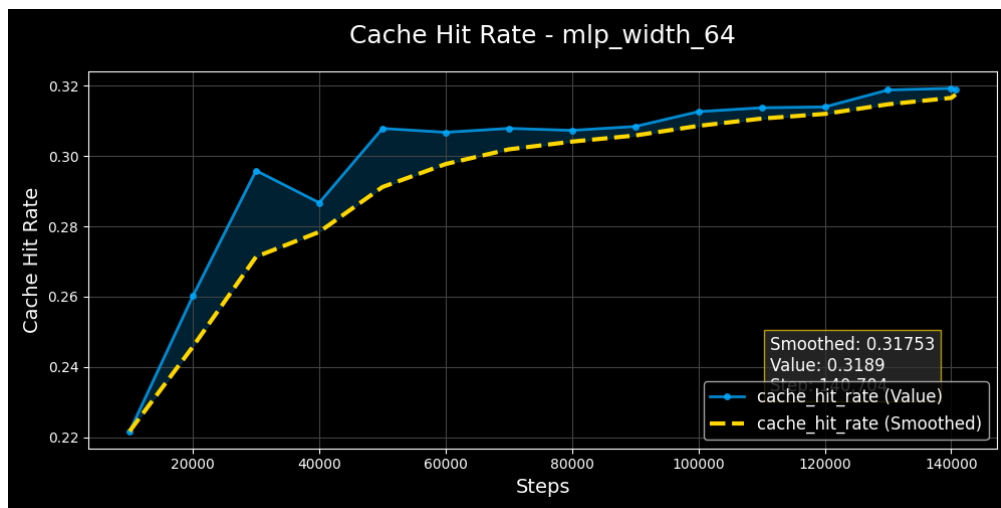
Figure 2: Cache hit rate analysis for Belady's algorithm

Section 3 - MLP

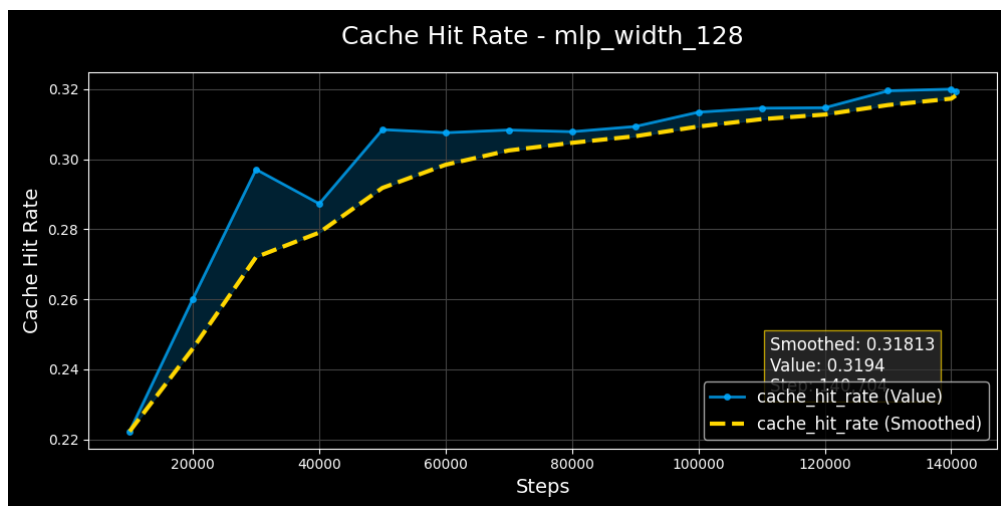
Task 1 - Hidden Layer Size



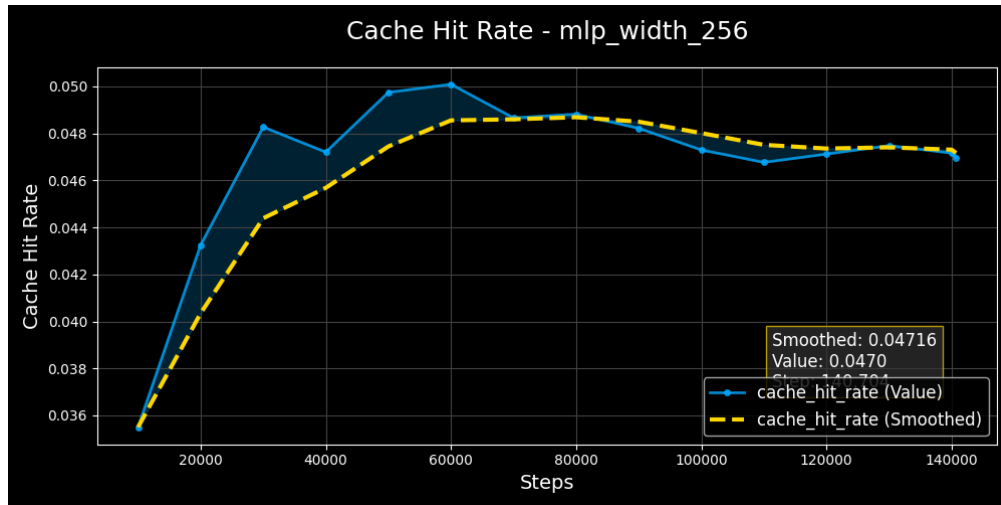
(a) 32 Neurons



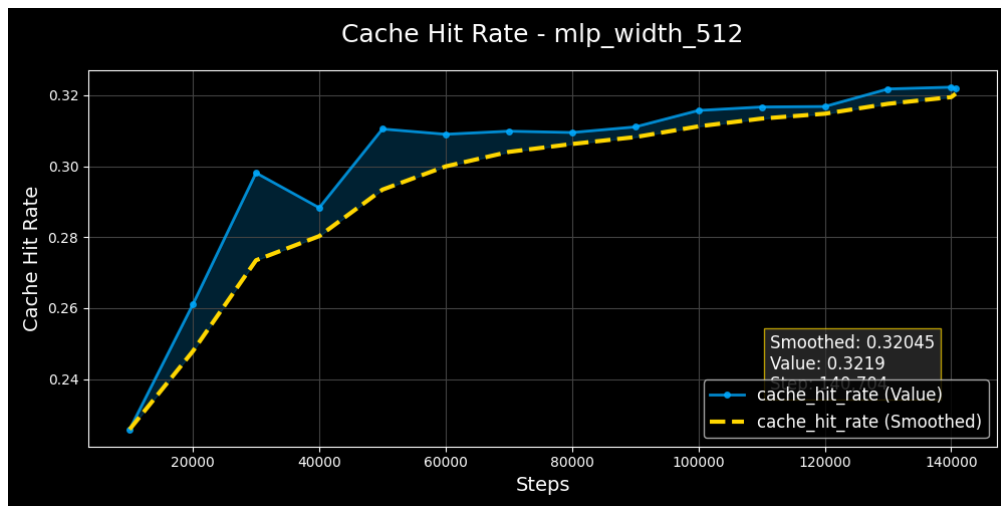
(b) 64 Neurons



(c) 128 Neurons



(a) 256 Neurons



(b) 512 Neurons

Figure 3: Effect of different numbers of neurons on cache hit rate

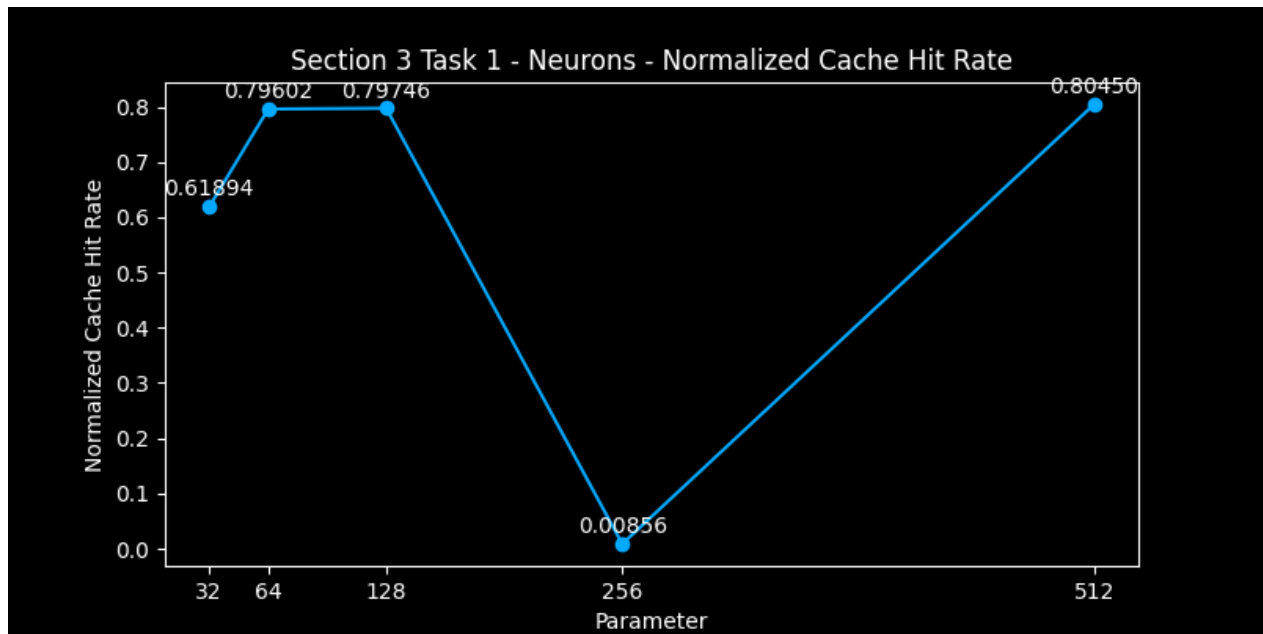
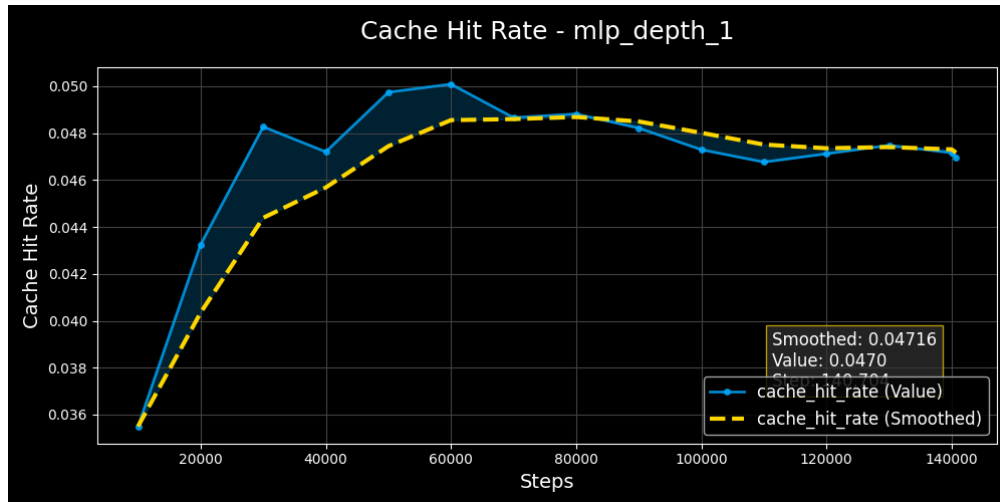
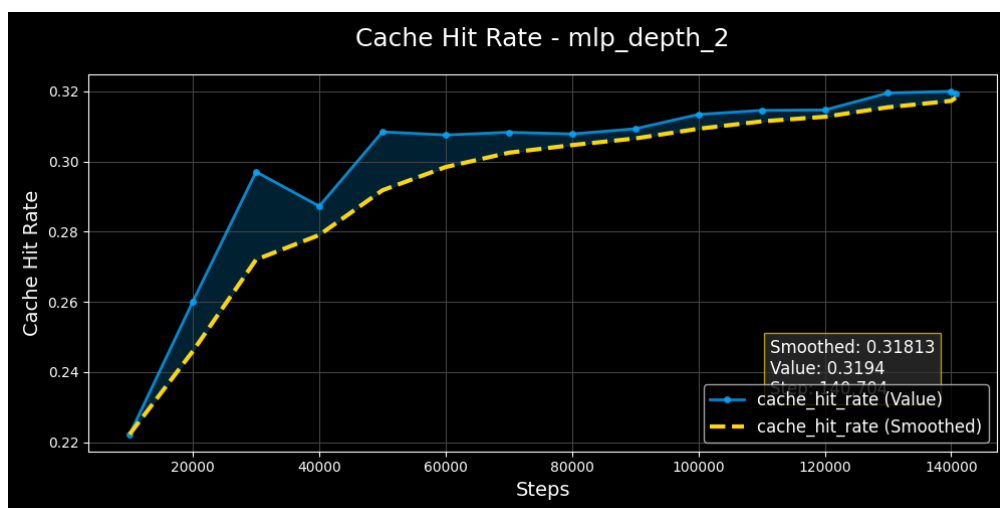


