	chooses the most probable next token, resulting in identical outputs every time. To balance randomness and determinism and ensure that the output is neither overly uniform and predictable nor overly random and chaotic, the default value is 0.8； ***float repeat_penalty:*** controls the occurrence of token sequence repetitions in the generated text, helping to prevent the model from generating repetitive or monotonous text. A higher value (e.g., 1.5) imposes a stronger penalty on repetitions, while a lower value (e.g., 0.9) is more lenient. The default value is 1.1; ***float frequency_penalty:*** a factor for penalizing word/phrase repetition, reducing the probability of using words/phrases with higher frequencies overall and increasing the likelihood of using those with lower frequencies. This may lead to more diversified generated text, but could also result in text that is difficult to understand or not as expected. The range is [-2.0, 2.0], with a default value of 0; ***int32_t mirostat:*** an algorithm actively maintaining the quality of generated text within the expected range during the text generation process. It aims to find a balance between coherence and diversity, avoiding low-quality output caused by excessive repetition (boredom trap) or incoherence (confusion trap). The values space is {0, 1, 2}, where 0 indicates not activating the algorithm, 1 indicates using the mirostat algorithm, and 2 indicates using the mirostat 2.0 algorithm; ***float mirostat_tau:*** an option setting the target entropy for mirostat, representing the expected perplexity value of the generated text. Adjusting the target entropy allows to control the balance between coherence and diversity in the generated text. Lower values will result in more concentrated and coherent text, while higher values will lead to more diversified text, possibly with lower coherence. The default value is 5.0; ***float mirostat_eta:*** an option setting the learning rate for mirostat, which influences the algorithm's responsiveness to feedback on generated text. A lower learning rate will result in slower adjustment, while a higher learning rate will make the algorithm