WebLLM : Adapting Large Language Models for Anti Tracking

Muhammad Jazlan, Shaoor Munir, Umar Iqbal, Zubair Shafiq, Sandra Siby

UC DAVIS WashU NYU | ABU DHABI

Problem Statement

Create an anti-tracking framework that:
- Has a **high accuracy**
- Relies **less on ground truth** (filterlist labels)
- Does **not require feature engineering** (e.g., length, domain names, etc.)
- Is **generalizable** (if it works for URL classification, it should work query parameters and cookie classification)

Prompting LLMs

URL classification on 1000 URLs using OpenAI's o4-mini:
- Zero shot accuracy **~90%** (**6%** below SOTA)
- **Prompt engineering approaches:** Role prompting and in-context learning
- **Prompt Optimization:** LLM-based feedback for iterative prompt refinement (Figure 1)
- Accuracy improves to **~92.8%**

Generalizability

- WebLLM can be applied to other anti tracking tasks by modifying the inputs
- Foundation LLM can be replaced as better ones come out, only adapters need to be re-trained

Results

- We implement the framework using Gemma 3 in two sizes, 1b and 4b
- For each size, we implement a text-only variation, and a text and graph implementation

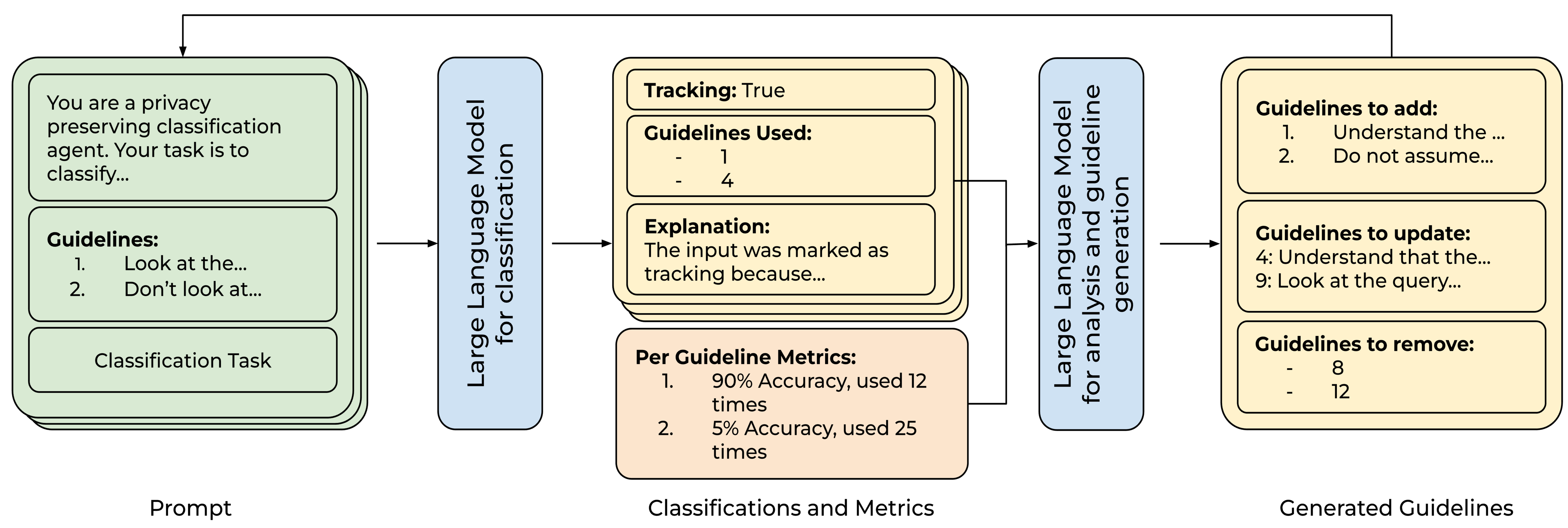

Figure 1. LLM-driven prompt optimization pipeline for automated feedback on classification