Figure 4: Cache Hit Rate vs. Number of Neurons

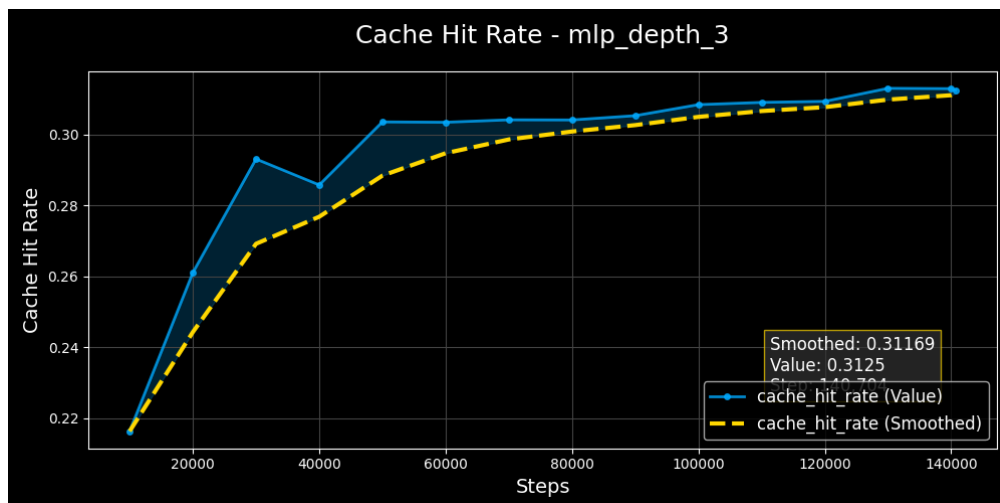
Task 2 - Number of Hidden Layers



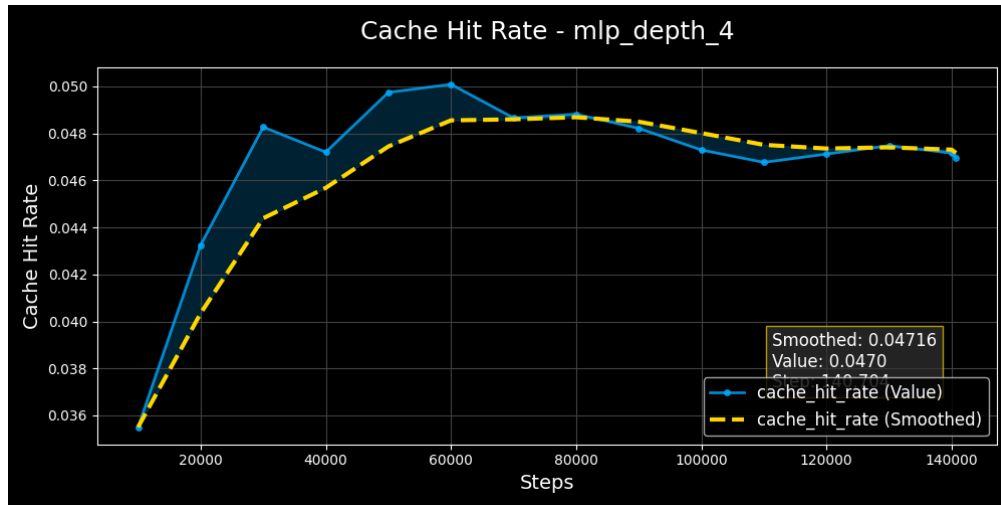
(a) Layer 1



(b) Layer 2



(c) Layer 3



(a) Layer 4

Figure 5: Comparison of cache hit rate for different layer depths

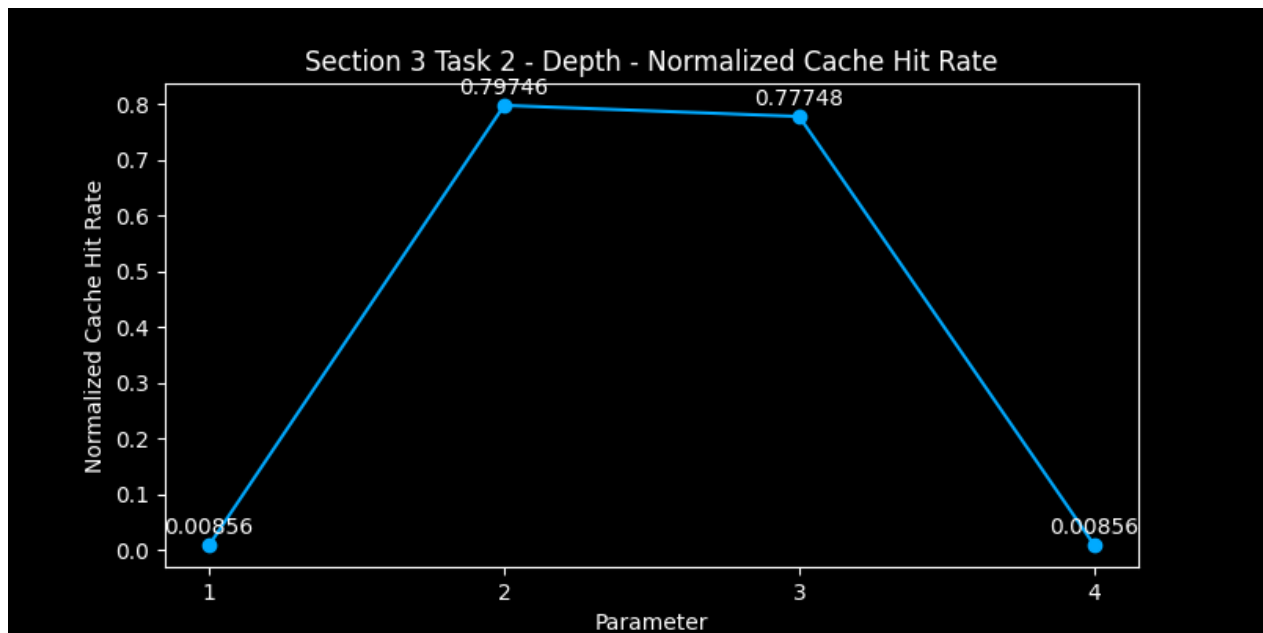
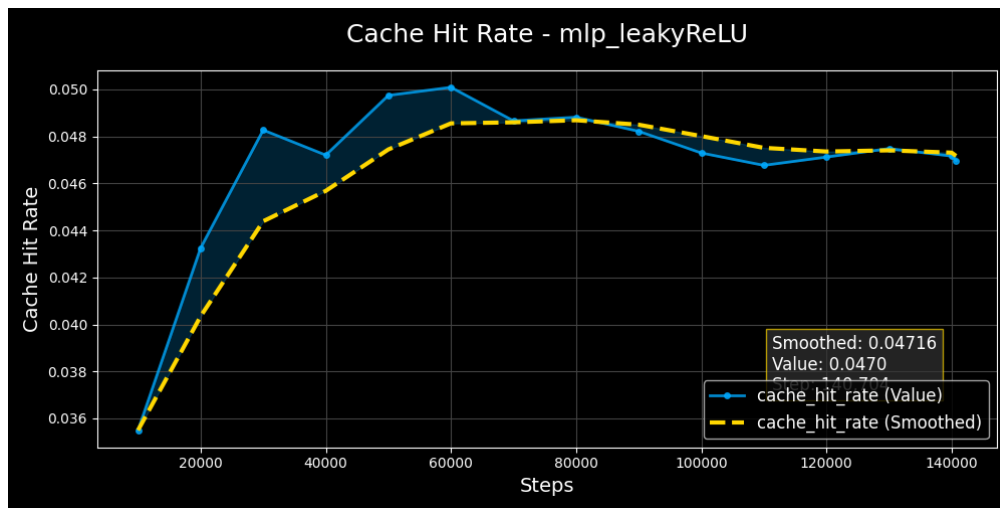
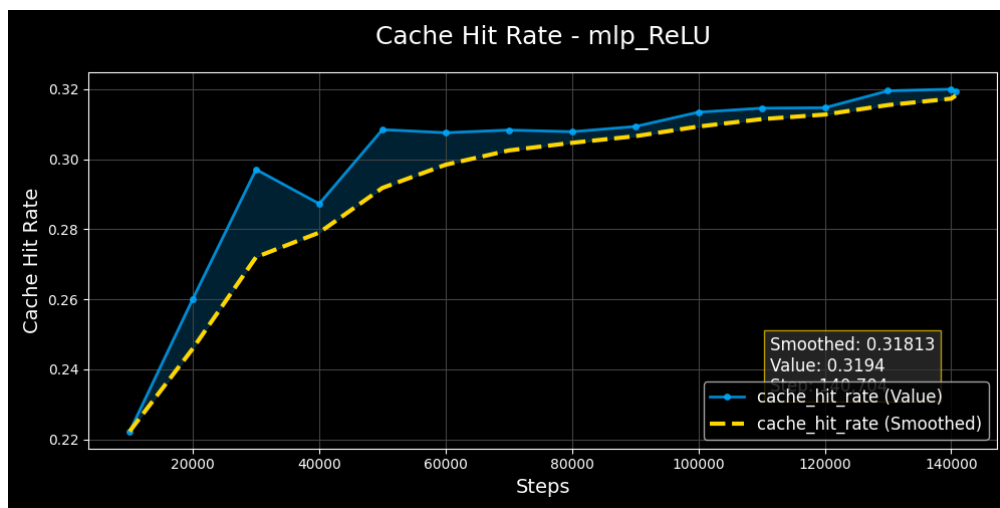


