24KIDS442 XAI LLM LAB

Stefi Shobika Sukumar

Understanding how large language models (LLMs) process and reason about text is a major challenge in modern AI research — especially when these models are deployed in black-box settings, where internal decision processes are hidden from users.

As part of the XAI LLM Lab Internship, this project focuses on building a lightweight, explainable interface for small, quantized LLMs (≤7B) that can run on CPU devices. The goal is to help users explore how the model processes input prompts, by combining visual neuron activation maps with natural language rationale explanations.

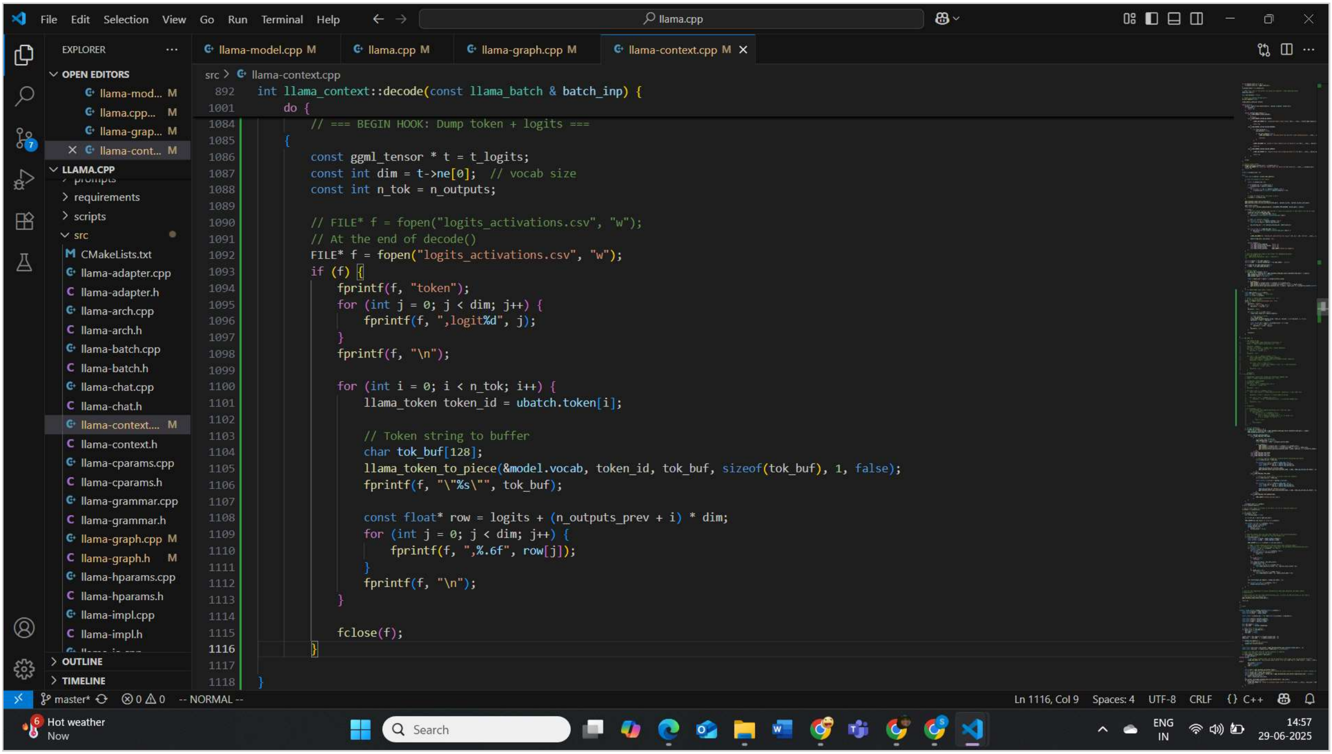
System Architecture and Methodology

This project is implemented as a two-stage pipeline:

1. **Offline Backend (Computation):** Prompts are executed using quantized GGUF models (TinyLlama Q3_K_M and Phi Q4_K_M) in `llama.cpp`. A custom-modified decoder caches layer-wise neuron activations into `logits_activations.csv`.
2. **Token Explanation:** The Zephyr model (Mistral-based) is used to generate natural language rationales for each token. These are saved in `rationales.jsonl`.
3. **Streamlit Dashboard:** The app loads precomputed activations and rationales. When the user enters a prompt, it retrieves the cached data and visualizes the token-wise attention using Plotly heatmaps alongside the generated rationales.

