Phase 2 Deliverables

Summary of Code-Base changes

Removing KV caching

llama/model.py:

- Commented out `kv_caching` flag in `ModelArgs`.
- Disabled all cache tensor pre-allocation and writes.
- Added non-cached fallback (`keys = xk`, `values = xv`).

llama/generation.py:

- Removed `kv_caching` argument from `Generation.generate()`.
- Simplified forward pass: always call `self(tokens[:, :cur_pos])`.
- Deleted cache-specific logic that updated `prev_pos`.

inference.py:

- Commented out old cache-enabled call; added new cache-free call.

Benchmark Testing

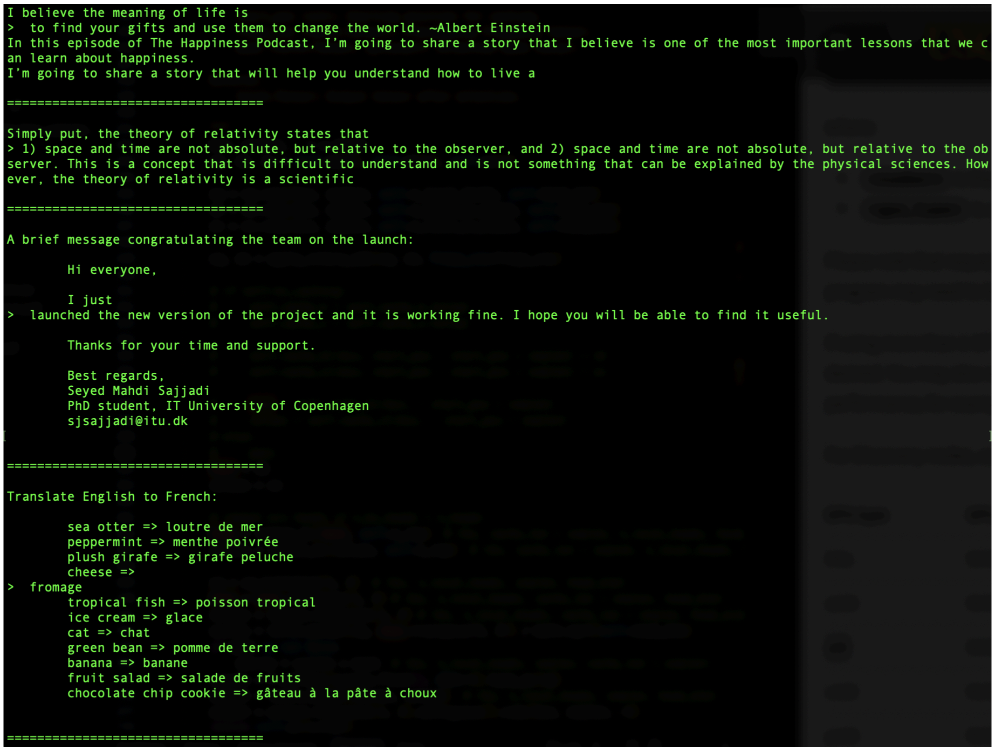
benchmark_inference.py:

- Updated call site to new `generate()` signature for testing purposes of this phase.

More detailed line-by-line changes can be viewed here:
https://github.com/uscmlsystems/ml-systems-final-project-Joshetaa/commit/09264895a68435e6d04ccc02746c7cc4d7a46a95