Figure 6: Cache Hit Rate vs. Number of Layers

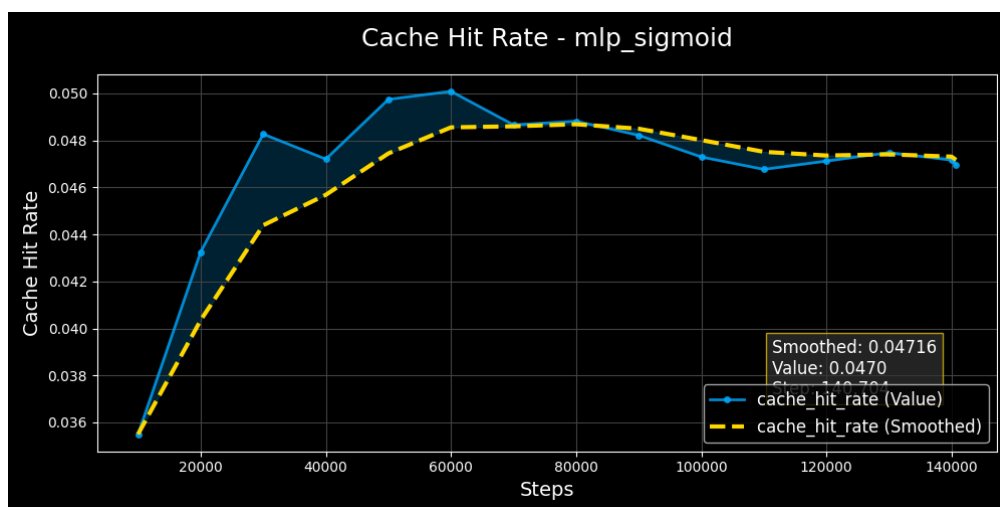
Task 3 - Activation Functions



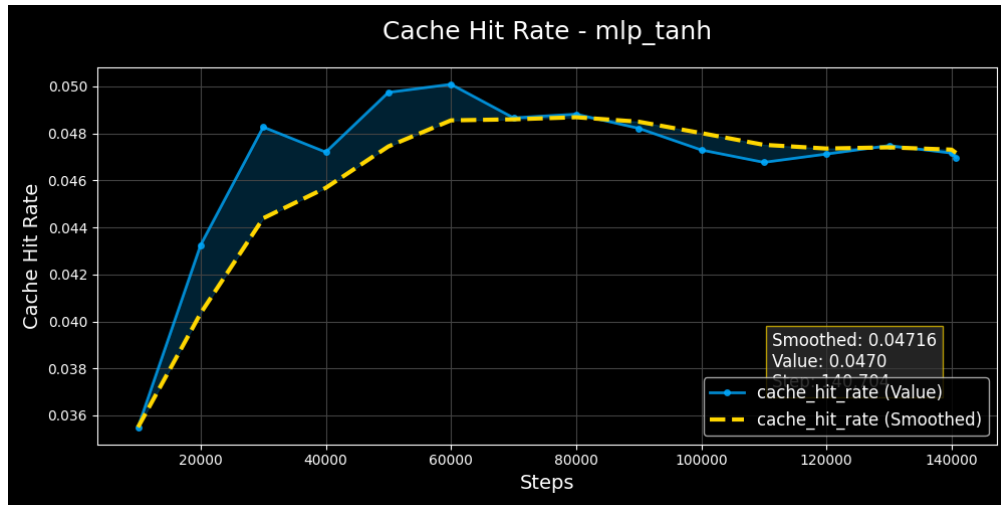
(a) Leaky ReLU



(b) ReLU



(c) Sigmoid



(a) Tanh

Figure 7: Comparison of activation functions on cache performance

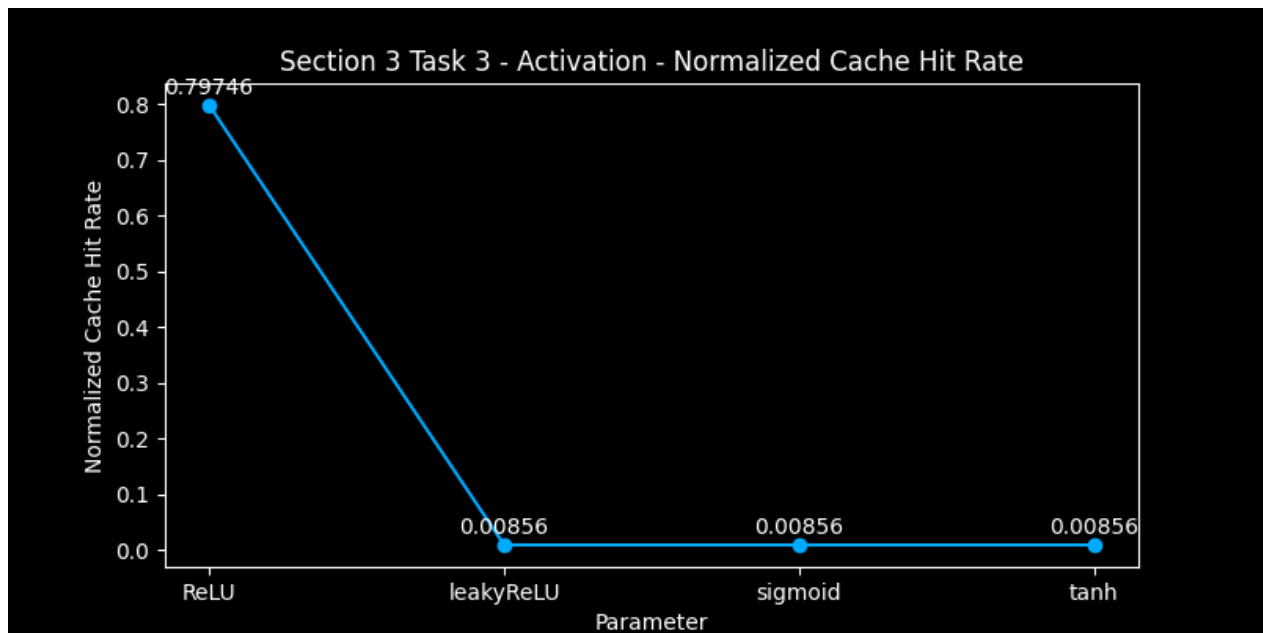
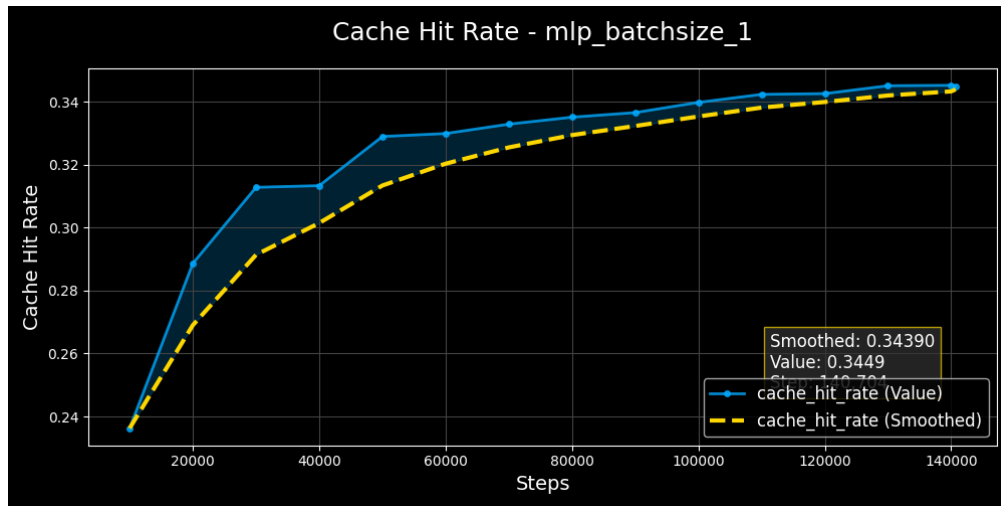
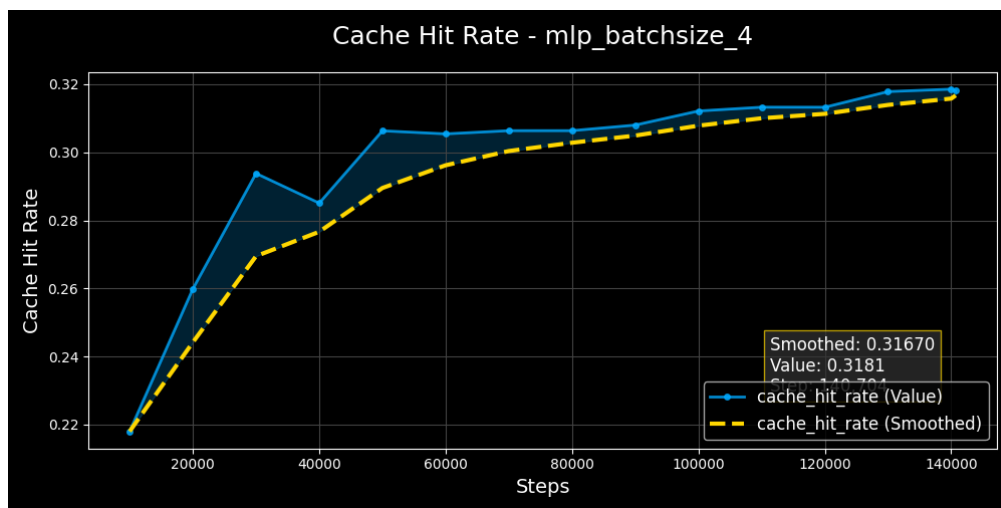


