Tutorial

Advances in Methods and Practices in Psychological Science
April-June 2026, Vol. 9, No. 2, pp. 1–36

DOI: 10.1177/25152459261442150
www.psychologicalscience.org/AMPPS

Fast-Track Your Abstract Screening: Mastering ASReview for Accelerating Abstract Screening and Evaluating Decisions From Automatic-Screening Methods

Tim Fütterer[1], Lars König[2], Diego G. Campos[3,4], Ronny Scherer[3,4], Steffen Zitzmann[5], and Martin Hecht[2]

[1]Hector Research Institute of Education Sciences and Psychology, University of Tübingen, Tübingen, Germany; [2]Helmut Schmidt University, Hamburg, Germany; [3]Centre for Educational Measurement, University of Oslo, Oslo, Norway; [4]Centre for Research on Equality in Education, University of Oslo, Oslo, Norway; and [5]Medical School Hamburg, Hamburg, Germany

Abstract

Research syntheses, such as systematic reviews and meta-analyses, are crucial for synthesizing research to support evidence-based decision-making. However, the abstract-screening phase, during which researchers evaluate titles and abstracts for inclusion, is highly time-consuming and often results in cognitive biases and fatigue. To address these challenges, machine-learning-assisted tools, particularly those using active learning, have gained prominence. One such tool is Active Screening Review (ASReview), an open-source software for semiautomating title and abstract screening in systematic reviews. ASReview incorporates user feedback to prioritize relevant studies, reducing screening time and improving efficiency. Despite its potential, many researchers remain uncertain about integrating ASReview into their workflows and making evidence-based decisions regarding the tool's configuration, training, and stopping criteria. In this tutorial, we provide a step-by-step guide to using ASReview, including practical examples from psychological research. We demonstrate the software's application in two use cases: screening unlabeled abstracts using active learning and verifying results from automated-screening methods. In the tutorial, we also offer evidence-based recommendations for selecting stopping rules to balance sensitivity and efficiency. We also outline strategies for prescreening, data-set preparation, model setup, and progress monitoring to ensure that researchers can maximize the tool's benefits while maintaining scientific rigor. By offering evidence-based guidance at each stage of the process for practitioners without coding skills, in this tutorial, we aim to help researchers harness artificial-intelligence-aided screening to enhance the quality and efficiency of research syntheses across disciplines.

Keywords

artificial intelligence, machine learning, research syntheses, systematic review, meta-analysis, ASReview, tutorial, open data, open materials

Received 5/11/25; revision accepted 3/10/26

Research syntheses, including systematic reviews and meta-analyses, are essential for gaining a comprehensive overview of the current literature and empirical findings on specific topics (Cooper et al., 2019). They provide a solid foundation for evidence-based decision-making by

Corresponding Author:
Tim Fütterer, Hector Research Institute of Education Sciences and Psychology, University of Tübingen, Tübingen, Germany
Email: tim.fuetterer@uni-tuebingen.de

researchers and policymakers by consolidating the current state of knowledge, highlighting emerging trends, and identifying gaps in the literature (Pigott & Polanin, 2020). However, producing research syntheses can be highly time-consuming, making it difficult to deliver timely answers to pressing questions (Smith et al., 2011). One particularly labor-intensive phase involves selecting relevant articles. For example, researchers have reported that screening a single title and abstract takes about 30 s on average (Gates et al., 2018), meaning that screening 14,923 abstracts may require as many as 89 days (Polanin et al., 2019). One key challenge researchers face during screening is ensuring they identify as many relevant articles as possible (an aspect of the quality of research syntheses) while limiting the pool of articles to maintain a feasible workload (an economic and ethical aspect; see Polanin et al., 2019). In light of this challenge, the rapid advancement of technological tools, especially those based on artificial intelligence (AI), holds considerable promise for optimizing research syntheses by making them more efficient, accurate, and reliable (Burgard & Bittermann, 2023).

Active Screening Review (ASReview) is a tool for title and abstract screening that meets important quality standards (e.g., open source, locally executable; van de Schoot et al., 2021; for a systematic review of AI-based tools for research syntheses, see Fütterer et al., 2026). This tool can be used to semiautomate the abstract-screening process, evaluate already classified abstracts, and simulate abstract screening using previously classified abstracts. In all cases, the tool relies on active learning, and the reviewer remains involved throughout the process (Fig. A1 in the Appendix). Screening begins with a set of abstracts, and the process continues until at least one is labeled as relevant and one is labeled as irrelevant. These initial labels serve as training data for a machine-learning (ML) model that ranks the remaining abstracts by predicted relevance. The reviewer then examines the highest ranked abstracts, adds new labels, and feeds the updated information back into the model. This cycle repeats, allowing the algorithm to progressively improve its predictions.

The semiautomation process enables users to identify all relevant abstracts without manually screening each one. Using already labeled abstracts enables comparison of multiple reviewers' screening decisions and a quick assessment of the accuracy of automated tools, such as ASReview (Bron et al., 2024). The simulation mode further enables researchers to compare the performances of different algorithms and develop recommendations for their use.

However, despite the availability of AI tools, many researchers remain unfamiliar with them and unsure how to operate or integrate them into their research workflows (e.g., Chai et al., 2021; Scott et al., 2021). In the case of ASReview, a variety of approaches to using the tool have been discussed (e.g., screening with different learning algorithms or training the model on a broader data set before screening begins; see ASReview forum: https://github.com/asreview/asreview/discussions). We provide a list of examples of approaches discussed in the forum on our OSF project page (https://osf.io/xrb9z/overview).

Although ASReview is a reliable, effective, and user-friendly tool that has been demonstrated to save time during abstract screening (Fütterer et al., 2026), practitioners still require guidance on how to use it in an evidence-based manner. Thus, unlike existing resources that offer mainly personal-experience-based advice, such as the SAFE (screen a random set for training data, apply active learning, find more relevant abstracts with a different model, evaluate quality) procedure proposed by Boetje and van de Schoot (2024), this tutorial is designed to provide structured, evidence-based guidance for each step of the process.

In the following sections, we briefly outline why we believe AI-supported reviews offer considerable potential to advise evidence-based research and practice. We then provide a step-by-step tutorial for ASReview using an example data set of articles identified through a systematic literature search.

Theoretical Background

ML in research syntheses

Research syntheses, such as systematic reviews and meta-analyses, are structured methodologies and statistical approaches designed to systematically retrieve and synthesize research evidence from primary studies (Cooper et al., 2019). They typically follow the first six key steps illustrated in Figure 1 (Carrasco-Labra et al., 2021). However, recent studies have suggested that completing these steps can be highly time-consuming, even for experienced researchers (Borah et al., 2017; Smith et al., 2011). One particularly labor-intensive step is title and abstract screening: After compiling potentially relevant literature from multiple databases, researchers must evaluate each article's title and abstract to determine whether it should be included. This process must be transparent and reproducible to ensure that all relevant studies are captured (Moreau & Gamble, 2022).

Recently, AI-aided abstract screening has gained increasing attention, particularly semiautomated screening, which has become a key focus in many research efforts. Whereas tools have long been used in fields such as health care (Harrison et al., 2020), they are increasingly being adopted in areas such as education research (e.g., for abstract screening, Bühler et al., 2025; e.g., for updating meta-analyses, Chernikova et al.,

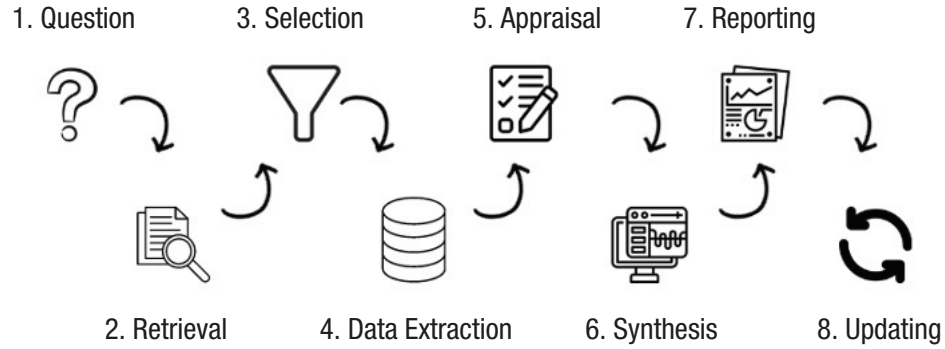

Fig. 1. Key steps in a systematic review and meta-analysis. This figure was adopted from Fütterer et al. (2026).

2024). Various tools employing different approaches to semiautomated screening have been developed, and most rely on active learning techniques now being applied in educational research (Fütterer et al., 2026). Specifically, active learning (see Fig. A1) uses an ML algorithm to sort unseen abstracts based on their predicted relevance, which is determined by weighting factors in the training data (e.g., one relevant abstract and one irrelevant abstract) and previous screening decisions (for an overview of the abbreviations used in this tutorial, see our legend in Appendix A). Such an approach can significantly reduce screening time by identifying the most relevant references before all references have been manually screened. That is, ML tools offer several benefits, such as accelerating processes by saving time and reducing costs associated with human labor, potentially improving replicability (pending further evidence), and mitigating cognitive biases (e.g., anchoring and adjustment effects) and mental fatigue, which often affect human decision-making. In this way, ML algorithms can support researchers by enhancing efficiency while preserving scientific rigor. However, ML algorithms and other aspects of AI-assisted screening, such as the data set and its specific characteristics, can lead to considerable variations in tool performance. These variations ultimately affect the number of abstracts that must be screened to identify the same number of relevant articles. Consequently, determining the stopping point in AI-aided abstract screening is a key challenge. Unfortunately, many users face uncertainties about how to use these tools effectively, select an appropriate ML algorithm, determine stopping points, and adopt optimal screening procedures.

Fully automated screening has been available in some tools for years (e.g., Gates et al., 2019). In this approach, ML algorithms automatically label abstracts using a training set, although they often lack sufficient accuracy (Zhang & Neitzel, 2024). Nonetheless, recent advances in AI, particularly in large language models (LLMs), have shown promising results (e.g., Li et al., 2024; Vembye et al., 2024). When LLMs are used for abstract screening, labeling is typically automated by providing the model with a prompt and the inclusion and exclusion criteria. A recent study examining sensitivity and specificity—that is, how accurately relevant and irrelevant abstracts are classified—found that these models could identify more than 85% of relevant literature without manual screening (Dai et al., 2024). For instance, Elicit (2025) provides users with fully automated reviews, and Deep Search in ChatGPT (OpenAI, 2025) synthesizes large volumes of online data through reasoning to produce a complete, fully automated research report.

However, automatic screening lacks transparency and controllability. For instance, researchers cannot determine how many relevant abstracts remain unidentified or have been misclassified. Consequently, recent research has advocated for semiautomated-screening methods (i.e., human-in-the-loop approaches; Bron et al., 2024). These approaches combine the benefits of automation with the reliability of validated stopping rules. One recommended tool that exemplifies semiautomatic screening and can handle both unlabeled and previously labeled abstracts is ASReview.

ASReview: a tool for semiautomated abstract screening

ASReview is a freely available, open-source tool designed by the AI-aided Knowledge Discovery Lab at Utrecht University (van de Schoot et al., 2021). It was designed to streamline the title- and abstract-screening stage in systematic reviews using ML. Through active learning, ASReview continuously refines its predictions based on user feedback, thereby accelerating the identification of relevant articles during the screening process and potentially reducing the total number of articles that require manual screening. The tool features two operational modes: the review mode, for conducting standard AI-aided screening

with unlabeled data, and the simulation mode, for testing the performance of different ML algorithms and stopping criteria. The ability to combine the simulation mode with ASReview's support of various ML algorithms is one of its key strengths. Because ASReview is open source and through the simulation functionality, simulation studies have evaluated diverse algorithms and screening procedures across multiple research domains, demonstrating that the tool can substantially reduce the effort required for manual screening (Campos et al., 2024; Ferdinands et al., 2020; König, Zitzmann, Fütterer, et al., 2024).