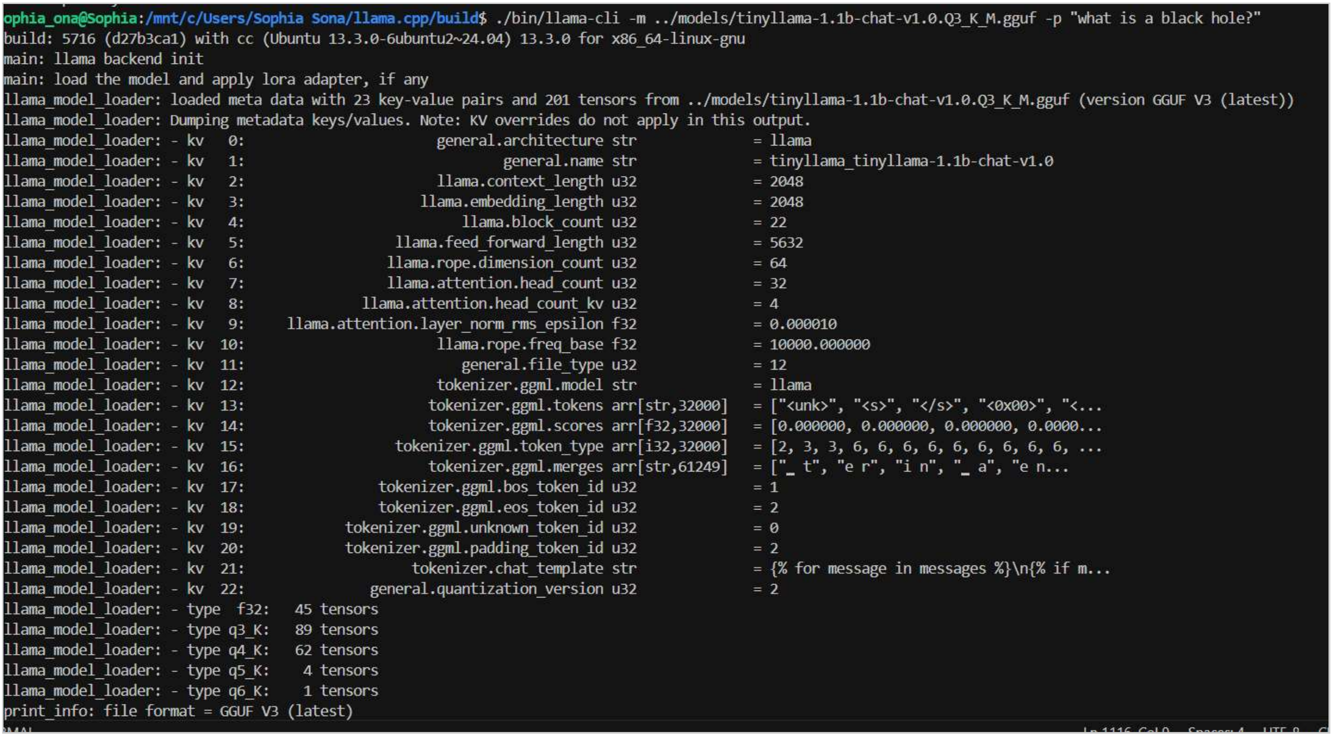
This decoupled design allows for CPU-efficient local analysis and avoids the need to re-run inference live inside the Streamlit app.

Backend Pipeline (Screenshots)

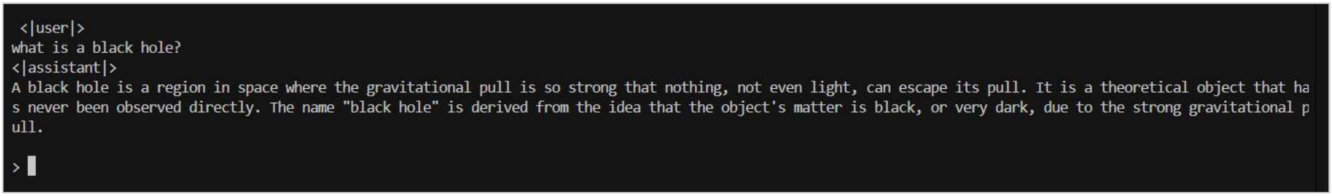


```cpp
int llama_context::decode(const llama_batch & batch_inp) {
    do {
        // === BEGIN HOOK: Dump token + logits ===
    {
        const ggml_tensor * t = t_logits;
        const int dim = t->ne[0];  // vocab size
        const int n_tok = n_outputs;

        // FILE* f = fopen("logits_activations.csv", "w");
        // At the end of decode()
        FILE* f = fopen("logits_activations.csv", "w");
        if (f) {
            fprintf(f, "token");
            for (int j = 0; j < dim; j++) {
                fprintf(f, ",logit%d", j);
            }
            fprintf(f, "\n");

            for (int i = 0; i < n_tok; i++) {
                llama_token token_id = ubatch.token[i];

                // Token string to buffer
                char tok_buf[128];
                llama_token_to_piece(&model.vocab, token_id, tok_buf, sizeof(tok_buf), 1, false);
                fprintf(f, "\"%s\"", tok_buf);

                const float* row = logits + (n_outputs_prev + i) * dim;
                for (int j = 0; j < dim; j++) {
                    fprintf(f, ",%.6f", row[j]);
                }
                fprintf(f, "\n");
            }

            fclose(f);
        }

}
```

