### Feature Overview

The proxy implements an intelligent product page detection and interactive Q&A system that enhances online shopping experiences. When users visit Target.com product pages, the system automatically detects product content and provides a floating chat interface for product-related inquiries.

### 

### Performs Well:

Answering specific product feature questions Explaining technical specifications, Comparing different product attributes, Summarizing product reviews, Providing usage suggestions.

### May Fall Short:

The current implementation requires retransmission of complete product data to the LLM for each query, even when multiple questions are asked about the same product. This may cause increased response time for repeated queries.

### Design

The proxy is implemented as a non-blocking server and uses a linked list structure to track all active connections, Certificate generation for HTTPS interception, Dynamic certificate creation for each domain. The proxy uses select() for event handling, All sockets set to non-blocking mode. The proxy intercepts all HTTP/HTTPS responses and analyzes HTML content for product-specific markers. Only processes responses identified as product pages.

### Technical Challenges:

First challenge that we encountered was the way we sourced product description data. First, we tried to parse it from JSON object that is statically loaded when user visits target.com for the first time. However, as target.com is a Single Page Application (SPA), we found out that data is not updated but rather is sourced from the redsky.target api. As such, we reverse engineered redsky target api and parsed data using their api.

The initial implementation attempted to send API requests directly from the injected JavaScript to the LLM API. We resolved this by implementing server-side API communication through the proxy instead of making direct requests from injected JavaScript, which improved security and performance.

### Performance Evaluation

Concurrency Analysis: Our proxy system underwent comprehensive testing with varying concurrent requests (10-100) of major websites:

* Under light load (10-60 concurrent requests): 100% success rate, response times 0.81-4.07 seconds
* Performance degraded at 70+ concurrent requests: success rates dropped to 84.29%
* At 100 concurrent requests: success rate decreased to 55.00%
* Median response time: 0.69-4.83 seconds under high load

Download Time Analysis (per webpage):

With Proxy:

* Median response time: ~0.25 seconds
* IQR: ~0.20 to ~0.30 seconds
* Maximum regular response time: ~0.40 seconds
* Outliers at ~0.45 seconds
* Minimum response time: ~0.20 seconds

Without Proxy:

* Median response time: ~0.22 seconds
* IQR: ~0.18 to ~0.25 seconds
* Maximum regular response time: ~0.35 seconds
* Outliers at ~0.58 and ~0.85 seconds
* Minimum response time: ~0.15 seconds

### 

### Performance Impact:

The proxy adds approximately 0.03 seconds to the median response time. These results demonstrate reliable performance under moderate load while maintaining acceptable response times for typical web browsing scenarios.

### Future Improvements:

Implement data caching to avoid repeated LLM data transmission. Add the ability to compare different products and look for products based on their description.