Another major advantage of ASReview is its intuitive, user-friendly design (Fütterer et al., 2026). Researchers without coding skills can install and use ASReview on Windows, Mac, and Linux, and the only technical requirement is a Python installation (see the installation instructions on the ASReview web page: https://asreview.nl/install/). In addition, users can adjust a range of settings to suit their specific research needs, as outlined below.

Finally, its open-source architecture further enhances reproducibility by enabling users to share workflows and configurations. Moreover, ASReview offers comprehensive documentation and case studies, facilitating a clear understanding of its features and applications (ASReview LAB Developers, 2022; van de Schoot et al., 2021). This documentation also includes instructions for using ASReview via the command line and its Python application programming interface (API).

For what purposes can ASReview be used? Whereas ASReview is primarily designed to accelerate the identification of relevant studies during the title- and abstract-screening phase of a research-synthesis project, user feedback suggests that it has been adapted to a wide range of workflows (https://github.com/asreview/asreview/discussions/; see also an exemplary overview of relevant ASReview discussion threads on our OSF project page, https://osf.io/xrb9z/overview). Users have reported that if the initial training set is too narrowly focused (e.g., on a single methodological approach), the algorithm may continue to suggest similarly narrow studies, potentially overlooking other relevant literature. To address this issue, research teams have developed various strategies, including expanding the training set to cover articles from diverse perspectives, screening a random subset of the entire data set (i.e., corpus of articles) to build the training data, or reusing screening decisions from previous reviews when updating a research-synthesis project. In ASReview, users can define both relevant and irrelevant abstracts of studies as priors or simply click "Quick Start" to begin screening randomly. This means that screening proceeds randomly until the first relevant abstract is identified, at which point, the model automatically activates and begins prioritizing abstracts based on predicted relevance. One area of considerable variation is the training phase of the learning algorithm, in which researchers identify at least one relevant article and one irrelevant article before the screening process begins.

Careful selection at this stage is essential because the training set influences the screening performance (Boetje & van de Schoot, 2024; König, Zitzmann, & Hecht, 2024). We recommend using only one relevant study for the initial training phase because adding more has been shown to have little impact on model performance (König, Zitzmann, Fütterer, et al., 2024).

During the main screening phase, ASReview can be used in both single-reviewer and multireviewer workflows. In a single-reviewer context, the tool can considerably reduce manual screening (van de Schoot et al., 2021), particularly in large-scale reviews or when resources are limited. In collaborative settings, some teams implement double screening by creating two separate ASReview projects—each with the same or different data sets, training data, and learning algorithms—and then comparing the results to identify areas of disagreement (for more ideas on how to use ASReview, see ASReview, n.d.). In addition to this solution, ASReview LAB supports a crowd-screening mode in which multiple reviewers can log in to the same project (hosted on a server) and jointly train the same active-learning model. Both labels and model updates are shared among reviewers, enabling a truly collaborative workflow (see https://asreview.nl/crowdscreen/).

ASReview also includes a feature that allows users to view others' screening decisions while screening. This functionality makes ASReview suitable for training reviewers on already labeled data or evaluating the decisions made by different reviewers. Furthermore, it enables the evaluation of prelabeled abstracts generated by automated screening tools while still applying active learning.

Finally, the question of when to stop screening has been widely discussed by both users and ASReview developers. Various heuristic-based (Campos et al., 2024; Wallace et al., 2010) and statistical (Callaghan & Müller-Hansen, 2020; König, Zitzmann, Fütterer, et al., 2024) approaches have been proposed. In the described version of ASReview, heuristically derived stopping rules are supported in the tool, as outlined below. In the following section, we outline the current state of the literature on both ML algorithms and stopping rules and provide evidence-based recommendations.

Key components of AI-aided screening

Deriving robust recommendations for the AI-aided screening process can be challenging. Numerous evaluation

studies have revealed considerable variation both between and within different screening tools (Burgard & Bittermann, 2023; König, Zitzmann, & Hecht, 2024). A major contributor to these performance differences is the data set itself. For example, whereas screening fewer than 20% of abstracts was sufficient to identify 95% of relevant articles in some data sets when ASReview was used, other data sets required screening more than 60% of abstracts (König, Zitzmann, & Hecht, 2024). However, beyond the data set, two key components of AI-aided screening influence the quality of the screening outcome: the ML algorithm used to rank abstracts and the stopping rules (König, Zitzmann, & Hecht, 2024). When combined effectively, these components increase the likelihood of identifying a large proportion of relevant articles while substantially reducing the time required for screening. Below and in our recommendation section, we summarize key findings from the literature, focusing on research in psychology and education.

ML algorithms. The ML algorithm used in AI-aided screening typically consists of two components: a feature extractor and a classifier. ASReview offers a variety of ML classifiers—including support vector machines (SVMs), naive Bayes, and random forests (RFs; Breiman et al., 2017)—and supports several feature-extraction methods, such as term frequency-inverse document frequency (TF-IDF) and word embeddings (Mikolov et al., 2013; Reimers & Gurevych, 2019). Feature extractors differ in how they identify linguistic elements to assess similarities between abstracts. For instance, the TF-IDF algorithm assigns weights to words on the basis of their frequency within a document (term frequency) and their rarity across the entire corpus (inverse document frequency; Manning et al., 2009; Salton & Buckley, 1988). By contrast, the Sentence Bidirectional Encoder Representations from Transformers (SBERT) algorithm uses a pretrained language model trained on extensive text data beyond the screened abstracts, enabling it to capture sentence-level semantics in an abstract (Reimers & Gurevych, 2019).

The classifier, by contrast, is responsible for calculating documents' relevance scores. It processes the numerical data generated by the feature extractor to predict the likelihood that an abstract is relevant. Variations in data processing affect how weights are assigned to textual elements, resulting in differences in how abstracts are ranked across algorithms (Teijema et al., 2023). Consequently, the sequence in which users label abstracts can vary depending on both which feature extractor and classifier are used.

Besides the feature extractor and classifier, ASReview allows users to choose different query strategies and balancing methods. A query strategy determines the sequence in which abstracts are screened. By default, ASReview uses certainty-based querying, which ranks abstracts by their predicted relevance, placing those most likely to be relevant at the top. Alternatively, users can opt for random ordering or introduce random abstracts in a certainty-based sequence. The balancing method addresses the common issue of imbalance caused by a surplus of irrelevant abstracts. ASReview employs dynamic resampling by default—a technique that adjusts the ratio of relevant and irrelevant articles by undersampling the latter and oversampling the former while keeping the size of the training data consistent. Other balancing options, including no balancing or simple undersampling, are also available but not discussed here (ASReview LAB Developers, 2022).

Several evaluation studies have explored the performances of different algorithms and the factors that influence their effectiveness (e.g., Ferdinands et al., 2020; van de Schoot et al., 2021). Among these, the combination of the logistic-regression (LR) classifier and the SBERT feature extractor showed superior performance compared with other algorithms (König, Zitzmann, & Hecht, 2024; Teijema et al., 2023). Although the RF classifier combined with the SBERT feature extractor also produced promising results (Campos et al., 2024), the LR+SBERT algorithm has been studied more extensively and outperformed RF+SBERT in terms of the number of abstracts to screen to identify 95% of the relevant literature (König, Zitzmann, Fütterer, et al., 2024). For instance, König, Zitzmann, Fütterer, et al. (2024) examined whether the performance of the LR+SBERT algorithm depends on the initial training data, prevalence of relevant abstracts, and data-set size. The authors found that the algorithm maintained a relatively stable performance across varying experimental conditions: An average of 30% to 39% of abstracts needed to be screened to identify 95% of the relevant articles.

Finally, we emphasize that existing evaluation studies of algorithm performance have focused on semiautomated screening, which typically provides ML algorithms with limited training data (i.e., 1–10 relevant and irrelevant abstracts). Thus, it remains unclear whether the LR+SBERT combination outperforms other algorithms when larger training data sets are available, as would be the case when combining automated- and semiautomated-screening methods. Nonetheless, findings indicating that the LR+SBERT algorithm also performs well in identifying the final relevant abstracts—even after many other abstracts have already been screened and added to the training set—suggest that this algorithm may be particularly beneficial in such screening scenarios (Teijema et al., 2023).

Stopping rules. A major challenge in AI-aided screening is determining a stopping point that ensures the most

relevant literature is covered. In AI-aided screening, stopping rules are typically designed to identify approximately 95% of the relevant literature, balancing comprehensiveness and efficiency (Burgard & Bittermann, 2023; Callaghan & Müller-Hansen, 2020; Campos et al., 2024; König, Zitzmann, Fütterer, et al., 2024). This threshold is widely accepted because retrieving the final 5% of relevant studies can considerably increase screening time (van de Schoot et al., 2021). By comparison, traditional screening methods have been shown to result in error rates of approximately 10%, often because of fatigue and the extended duration of the process (Wang et al., 2020). AI-aided screening may help reduce such errors by shortening screening times and thereby lowering fatigue, ultimately improving accuracy (Boetje & van de Schoot, 2024). Consequently, identifying 95% of the relevant literature may approximate the level of accuracy achieved when screening all abstracts. Moreover, Oude Wolcherink et al. (2023) found that ASReview suggested articles that ultimately passed full-text screening earlier in the screening process than those that were judged relevant based only on abstract screening. This finding indicates that the final portion of abstracts may add limited value for identifying additional actual relevant studies. Nevertheless, determining an appropriate stopping point requires careful methodological consideration.

Several aspects of the screening process can help users determine stopping rules that balance efficiency, defined as the number of abstracts screened (screening cost), and effectiveness, defined as the number of relevant abstracts identified (sensitivity). For instance, the algorithms' performance, measured as the ratio of identified relevant abstracts to screened abstracts, generally improves at the beginning but declines toward the end (Cormack & Grossman, 2016). As performance declines, increasingly long sequences of irrelevant abstracts tend to appear between relevant ones, a general phenomenon in AI-aided screening that can be used to guide the determination of an appropriate stopping point. The data-driven heuristic, for example, proposes that the screening process can be terminated once a predefined number of consecutive irrelevant abstracts has been encountered (Ros et al., 2017). For instance, in a collection of 1,000 abstracts, a cutoff value of 2.5% would correspond to stopping after 25 consecutive irrelevant abstracts. Recent studies have demonstrated that this approach can identify 95% of the relevant literature on average when the prevalence of relevant abstracts is at least 5% (König, Zitzmann, & Hecht, 2024). However, at lower prevalence rates, this cutoff has been shown to identify fewer than 80% of the relevant abstracts. Although easy to apply, the rule can therefore be problematic in such settings or in the early phases of screening, when long runs of irrelevant abstracts may appear before the algorithm has stabilized, increasing the risk of premature stopping.

In contrast, the time-based heuristic defines a stopping point after a fixed proportion of abstracts has been screened—for example, a 30% threshold means screening ends once 30% of the data set has been reviewed (Wallace et al., 2010). This rule helps prevent very early stopping and performs well when the algorithm successfully prioritizes relevant abstracts early in the process. However, it can lead to premature stopping when algorithm performance is poor and to overscreening when performance is strong.

Combining the two heuristics can mitigate their respective weaknesses: The time-based rule serves as a safeguard against premature triggering of the data-driven rule, and the data-driven rule helps reduce unnecessary continuation once screening has progressed (Campos et al., 2024). In such combinations, the time-based rule primarily prevents the data-driven rule from being activated prematurely before the algorithm has had enough training data to rank effectively. At the same time, when algorithm performance is strong, the time-based rule may lead to unnecessary overscreening. To address this, a breakout rule has been proposed that doubles the data-driven threshold or sets it to a higher value (König, Zitzmann, Fütterer, et al., 2024). Whereas this may result in overscreening under typical conditions, it helps ensure that the rule is triggered only once nearly all relevant studies have already been retrieved. In some data sets, fewer than 10% of abstracts needed to be screened to identify all relevant studies, illustrating the potential of breakout strategies to reduce screening time.