Loading quantized GGUF models with llama.cpp

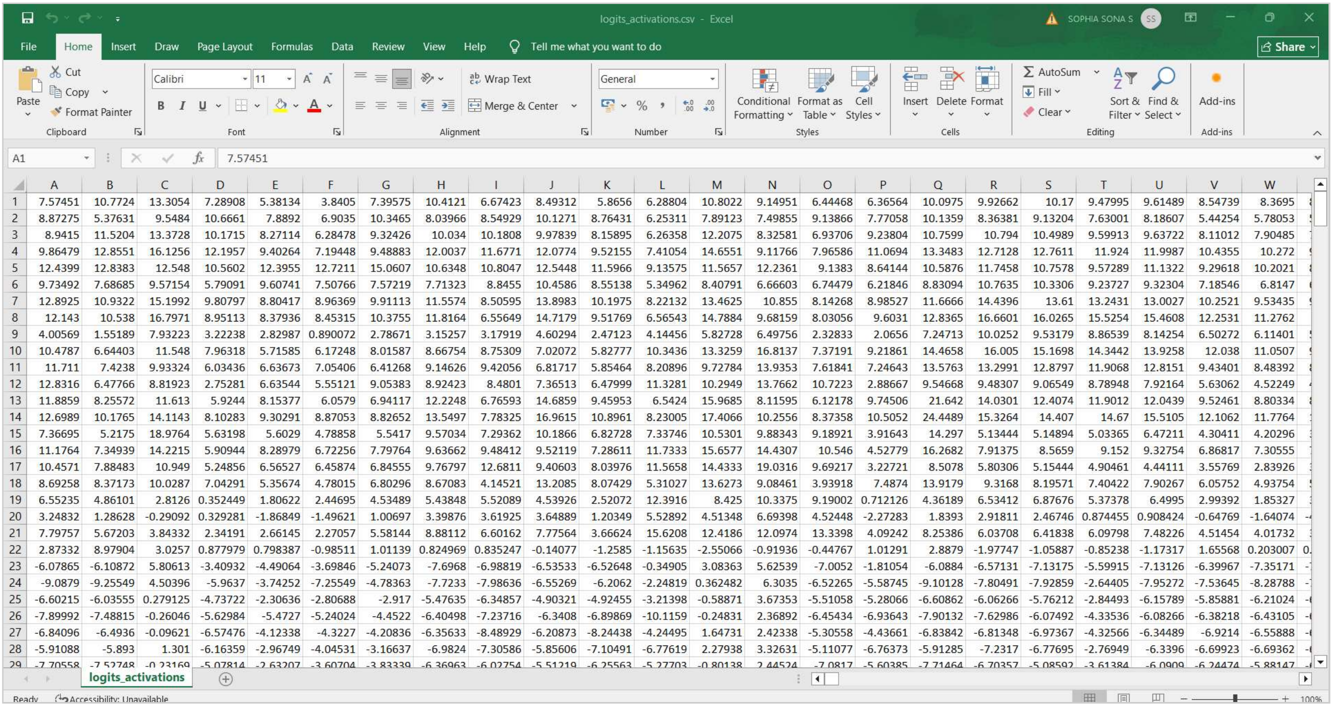


```
ophia_ona@Sophia:/mnt/c/Users/Sophia Sona/llama.cpp/build$ ./bin/llama-cli -m ../models/tinyllama-1.1b-chat-v1.0.Q3_K_M.gguf -p "what is a black hole?"
build: 5716 (d27b3ca1) with cc (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0 for x86_64-linux-gnu
main: llama backend init
main: load the model and apply lora adapter, if any
llama_model_loader: loaded meta data with 23 key-value pairs and 201 tensors from ../models/tinyllama-1.1b-chat-v1.0.Q3_K_M.gguf (version GGUF V3 (latest))
llama_model_loader: Dumping metadata keys/values. Note: KV overrides do not apply in this output.
llama_model_loader: - kv   0:                       general.architecture str              = llama
llama_model_loader: - kv   1:                               general.name str              = tinyllama_tinyllama-1.1b-chat-v1.0
llama_model_loader: - kv   2:                       llama.context_length u32              = 2048
llama_model_loader: - kv   3:                     llama.embedding_length u32              = 2048
llama_model_loader: - kv   4:                          llama.block_count u32              = 22
llama_model_loader: - kv   5:                  llama.feed_forward_length u32              = 5632
llama_model_loader: - kv   6:                 llama.rope.dimension_count u32              = 64
llama_model_loader: - kv   7:                 llama.attention.head_count u32              = 32
llama_model_loader: - kv   8:              llama.attention.head_count_kv u32              = 4
llama_model_loader: - kv   9:     llama.attention.layer_norm_rms_epsilon f32              = 0.000010
llama_model_loader: - kv  10:                       llama.rope.freq_base f32              = 10000.000000
llama_model_loader: - kv  11:                          general.file_type u32              = 12
llama_model_loader: - kv  12:                       tokenizer.ggml.model str              = llama
llama_model_loader: - kv  13:                      tokenizer.ggml.tokens arr[str,32000]   = ["<unk>", "<s>", "</s>", "<0x00>", "<...
llama_model_loader: - kv  14:                      tokenizer.ggml.scores arr[f32,32000]   = [0.000000, 0.000000, 0.000000, 0.0000...
llama_model_loader: - kv  15:                  tokenizer.ggml.token_type arr[i32,32000]   = [2, 3, 3, 6, 6, 6, 6, 6, 6, 6, 6, 6, ...
llama_model_loader: - kv  16:                      tokenizer.ggml.merges arr[str,61249]   = ["▁ t", "e r", "i n", "▁ a", "e n...
llama_model_loader: - kv  17:                tokenizer.ggml.bos_token_id u32              = 1
llama_model_loader: - kv  18:                tokenizer.ggml.eos_token_id u32              = 2
llama_model_loader: - kv  19:            tokenizer.ggml.unknown_token_id u32              = 0
llama_model_loader: - kv  20:            tokenizer.ggml.padding_token_id u32              = 2
llama_model_loader: - kv  21:                    tokenizer.chat_template str              = {% for message in messages %}\n{% if m...
llama_model_loader: - kv  22:               general.quantization_version u32              = 2
llama_model_loader: - type  f32:   45 tensors
llama_model_loader: - type q3_K:   89 tensors
llama_model_loader: - type q4_K:   62 tensors
llama_model_loader: - type q5_K:    4 tensors
llama_model_loader: - type q6_K:    1 tensors
print_info: file format = GGUF V3 (latest)
```