Figure 8: Normalized Cache Hit Rate vs. Activation Function

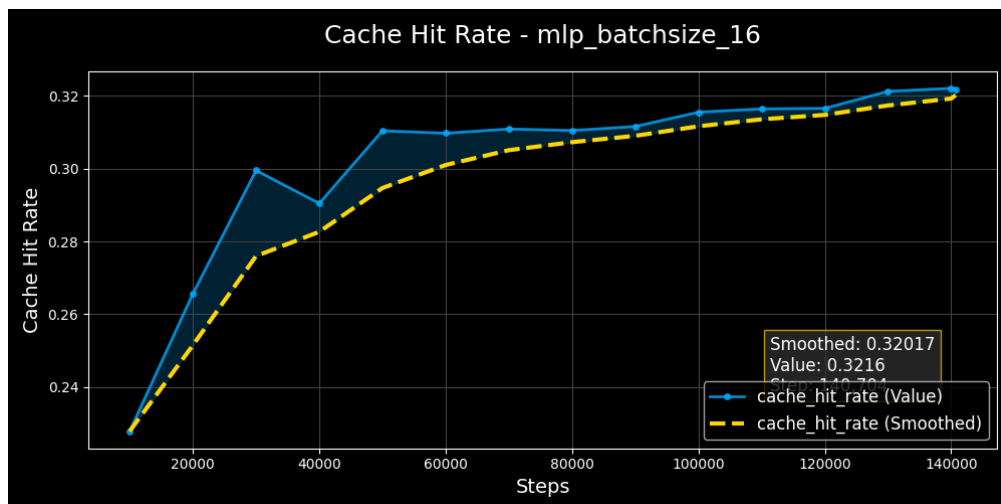
Task 4 - Batch Size



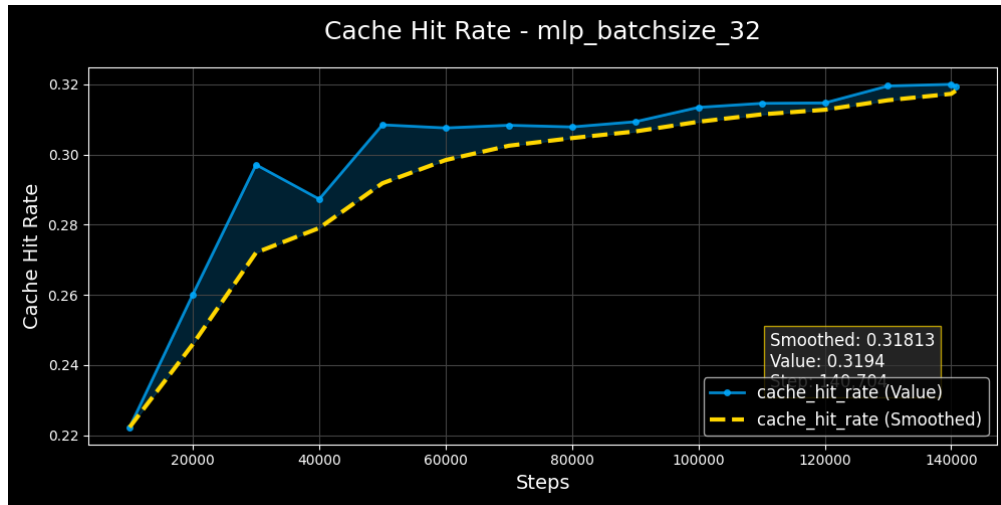
(a) Batch Size 1



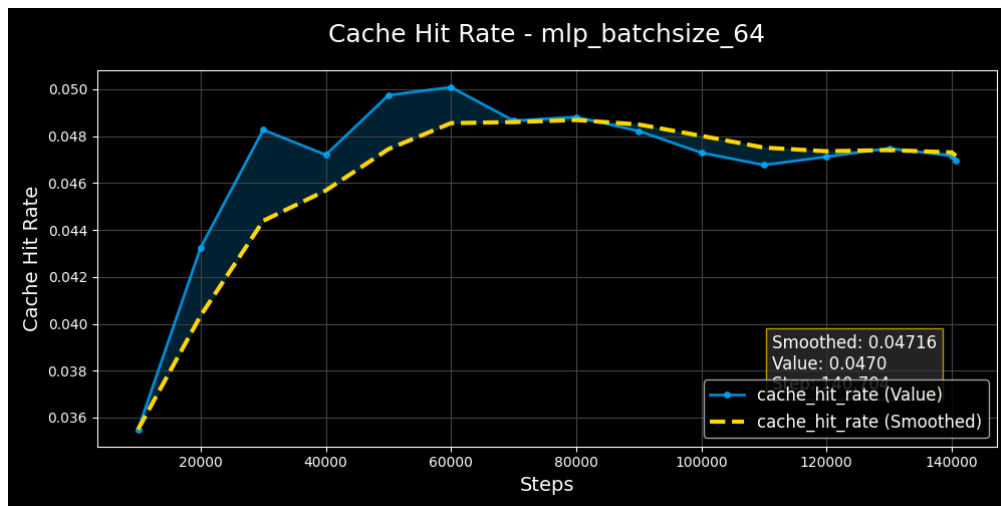
(b) Batch Size 4



(c) Batch Size 16



(a) Batch Size 32



(b) Batch Size 64

Figure 9: Effect of different batch size on cache hit rate

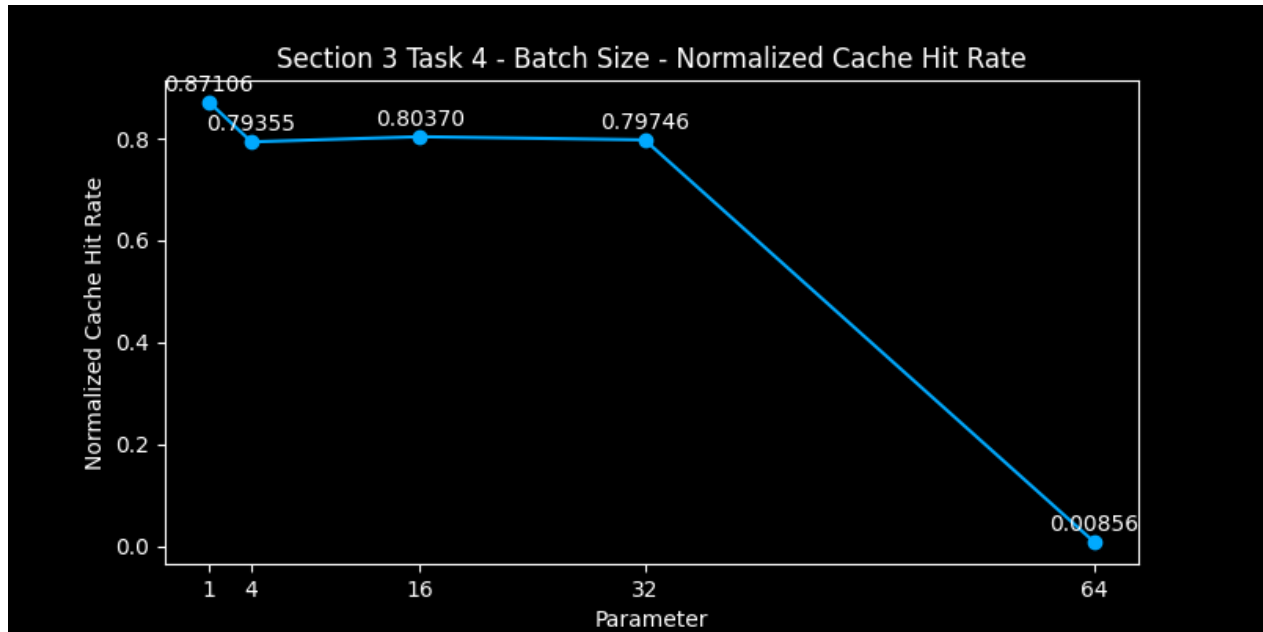
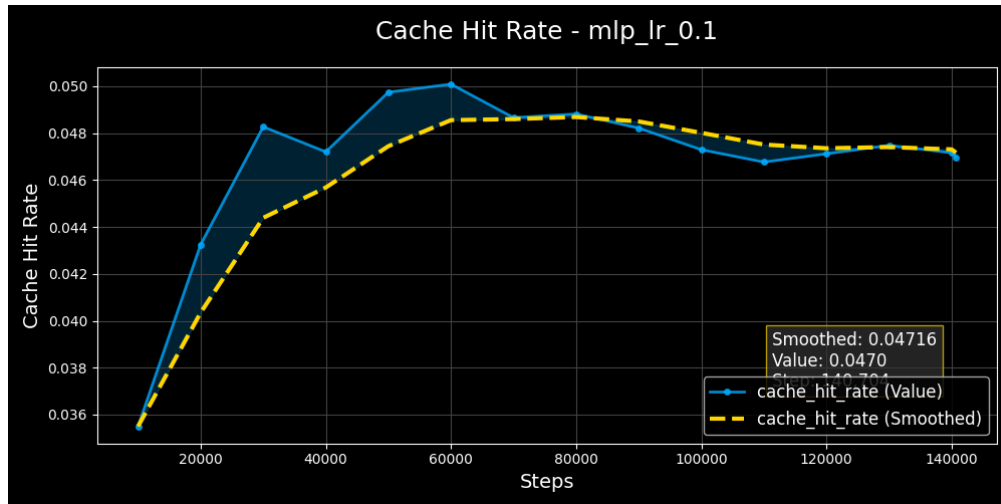
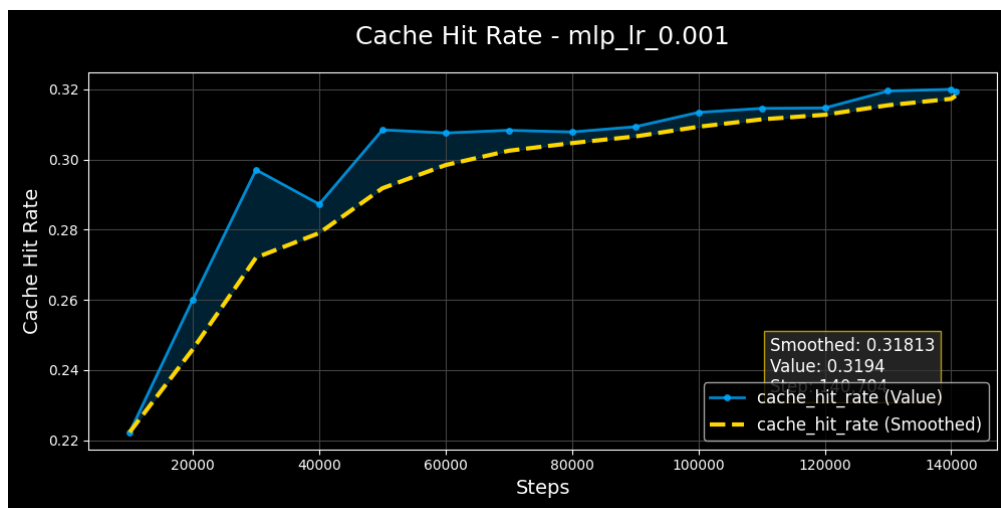


Figure 10: Normalized Cache Hit Rate vs. Batch Size

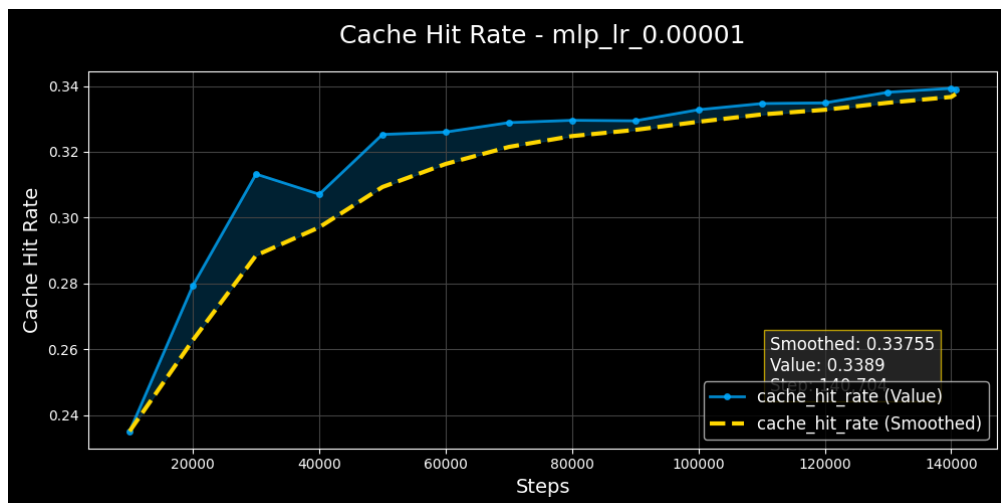
Task 5 - Learning Rates



(a) Learning Rate 0.1



(b) Learning Rate 0.001



(c) Learning Rate 0.00001

Figure 11: Effect of different batch size on cache hit rate

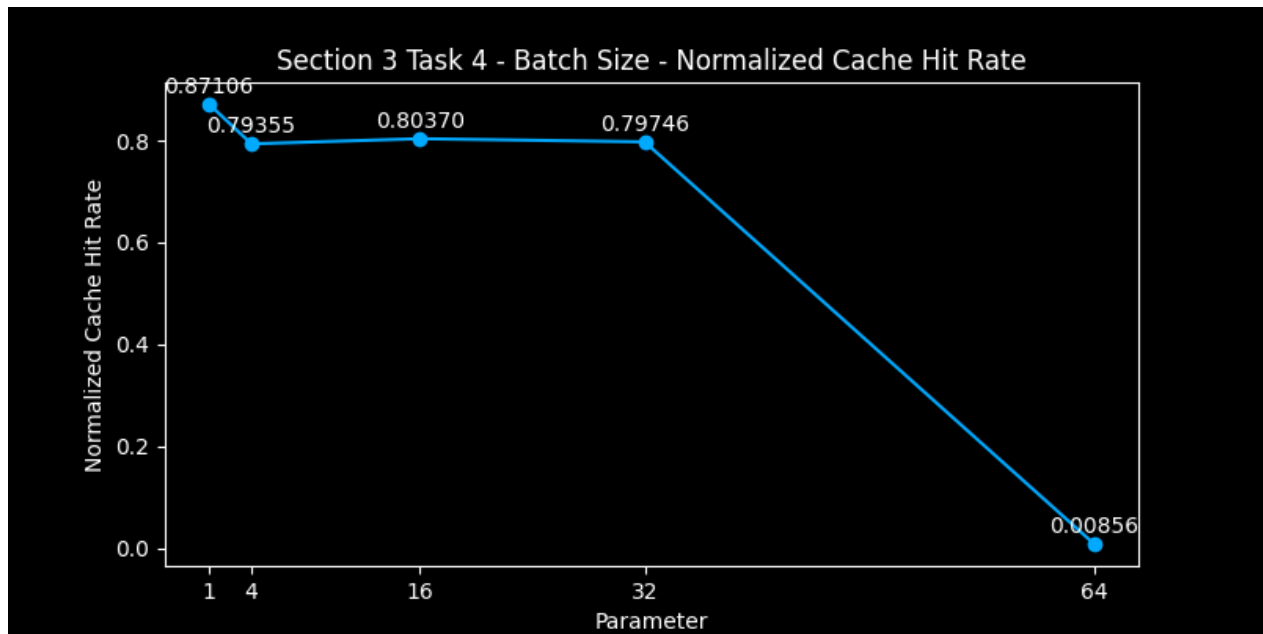
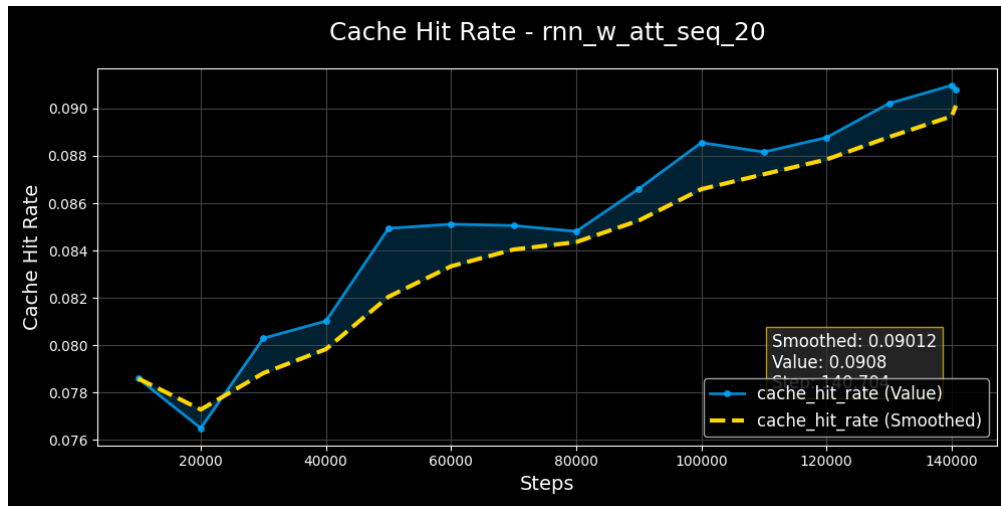


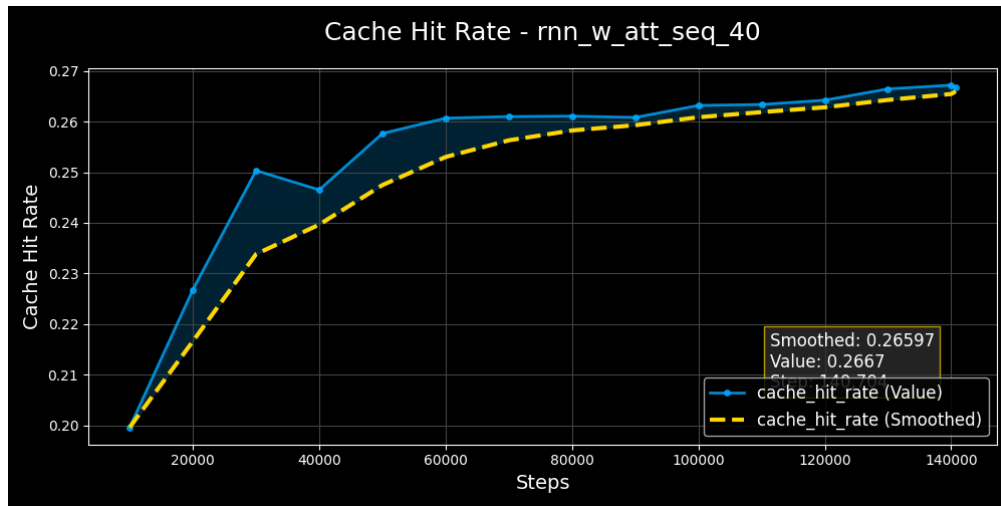
Figure 12: Normalized Cache Hit Rate vs. Learning Rate

