

Handout to *Chat de Schrödinger: Both Adept and Naïve*

konrad.jan.szymanski@gmail.com

May 10, 2023

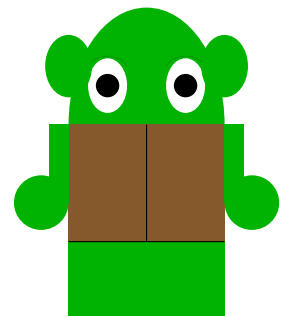
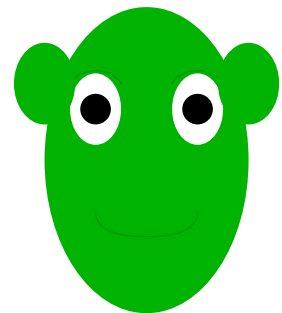
How to access ChatGPT

Go to <https://chat.openai.com/> and create an account. As of writing this text, the basic model is available for free (with quotas), while the access to more capable ones (with longer working memory etc.) costs 20 USD/month. All of the conversations are accessible from any logged in device, there are no subaccounts.

Basic usage

Select a previous conversation or click *New chat* and type in your request. It is particularly useful in dealing with unknown API/libraries and simple text generation:

- *Please draw Shrek – the green ogre – using TikZ.*
- *Write a function in Python that downloads the last week entries from quant-ph section of arxiv and returns them as a list of dicts.*
- *Read the following exercise for a quantum optics tutorial and write an introduction to it (about 3-4 sentences, providing the context and purpose of the calculation).*
- *Reply to the following e-mail – politely reject the offer.* (Note: the response is likely to end up being flagged as spam – I think the text written by ChatGPT is ‘watermarked’ as AI-generated.)



Quirks and limitations

The AI has sometimes struggles to do what is expected from it for various reasons:

- **Hallucinations:** The model can generate plausible-sounding but incorrect or nonsensical information.
- **Memory loss:** In prolonged discussions, ChatGPT may lose track of earlier parts of the conversation.
- **No Internet access:** The model is not able to access the web to fetch up-to-date information or verify claims.
- **Fitting to expectations:** If prompted with a buggy code, it might produce more bugs later even though it knows better.
- **Limited precision:** It may struggle with tasks requiring high precision, such as matching long parentheses strings or performing multi-qubit Pauli multiplication.
- **Limited training data** for certain tasks, e.g. writing Mathematica code.

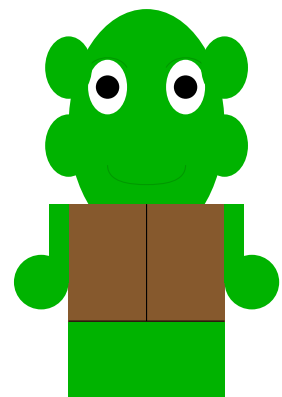


Figure 1: *Top:* Result of the first prompt example (*TikZ Shrek*). *Center, Bottom:* Successive iterations after asking for more features and improvements. (Prompt: How do I add a figure on the side of a standard LaTeX document?)

Advanced usage strategies

To make the most out of ChatGPT in scientific work, consider the following strategies:

- **First prompt priority:** The initial prompt sets the context and focus for the AI's response, so make it as clear and concise as possible to get the desired outcome.
- **Bite-sized prompts:** Break down complex tasks into smaller, more manageable steps, and request ChatGPT to complete each step individually.

Programming assistance

- For complex programming tasks specify the goal, context and other requirements (e.g. that the code should be commented) clearly in the first prompt. For example, I used the following one to (co)develop a graph state simulations library:

Your job is to help me with Python code development. I'm writing a library (using qutip) to simulate graph states. I will ask you to propose solutions, refactor code, and write parts of the code for me. Please be concise and write only the parts that need to change + very little context. Nothing more, that's very important so that I understand what you're doing. In the code you generate, please provide comments (both function/class descriptions and next to the instructions). If you spot any bug, please let me know.

- Think about the structure of what you want to build. Ask the AI to write the code for the basic functions first, then to pack it into classes. Iterate for whatever you want to build.
- Finally, debug the code (very important). Bonus points if you ask ChatGPT to write the checks.

Keyword search

- **Detailed problem description:** Describe your problem as comprehensively as possible, without including specific mathematical notation – the AI doesn't excel at mathematics (yet), but seems to understand the general structure of physics research. Consider mentioning the relations between your problem and other research areas or mathematical concepts, even if hypothetical (but clearly state that they are so).
- **Multiple angles:** Present your problem from different perspectives to cover various aspects and provide a broader context for the AI to work with.
- **Request for keywords:** Ask ChatGPT to suggest relevant keywords or concepts from physics, mathematics, or computer science that may be related to your problem. This can help you find similar structures or ideas in existing research, so you have a starting point for more detailed search.

For example, this is the prompt I used when optimizing a qubit measurement scheme, and it identified several approaches, one of which I implemented in the end (Monte Carlo Tree Search):

I have a game-theoretical problem. There's only a single player - me, and initially there is a list of length N , containing copies of a symbol I . So, it looks like $[I, I, \dots, I]$. At every step I need to choose an index which hasn't been chosen before, and one of the two actions: X , or Y . Then, a sign (+ or -) is chosen at random, with probability depending on the state of the list just before the action and the choice of X or Y . The symbol I is then replaced with tuple $(X/Y,+/-)$. The game ends when all indices have been used, and with every end state of the list there is an associated score S . The goal is to maximize the expected score, knowing the probabilities of measuring +/- depending on choice of X/Y and previous state of the list. The entire game tree with transition probabilities is therefore known. So, the game for a list of size $N=3$ might go as follows:

1. Initially there is a list $[I, I, I]$.
2. I choose the index 1, action X . By random I get the sign +, and the list gets replaced with $[(X,+), I, I]$.
3. The game continues: I choose index 3, action Y , get minus. The list is now $[(X,+), I, (Y,-)]$.
4. In the end I choose the only valid (remaining) index 2, action X , and get minus sign: $[(X,+), (X,-), (Y,-)]$. This is the end state of the game, it has some score S .

How to maximize the expected score? Are there some general methods or algorithms?

Additional links

- **Early example of a language model:**

karpathy.github.io/2015/05/21/rnn-effectiveness

- **Stephen Wolfram article:**

writings.stephenwolfram.com/2023/02/what-is-chatgpt-doing-and-why-does-it-work