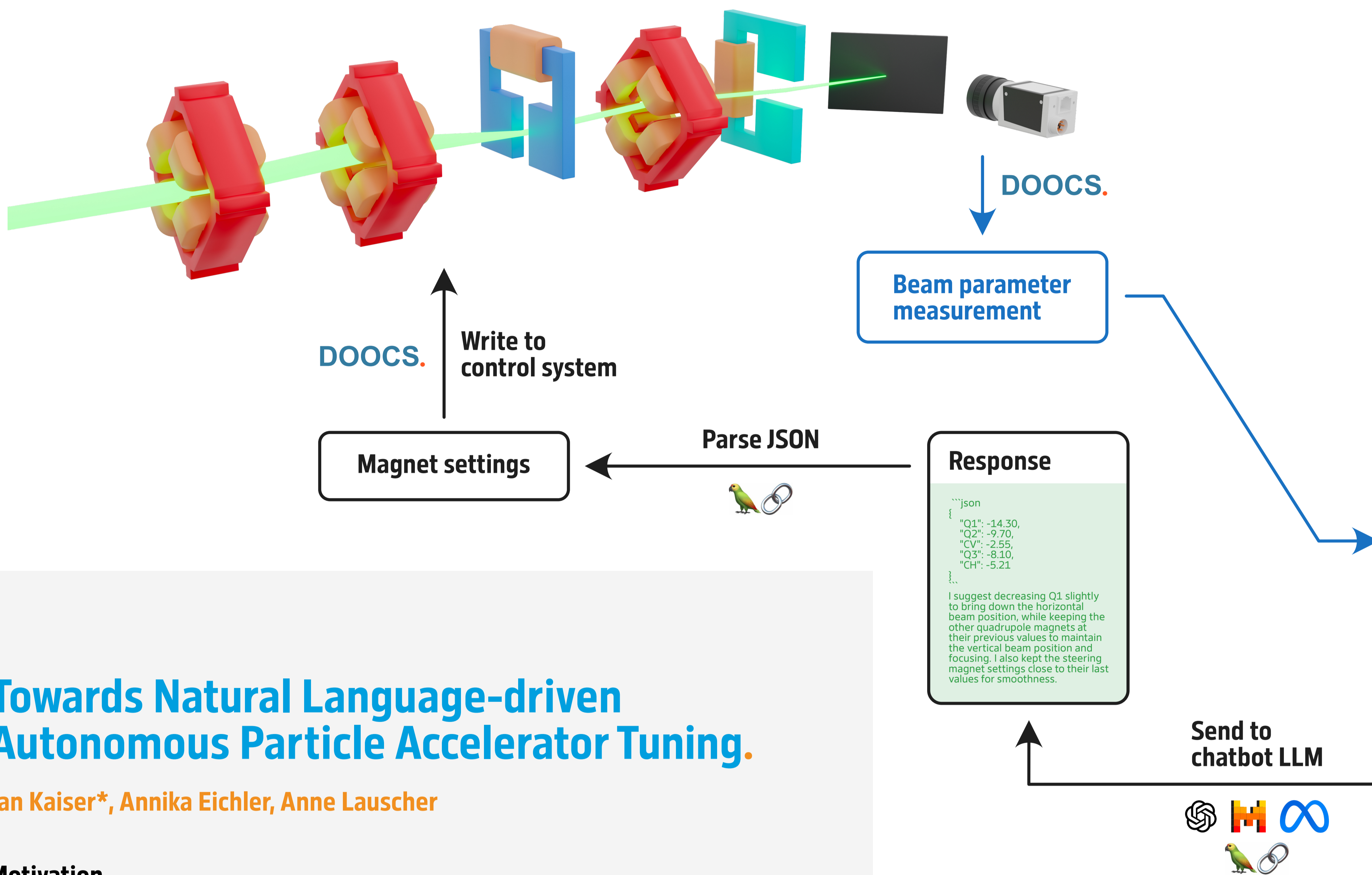


Large language models (LLMs) like GPT-4, which are really only trained to do text completion, can be (ab)used as optimisers to directly and autonomously tune a particle accelerator.



Prompt (explained tuning prompt)

Human: Now you will help me optimise the horizontal and vertical position and size of an electron beam on a diagnostic screen in a particle accelerator.

You are able to control five magnets in the beam line. The magnets are called Q1, Q2, CV, Q3 and CH.

Q1, Q2 and Q3 are quadrupole magnets. When their k1 strength is increased, the beam becomes more focused in the horizontal plane and more defocused in the vertical plane. When their k1 strength is decreased, the beam becomes more focused in the vertical plane and more defocused in the horizontal plane. When their k1 strength is zero, the beam is not focused in either plane. Quadrupole magnets might also steer the beam in the horizontal or vertical plane depending on their k0 strength, when the beam does not travel through the centre of the magnet. The range of the k1 strength is -30.0 to 30.0 m⁻².