Section 4 - RNN

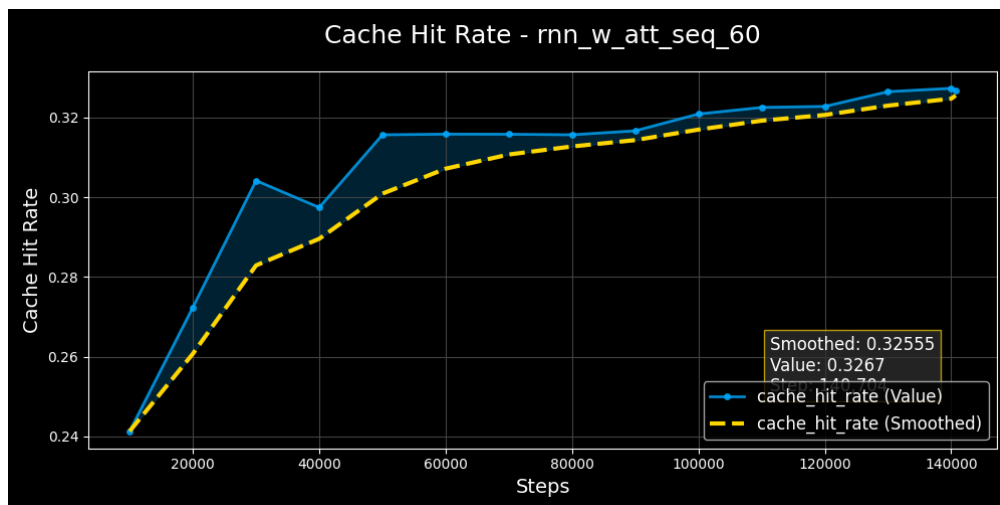
Task 1 - With Attention History



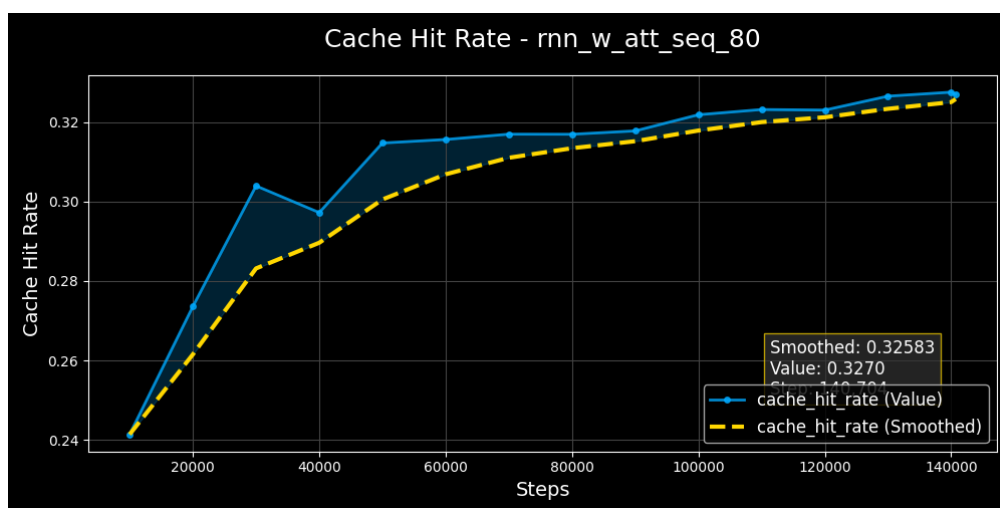
(a) 20



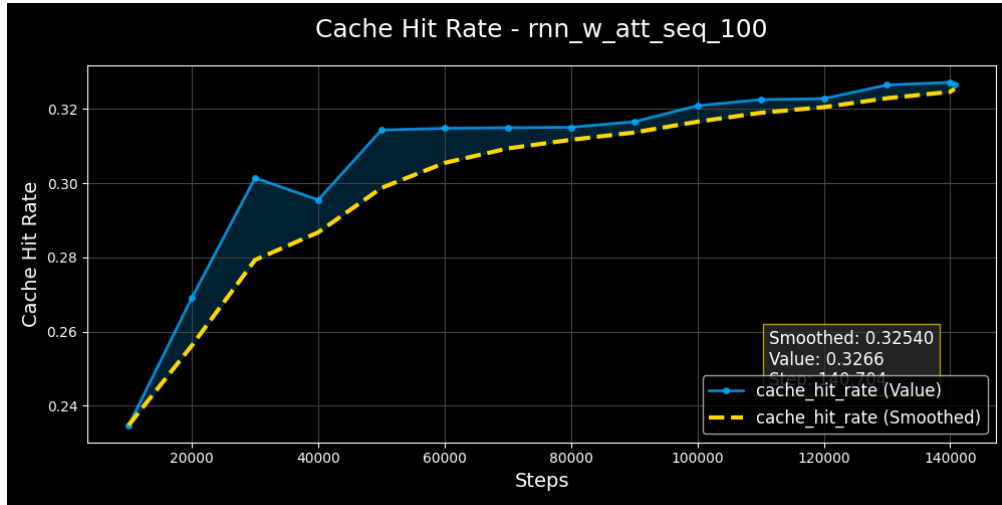
(b) 40



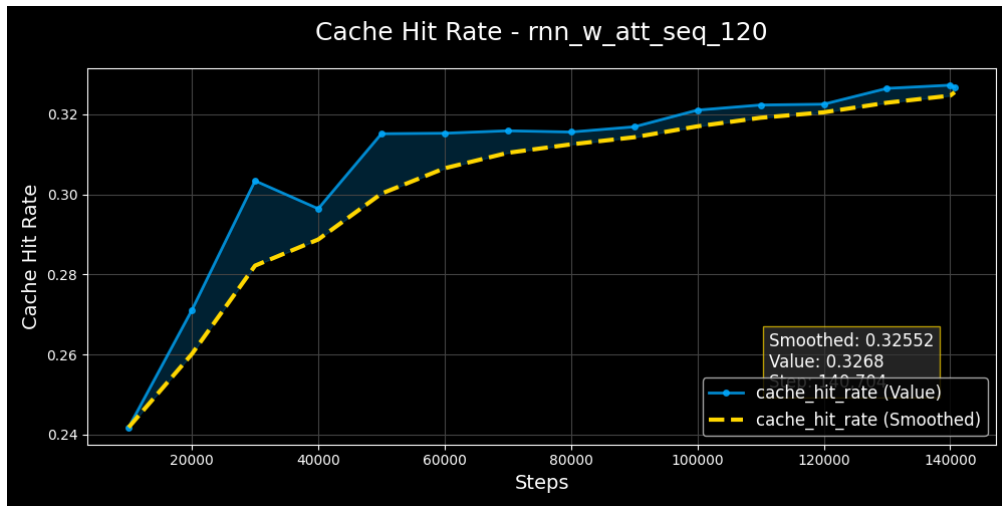
(a) 60



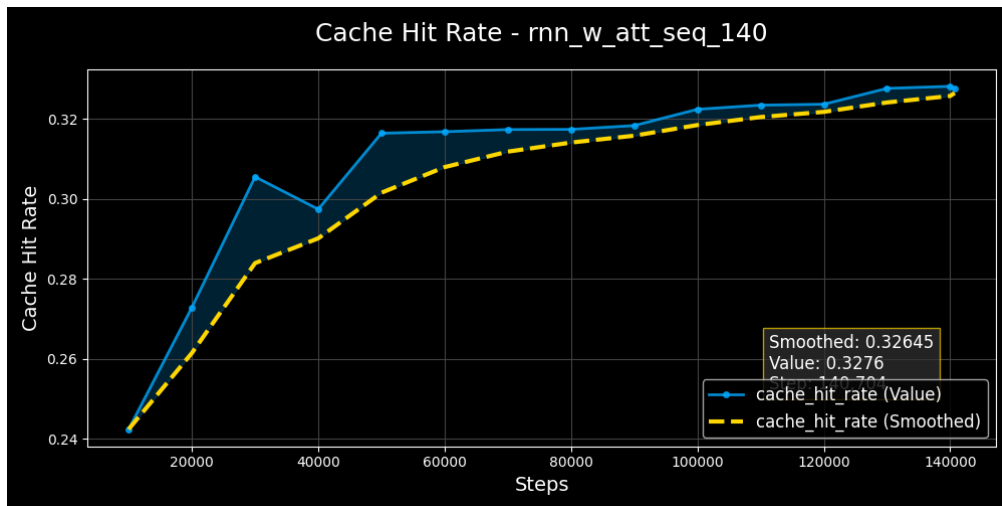
(b) 80



(a) 100



(b) 120



(c) 140

Figure 13: Effect of different batch size on cache hit rate

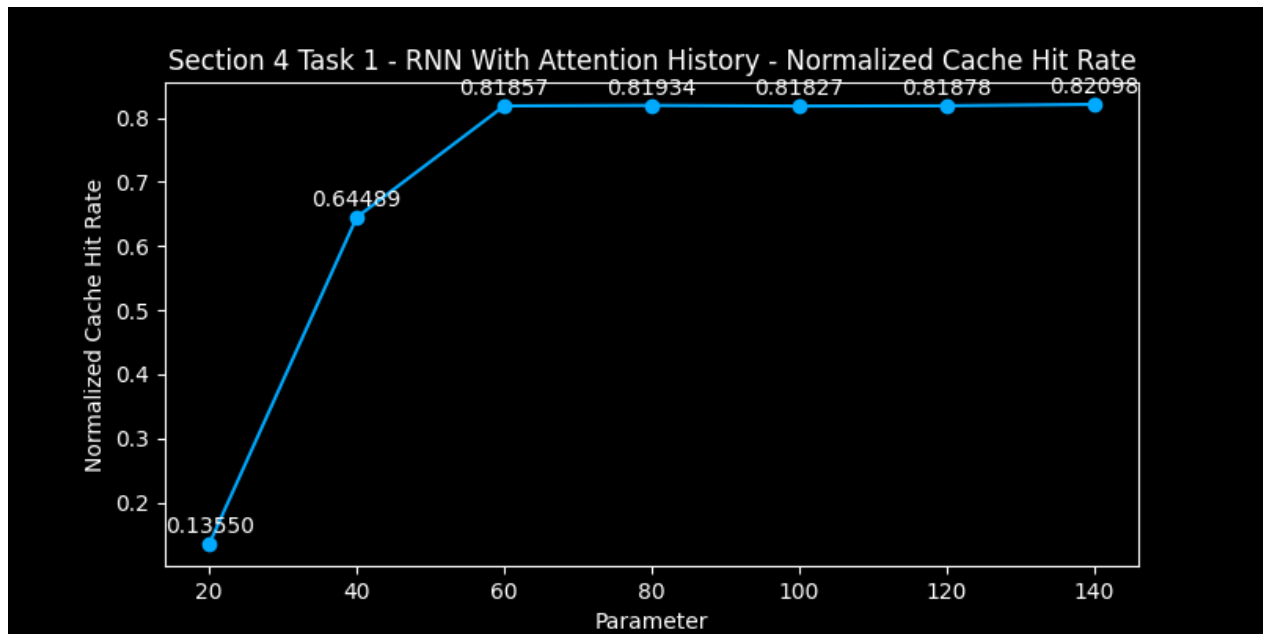
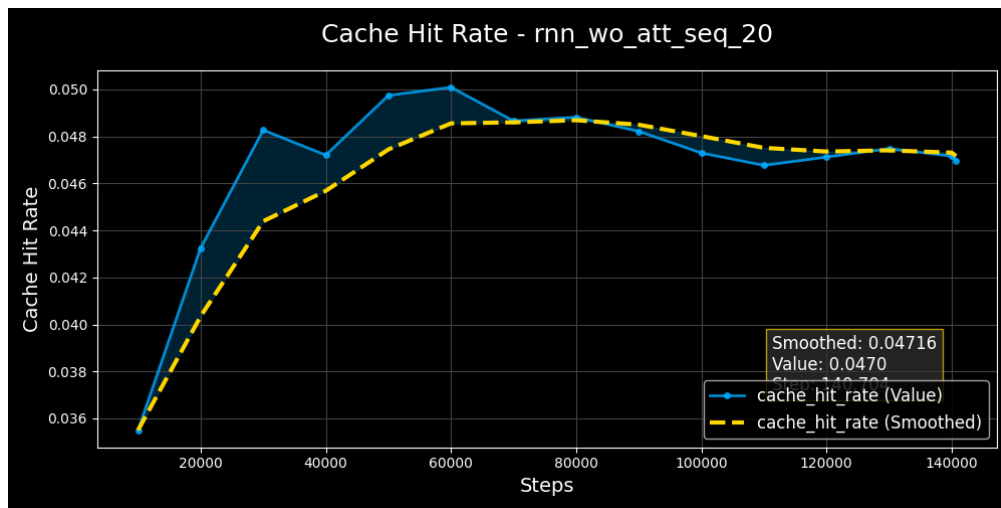
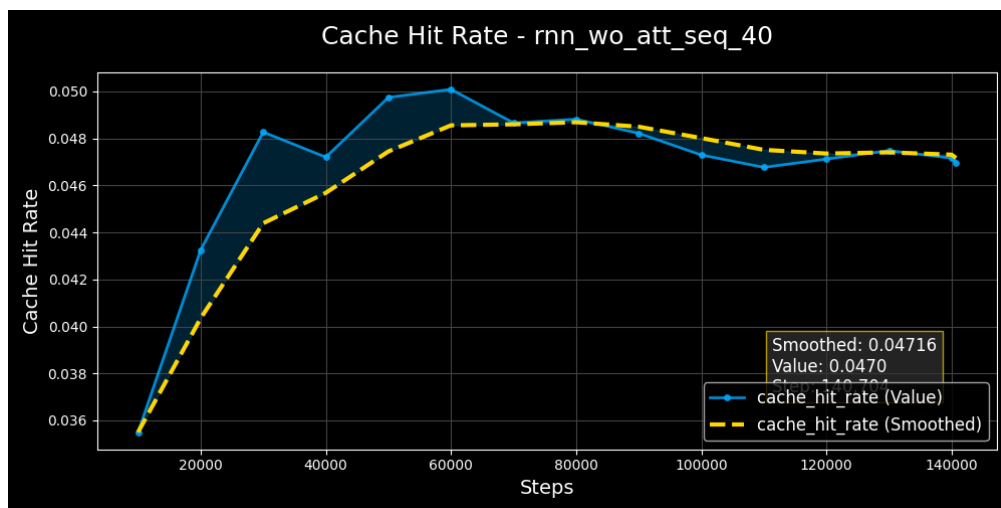