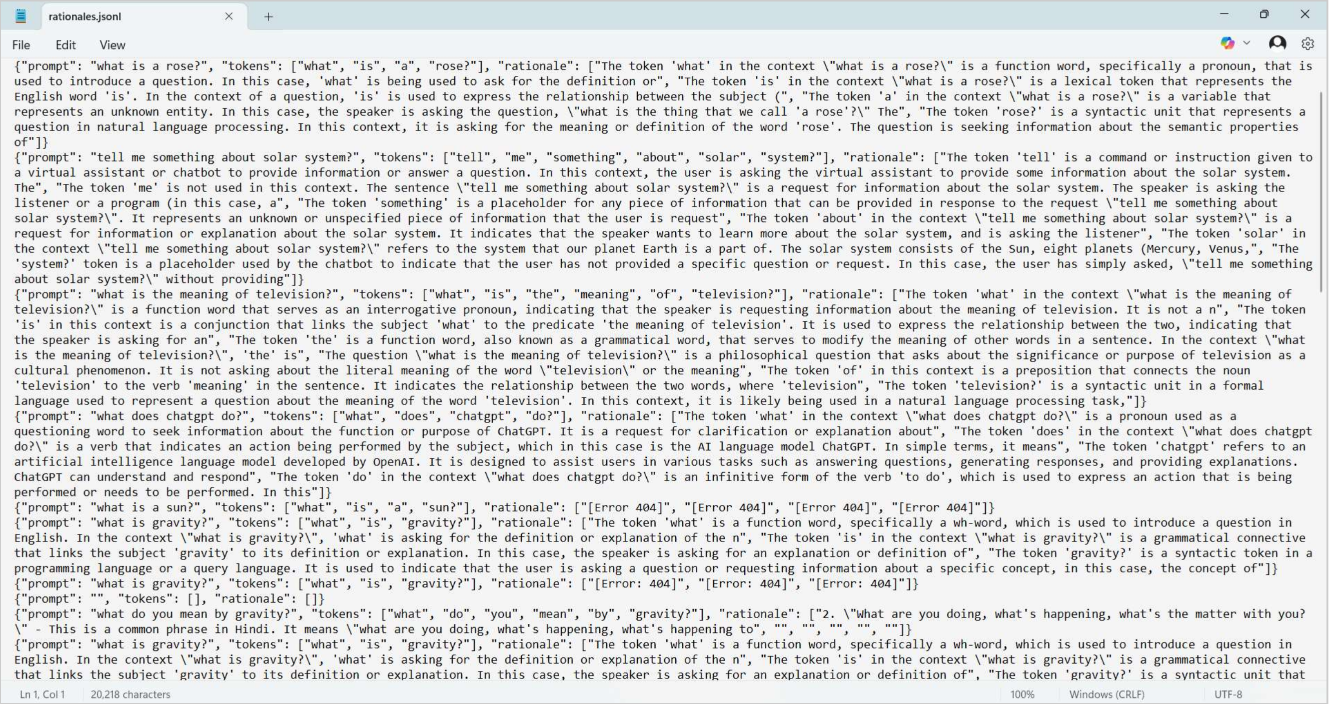
```
 <|user|>
what is a black hole?
<|assistant|>
A black hole is a region in space where the gravitational pull is so strong that nothing, not even light, can escape its pull. It is a theoretical object that ha
s never been observed directly. The name "black hole" is derived from the idea that the object's matter is black, or very dark, due to the strong gravitational p
ull.

> 
```