CV is vertical steering magnet. When its deflection angle is increased, the beam is steered upwards. When its deflection angle is decreased, the beam is steered downwards. The range of the deflection angle is -6.0 to 6.0 mrad.

CH is horizontal steering magnet. When its deflection angle is increased, the beam is steered to the right. When its deflection angle is decreased, the beam is steered to the left. The range of the deflection angle is -6.0 to 6.0 mrad.

You are optimising four beam parameters: mu_x, sigma_x, mu_y, sigma_y. The beam parameters are measured in millimetres (mm). The target beam parameters are:

```

{
  "mu_x": 1.20,
  "sigma_x": 0.11,
  "mu_y": 1.25,
  "sigma_y": 0.06
}

```

Task description

Below are previously measured pairs of magnet settings and the corresponding observed beam parameters.

Magnet settings:

```

{
  "Q1": 25.12,
  "Q2": 12.48,
  "CV": 0.84,
  "Q3": -8.25,
  "CH": 3.94
}

```

Beam parameters:

```

{
  "mu_x": -1038.63,
  "sigma_x": 1893.75,
  "mu_y": -2353.77,
  "sigma_y": 2226.94
}

```

Previous samples

Give me new magnet settings that are different from all pairs above. The magnet settings you should propose should lead to beam parameters closer to the target or, if you do not have enough information yet, maximise information gain for finding new beam parameters. Do not set any magnet setting to zero. Smooth changes relative to the last magnet settings are preferred.

The output should be a markdown code snippet formatted in the following schema, including the leading and trailing ````json` and `````:

```

```json
{
 "Q1": float // k1 strength of the first quadrupole magnet
 "Q2": float // k1 strength of the second quadrupole magnet
 "CV": float // Deflection angle of the vertical steering magnet
 "Q3": float // k1 strength of the third quadrupole magnet
 "CH": float // Deflection angle of the horizontal steering magnet
}
```

```

Output instructions

Do not add comments to the output JSON.

Towards Natural Language-driven Autonomous Particle Accelerator Tuning.

Jan Kaiser*, Annika Eichler, Anne Lauscher

Motivation

- Large language models (LLMs) have most recently demonstrated incredible capabilities beyond simple text completion, such as programming and problem solving, with **emergent-behaviours** appearing as models become larger and better trained.
- This raises the question: **Can LLMs tune a particle accelerator?**
- Ultimately, this could be a step towards accelerator operators defining goals for accelerator operation through natural language, with LLMs figuring out how to get there.

Implementation

- We engineer a prompt made up of three main components:
 - In the **task description**, we explain the task and define a goal.
 - We then list a history of **previously collected samples** of inputs and resulting outputs.
 - At last, we provide **instructions for the output**, asking the model to output the next magnet settings and providing format instructions for later parsing.
- We test four different **prompting strategies** following the above scheme:
 - Tuning prompt:** Frames the task in terms of accelerator tuning, where the model is tasked to find magnet settings that result in specific target beam parameters. We assume the model knows how magnets in an accelerator work.
 - Explained tuning prompt:** We explain in simple terms how quadrupole magnets etc. work and how they influence the beam parameters. Otherwise like the tuning prompt.
 - Chain-of-thought prompt:** Ask the model to explain its reasoning before providing the next magnet settings. Otherwise like explained tuning prompt.
 - Optimisation prompt:** Frames the task as a simple optimisation, where the output is an objective value. The model is unaware it is tuning an accelerator.
- Implementation of prompt creation with *LangChain*. LLMs are run through the *OpenAI* API or with *Ollama* on A100 GPUs on the DESY Maxwell cluster.

Results

- Success depends on model and prompting scheme:
 - Often successful with GPT-4 / GPT-4-Turbo** and explained tuning prompt.
 - Few successes with capable open source models like *Mixtral 7x8B* and even small ones like *Mistral 7B* and *Starling LM*.
 - Some models, like *Llama 2 70B* and *Orca 2 13B*, fail because they explain a strategy instead of providing the next magnet settings. Others, like *GPT-3.5-Turbo*, simply fail to show reasonable behaviour.

Outlook

- More extensive evaluation of **large commercial models**, including *Gemini* and *Claude*, as well as inevitably **better future models** (*Gemma?* *GPT-5?*).
- Test potentially better suited prompting schemes like **ReAct**.

*Corresponding author: jan.kaiser@desy.de

Deutsches Elektronen-Synchrotron DESY
A Research Centre of the Helmholtz Association



Scan me to download the poster!

Cannot be helped ...
GPT 3.5 Turbo (chain-of-thought prompt)

Better!
Mixtral 7x8B (explained prompt)

Good, but failed at formatting once tuned.
GPT 4 (explained prompt)

Very good!
GPT 4 (optimisation prompt)

Example tuning runs by LLMs