Figure 14: Normalized Cache Hit Rate vs. RNN with attention History

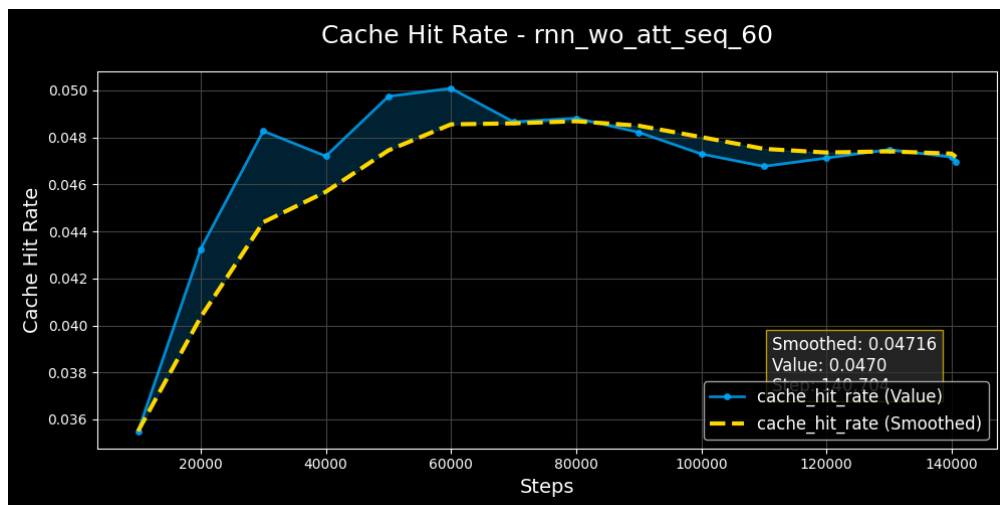
Task 2 - Without Attention History



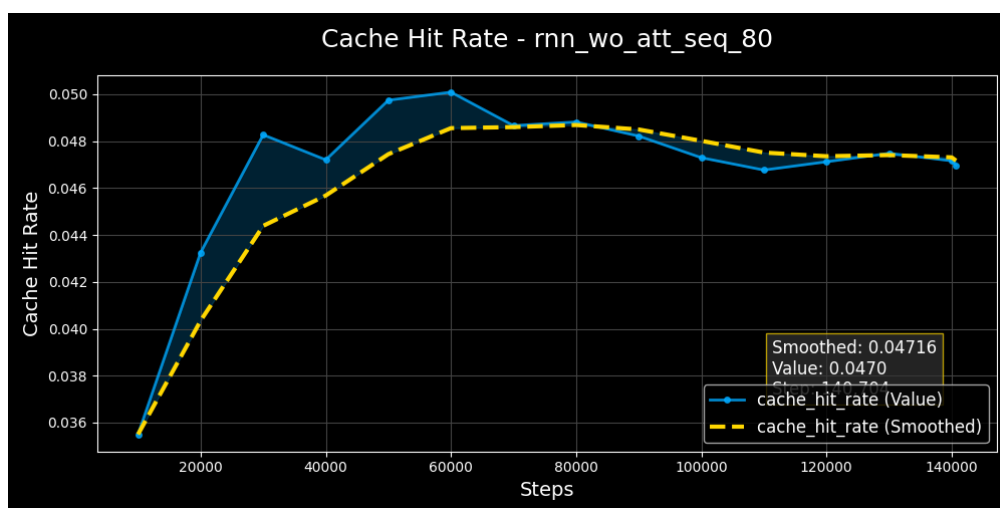
(a) 20



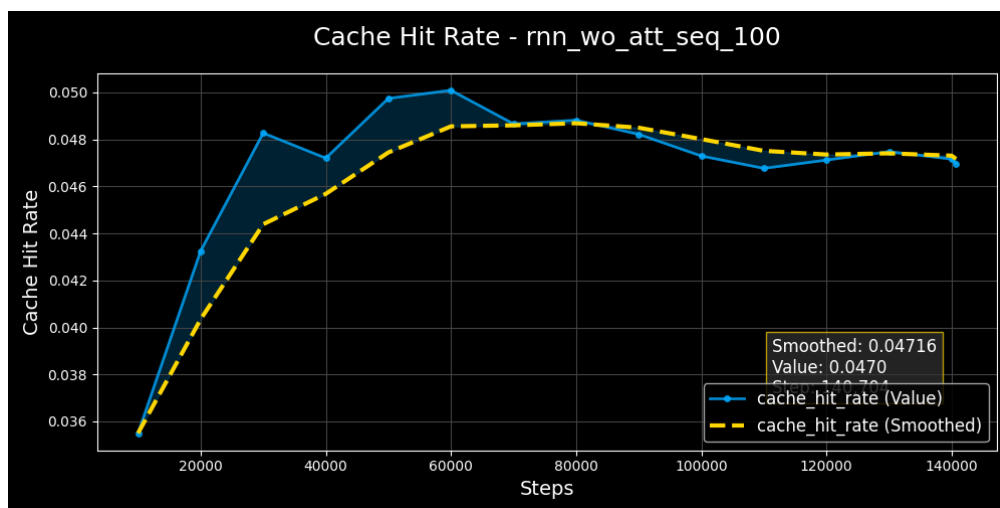
(b) 40



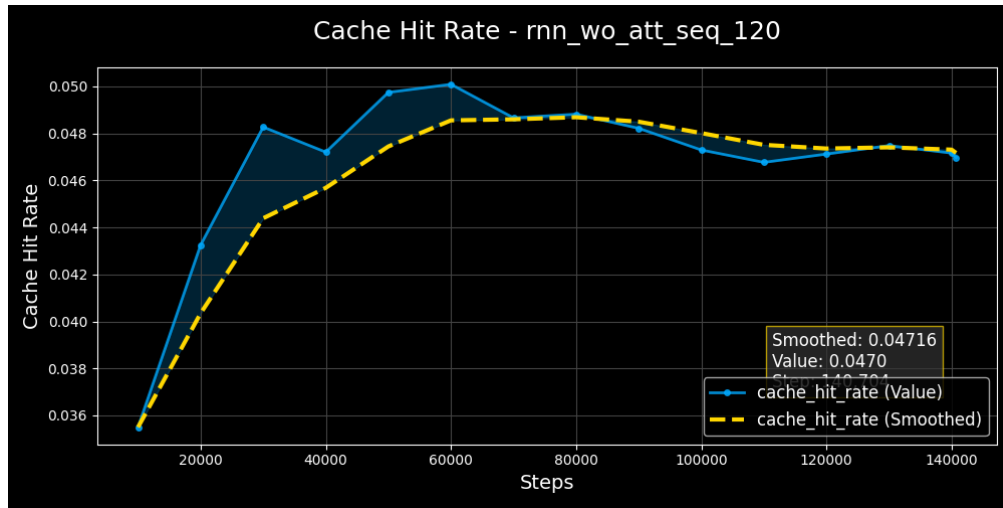
(a) 60



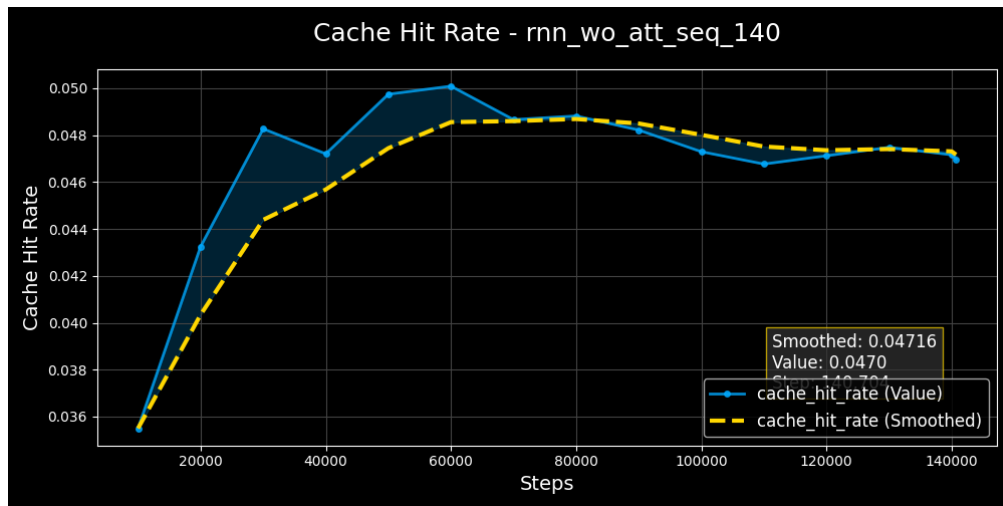
(b) 80



(c) 100



(a) 120



(b) 140

Figure 15: Effect of different batch size on cache hit rate

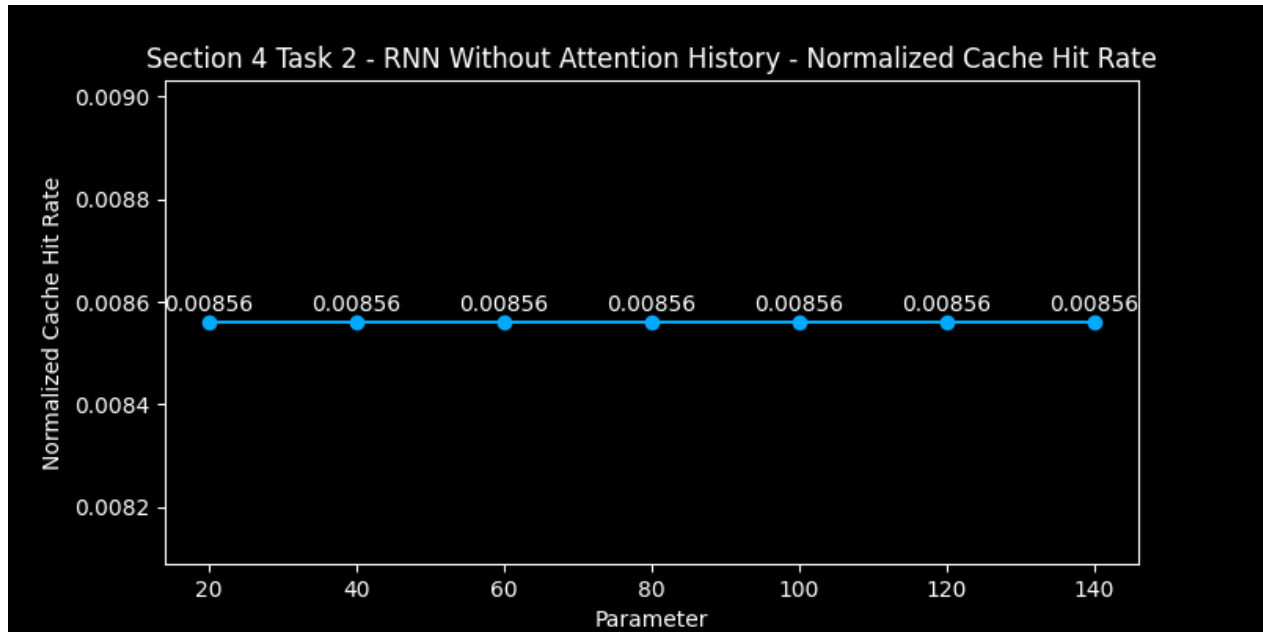
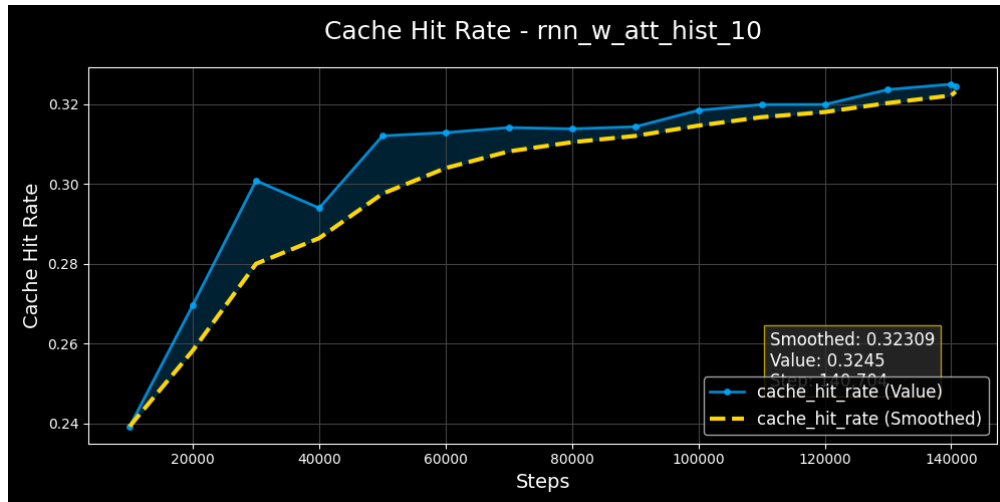
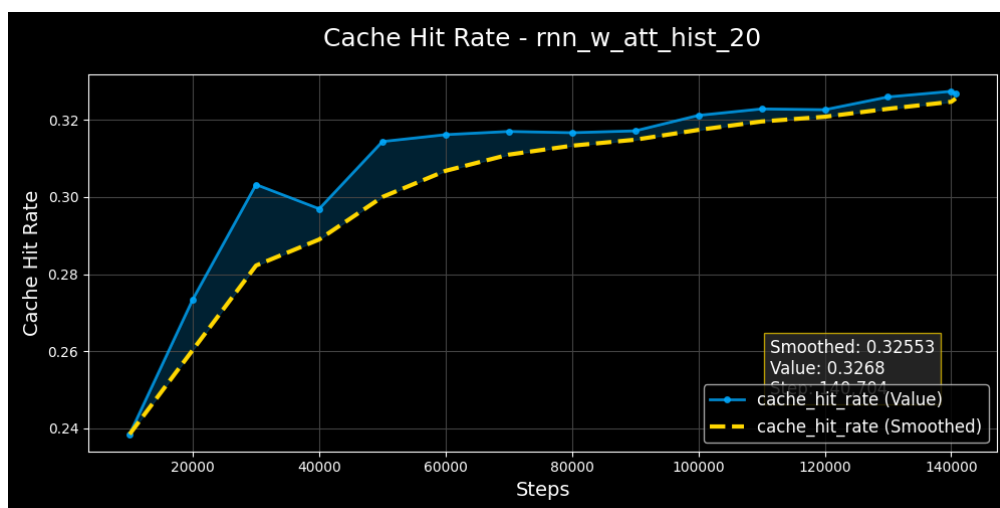