Quantized TinyLlama and Phi models running in CPU with llama.cpp



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	7.57451	10.7724	13.3054	7.28908	5.38134	3.8405	7.39575	10.4121	6.67423	8.49312	5.8656	6.28804	10.8022	9.14951	6.44468	6.36564	10.0975	9.92662	10.17	9.47995	9.61489	8.54739	8.3695
2	8.87275	5.37631	9.5484	10.6661	7.8892	6.9035	10.3465	8.03966	8.54929	10.1271	8.76431	6.25311	7.89123	7.49855	9.13866	7.77058	10.1359	8.36381	9.13204	7.63001	8.18607	5.44254	5.78053
3	8.9415	11.5204	13.3728	10.1715	8.27114	6.28478	9.32426	10.034	10.1808	9.97839	8.15895	6.26358	12.2075	8.32581	6.93706	9.23804	10.7599	10.794	10.4989	9.59913	9.63722	8.11012	7.90485
4	9.86479	12.8551	16.1256	12.1957	9.40264	7.19448	9.48883	12.0037	11.6771	12.0774	9.52155	7.41054	14.6551	9.11766	7.96586	11.0694	13.3483	12.7128	12.7611	11.924	11.9987	10.4355	10.272
5	12.4399	12.8383	12.548	10.5602	12.3955	12.7211	15.0607	10.6348	10.8047	12.5448	11.5966	9.13575	11.5657	12.2361	9.1383	8.64144	10.5876	11.7458	10.7578	9.57289	11.1322	9.29618	10.2021
6	9.73492	7.68685	9.57154	5.79091	9.60741	7.50766	7.57219	7.71323	8.8455	10.4586	8.55138	5.34962	8.40791	6.66603	6.74479	6.21846	8.83094	10.7635	10.3306	9.23727	9.32304	7.18546	6.8147
7	12.8925	10.9322	15.1992	9.80797	8.80417	8.96369	9.91113	11.5574	8.50595	13.8983	10.1975	8.22132	13.4625	10.855	8.14268	8.98527	11.6666	14.4396	13.61	13.2431	13.0027	10.2521	9.53435
8	12.143	10.538	16.7971	8.95113	8.37936	8.45315	10.3755	11.8164	6.55649	14.7179	9.51769	6.56543	14.7884	9.68159	8.03056	9.6031	12.8365	16.6601	16.0265	15.5254	15.4608	12.2531	11.2762
9	4.00569	1.55189	7.93223	3.22238	2.82987	0.890072	2.78671	3.15257	3.17919	4.60294	2.47123	4.14456	5.82728	6.49756	2.32833	2.0656	7.24713	10.0252	9.53179	8.86539	8.14254	6.50272	6.11401
10	10.4787	6.64403	11.548	7.96318	5.71585	6.17248	8.01587	8.66754	8.75309	7.02072	5.82777	10.3436	13.3259	16.8137	7.37191	9.21861	14.4658	16.005	15.1698	14.3442	13.9258	12.038	11.0507
11	11.711	7.4238	9.93324	6.03436	6.63673	7.05406	6.41268	9.14626	9.42056	6.81717	5.85464	8.20896	9.72784	13.9353	7.61841	7.24643	13.5763	13.2991	12.8797	11.9068	12.8151	9.43401	8.48392
12	12.8316	6.47766	8.81923	2.75281	6.63544	5.55121	9.05383	8.92423	8.4801	7.36513	6.47999	11.3281	10.2949	13.7662	10.7223	2.88667	9.54668	9.48307	9.06549	8.78948	7.92164	5.63062	4.52249
13	11.8859	8.25572	11.613	5.9244	8.15377	6.0579	6.94117	12.2248	6.76593	14.6859	9.45953	6.5424	15.9685	8.11595	6.12178	9.74506	21.642	14.0301	12.4074	11.9012	12.0439	9.52461	8.80334
14	12.6989	10.1765	14.1143	8.10283	9.30291	8.87053	8.82652	13.5497	7.78325	16.9615	10.8961	8.23005	17.4066	10.2556	8.37358	10.5052	24.4489	15.3264	14.407	14.67	15.5105	12.1062	11.7764
15	7.36695	5.2175	18.9764	5.63198	5.6029	4.78858	5.5417	9.57034	7.29362	10.1866	6.82728	7.33746	10.5301	9.88343	9.18921	3.91643	14.297	5.13444	5.14894	5.03365	6.47211	4.30411	4.20296
16	11.1764	7.34939	14.2215	5.90944	8.28979	6.72256	7.79764	9.63662	9.48412	9.52119	7.28611	11.7333	15.6577	14.4307	10.546	4.52779	16.2682	7.91375	8.5659	9.152	9.32754	6.86817	7.30555
17	10.4571	7.88483	10.949	5.24856	6.56527	6.45874	6.84555	9.76797	12.6811	9.40603	8.03976	11.5658	14.4333	19.0316	9.69217	3.22721	8.5078	5.80306	5.15444	4.90461	4.44111	3.55769	2.83926
18	8.69258	8.37173	10.0287	7.04291	5.35674	4.78015	6.80296	8.67083	4.14521	13.2085	8.07429	5.31027	13.6273	9.08461	3.93918	7.4874	13.9179	9.3168	8.19571	7.40422	7.90267	6.05752	4.93754
19	6.55235	4.86101	2.8126	0.352449	1.80622	2.44695	4.53489	5.43848	5.52089	4.53926	2.52072	12.3916	8.425	10.3375	9.19002	0.712126	4.36189	6.53412	6.87676	5.37378	6.4995	2.99392	1.85327
20	3.24832	1.28628	-0.29092	0.329281	-1.86849	-1.49621	1.00697	3.39876	3.61925	3.64889	1.20349	5.52892	4.51348	6.69398	4.52448	-2.27283	1.8393	2.91811	2.46746	0.874455	0.908424	-0.64769	-1.64074
21	7.79757	5.67203	3.84332	2.34191	2.66145	2.27057	5.58144	8.88112	6.60162	7.77564	3.66624	15.6208	12.4186	12.0974	13.3398	4.09242	8.25386	6.03708	6.41838	6.09798	7.48226	4.51454	4.01732
22	2.87332	8.97904	3.0257	0.877979	0.798387	-0.98511	1.01139	0.824969	0.835247	-0.14077	-1.2585	-1.15635	-2.55066	-0.91936	-0.44767	1.01291	2.8879	-1.97747	-1.05887	-0.85238	-1.17317	1.65568	0.203007
23	-6.07865	-6.10872	5.80613	-3.40932	-4.49064	-3.69846	-5.24073	-7.6968	-6.98819	-6.53533	-6.52648	-0.34905	3.08363	5.62539	-7.0052	-1.81054	-6.0884	-6.57131	-7.13175	-5.59915	-7.13126	-6.39967	-7.35171
24	-9.0879	-9.25549	4.50396	-5.9637	-3.74252	-7.25549	-4.78363	-7.7233	-7.98636	-6.55269	-6.2062	-2.24819	0.362482	6.3035	-6.52265	-5.58745	-9.10128	-7.80491	-7.92859	-2.64405	-7.95272	-7.53645	-8.28788
25	-6.60215	-6.03555	0.279125	-4.73722	-2.30636	-2.80688	-2.917	-5.47635	-6.34857	-4.90321	-4.92455	-3.21398	-0.58871	3.67353	-5.51058	-5.28066	-6.60862	-6.06266	-5.76212	-2.84493	-6.15789	-5.85881	-6.21024
26	-7.89992	-7.48815	-0.26046	-5.62984	-5.4727	-5.24024	-4.4522	-6.40498	-7.23716	-6.3408	-6.89869	-10.1159	-0.24831	2.36892	-6.45434	-6.93643	-7.90132	-7.62986	-6.07492	-4.33536	-6.08266	-6.38218	-6.43105
27	-6.84096	-6.4936	-0.09621	-6.57476	-4.12338	-4.3227	-4.20836	-6.35633	-8.48929	-6.20873	-8.24438	-4.24495	1.64731	2.42338	-5.30558	-4.43661	-6.83842	-6.81348	-6.97367	-4.32566	-6.34489	-6.9214	-6.55888
28	-5.91088	-5.893	1.301	-6.16359	-2.96749	-4.04531	-3.16637	-6.9824	-7.30586	-5.85606	-7.10491	-6.77619	2.27938	3.32631	-5.11077	-6.76373	-5.91285	-7.2317	-6.77695	-2.76949	-6.3396	-6.69923	-6.69362
29	-7.70558	-7.52748	-0.23169	-5.07814	-2.63207	-3.60704	-3.83339	-6.36963	-6.02754	-5.51219	-6.25563	-5.27703	-0.80138	2.44524	-7.0817	-5.60385	-7.71464	-6.70357	-5.08592	-3.61384	-6.0909	-6.24474	-5.88147