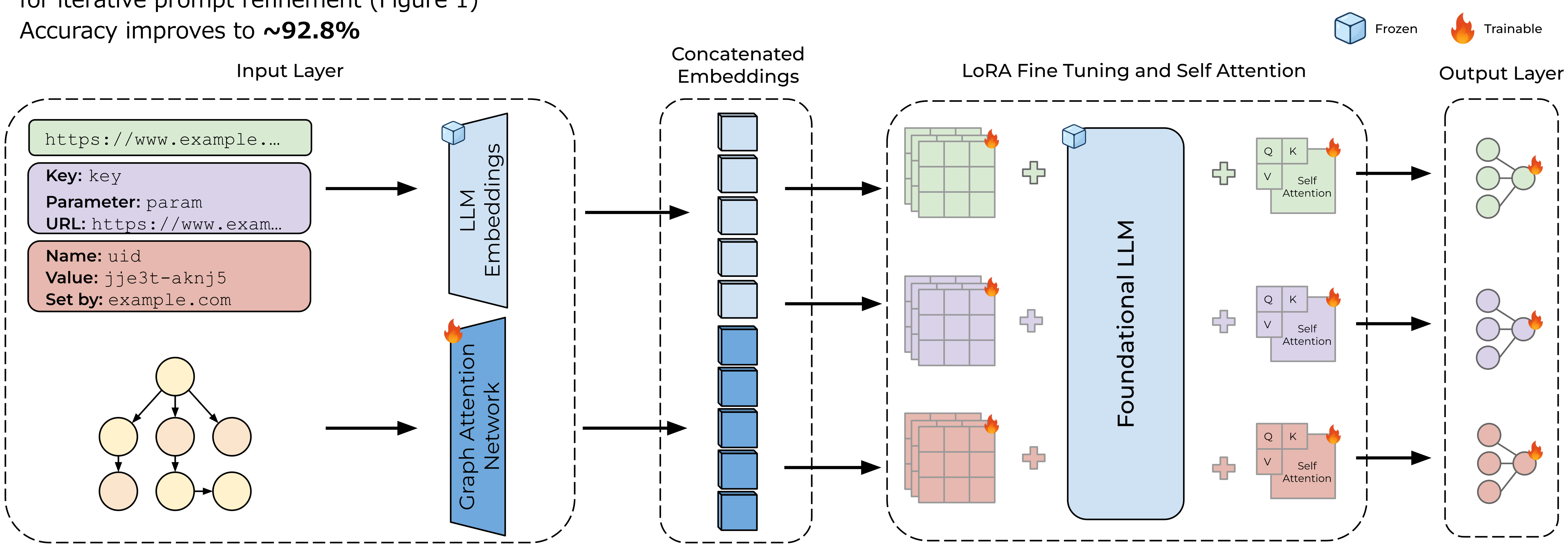

Figure 2. Parameter efficient fine-tuning pipeline. We encode graph representations of the webpage by using a GAT.

Context Matters

- SOTA models use graph representations of webpages
- Provide graph representations of webpages in addition to URLs
- Accuracy **drops** when graphs are provided as context
- **Explanation**: LLMs have limited understanding of structured data

Low Rank Fine-Tuning

To adapt LLMs for anti tracking (Figure 2), we do the following:
- **Parameter efficient fine-tuning**:
  - Reduce the total parameters that need to be trained (~10s of millions vs ~ billions)
- **Restrict the output space**:
  - Use the LLM as an encoder of information
  - Large output space leads to more errors
  - No need for the "LM head"; we only require a binary output label
- **Addressing the modality gap:**
  - Convert graphs into embeddings that can be understood by the LLM

Robustness

- SOTA models are robust to evasion attempts like domain obfuscation, query parameter obfuscation, query parameter encryption
- We replicate domain and query parameter obfuscation, and add a new evasion technique: path obfuscation
- We use these obfuscation techniques as forms of data augmentation during the training process and evaluate only for fine tuned LLM.

Future Work

- Improve graph-based models:
  - Performance is similar to non-graph models, but prior work demonstrates that graphs are significantly more robust
- Evaluate the framework on two anti tracking tasks
  - Query Parameter sanitization
  - Cookie Classification
- Implement a browser extension to calculate impact on QoE.

Variant	Plain URLs			Partially Obfuscated			Fully Obfuscated		
	Accuracy	FPR	FNR	Accuracy	FPR	FNR	Accuracy	FPR	FNR
1b	93.15	6.19	7.49	88.50	10.87	12.10	83.70	12.88	19.13
1b with Graph	91.85	6.36	9.88	88.40	9.66	13.36	83.45	11.95	20.16
4b	95.40	3.58	5.58	92.45	7.42	7.68	85.40	14.95	14.24
4b with Graph	94.90	4.18	5.98	91.05	8.07	9.79	84.55	11.57	18.62

Table 1. Performance metrics for fine-tuned LLM pipeline (%)