Another safeguard is the key-study rule, which requires that all known relevant studies be identified before screening stops (Boetje & van de Schoot, 2024). In practice, this involves mixing known relevant studies into the unlabeled pool. For example, when five relevant studies are known beforehand, one abstract of such may be used for initial training and the other four included in the pool of unlabeled abstracts and screening ending only once all four have appeared. This rule is not intended to be applied in isolation but in combination with other stopping rules. Empirical evidence remains limited, but recent findings indicate that training an algorithm with more than one known relevant study does not markedly enhance performance in terms of screening cost or reduce variability in corresponding sensitivity values (König, Zitzmann, Fütterer, et al., 2024). Thus, using only one relevant study and one irrelevant study for training while retaining additional known studies for the key-study rule appears advantageous because it reduces the likelihood that screening terminates prematurely.

In contrast to these rules, approaches such as prevalence-estimation techniques (Callaghan & Müller-Hansen,

2020) and the knee method (Cormack & Grossman, 2016) apply statistical criteria to determine when to stop screening. However, estimation techniques and the knee method carry the risk of oversampling, potentially leading to a screening process that never terminates (König, Zitzmann, & Hecht, 2024). In addition, they often require technical implementation in the screening software or alternatively, repeated data download and rule application in third-party software and are not supported by ASReview. Considering this limitation and the fact that when combined and applied appropriately heuristic methods provide a reliable and straightforward basis for defining stopping points, in our tutorial, we focus on heuristic stopping rules. Nonetheless, the question arises of how to implement these methods correctly and define cutoff values that are sufficiently reliable to limit unnecessary screening while identifying at least 95% of the relevant literature. Therefore, we summarize key findings from the literature in the recommendations below.

A Step-by-Step Tutorial on ASReview for Efficient Semiautomated Abstract Screening

In this tutorial,[1] we provide a step-by-step guide to using ASReview for semiautomated screening, focusing on two distinct use cases. The first use case involves using active learning to review unlabeled abstracts and accelerate the screening process. The second approach, the mixed approach, focuses on verifying results from an automated-screening method, such as an LLM that classifies abstracts. The recommended screening procedure is broadly consistent across both. Each begins with a prescreening step, which prepares the data set for AI-aided screening and provides preliminary insights (Boetje & van de Schoot, 2024). The second step involves selecting stopping rules that align with the insights gained during prescreening (König, Zitzmann, Fütterer, et al., 2024). Finally, AI-aided screening is conducted. The following sections outline the application of the evidence-based recommendations.

Preliminaries

We do not address general ASReview operations in detail because these are thoroughly documented in the official ASReview software guide, which is regularly updated with new releases (see the official ASReview documentation, which is maintained as a living document that is continuously updated with each new software release: https://asreview.readthedocs.io/en/latest/index.html and https://github.com/asreview/asreview). However, we highlight and explain all functionalities required to follow the recommended workflow. Because the version of ASReview (Version 2.1.1) employed here requires certain processing steps to be performed in external software, we illustrate these steps in R (R Core Team, 2023) to remain consistent with ASReview's open-source approach. Equivalent processing, however, can also be performed using Microsoft Excel (Version 2301) or IBM SPSS (Version 28.0), among others.

Metrics and formulas. Throughout this tutorial, we compute the prevalence of relevant abstracts, the maximum number of consecutive irrelevant abstracts, sensitivity, specificity, and interrater agreement. All metrics are derived from descriptive counts (e.g., true positives, false negatives, true negatives, and false positives). We use the terms "labeled" to refer to abstracts that were seen during screening and assigned a decision (relevant or irrelevant) and "unlabeled" to refer to records that were not yet reviewed and therefore have no decision.

Prevalence is the proportion of relevant records in the full data set (N). The total number of relevant abstracts can be expressed as the sum of abstracts labeled as relevant (TP) and relevant abstracts that remain unseen and therefore unlabeled (FN). In practice, these false negatives remain unidentified; however, when a subset is fully labeled (i.e., all abstracts in that subset were seen), FN is zero by definition:

$$Prevalence = \frac{TP + FN}{N} \times 100\%. \quad (1)$$

Sensitivity is the proportion of truly relevant abstracts that are correctly labeled as relevant. It is defined as the number of abstracts labeled as relevant (TP) divided by the total number of relevant abstracts ($TP + FN$):

$$Sensitivity = \frac{TP}{TP + FN} \times 100\%. \quad (2)$$

Specificity is the proportion of unseen abstracts that are unlabeled but irrelevant. It is defined as the number of unseen irrelevant abstracts (TN) divided by the total number of irrelevant abstracts ($TN + FP$), where FP are abstracts that were seen and labeled as irrelevant:

$$Specificity = \frac{TN}{FP + TN} \times 100\%. \quad (3)$$

In the context of using an LLM to preclassify abstracts and active learning to evaluate these results, we use two additional formulas. Interrater agreement (Agr) indicates how often the LLM and the human reviewer give the same label to the same abstract. It is calculated on the subset labeled by counting the abstracts the LLM and the human reviewer both labeled as relevant and the abstracts they both labeled as irrelevant. The total

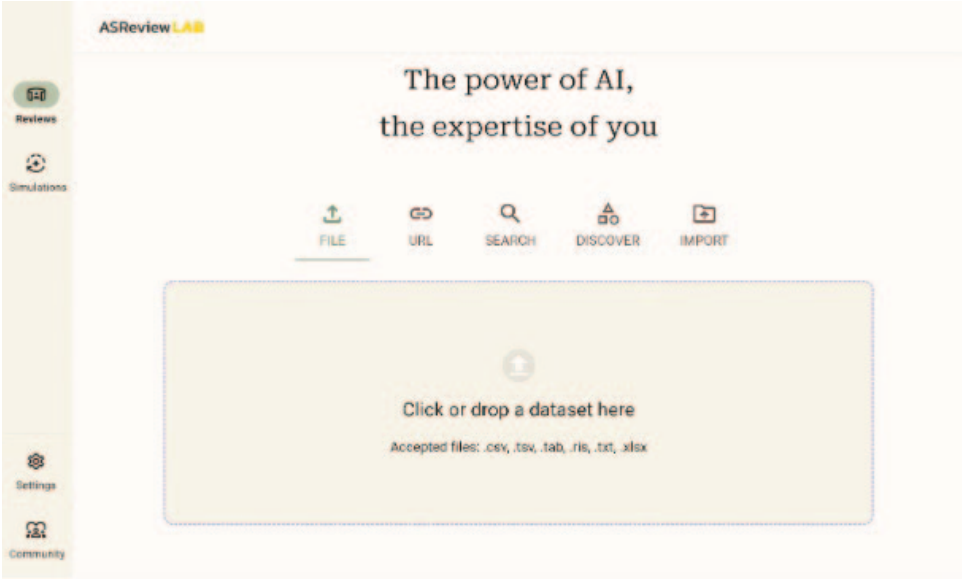

Fig. 2. Landing page of ASReview.

number of agreements is then divided by the size of this shared subset:

$$Percentage\ Aagreement = \frac{TP_{Agr} + TN_{Agr}}{N_{Agr}} \times 100\%. \quad (4)$$

The number of potentially missed relevant abstracts estimates how many relevant abstracts may still be hidden among abstracts that the human has not reviewed. It is computed by first estimating how often an abstract is labeled irrelevant by the LLM and is actually relevant according to the human in the overlap subset labeled by both. This rate is given by $FN/(FN+TN)$. The estimated number of potentially missed relevant abstracts is then obtained by multiplying this rate by n_u, the number of abstracts labeled as irrelevant by the LLM that were not reviewed by the human:

$$Percentage\ missed\ relevant\ abstracts = \frac{FN}{FN + TN} \times n_u. \quad (5)$$

Data set. In this tutorial, we use examples based on a data set originally screened by Tang et al. (2022) for their meta-analysis of questionnaire-based studies examining the relationship between curiosity and interest. The data set was subsequently requested by König, Zitzmann, Fütterer, et al. (2024), who cleaned the data, resulting in slight deviations in reference counts compared with those reported in Tang et al. The data set used here is available for download from our OSF project page (https://osf.io/xrb9z/overview). It contains 2,035 unique references, of which 53 were deemed relevant after abstract screening. Each study includes a corresponding abstract. Different versions of the data set are used throughout this tutorial and are referenced accordingly.

For the mixed approach, which requires a fully automated screening step, we simulated the classification accuracy of an automated tool with 75% sensitivity and 90% specificity. This simulation was based on random sampling of abstracts and is therefore not representative of fully automated screening, which would likely yield a more systematic selection. Because in this tutorial we focus on ASReview, we did not include any recommendations or explanations for using automated-screening tools. However, using the simulated data enabled us to provide a detailed, step-by-step tutorial on how to process results from an automated-screening tool in ASReview.

ASReview: interface

Before introducing the tutorial in detail, we provide an overview of the ASReview interface, which we refer to throughout the different screening phases. After installation, ASReview can be launched from the command prompt using the following command:

```
asreview lab
```

This command opens a browser-based interface that automatically loads in the system's default web browser. Upon launch, the ASReview landing page presents several options (see Fig. 2).

Landing page

Review and simulation modes. The left panel of ASReview offers two primary modes: review and simulation. Review mode is designed for real screening tasks and is the primary focus of this tutorial. In this mode, users can import nonlabeled, partially labeled, or fully labeled data sets and use ASReview to screen abstracts, either randomly or systematically, employing AI-aided screening.

Simulation mode, in contrast, enables users to explore ASReview using only fully labeled data sets. Users can upload previously screened abstracts or select data sets from ASReview's Synergy collection, a collection of data sets from different fields used to benchmark ASReview (van de Schoot et al., 2021). This mode allows testing different model configurations in a simulation environment to evaluate model performance and identification rates for specific cutoff values of the data-driven heuristic. In essence, simulation mode mimics the screening of unlabeled data by using preexisting labels to simulate user decisions during screening. Consequently, this mode supports the examination of the theoretical performance of various screening configurations and was instrumental in the large-scale simulation studies that informed the recommendations presented in this tutorial (Campos et al., 2024; de Bruin et al., 2025; König, Zitzmann, Fütterer, et al., 2024; König, Zitzmann, & Hecht, 2024).

Data upload. The data set should be a tabular list of references, with each reference containing an abstract in a column explicitly named "abstract" and a column named "data_id" that provides a unique number for each record. The data must be stored in one of the following file formats: .csv, .tsv, .tab, .ris, .txt, or .xlsx. Before initiating AI-aided screening, the data set should be deduplicated.

To initiate a new project in ASReview, a data set must be uploaded using the options in the center of the landing page (see Fig. 2). By default, the "File" tab is selected. In this view, users can upload data files by either dragging and dropping them into the upload area or selecting a directory after clicking the upload area. Alternatively, data sets can be uploaded via the URL option by providing a URL or DOI. The "Search" option, designed to search OpenAlex (Priem et al., 2022) for data sets, is inactive in the version used for this tutorial (Version 2.1.1). The "Discover" option, in contrast, is active and provides access to ASReview's built-in data sets from the Synergy collection (van de Schoot et al., 2021), which are particularly useful for using simulation mode.

The "Import" option allows loading existing ASReview project files. Each ASReview project is stored in an .asreview file that contains all data, model configurations, and screening decisions for that project. Project files are stored in the default installation directory (C:/user/[username]/.asreview), persist through software upgrades or reinstallation, and are automatically detected and reloaded when ASReview is reopened. These files can be shared among researchers, providing a straightforward way to ensure reproducibility and facilitate collaborative screening.

All active review projects appear under "Current Reviews" on the landing page once a data set has been uploaded. All completed review projects are listed below this section under "Finished Reviews."