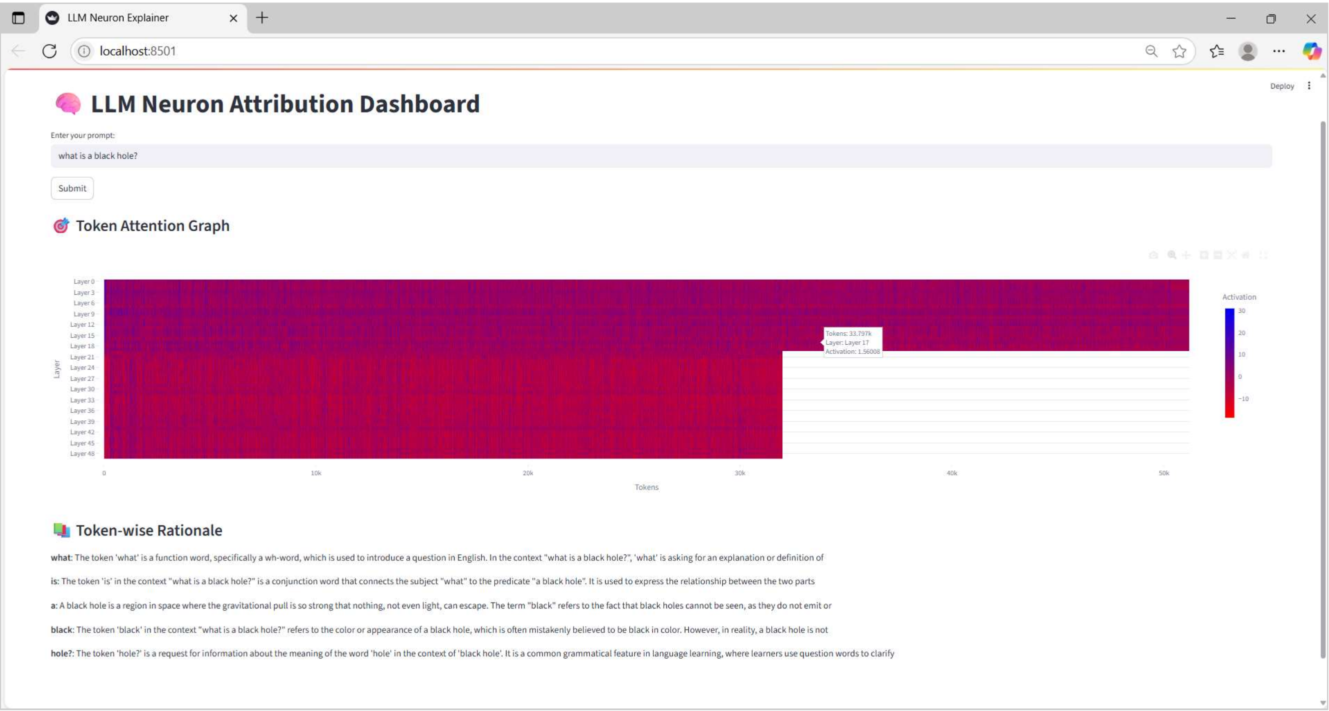
Neuron activations cached in logits_activations.csv



```
{"prompt": "what is a rose?", "tokens": ["what", "is", "a", "rose?"], "rationale": ["The token 'what' in the context \"what is a rose?\" is a function word, specifically a pronoun, that is
used to introduce a question. In this case, 'what' is being used to ask for the definition or", "The token 'is' in the context \"what is a rose?\" is a lexical token that represents the
English word 'is'. In the context of a question, 'is' is used to express the relationship between the subject (", "The token 'a' in the context \"what is a rose?\" is a variable that
represents an unknown entity. In this case, the speaker is asking the question, \"what is the thing that we call 'a rose'?\" The", "The token 'rose?' is a syntactic unit that represents a
question in natural language processing. In this context, it is asking for the meaning or definition of the word 'rose'. The question is seeking information about the semantic properties
of"]}
{"prompt": "tell me something about solar system?", "tokens": ["tell", "me", "something", "about", "solar", "system?"], "rationale": ["The token 'tell' is a command or instruction given to
a virtual assistant or chatbot to provide information or answer a question. In this context, the user is asking the virtual assistant to provide some information about the solar system.
The", "The token 'me' is not used in this context. The sentence \"tell me something about solar system?\" is a request for information about the solar system. The speaker is asking the
listener or a program (in this case, a", "The token 'something' is a placeholder for any piece of information that can be provided in response to the request \"tell me something about
solar system?\". It represents an unknown or unspecified piece of information that the user is request", "The token 'about' in the context \"tell me something about solar system?\" is a
request for information or explanation about the solar system. It indicates that the speaker wants to learn more about the solar system, and is asking the listener", "The token 'solar' in
the context \"tell me something about solar system?\" refers to the system that our planet Earth is a part of. The solar system consists of the Sun, eight planets (Mercury, Venus,", "The
'system?' token is a placeholder used by the chatbot to indicate that the user has not provided a specific question or request. In this case, the user has simply asked, \"tell me something
about solar system?\" without providing"]}
{"prompt": "what is the meaning of television?", "tokens": ["what", "is", "the", "meaning", "of", "television?"], "rationale": ["The token 'what' in the context \"what is the meaning of
television?\" is a function word that serves as an interrogative pronoun, indicating that the speaker is requesting information about the meaning of television. It is not a n", "The token
'is' in this context is a conjunction that links the subject 'what' to the predicate 'the meaning of television'. It is used to express the relationship between the two, indicating that
the speaker is asking for an", "The token 'the' is a function word, also known as a grammatical word, that serves to modify the meaning of other words in a sentence. In the context \"what
is the meaning of television?\", 'the' is", "The question \"what is the meaning of television?\" is a philosophical question that asks about the significance or purpose of television as a