Figure 16: Normalized Cache Hit Rate vs. RNN without attention History

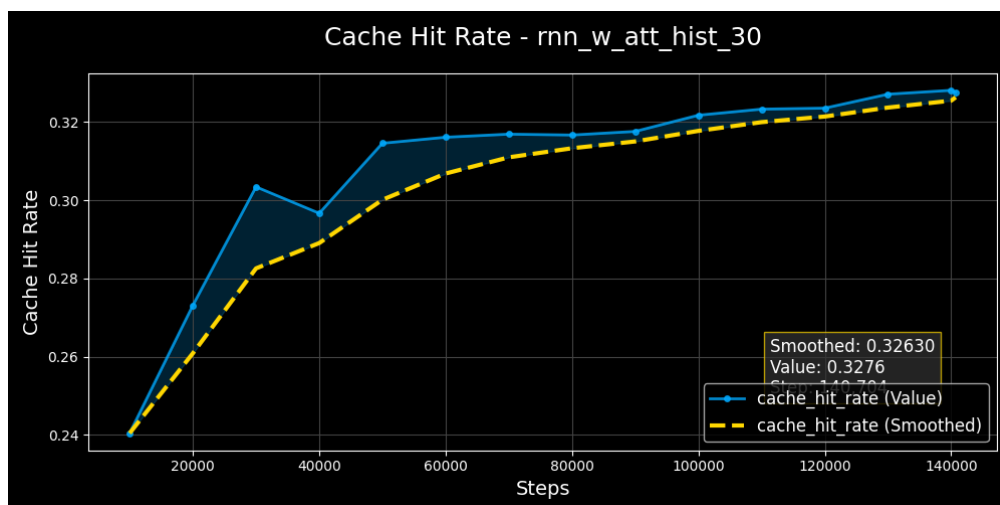
Task 3 - Maximum Attention History



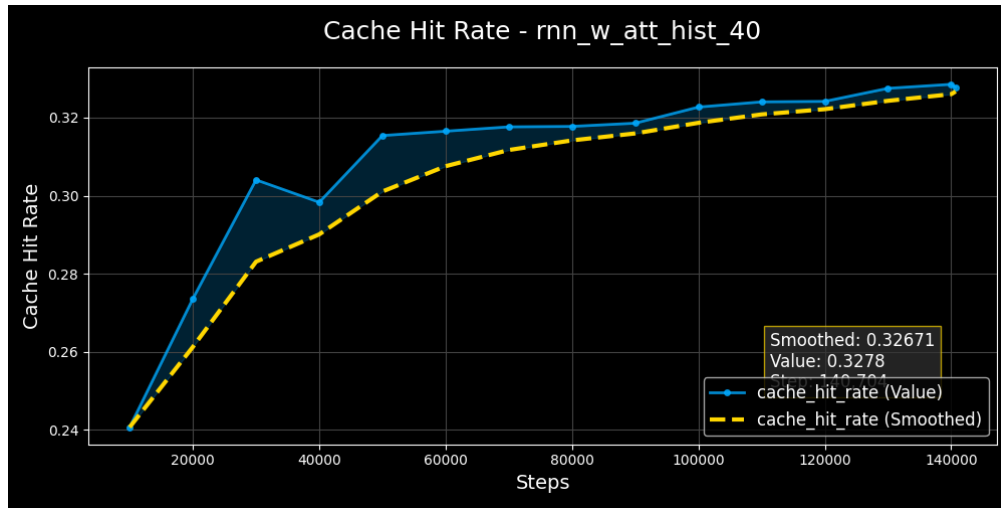
(a) 10



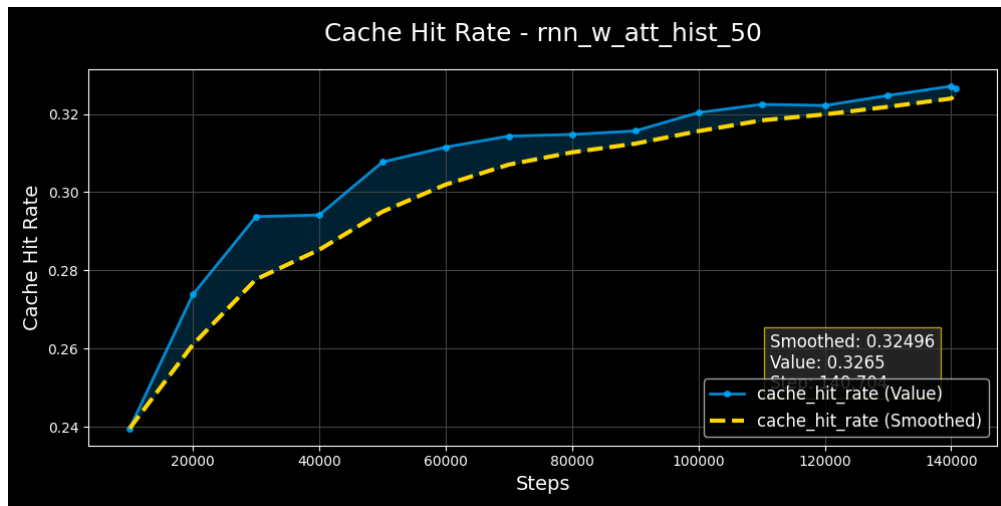
(b) 20



(c) 30



(a) 40



(b) 50

Figure 17: Effect of different batch size on cache hit rate

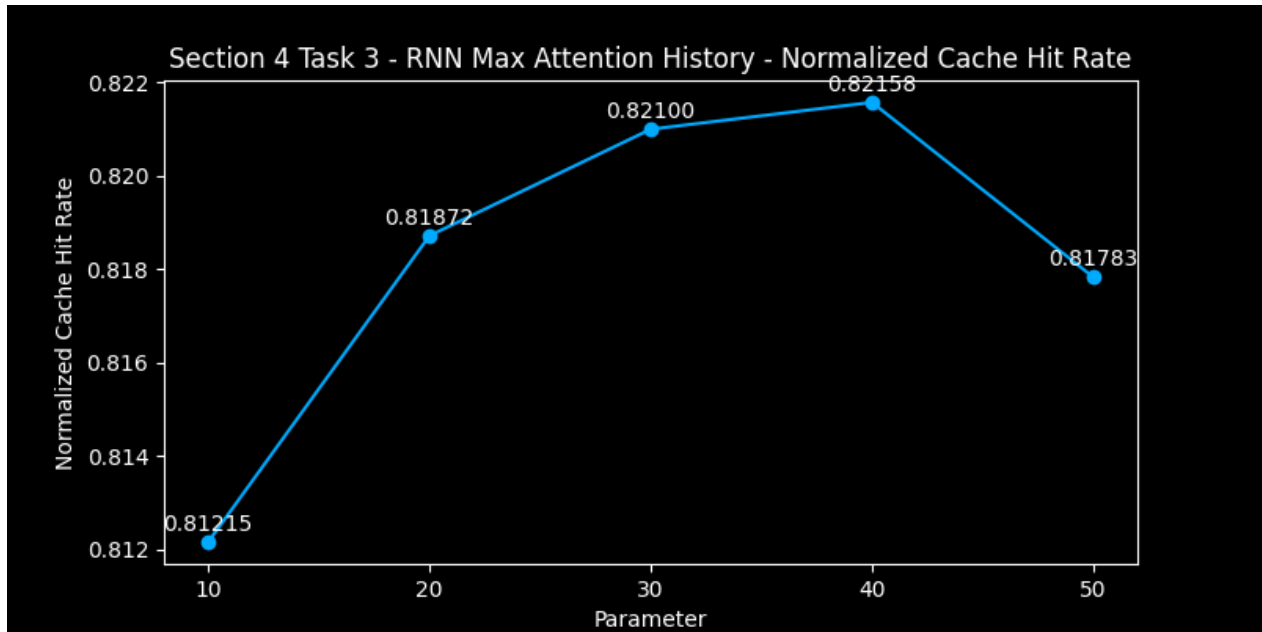


Figure 18: Normalized Cache Hit Rate vs. RNN Max attention History

.