General settings. Additional configuration options are available in the lower left panel of the landing page under "Settings" and "Community" (see Fig. 2). In "Settings," users can customize ASReview's appearance, including switching between "System," "Always light," and "Always dark" display modes. Font size can be adjusted, the screening layout can be set to "Landscape" view, and the interface can be configured to always display model information. We recommend enabling "Show model information" because this feature provides useful context during screening and is referenced multiple times in this tutorial. However, all of these settings can be modified at any stage of the screening process.

Creating a project. Once a data set is uploaded, ASReview automatically creates a project folder named after the data set (Fig. 3). The displayed data set name can be changed by clicking the pen icon next to it. The project files are central in ASReview, storing all information related to a screening endeavor. Because they can be easily shared with other researchers, project files provide a reliable foundation for reproducibility and promote collaborative approaches to screening. Moreover, they are maintained in the default installation directory (C:/user/[username]/.asreview), where they remain unaffected by software upgrades or reinstallation. When the program is reopened, these files are automatically located and reloaded.

The initial page of the project setup presents the data set's metadata and verifies that each article includes a title, an abstract, and either a URL or a DOI. In our example, approximately half of the data entries lacked a DOI; however, all contained both a title and an abstract. ASReview also checks for duplicate entries and shows the number of potential duplicates after "This dataset contains X records." According to ASReview's GitHub documentation (see the official ASReview documentation, which is maintained as a living document continuously updated with each new software release: https://asreview.readthedocs.io/en/latest/index.html and https://github.com/asreview/asreview), duplication checks primarily compare titles and DOIs (https://github.com/asreview/asreview-datatools?tab=readme-ov-file#data-dedup). When duplicates are detected, we

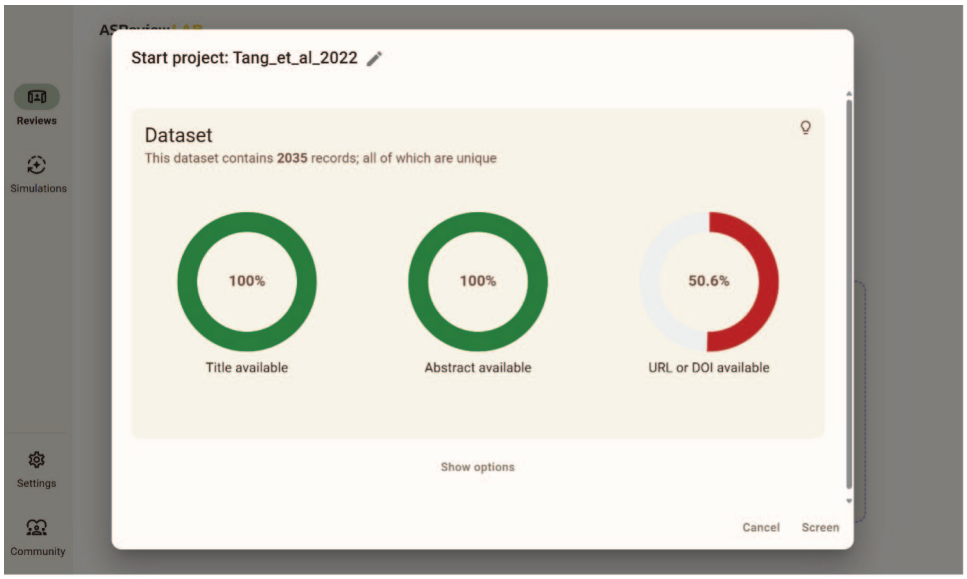

Fig. 3. Data-set upload.

recommend removing them before proceeding (using external software, as noted above). If duplicates remain in the data set, they do not necessarily reduce performance: Relevant duplicates will typically appear next to each other, and once an irrelevant abstract is labeled, the other is effectively deprioritized after retraining and pushed toward the end of the ranking. However, when multiple duplicates are present, the model may overemphasize features repeated across those duplicates, which can bias learning and affect ranking.

Although screening can be initiated immediately by selecting "Screen" in the lower right corner, we strongly recommend first selecting "Show options" in the center to define project-specific settings (see Fig. 3), such as "tags," screening "model," and "prior knowledge" (Fig. 4). These settings can also be adjusted after the project has been created—either before screening the first abstract or at any point during the screening process; we summarize these options in the following section.

Tags. We recommend creating specific tags before initializing the screening (see Fig. 5). For example, users can create a tag group named "Exclusion Reason" and add specific exclusion categories within it—such as those based on the population, intervention, comparison, outcome, and study design (PICO-S) criteria or other criteria (Methley et al., 2014)—allowing reasons to be assigned during screening by simply clicking the corresponding tag. In the exported data, each tag is represented in a dedicated column with values of 1 ("tag applies") or 0 ("default"; "tag does not apply") to indicate its presence. Therefore, tags provide valuable support for reproducibility by streamlining documentation of screening decisions. Unfortunately, tags can only be altered and added but not removed after they have been created.

Model selection. ASReview allows users to select from several preconfigured AI models—called "Learners"—or to build a custom configuration by selecting individual components (see Fig. 6). The default learner ("ELAS u4") in this version uses TF-IDF with bigrams as the feature extractor, an SVM classifier, a maximum (certainty-based) querier, and a balancer. In addition, the multilingual ("ELAS l2") and high-capacity ("ELAS h3") variants, available via the Dory extension, replace TF-IDF with transformer-based embeddings and require greater computational resources. Although the default learner has performed well across the Synergy data sets used to benchmark ASReview (https://github.com/asreview/synergy-dataset), studies in the psychological and educational literature suggest that alternative settings may be more effective in these research areas (de Bruin et al., 2025). In this tutorial, we demonstrate how to adjust these settings to suit psychological reviews as well.

To build a custom learner, four components must be selected: feature extractor, classifier, querier, and balancer (see Fig. 7). The feature extractor converts abstracts into numerical representations. Built-in options include One-Hot and TF-IDF, whereas installing the Dory extension provides access to additional extractors, including various BERT-based models. Although ASReview recommends the default "ELAS u4" model (see the official ASReview documentation: https://asreview.readthedocs.io/en/latest/index.html) because it is lightweight and

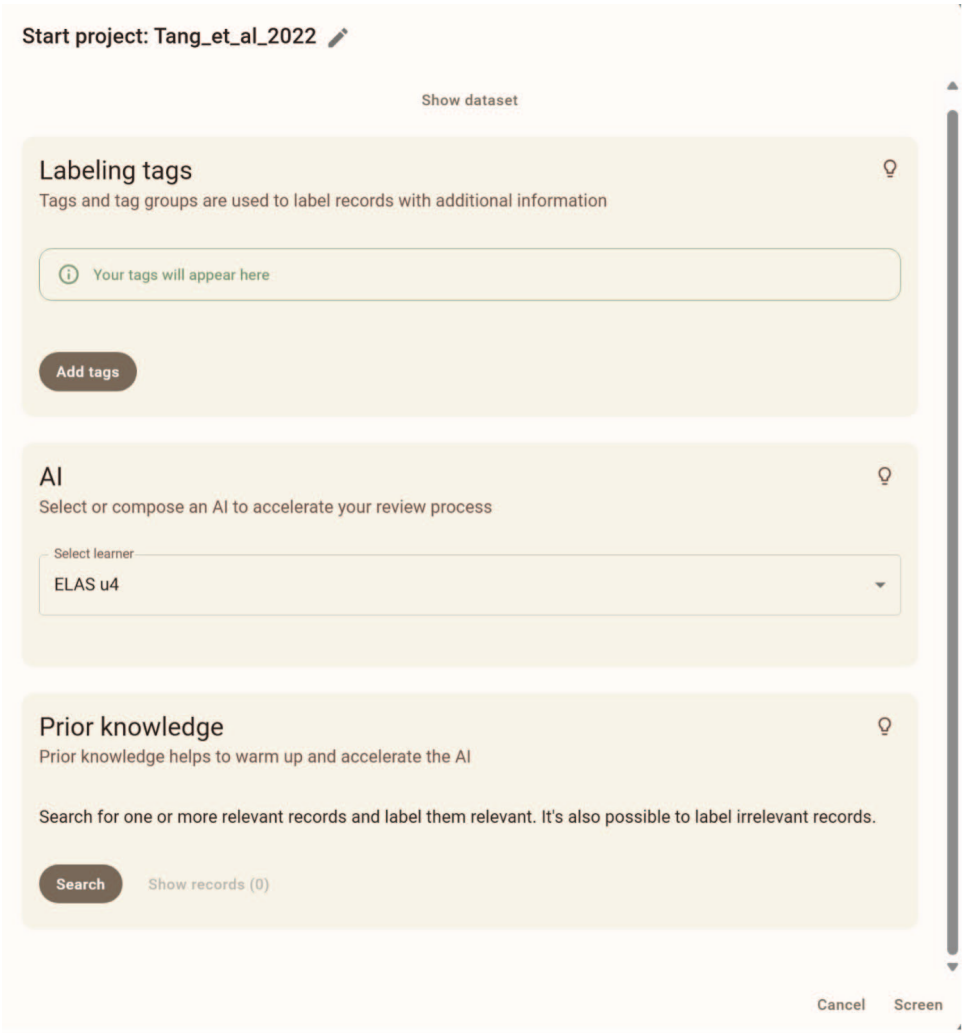

Fig. 4. Options relevant for the screening process in ASReview.

fast, we advise selecting a custom configuration that combines the logistic regression classifier with the SBERT feature extractor (default: "all-mpnet-base-v2") for the AI-aided screening phase (see below). This combination not only achieved the shortest screening time to identify 95% of the relevant literature but also showed the least variability across data sets and across different prevalence rates of relevant abstracts. In addition, it consistently ranked first or second in performance when the data-driven heuristic was applied to psychologically oriented literature (Campos et al., 2024; König, Zitzmann, Fütterer, et al., 2024). Other transformer-based extractors include "LaBSE" and "multilingual-e5-large," which support more than 100 languages, and "mxbai-embed-large-v1," a high-capacity model designed for nuanced semantic matching.

The classifier predicts whether an abstract is relevant based on the extracted features. ASReview includes naive Bayes, SVM, RF, and LR as built-in options. The Dory extension adds more advanced algorithms, such as AdaBoost, gradient boosting (e.g., XGBoost), and neural networks. However, previous evaluation studies found that the SBERT feature extractor performed best when paired with the LR classifier; this combination is therefore recommended (Campos et al., 2024; König, Zitzmann, Fütterer, et al., 2024).

The querier controls the order in which abstracts are presented for screening. Options include maximum (showing the most likely relevant abstracts first), uncertainty (showing abstracts for which the model is least certain), random, and top-down. Mixed strategies, which interleave maximum selection with a small proportion of random or uncertainty-based choices, can help balance exploration and exploitation. In most evaluation studies, the maximum querying was used. Therefore, we recommend this option.

Fig. 5. Defining tags.

Finally, the balancer determines how training data are sampled during model updates. The "Balanced" option is recommended for most reviews because it increases the proportional influence of relevant studies, thereby counteracting the dominance of irrelevant studies in the data set. Whereas pre-2.x versions of ASReview supported multiple balancing methods, the version used here supports balancing only sample weights. Although users can disable the "Balanced" option, doing so is not recommended. In data sets with substantial class imbalance—a common situation in abstract screening—disabling it may reduce models' capability to identify

Fig. 6. Selecting the machine-learning model.

Fig. 7. Customizing one's own machine-learning model.

relevant abstracts because the algorithm is more likely to focus on features associated with irrelevance given the much larger proportion of irrelevant abstracts.

Prior knowledge. Before initiating screening, users may search for specific studies they already know to be relevant or irrelevant. By selecting "Search," users can enter keywords or phrases, and ASReview will return matching abstracts using pattern matching (see Fig. 8). Any identified prior knowledge can be used to train the model during its first iteration.

This step is optional, and we generally recommend not using this functionality. Although it might be beneficial in cases in which prevalence is anticipated to be really low, randomly identifying prior knowledge can reduce the risk of a selection bias that influences the ranking of abstracts. When no prior knowledge is provided, screening begins with the random presentation of records until one relevant abstract and one irrelevant abstract have been identified. Reaching this threshold activates the selected model, which is then trained on the labeled abstracts and orders the remaining records according to predicted relevance. However, in our recommended workflow, we do not use this functionality either.

