We have built the AIscreenR as a flexible software, allowing users to conduct multiple screenings simultaneously based on multiple prompts, API models, iterations of the same request, and nucleus samples (i.e., different top\_p or temperature values). The software further allows the user to send the same request (i.e., repeatedly asking the same question) multiple times to avoid random noise in individual model responses (especially relevant when using GPT-3.5 models). When this feature is used, the final GPT decision is based on the probability of inclusion across the iterated requests, and the specific inclusion threshold can then be determined by the user (see Figure 4 of this paper). This also allows the users to test model response consistency. Moreover, the software has been built to draw on multi-core processing, thereby allowing users to significantly speed up the screening. Finally, we built the package so that reviewers can work with two different types of function calls; one yielding simple/trinary (i.e., 1 = {include}, 1.1 ={uncertain}, and 0 = {exclude}) decisions and/or another yielding descriptive responses to the screening requests. We consider the former to be the main work engine, whereas the latter can be pivotal when examining discrepancies between GPT and human screener decisions.

Although we have tried to accommodate the requirements set forth by evidence organizations, we do not consider our solution to be a final one. Our aim has merely been to show one way in which GPT API models can be used for TAB screening in large-scale systematic reviews that can inspire and be transferred to future applications of TAB screening with all kinds of LLMs

A side-effect of such research would further be that the costs of using GPT models may be substantially reduced, which can be a major barrier to using GPT-4 models for TAB screening at the current point in time. These models are still rather expensive (in absolute terms, not compared to hiring a human screener). Thus, another line for future research could be to investigate the performance of cheaper GPT-4 models, such as GPT-4o and GPT-4-turbo.