cultural phenomenon. It is not asking about the literal meaning of the word \"television\" or the meaning", "The token 'of' in this context is a preposition that connects the noun
'television' to the verb 'meaning' in the sentence. It indicates the relationship between the two words, where 'television", "The token 'television?' is a syntactic unit in a formal
language used to represent a question about the meaning of the word 'television'. In this context, it is likely being used in a natural language processing task,"]}
{"prompt": "what does chatgpt do?", "tokens": ["what", "does", "chatgpt", "do?"], "rationale": ["The token 'what' in the context \"what does chatgpt do?\" is a pronoun used as a
questioning word to seek information about the function or purpose of ChatGPT. It is a request for clarification or explanation about", "The token 'does' in the context \"what does chatgpt
do?\" is a verb that indicates an action being performed by the subject, which in this case is the AI language model ChatGPT. In simple terms, it means", "The token 'chatgpt' refers to an
artificial intelligence language model developed by OpenAI. It is designed to assist users in various tasks such as answering questions, generating responses, and providing explanations.
ChatGPT can understand and respond", "The token 'do' in the context \"what does chatgpt do?\" is an infinitive form of the verb 'to do', which is used to express an action that is being
performed or needs to be performed. In this"]}
{"prompt": "what is a sun?", "tokens": ["what", "is", "a", "sun?"], "rationale": ["[Error 404]", "[Error 404]", "[Error 404]", "[Error 404]"]}
{"prompt": "what is gravity?", "tokens": ["what", "is", "gravity?"], "rationale": ["The token 'what' is a function word, specifically a wh-word, which is used to introduce a question in
English. In the context \"what is gravity?\", 'what' is asking for the definition or explanation of the n", "The token 'is' in the context \"what is gravity?\" is a grammatical connective
that links the subject 'gravity' to its definition or explanation. In this case, the speaker is asking for an explanation or definition of", "The token 'gravity?' is a syntactic token in a
programming language or a query language. It is used to indicate that the user is asking a question or requesting information about a specific concept, in this case, the concept of"]}
{"prompt": "what is gravity?", "tokens": ["what", "is", "gravity?"], "rationale": ["[Error: 404]", "[Error: 404]", "[Error: 404]"]}
{"prompt": "", "tokens": [], "rationale": []}
{"prompt": "what do you mean by gravity?", "tokens": ["what", "do", "you", "mean", "by", "gravity?"], "rationale": ["2. \"What are you doing, what's happening, what's the matter with you?
\" - This is a common phrase in Hindi. It means \"what are you doing, what's happening, what's happening to", "", "", "", "", ""]}
{"prompt": "what is gravity?", "tokens": ["what", "is", "gravity?"], "rationale": ["The token 'what' is a function word, specifically a wh-word, which is used to introduce a question in
English. In the context \"what is gravity?\", 'what' is asking for the definition or explanation of the n", "The token 'is' in the context \"what is gravity?\" is a grammatical connective
that links the subject 'gravity' to its definition or explanation. In this case, the speaker is asking for an explanation or definition of", "The token 'gravity?' is a syntactic unit that
```