Project interface. After users complete the project setup and select "Screen," the model is trained. The project interface then opens, displaying the first abstract to be screened (Fig. 9).

Navigation pane. The navigation pane is displayed or hidden depending on whether landscape view is enabled or the user's screen is too small. When hidden, it can be opened by clicking the three-line icon in the top left corner. At the top of the navigation pane, users can return to the landing page, where all projects are displayed. Below this, the pane contains the options "Dashboard," "Reviewer," "Collection," and "Customize," with "Reviewer" serving as the default screening view (see Fig. 9).

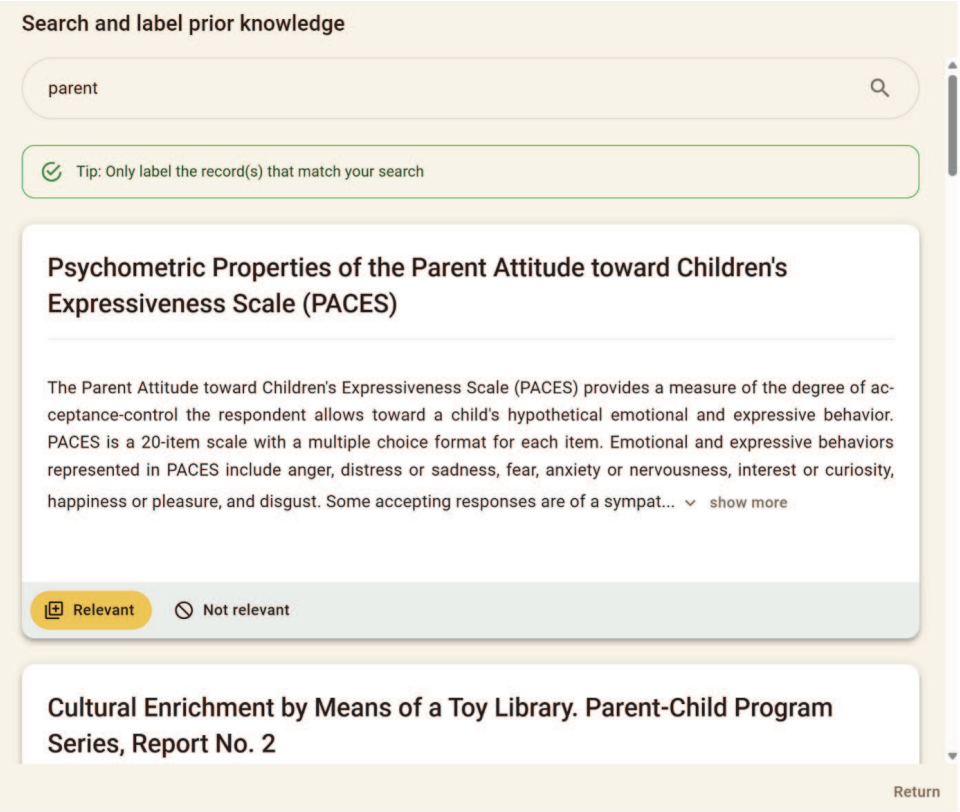

Fig. 8. Manual prior-knowledge selection.

Reviewing. The default page displayed after project creation is the "Review" mode, which enables users to classify abstracts. Alongside the abstract, ASReview displays the title and when available, a link to the study via its URL or DOI. When "Show model information" is enabled in the "Settings" menu, the active screening model is displayed

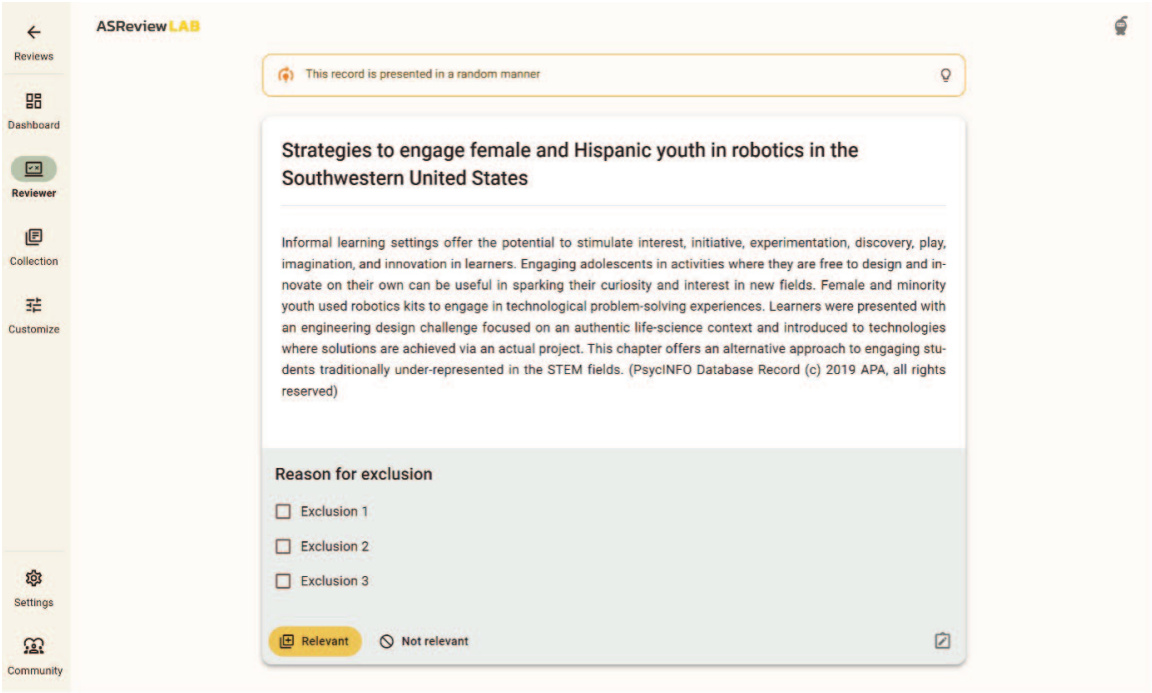

Fig. 9. Default page of the project interface: reviewing.

Fig. 10. Dashboard: screening progress, stopping, and history chart.

above the title. When screening randomly, this area reads as follows: "This record is presented in a random manner" (see Fig. 9). When screening with an ML model, it instead shows the number of labeled abstracts, the feature extractor, the classifier, and the balancing method. When a selected model cannot be fitted, a warning appears—for example: "Model training error: TypeError: sparse array length is ambiguous; use getnnz() or shape[0]. Change model on the Customize tab."

Below the abstract, any predefined user tags are displayed. During screening, users can add tags to individual abstracts by clicking on them—an especially useful feature for documenting exclusion reasons or flagging abstracts for reconsideration or team discussion. Tags can also be modified or added later in the project interface via the "Customize" page, as outlined below. In addition, users can add notes to an abstract by clicking the icon in the bottom right corner (depicted as a sheet with a pen).

Dashboard. The "Dashboard" allows users to monitor the screening process. It displays the number of relevant, irrelevant, and unlabeled abstracts and a pie chart showing their proportions (Fig. 10). A stopping rule can also be defined here. By clicking "Set Threshold," users can specify the cutoff value for a data-driven heuristic as either an integer or a percentage. This feature streamlines the screening experience by eliminating the need to manually check whether the threshold has been reached.

In addition, the "Dashboard" provides automatically generated and continuously updated visualizations (Fig. 10). These include "History," which shows the sequence in which abstracts were labeled; "Density," which displays the proportion of relevant abstracts within the last 10 screened abstracts (see Fig. 11); "Wave," which shows the number of consecutive irrelevant abstracts in relation to the number of screened abstracts including any defined cutoff for the data-driven heuristic; and "Recall," which visualizes screening progress over time. Among these, the "Wave" and "Recall" charts are the most informative for assessing model performance and will be explained in the AI-aided screening phase for semiautomatic abstract screening.

The "Dashboard" also offers a "Words of Importance" view that lists the terms the AI model considers most influential during classification, grouped into relevant and irrelevant categories. Although these cannot be altered, they may provide users with some insight into how the model is functioning. For the purposes of this tutorial, however, only the "Progress" statistics ("n_relevant," "n_irrelevant," "n_unlabeled") and the "Stopping Rule" are of primary interest. For detailed descriptions of all dashboard charts and visualizations, we refer users to the official ASReview documentation.

Collection. Navigating to "Collection" allows users to reexamine screening decisions. ASReview provides information on which model was used to screen a specific abstract and allows filtering by studies labeled relevant or

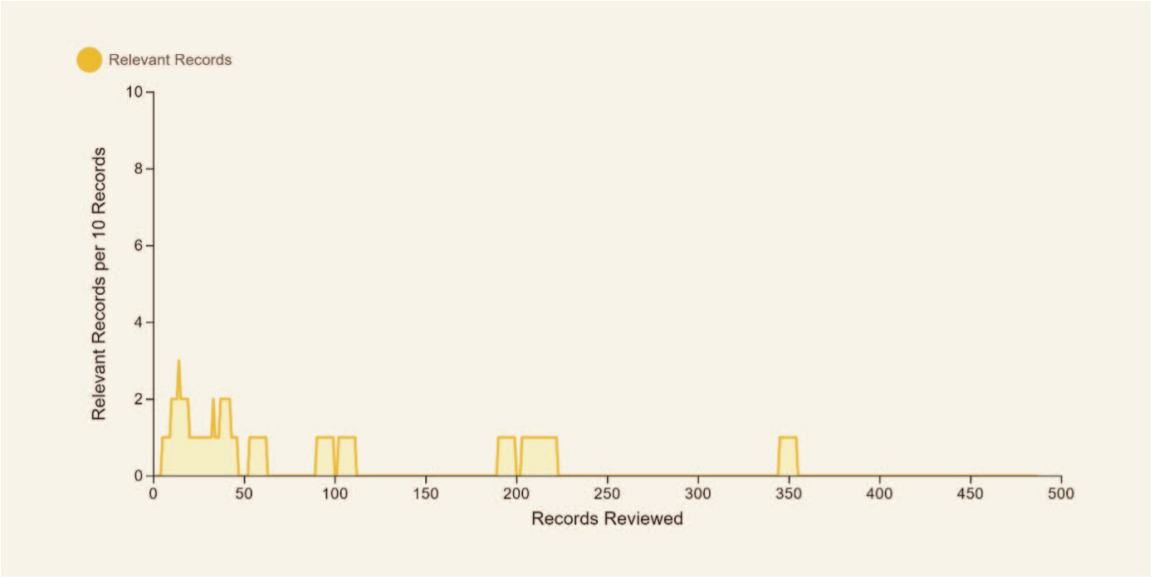

Fig. 11. Dashboard: density visualization.

irrelevant. In addition, users can search for notes and for abstracts used as prior knowledge (Fig. 12). All abstracts of studies that were selected or identified by ASReview as prior knowledge can thus be reviewed. In addition, looking for abstracts that contain a note can be useful for identifying key studies when marked accordingly or for identifying any other abstracts that contain a note. Furthermore, by clicking the three dots in the bottom left corner of an abstract, notes can be added, screening decisions can be reverted, and labels can be removed; the latter functionality is displayed but not yet available. This feature is likely to be fully supported in future versions. Finally, users can export their results by clicking "Export" in the top right corner of the screen.

