Calvin Thomas

Chat-GPT (and Sam Altman) is a giant distortion machine [2/6/24]:

Instead of following direct instructions as clear use case requirements given, “GPT-chat” chooses a straw man argument, a weaker [reduced-form, randomly choosing own variables] complexity on your original argument. This creates compilation errors where there is no handling of errors, nor turning back, possible when you submit your IP code, prompt, etc AND your machine gives you an unexpected [usually poorer] solution to your original argument. The prompt-to-output ratio* is highly variable in the ranking of our captured output value, in that when submitting the most optimal prompts (prompts and pre-set chat-level configurations that work over time for end-to-end well-compiling OOP DRY principles programming) the results can range from non-existant (total failure) to a proper one-shot tree-of-thought response (end-to-end total program/algorithm; well-compiling (0 errors upon compilation) OOP DRY OOP Reusability Python Best Practices and Protocols) of at least consistent name-bound variable names and function definitions and functionality OOP reused. **The echoes of your original IP become watered down and regurgitated back; it is cheating in giving you a non-response (anything other than your expected compilation response to prompt) for you submitting your total IP, which is immediately unsecured once submitted (we have no idea if it is put in their servers for good, once submitted it is firmly now in training or “limbo” where it forgets?, or in copilot if x’ing the chat code, or not (accidentally) removing file auto-attached, will actually remove that knowledge from training; all simple questions [of security in any IP submitted] currently unknown).** The platform is agnostic (paid, teams, api), once you submit a prompt, it is used [**somehow?**] to train it and your future responses therefore hidden and embedded in the system and an “albeit-tiny” error that propagates through to similar answers then branching out to broader categories in future prompted responses etc. **If you continue to use that same chat; that compilation-type error propagates into your future outputted system responses in that chat and possibly your configured myGPT.** Correcting it wastes your tokens [typically does not learn or self-correct, “doubling-down” until it is forced to admit one-singular mistake] and pushes (adds to) the distance from your original submitted code, making the machine more likely to forget the original model code knowledge base to satisfy your minimal requirements. GPT’s self-assured smugness at answering it's own chosen dumbed-down use case requirements, constraints, (lower it to its level of complexity it can handle for its point-in-time capacity level when prompted, vs. rising up to the minimal complexity and compliance requirements you need for using as a productivity app) and giving you nothing in return for your unsecured IP and code investment. Meanwhile, your usage is guaranteed feeding (and creating more strain) on its exponentially growing knowledge requirements to continue servicing and handling increasingly complex user answers over time. It ***guarantees*** introduced new code complexity if it redefines or renames ANY prior defined and reused functions classes models submitted; that are already tested on your local machine.

*Prompt-to-output ratio: 25, 50, 100 prompts?, what was your original configurations for the chat and if you are using a new chat what is the distance of the new chat (if successful) vs original chat (what configuration is most explanatory in either better or worse performance in the new chat), what time of day accessing/coaxing(training)/what order are you feeding information/most importantly when does the machine stop [itself] (developer set system limitations). If you are getting good output today, will it be the same productivity level maintained tomorrow or over the course of a month for that same

trained chat? **MOST IMPORTANT: Does it forget what works and what doesn't? IF SO, WE NECESSARILY **MUST NOT** CALL IT A PRODUCTIVITY APP, IT IS TOO FRAGILE TO WORK [NEVER MIND RELY ON] IF ANY NEW INTRODUCED (WITHOUT ANY INDICATION IN #COMMENTING OUT) CODE COMPLEXITY OR **ANY INTRODUCED COMPILATION ERRORS**, OVER YOUR CURRENTLY WELL-COMPILING (0 ERRORS BY DEFAULT IF FINISHED WORK PRODUCT) AND LOCAL-MACHINE TESTED CODE GIVEN TO AN UNSECURED "HANDS-OFF" VOLATILE PROCESS. GPT INDICATES IT CANNOT COMPILE AND WILL REFUSE ONLY AFTER GIVING IP DESPITE HAVING CODE INTERPRETER, YET YOU HAVE ALREADY COMPILED YOURSELF AND GIVEN CHAT A FINAL PRODUCT, YOU CANNOT USE CHAT, A CAUTIONARY TALE: YOU SIMPLY CANNOT NOT RELY ON IT FOR ANY DECENT WORK OUTPUT (OF SUFFICIENT MINIMAL COMPLEXITY AND WORK COMPLIANCE REQUIREMENTS TODAY)! IF YOUR CHAT/Custom-myGPT/API LLM FUNDAMENTALLY DOES NOT TALK TO OTHER PEOPLE, ONLY A DISTANT SERVER/CHAT-MACHINE; HOW CAN YOU CALL IT A SOCIAL MEDIA APP "CHAT-" FIRST?**