Zephyr generates token-level rationales for each prompt

Streamlit Visualization Dashboard



🧠 LLM Neuron Attribution Dashboard

Enter your prompt:

what is a black hole?

🎯 Token Attention Graph

📚 Token-wise Rationale

what: The token 'what' is a function word, specifically a wh-word, which is used to introduce a question in English. In the context "what is a black hole?", 'what' is asking for an explanation or definition of

is: The token 'is' in the context "what is a black hole?" is a conjunction word that connects the subject "what" to the predicate "a black hole". It is used to express the relationship between the two parts

a: A black hole is a region in space where the gravitational pull is so strong that nothing, not even light, can escape. The term "black" refers to the fact that black holes cannot be seen, as they do not emit or

black: The token 'black' in the context "what is a black hole?" refers to the color or appearance of a black hole, which is often mistakenly believed to be black in color. However, in reality, a black hole is not

hole?: The token 'hole?' is a request for information about the meaning of the word 'hole' in the context of 'black hole'. It is a common grammatical feature in language learning, where learners use question words to clarify

- User enters a prompt into the dashboard.
- The app looks up cached activations and explanations.
- Plotly is used to render the token-by-layer attention heatmap.
- Below the heatmap, Zephyr-generated rationales are shown for each token.

This architecture ensures the system remains CPU-friendly, while still offering rich interpretability features for education, debugging, and LLM transparency.