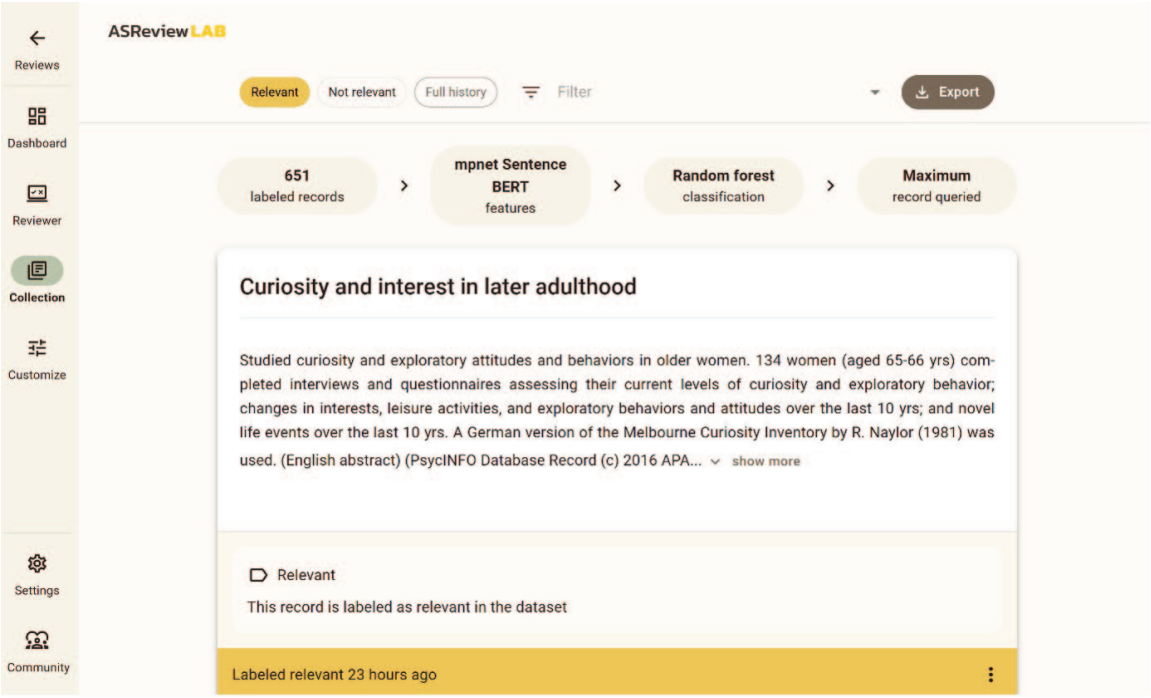

Fig. 12. Collection.

Table 1. ASReview Checklist for Semiautomatic Screening

Phases	Description	Examples
Prescreening	During the random screening of unlabeled abstracts, it is recommended that researchers document the data-set name, the number of abstracts, the number of identified relevant and irrelevant abstracts, the estimated prevalence, and the IDs of the key studies and training data.	• We will stop screening after at least 100 abstracts have been screened and at least one relevant abstract has been identified. • Data-set name: Tang_et_al_2022_unlabeled_pre_completed.csv • N_abstracts = 2,034 • n_screened = 100 • n_irr = 96 • n_rel = 4 • Prevalence = 4% • Key-Study record_id: 18, 660, 15
Stopping-rule selection	Explain and define stopping rule and cutoff value selection.	• On the basis of the prevalence of 4%, we decided to screen a minimum of 35% (time-based heuristic) of the abstracts and stop when 7% of the abstracts were irrelevant in a row (data-driven heuristic) and all key studies had been identified. • Breakout stopping rule: Stop when 15% of the abstracts in a row are irrelevant (data-driven heuristic).
AI-aided screening	Specify details of the screening process.	• One reviewer; ambiguous cases are discussed by the team.
	Document the model details.	• Feature extractor: SBERT • Classifier: LR • Query strategy: certainty-based (maximum) • Balance strategy: balanced
	Document final results.	• Data-set name: Tang_et_al_2022_unlabeled_pre_completed • N_abstracts = 2,034 • n_screened = 796 • n_irr = 744 • n_rel = 51 • Prevalence = 2.5%

Note: SBERT = Sentence-BERT; LR = logistic regression.

Customize. Finally, users can customize their screening configurations. These options provide the same functionality as during project creation (see Fig. 4). Users can create or modify tags, add prior knowledge, and define or change the screening model. When a model is changed, it will be trained using all previously labeled abstracts and the selected prior knowledge. The new model will be available as soon as its training is complete. However, users can continue screening with the old configuration until the new configuration is set up.

Semiautomated abstract screening

In ASReview, users can choose between randomly screening abstracts or using AI-aided abstract screening, which orders abstracts by predicted relevance. Since Version 2.x, the software has supported a multistep screening procedure. When no prior knowledge is selected before initiating AI-aided abstract screening, the software begins with random screening and then switches to AI-aided screening once at least one relevant abstract and one irrelevant abstract have been identified. Thus, ASReview incorporates a built-in multistep procedure consistent with recommendations from the literature. One example is the SAFE procedure (Boetje & van de Schoot, 2024), which organizes the process into four phases. The first phase consists of random prescreening, during which a subset of abstracts is selected at random for review. The second phase applies active learning, with screening guided by multiple stopping rules. In the third phase, a neural network or another complex ML algorithm may be introduced to identify additional relevant studies by leveraging its capacity to detect more complex patterns in abstracts (e.g., meaning and context within texts). The fourth and final phase involves a quality check, during which previously excluded abstracts are rescreened using active learning to ensure that no relevant studies were misclassified.

In this tutorial, however, we focus primarily on the random-prescreening and active-learning phases. Specifically, our recommended screening procedure comprises three major steps: prescreening, selection of the stopping rule, and AI-aided screening. To enhance transparency, we also provide a checklist of key information to be documented at each step, summarized in Table 1.

Step 1: prescreening. Random prescreening serves three primary purposes: (a) identifying relevant abstracts to train the learning algorithm, (b) collecting key studies, and (c) estimating the prevalence of relevant abstracts in the

data set (Boetje & van de Schoot, 2024). Random sampling also helps reduce the risk of overfitting, which can occur when training relies exclusively on abstracts of known relevant studies that share systematic features, such as journal guidelines, study designs, or research-group affiliations. However, recent findings indicate that using up to five relevant abstracts for training does not substantially improve performance (König, Zitzmann, Fütterer, et al., 2024). Therefore, training the algorithm with a single relevant abstract may be sufficient while reserving additional known relevant abstracts for the key-study rule. Estimating prevalence also informs the selection of a stopping procedure to maximize the probability of identifying at least 95% of the relevant literature, as described below.

The SAFE procedure recommends screening at least 1% of abstracts and ensuring that at least one relevant study is identified. If no relevant study is found within that first 1%, screening should continue until at least one relevant study is detected. Building on this guideline, König, Zitzmann, Fütterer, et al. (2024) suggested screening a minimum of 100 abstracts. They argued that this threshold yields more stable and reliable prevalence estimates in smaller data sets. For example, 100 abstracts correspond to 1% of a corpus of 10,000 abstracts, whereas most psychological meta-analyses require screening fewer abstracts. At the same time, screening 100 abstracts provides a practical balance in larger data sets because relying solely on a minimum percentage may result in unnecessarily long prescreening phases. Moreover, when the true prevalence is 10%, this proportion should be reflected in the sample regardless of whether the total collection contains 1,000 or 10,000 abstracts.

Project setup. To initialize random prescreening, users must first upload the data set as described above. In the following example, we use the data set "Tang_et_al_2022_unlabeled_pre.csv," which is available exclusively through our materials (https://osf.io/xrb9z/overview). To follow our tutorial, users must ensure the data set includes a column named "data_id" that stores a unique identifier for each abstract. When this information is stored in a differently named column, ASReview may delete it.

After uploading the data, ASReview will evaluate the completeness and uniqueness of the abstracts. When duplicates are detected, we strongly recommend removing them using a deduplication tool of your choice and then uploading the cleaned data set as a new project. When the data set contains no duplicates and includes 100% of abstracts and titles, users can proceed to create tags and define the screening model. For tagging, we recommend that all exclusion criteria—or at least exclusion categories (i.e., PICOS)—are represented as tags.

For model selection, a custom learner (AI model) should be defined by setting the query strategy to "random" (see Fig. 13). This ensures that references are presented in a purely random order, thereby disabling AI-aided screening. Although ASReview defaults to random presentation when no prior knowledge is provided, explicitly setting the query strategy to "random" prevents the system from switching to systematic querying once the first relevant and irrelevant abstracts have been identified. All other model settings can remain at their default values during this screening phase. We do not recommend searching for prior knowledge at this step because it will be identified through random screening of a subset of abstracts. Once tags are specified and the model is defined, users can continue by clicking "Screen" in the bottom right corner (see Fig. 13).

Screening. After the project setup is complete, screening can begin. As noted above, we recommend screening at least 100 abstracts and continuing until at least one relevant abstract and one irrelevant abstract have been identified. All relevant abstracts should be annotated with notes to mark them as key studies, which facilitates the later application of the key-study stopping rule.

The "Dashboard" can be accessed at any time to monitor screening progress. Note, however, that dashboard visualizations are not informative during the prescreening stage. Users should nonetheless document the number of screened abstracts, the number labeled as relevant, and the number labeled as irrelevant because this information is used to estimate the prevalence of relevant abstracts in the data set (see metrics above).

To apply the key-study stopping rule, future versions of ASReview will allow users to prepare the data directly in the "Collection" tab. There, relevant abstracts with attached notes can be filtered and labels removed by clicking the three dots in the upper left corner of an abstract. In the version of ASReview used here, label removal is not yet supported, even though the option is displayed. As a temporary workaround, we recommend exporting the data, including all relevant, irrelevant, and yet unlabeled abstracts. Note that during screening, ASReview automatically adds several columns to the data set: "asreview_label" stores the screening decision (1 = relevant, 0 = not relevant, missing = not screened), "asreview_time" abstracts the date and time of each decision, and "asreview_note" contains user-added notes. For each user-defined tag, ASReview also generates a column indicating whether the tag was applied (1) or not applied (0).

When opening the data, all relevant labeled abstracts will be placed at the top, and all irrelevant labeled abstracts will be placed at the bottom. Thus, users can remove the labels in "asreview_labels" column from all but one relevant abstract and document the "data_ids" of these abstracts. The manipulated data set can then be uploaded to a new project. When labels remain unchanged, all labeled abstracts will be treated as prior

Fig. 13. Model setup for random prescreening for semiautomated abstract screening.

knowledge. When only one relevant article is identified during prescreening or when the key-study stopping rule is not applied, the data set is already prepared for AI-aided screening.

Finally, the information gathered during this phase should be documented (see Table 1). In our example, we screened 100 abstracts, of which four were relevant, resulting in an estimated prevalence of 4%. We used three relevant abstracts as key studies. In our repository, the exported file is labeled "Tang_et_al_2022_unlabeled_pre_completed.csv," and the file with removed labels for key studies is named "Tang_et_al_2022_unlabeled_AI.csv."

Step 2: stopping-rule selection. In the following section, we summarize the evidence that informed our recommendations for selecting stopping rules. We then present these recommendations and illustrate their implementation with the example data set.

Evidence. When selecting stopping rules, both logical reasoning and empirical evidence underscore the importance of aligning thresholds with the estimated prevalence of relevant literature. From a logical perspective, higher prevalence reduces the intervals between relevant and irrelevant abstracts, meaning that even under random screening, any cutoff for the data-driven heuristic will largely depend on prevalence. In addition, because ML algorithms improve with larger training data sets, lower-prevalence conditions—by providing fewer relevant examples when the overall sample size is held constant—slow learning and delay the identification of relevant studies. This ultimately influences the performance of the time-based heuristic.

Empirical findings support these mechanisms. König, Zitzmann, Fütterer, et al. (2024) found that a 2.5% data-driven cutoff produced median identification rates above 95% when prevalence was 5% to 10% but only around 60% when prevalence was 0.5% to 1%. They also

observed that larger data sets improved performance and prevalence effects were nonlinear: Increasing prevalence from 0.5% to 5% reduced screening time to reach 95% recall, but further increasing prevalence to 10% increased screening time again because of the larger number of relevant abstracts to retrieve. Campos et al. (2024) reported similar findings: Higher prevalence reduced performance, whereas larger data sets improved it. In their sample of data sets with a median prevalence of 18% (ranging from 2% to 50%), conservative thresholds were required—at least 70% for the time-based heuristic and 7% for the data-driven heuristic—to reach 95% recall. Importantly, they showed that combining heuristics could achieve the same goal at lower cost; the most efficient combination was 20% of abstracts screened and a 5% irrelevant-abstract threshold. They also observed that all combinations using a 10% data-driven cutoff or the 10% cutoff alone consistently achieved the 95% threshold, whereas a 5% cutoff required higher time-based thresholds to perform reliably.