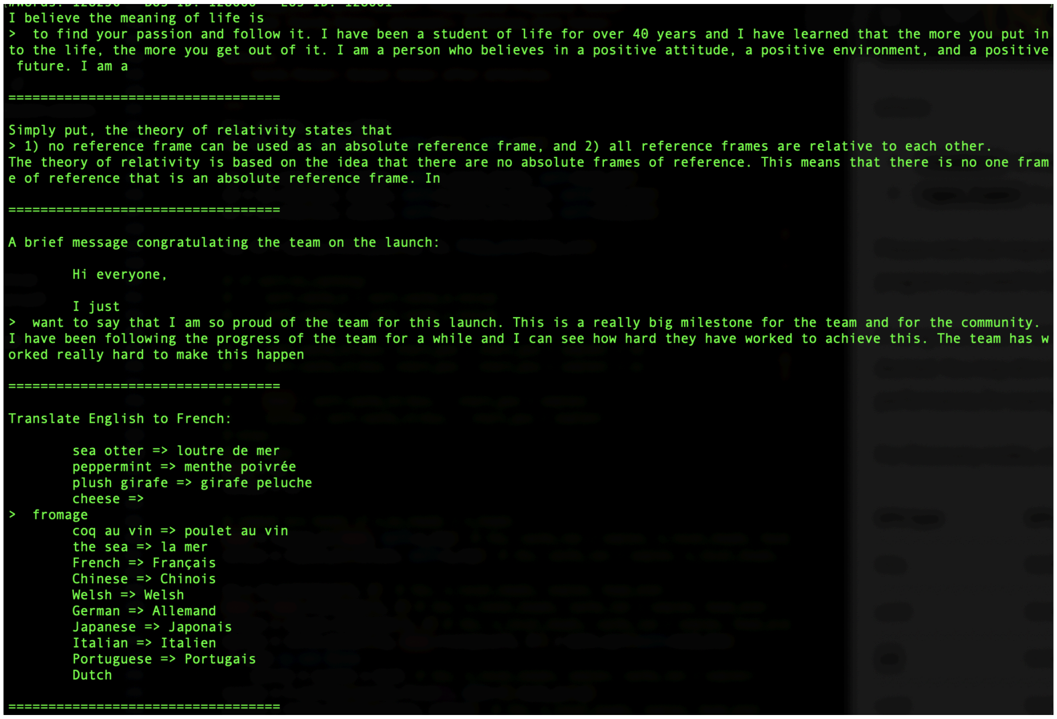
Comparison of model before and after KV cache removal

Before Cache removal



```
I believe the meaning of life is
>  to find your gifts and use them to change the world. ~Albert Einstein
In this episode of The Happiness Podcast, I’m going to share a story that I believe is one of the most important lessons that we c
an learn about happiness.
I’m going to share a story that will help you understand how to live a

==================================

Simply put, the theory of relativity states that
> 1) space and time are not absolute, but relative to the observer, and 2) space and time are not absolute, but relative to the ob
server. This is a concept that is difficult to understand and is not something that can be explained by the physical sciences. How
ever, the theory of relativity is a scientific

==================================

A brief message congratulating the team on the launch:

        Hi everyone,

        I just
>  launched the new version of the project and it is working fine. I hope you will be able to find it useful.

        Thanks for your time and support.

        Best regards,
        Seyed Mahdi Sajjadi
        PhD student, IT University of Copenhagen
        sjsajjadi@itu.dk


==================================

Translate English to French:

        sea otter => loutre de mer
        peppermint => menthe poivrée
        plush girafe => girafe peluche
        cheese =>
>  fromage
        tropical fish => poisson tropical
        ice cream => glace
        cat => chat
        green bean => pomme de terre
        banana => banane
        fruit salad => salade de fruits
        chocolate chip cookie => gâteau à la pâte à choux


==================================
```

After Cache Removal



```
I believe the meaning of life is
>  to find your passion and follow it. I have been a student of life for over 40 years and I have learned that the more you put in
to the life, the more you get out of it. I am a person who believes in a positive attitude, a positive environment, and a positive
 future. I am a

==================================

Simply put, the theory of relativity states that
> 1) no reference frame can be used as an absolute reference frame, and 2) all reference frames are relative to each other.
The theory of relativity is based on the idea that there are no absolute frames of reference. This means that there is no one fram
e of reference that is an absolute reference frame. In

==================================

A brief message congratulating the team on the launch:

        Hi everyone,

        I just
>  want to say that I am so proud of the team for this launch. This is a really big milestone for the team and for the community.
I have been following the progress of the team for a while and I can see how hard they have worked to achieve this. The team has w
orked really hard to make this happen

==================================

Translate English to French:

        sea otter => loutre de mer
        peppermint => menthe poivrée
        plush girafe => girafe peluche
        cheese =>
>  fromage
        coq au vin => poulet au vin
        the sea => la mer
        French => Français
        Chinese => Chinois
        Welsh => Welsh
        German => Allemand
        Japanese => Japonais
        Italian => Italien
        Portuguese => Portugais
        Dutch

==================================
```

Qualitative Output Comparison (descriptive)