Section 5 - LSTM

Task 1 - Baseline

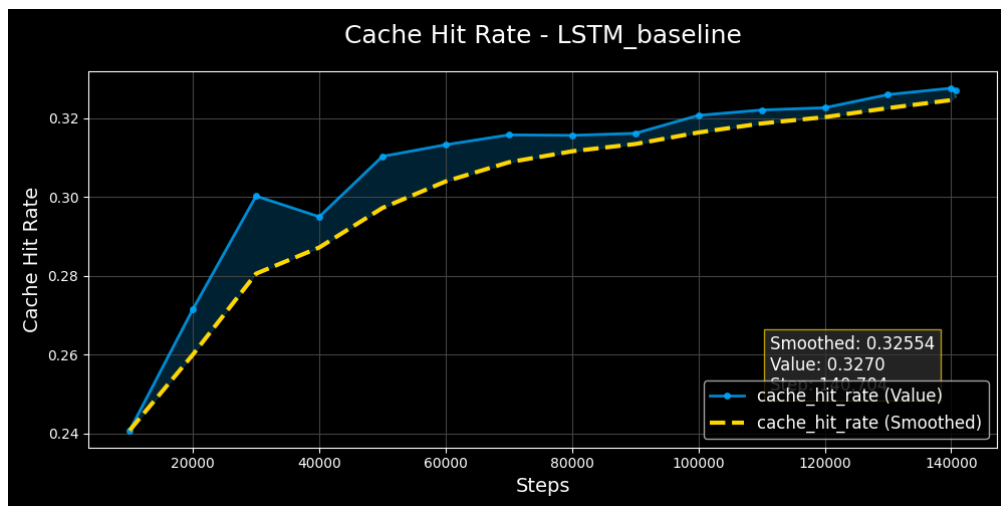


Figure 19: Baseline Task Image for LSTM

Task 2 - Byte Embedder

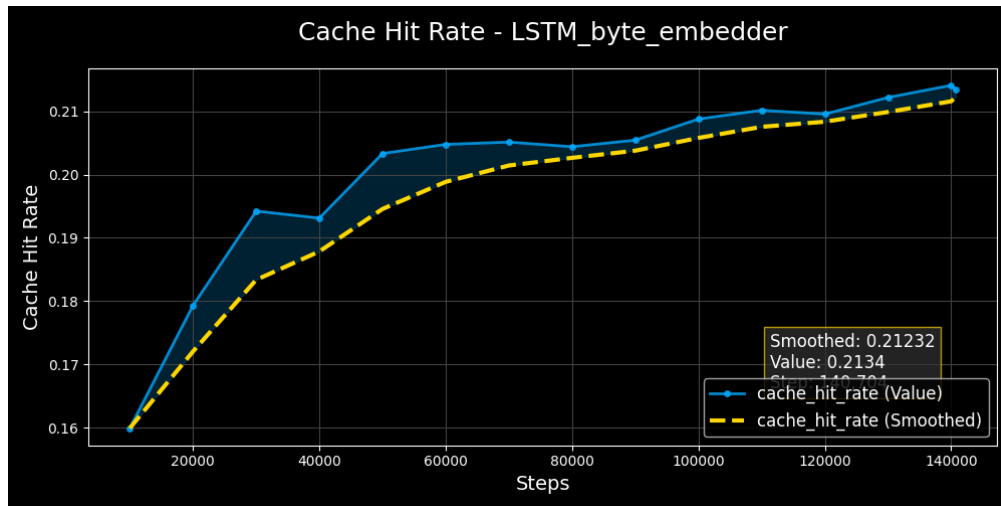


Figure 20: Byte Embedder Task Image for LSTM

Task 3 - Ablation - Reuse Distance Loss

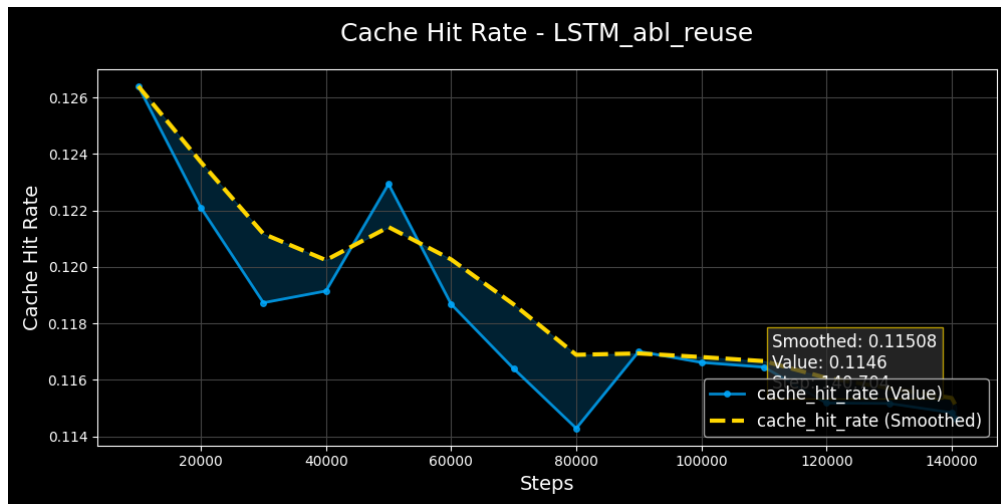


Figure 21: Ablation - Reuse Distance Loss Task Image for LSTM

Task 4 - Ablation - Ranking Loss

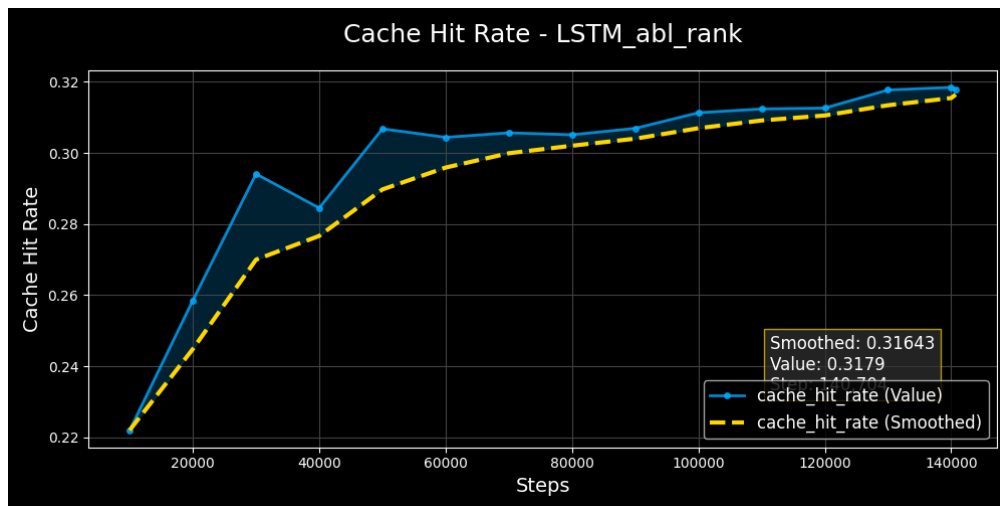


Figure 22: Ablation - Ranking Loss Task Image for LSTM

Comparison

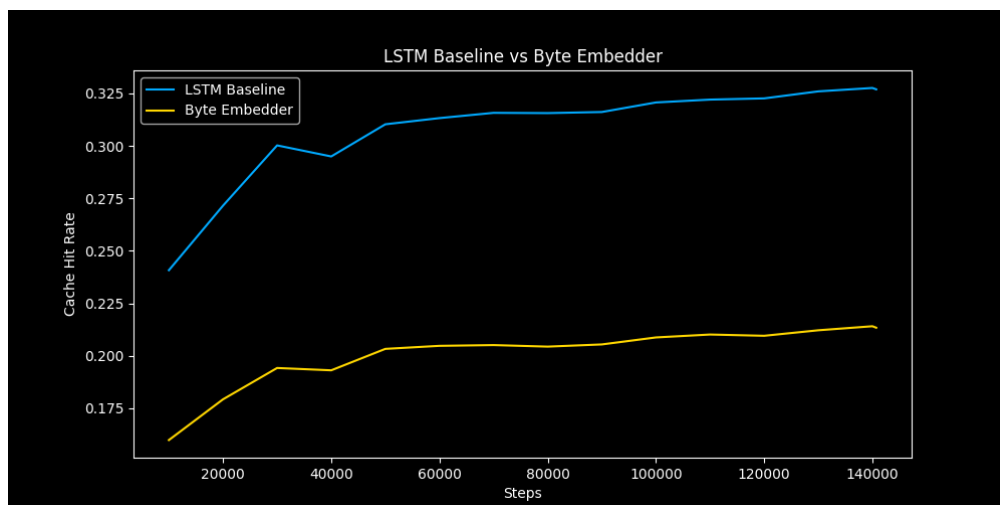


Figure 23: LSTM Baseline Vs Byte Embedder

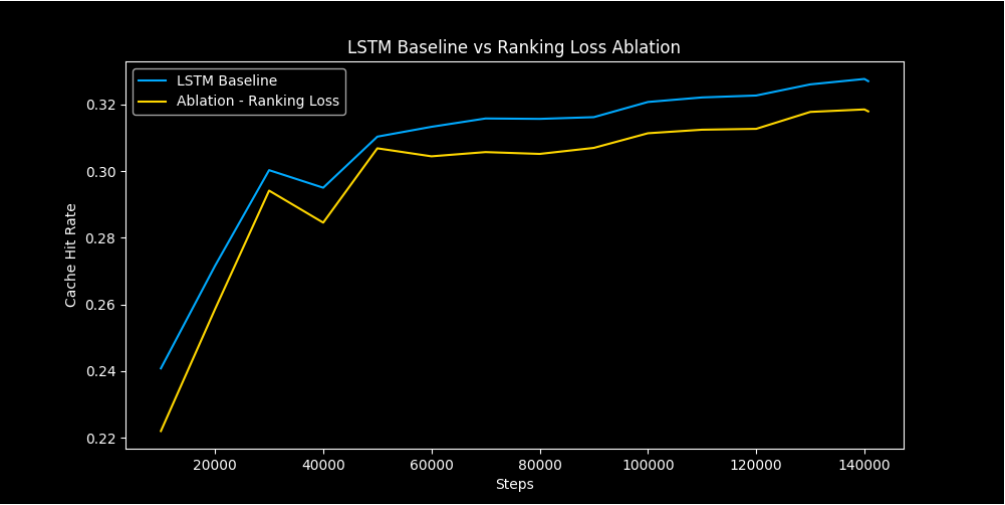


Figure 24: LSTM Baseline Vs Ranking Loss Ablation

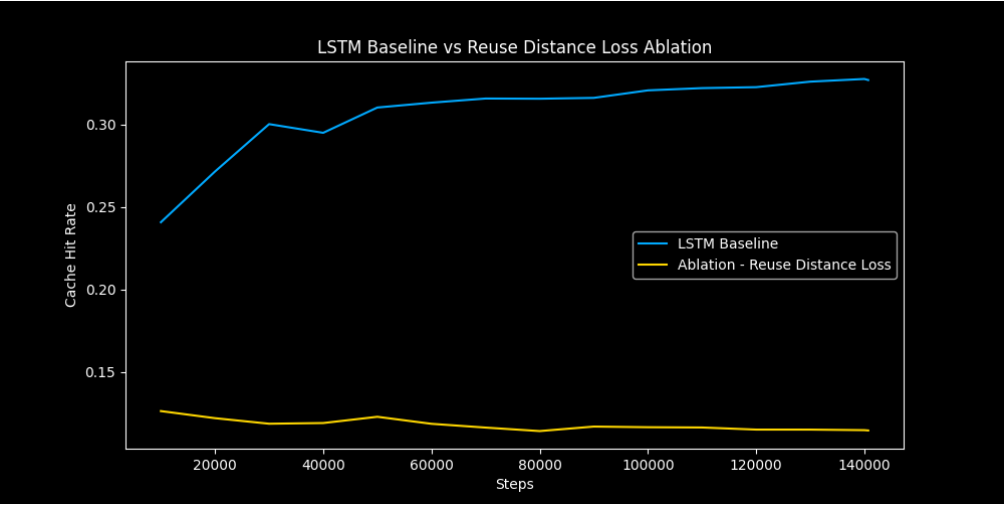
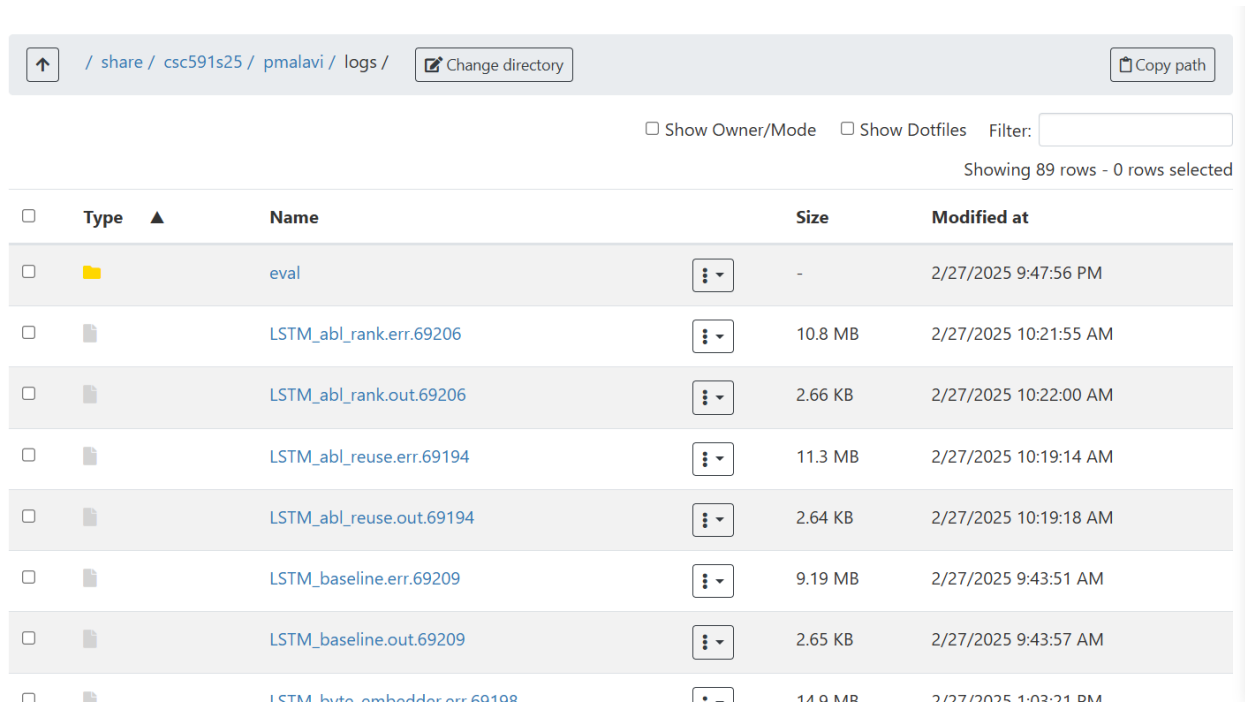


Figure 25: LSTM Baseline Vs Reuse Distance Loss Ablation

.

Log File Screenshot



The screenshot shows a file explorer interface with a breadcrumb path: / share / csc591s25 / pmalavi / logs /. It includes a 'Change directory' button and a 'Copy path' button. Below the path, there are checkboxes for 'Show Owner/Mode' and 'Show Dotfiles', and a 'Filter:' input field. A status bar indicates 'Showing 89 rows - 0 rows selected'. The main content is a table with columns: Type, Name, Size, and Modified at. The table lists several files, including 'eval' (a directory), and several log files with names like 'LSTM_abl_rank.err.69206', 'LSTM_abl_rank.out.69206', 'LSTM_abl_reuse.err.69194', 'LSTM_abl_reuse.out.69194', 'LSTM_baseline.err.69209', and 'LSTM_baseline.out.69209'. Each file row has a checkbox on the left and a three-dot menu icon on the right.

<input type="checkbox"/>	Type ▲	Name	Size	Modified at
<input type="checkbox"/>	Folder	eval	-	2/27/2025 9:47:56 PM
<input type="checkbox"/>	File	LSTM_abl_rank.err.69206	10.8 MB	2/27/2025 10:21:55 AM
<input type="checkbox"/>	File	LSTM_abl_rank.out.69206	2.66 KB	2/27/2025 10:22:00 AM
<input type="checkbox"/>	File	LSTM_abl_reuse.err.69194	11.3 MB	2/27/2025 10:19:14 AM
<input type="checkbox"/>	File	LSTM_abl_reuse.out.69194	2.64 KB	2/27/2025 10:19:18 AM
<input type="checkbox"/>	File	LSTM_baseline.err.69209	9.19 MB	2/27/2025 9:43:51 AM
<input type="checkbox"/>	File	LSTM_baseline.out.69209	2.65 KB	2/27/2025 9:43:57 AM
<input type="checkbox"/>	File	LSTM_hydr_embdder.err.69198	14.9 MB	2/27/2025 1:03:21 PM

Figure 26: Log File Screenshot