Other studies reinforce these patterns. Research on medical- and health-economics data sets has shown that thresholds of 100 to 200 consecutive irrelevant abstracts generally improved performance and reduced variability across replications, although at the cost of additional screening (Callaghan & Müller-Hansen, 2020; Oude Wolcherink et al., 2023; Scherhag & Burgard, 2023). For example, raising the cutoff from 50 to 100 markedly increased recall, and further increases did not improve median sensitivity. Even though such thresholds missed roughly 20% of potentially relevant abstracts, they still identified nearly 100% of studies deemed relevant after full-text screening. In another study, Campos et al. (2024) evaluated an adaptive data-driven heuristic in which cutoffs were defined as percentages of the data set. Setting the cutoff to 7% consistently achieved 95% recall across multiple educational and psychology data sets when using algorithms such as LR+SBERT. In contrast, other algorithms required a 10% cutoff to achieve a similar identification rate.

Taken together, this evidence highlights both the potential and variability of stopping rules across domains. Performance is strongly shaped by prevalence, data-set size, and algorithm stability. Conservative thresholds tend to improve recall but may require additional screening, reducing efficiency. Conversely, more liberal thresholds may save time but risk missing relevant studies that appear late. For meta-analytical research, comprehensiveness should be prioritized over efficiency, making conservative thresholds preferable. For exploratory work, efficiency may be prioritized instead. In both cases, we recommend aligning cutoffs with prevalence estimates obtained from an initial random subsample.

Recommendation. Following König, Zitzmann, Fütterer, et al. (2024) and the SAFE method (Boetje & van de Schoot, 2024), we endorse combining the key-study, data-driven, and time-based heuristics as primary rules and the data-driven heuristic also serving as a breakout rule. To further encourage conservative practice, we recommend adjusting the prevalence categories suggested by König, Zitzmann, Fütterer, et al. from < 2.5%, 2.5% to 7.5%, and > 7.5% to < 5%, < 10%, and > 10%. As prevalence increases, both data-driven and time-based cutoffs should be reduced. Building on these principles, we translate them into concrete guidelines for different prevalence levels, specifying how each stopping rule can be applied in practice and how breakout strategies can be incorporated. Based on findings from the prescreening phase, a prevalence estimate can be calculated. Corresponding to this estimate, we suggest the following guidelines:

- Prevalence ≤ 5% (Rule A): Stop screening when all three primary rules are satisfied—the time-based heuristic (35% cutoff), the data-driven heuristic (7% cutoff), and the key-study rule. Alternatively, screening may be stopped once 15% of the entire data set is irrelevant in a row provided that all key studies have already been identified (breakout rule).
- Prevalence > 5% to 10% (Rule B): Stop screening when the time-based heuristic (25% cutoff), the data-driven heuristic (5% cutoff), and the key-study rule are all met. Alternatively, screening may be stopped once 10% of the entire data set is irrelevant in a row, again assuming all key studies have been identified (breakout rule).
- Prevalence > 10% (Rule C): Stop screening when the time-based heuristic (15% cutoff), the data-driven heuristic (3% cutoff), and the key-study rule are satisfied. Alternatively, screening may be stopped once 10% of the data set is irrelevant in a row, provided all key studies have been identified (breakout rule).

Evidence of the effectiveness of the recommendations. Although in this tutorial we focus on implementing ASReview, we also conducted a simulation study using 36 data sets from Campos et al. (2024) and König, Zitzmann and Hecht (2024) to verify the outcomes of our recommendations. In each simulation, we first sampled studies until at least 100 abstracts had been screened and at least one relevant study had been identified. We then unlabeled the remaining relevant studies, marked them as key studies, and used one relevant abstract together with all irrelevant abstracts from prescreening as training data. This process was repeated 100 times for each data set. Screening was simulated with ASReview's built-in simulation mode via the

Table 2. Simulated Performance of the Stopping Rules Across Prevalence Categories

Category	Estimate	N_s	Rule	M_c	sd_c	m_s	sd_s	≥95	=100	n_b	n_key
≤ 5%	2.97%	1,174	A	46.01	11.42	99.04	1.42	100.00	46.85	79	26
			B	38.91	12.24	98.50	1.55	92.42	25.38	5	35
			C	31.15	12.64	96.00	4.74	75.89	3.49	0	103
5% to 10%	7.76%	910	A	59.35	16.07	99.53	0.73	100.00	43.85	46	30
			B	54.98	16.55	99.33	0.82	99.56	29.12	1	41
			C	47.10	15.82	97.74	3.24	90.88	15.71	0	169
> 10%	19.54%	1,100	A	79.02	16.81	99.74	0.47	100.00	58.55	0	57
			B	75.06	16.67	99.55	0.61	99.91	34.55	0	106
			C	69.22	17.21	98.92	1.65	97.09	28.36	0	267

Note: Estimate = average prevalence estimate based on the training sample; N_s = number of simulation runs within that category; rule = stopping procedure (Rules A, B, or C, as described in text); c = screening cost; s = sensitivity; ≥ 95 = percentage of simulation runs in which the rule achieved a sensitivity of at least 95%; ≥ 100 = percentage of simulation runs in which the rule achieved a sensitivity of 100%; n_b = number of times the breakout rule was triggered; n_key = number of times not all key studies were identified after the rules were met.

Python API, which mirrors reviewer decisions by accessing the column containing the original labels. For example, when the algorithm recommended screening an abstract marked as irrelevant, the system classified it as such before updating its predictions (see above). We then used the screening order of the labeled abstracts to apply the stopping rules.

The time-based heuristic was established as a reference point in the screening process. Before this point was reached, screening generally continued unless the breakout condition or the key-study rule was met. The data-driven heuristic was applied once the time threshold had been reached. Specifically, we counted the number of consecutive irrelevant abstracts since the last relevant abstract before the time-based heuristic was applied. When the proportion of this sequence exceeded the cutoff value, the key-study rule was applied. When all key studies had already been identified, screening was stopped; when not, screening continued until all key studies were found. Each rule (Rules A–C) was applied to every data set. We then calculated sensitivity and screening cost and documented whether the breakout and key-study rules had been triggered.

Next, we grouped the data sets by prevalence, estimated from random prescreening, to examine whether the rules performed differently across prevalence levels. We documented the average sensitivity and screening cost for each rule in each category (Table 2). In addition, we recorded the percentage of simulation runs in which each rule achieved a sensitivity above 95% and 100%. We provide the full documentation of this simulation, including code and data, on our OSF project page (https://osf.io/xrb9z/overview).

As shown in Table 2, all rules achieved a mean identification rate above 95% across prevalence levels. However, not all rules consistently reached this threshold in at least 95% of simulation runs. The most conservative approach, Rule A, achieved this threshold across all prevalence categories, Rule B did so only in the 5% to 10% and > 10% prevalence categories, and Rule C did so only in the latter. When screening costs are also considered, the most efficient rule, which achieved sensitivity above 95% in at least 95% of cases, was consistently the one tailored to the respective prevalence category.

Another important observation is that none of the rules succeeded in identifying 100% of relevant abstracts in 95% of the cases even when applying the most conservative option. This finding is not unexpected because the rules are derived from the literature, in which a 95% sensitivity threshold is typically the goal. Nonetheless, researchers seeking to identify all relevant abstracts should consider raising our recommended cutoff values.

Finally, the results highlight the contribution of the key-study stopping rule, which was triggered up to 292 times depending on the method and prevalence category. Likewise, the breakout rule was occasionally triggered even under the most conservative condition, underscoring its usefulness in reducing screening effort once most relevant abstracts had already been identified.

All in all, our rules performed as expected. Nonetheless, we wish to emphasize that although we view these recommendations as conservative, they do not guarantee that at least 95% of the relevant literature will be identified, and our sample was too small to be generalizable. For instance, some findings suggest that the performance of these rules depends not only on prevalence but also on data-set size (Campos et al., 2024; König, Zitzmann, Fütterer, et al., 2024). Researchers may therefore adjust the cutoff values upward to increase the likelihood of identifying all relevant articles or downward to improve screening efficiency when minimizing time is a priority. Moreover, when applying stopping rules other than

those included here, alternative algorithms may be more appropriate (see König, Zitzmann, & Hecht, 2024).

Implementation. Considering the prescreening results displayed on the "Dashboard," screening 100 studies yielded four relevant abstracts and 96 irrelevant abstracts, corresponding to an estimated prevalence of 4%. Using the cutoff values for stopping-procedure selection, this outcome would align with Rule A. For guidance on selecting appropriate procedures across different prevalence levels of relevant abstracts, see the Recommendations section. With this strategy, screening would stop once 35% of the abstracts have been reviewed, 7% have been consecutively labeled as irrelevant, and all key studies have been identified. Alternatively, the breakout strategy could be applied, which, in this case, would require 15% consecutive irrelevant abstracts. Note that ASReview does not display relative screening progress in the "Analytics" tab; instead, it reports the absolute number of screened abstracts and the number of consecutively irrelevant abstracts. Consequently, users must calculate the required thresholds manually. For example, in our data set containing 2,035 abstracts with an estimated prevalence of 4%, the time-based heuristic with a 35% cutoff would require screening at least $2{,}035 \times 0.35 = 712$ abstracts. The threshold for consecutively irrelevant abstracts—after this point—would be $2{,}035 \times 0.07 = 142$ abstracts in a row (data-driven heuristic). For the breakout strategy, the corresponding threshold would be $2{,}035 \times 0.15 = 305$ consecutive irrelevant abstracts.

Step 3: AI-aided screening. Although AI-aided screening can be initiated by simply changing the prediction model, creating a new project may be advantageous. This approach ensures that ASReview correctly labels prior knowledge and allows both screening phases to be shared among researchers. When continuing in the same project, prior knowledge can be identified by examining the screening IDs from 1 to *n* (labeled) after prescreening. The main advantage of continuing in the same project is that previously created tag groups do not need to be recreated. However, because the ASReview version used here allows key studies to be marked as unlabeled only outside the software, we focus on creating two separate projects for each screening phase.

For the data set used in the following example ("Tang_et_al_2022_unlabeled_AI.csv"), we removed all labels for key studies and documented their "data_ids" (stored in Table 1), which can be used to search for them after other stopping rules are triggered. The data set is available on our OSF project page (https://osf.io/xrb9z/overview).

Project setup. Model configuration is critical during the AI-aided screening phase. Because our recommendations are based on the LR+SBERT configuration, we encourage users to select the SBERT feature extractor ("all-mpnet-base-v2") in combination with the LR classifier (for results on other algorithms in psychology- and education-related literature, see Campos et al., 2024; König, Zitzmann, & Hecht, 2024). We further recommend using the "Maximum" (certainty-based) query strategy and activating the "Balancer" (Fig. 14). To the best of our knowledge, most evaluation studies on learning algorithms and stopping rules in ASReview have employed this setup (e.g., Campos et al., 2024; König, Zitzmann, Fütterer, et al., 2024; König, Zitzmann, & Hecht, 2024; Teijema et al., 2023).

Once the model is defined, ASReview proceeds with initialization, which involves training the model and generating the first ranking of abstracts by predicted relevance. Because the SBERT feature extractor is computationally intensive, this process may take up to 20 min, depending on data-set size and hardware specifications. During this time, users can either wait or continue random screening by clicking on "I can't wait." When initialization is complete, the model configuration is displayed above the abstract provided that "Show model information" is enabled in the "Settings" menu (Fig. 15).