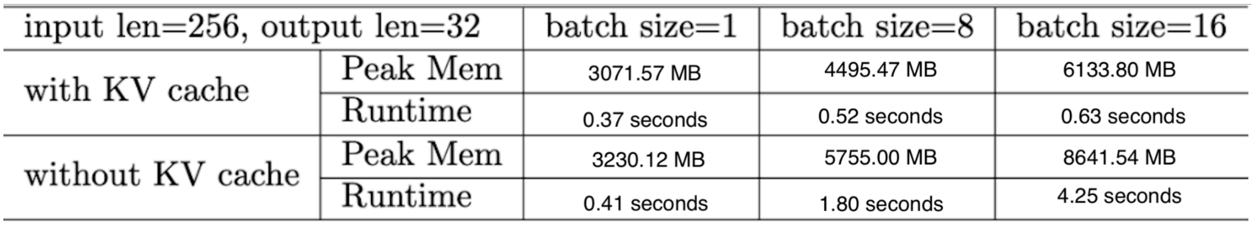
When KV-caching was enabled, the model's continuations tended toward longer, slightly more polished prose. For the prompt **"I believe the meaning of life is …"**, the cached run completed the sentence with an inspirational style that even quoted Einstein ("… to find your gifts and use them to change the world — Albert Einstein"). After removing the cache, the same prompt produced a shorter, more conversational answer ("… to find your passion and follow it. I have been a student of life for over 40 years…"). Both are on-topic, but the cache-free output feels less 'flowery' language.

A similar shift shows up in the physics prompt. With caching, the model began, "Simply put, the theory of relativity states that space and time are not absolute, but relative to the observer…," whereas the cache-free version reverted to the textbook framing about reference frames ("… no reference frame can be used as an absolute reference frame, and all frames are relative to each other…"). Content remains correct; only the phrasing differs. Although pre-cache removal, there seems to be redundant repetition.

For the **team-launch congratulatory email**, the cached model generated a full note complete with sign-off, name, and academic affiliation—clearly elaborated beyond the minimal ask. The cache-free model kept things shorter and more generic, omitting the formal signature block. Again, semantics align, but verbosity is reduced without the cache.

Finally, in the **English→French lexicon prompt** ("cheese ⇒ ?"), both versions correctly answered *fromage* and then free-associated additional word pairs. The cached run listed food items like *tropical fish* and *ice cream*, while the cache-free run chose examples such as *coq au vin* and language names (Welsh, German). The differences are arbitrary expansions rather than errors, suggesting here that disabling KV-caching affects stylistic sampling choices more than factual accuracy.

Benchmarking



input len=256, output len=32		batch size=1	batch size=8	batch size=16
with KV cache	Peak Mem	3071.57 MB	4495.47 MB	6133.80 MB
	Runtime	0.37 seconds	0.52 seconds	0.63 seconds
without KV cache	Peak Mem	3230.12 MB	5755.00 MB	8641.54 MB
	Runtime	0.41 seconds	1.80 seconds	4.25 seconds

1. Memory behaviour

At every batch size the cache-enabled run consumes **less** memory than the cache-free run.

- **Batch 1:** 3.07 GB with cache vs 3.23 GB without (≈ +5 %).
- **Batch 8:** 4.50 GB with cache vs 5.76 GB without (≈ +28 %).
- **Batch 16:** 6.13 GB with cache vs 8.64 GB without (≈ +41 %).

The gap widens as the batch size grows because, without caching, each forward step must materialise fresh key/value projections for *all* decoder layers, whereas the cached variant stores them once and simply appends new tokens. Consequently, memory overhead scales almost linearly with sequence-length × batch-size in the cache-free setting.

2. Runtime behaviour

Caching also delivers consistent speed-ups, and the advantage becomes bigger for larger batches:

- **Batch 1:** 0.37 s (cached) vs 0.41 s (cache-free) — a modest 11 % slowdown.
- **Batch 8:** 0.52 s vs 1.80 s — roughly **3.5 ×** slower without the cache.
- **Batch 16:** 0.63 s vs 4.25 s — roughly **6.7 ×** slower.

Because the cache-free path recomputes attention for the entire prefix at every decoding step, its cost grows with both *sequence length* and *batch size*. In contrast, the cached version reuses previously computed key/value tensors and only attends over the newly generated token, hence the nearly flat runtime curve.

My Take-aways

- **When VRAM is plentiful:** keep KV caching turned **on**—it both lowers peak memory and accelerates generation, with benefits that compound for larger batches or longer outputs.
- **When VRAM is extremely limited or on CPU:** disabling the cache can still work (no correctness loss), but expect noticeably higher latency; you are trading compute for model-state simplicity.
- The latency penalty grows super-linearly with batch size, so cache-free inference is best reserved for single-request, low-throughput scenarios.