Before starting active learning, users should enable ASReview's built-in stopping rule functionality via the "Dashboard." They can specify either the number or the percentage of consecutive irrelevant abstracts to activate the data-driven heuristic. Initially, we recommend using the value defined by the breakout strategy, which in our example corresponds to 305 abstracts (see above). Once this is set, screening can begin. During this phase, careful and accurate classification is essential because misclassifications can reduce the model's predictive accuracy.

Stopping. To determine whether the time-based heuristic has been met, users should check the number of labeled abstracts displayed above each abstract during screening when "Show model information" is enabled in "Settings." In our example, this rule would be satisfied once $0.35 \times 2{,}035 = 712$ abstracts had been screened. Once this threshold is reached, the cutoff for the data-driven heuristic should be adjusted according to the selected strategy. In our example, this corresponded to a percentage of 7%, which meant stopping after encountering 142 consecutive irrelevant abstracts.

Once these heuristic thresholds have been applied, ASReview will automatically stop (see Fig. 16). ASReview also asks whether users wish to take additional measures to further improve screening quality, such as increasing the cutoff value or selecting a different algorithm. Additional measures to increase sensitivity are described below under Additional Screening Measures. We do not recommend marking the project as finished until the

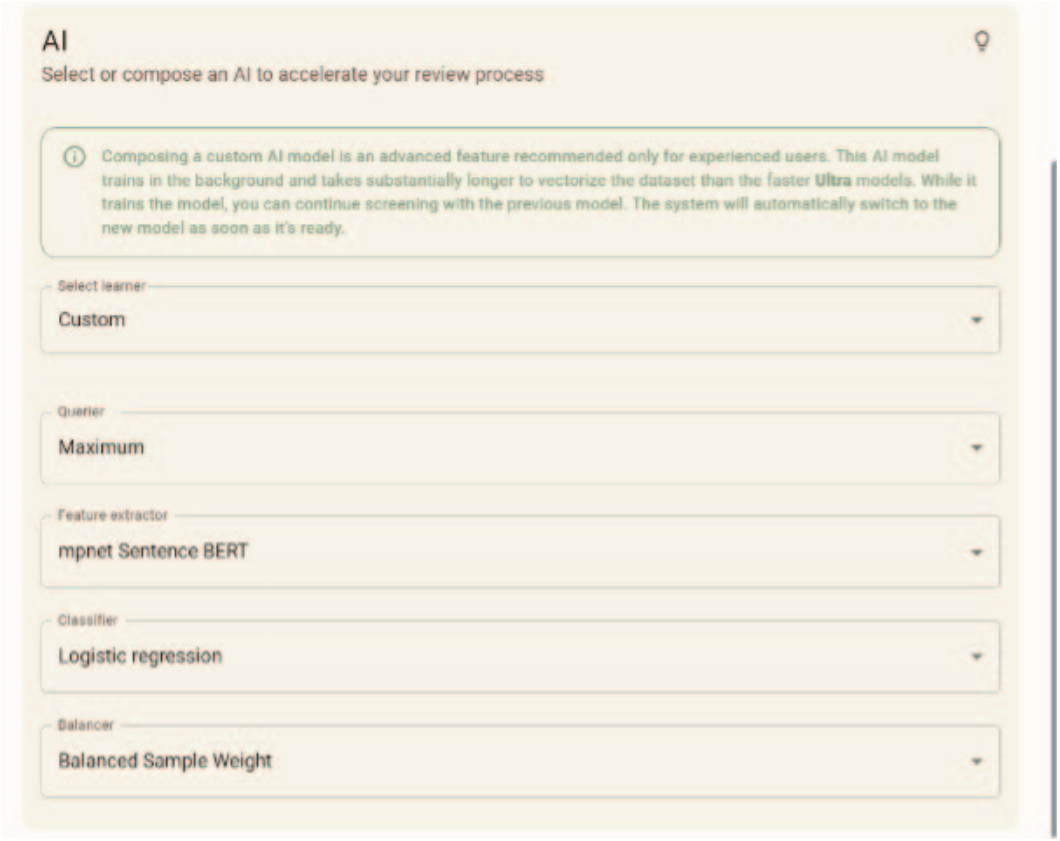

Fig. 14. Model setup for artificial-intelligence-aided semiautomated abstract screening.

results are verified. However, users can always change their project's status back to "unfinished" if they decide to continue screening.

To evaluate screening success, users can, for example, examine the "Dashboard" visualizations. The "Recall" chart (Fig. 17a) plots the cumulative number of relevant

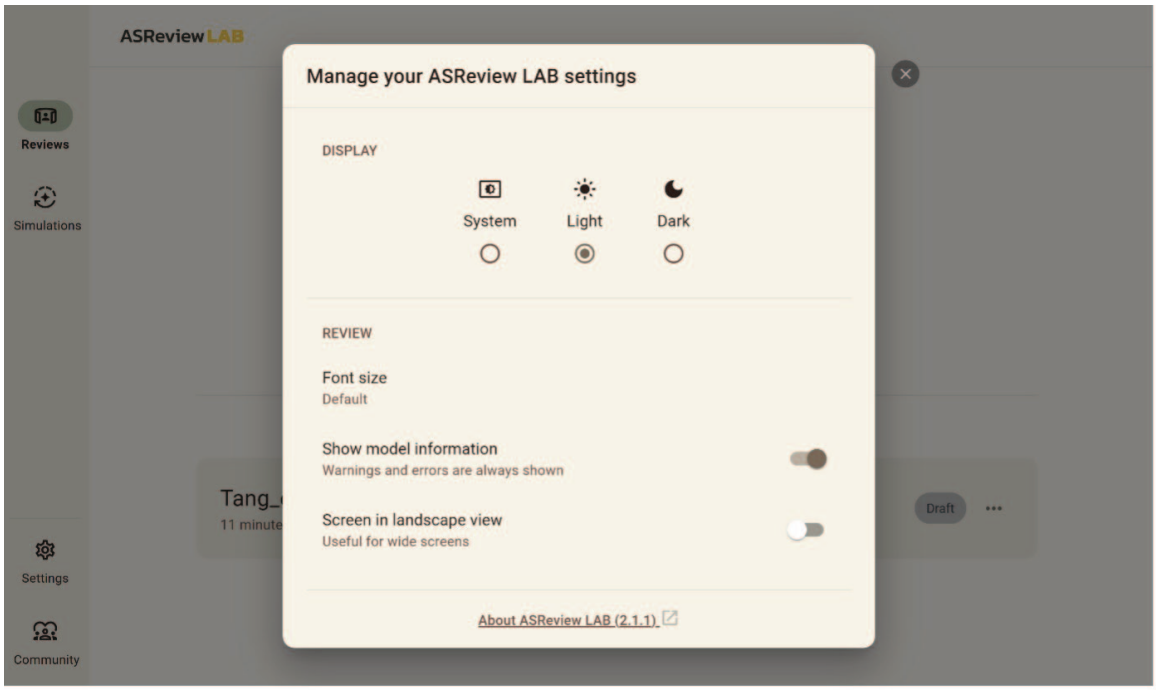

Fig. 15. Settings.

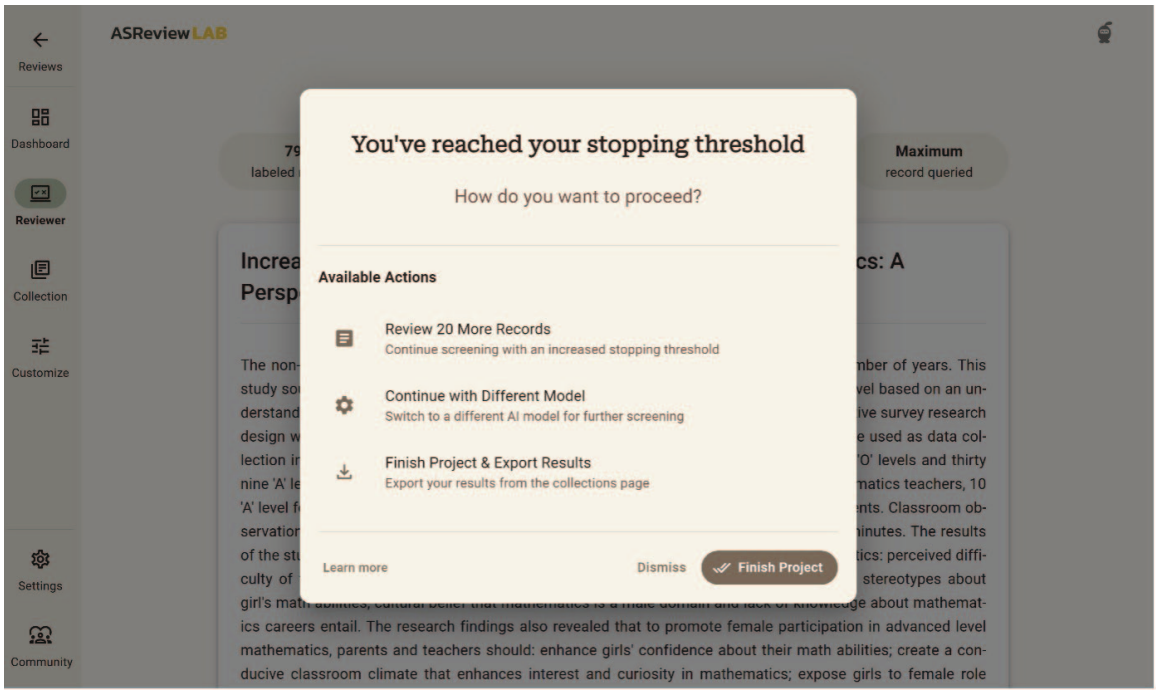

Fig. 16. Automatic stopping of ASReview and additional screening measures.

abstracts (*y*-axis) against the total number of screened abstracts (*x*-axis), providing a clear representation of progress over time. Researchers can focus on identifying the "knee" of the curve—the inflection point at which the model's ability to find relevant abstracts begins to decline, leading to longer sequences of irrelevant abstracts between relevant ones, which we show in the AI-aided screening section. This reflection point suggests that a large portion of the relevant abstracts have been identified. In our example, this inflection point is anywhere around 40 relevant abstracts, which roughly corresponds to 80% sensitivity. By hovering over the "Recall" chart with the mouse, users can also see how many more relevant abstracts have been identified compared with random screening. The "Wave" chart (Fig. 17b) can complement the "Recall" analysis by showing whether sequences of consecutive irrelevant abstracts are increasing or decreasing and whether the data-driven heuristic has been met. In our example, the intervals increase, clearly illustrating how the data-driven heuristic operates.

However, before concluding the screening, it is essential to confirm that all key studies have been identified by searching for their titles or notes. This can be done most efficiently by exporting the relevant labeled abstracts and then searching for their titles or "data_ids." When any key studies remain unidentified, screening should be extended until this condition is met. In our example, all key studies had already been identified. After a project is marked as finished in ASReview, the software automatically estimates the number of hours saved, which in our example corresponded to 25 working hours (see Fig. 18).

Results. With these conditions satisfied, the AI-aided screening process achieved a sensitivity of 98% and a screening cost of 39%. All but one relevant abstract were identified without the need to screen 61% of the irrelevant abstracts.

Finally, users can document their results as suggested in Table 1. On our OSF project page (https://osf.io/xrb9z/overview), the data set after screening is named "Tang_et_al_2022_unlabeled_AI_completed.csv."

Combining automated and semiautomated screening of abstracts

Advances in AI—particularly the development of LLMs—have further enhanced AI-aided abstract screening, especially in fully automated abstract labeling. Recent evidence suggests that state-of-the-art LLMs can reliably classify most irrelevant abstracts and a substantial proportion of relevant ones. For instance, ChatGPT (Version 4.0) has demonstrated sensitivity and specificity exceeding 90% (e.g., Li et al., 2024; Vembye et al., 2024). However, because these tools lack control over the selection process, combining automated- and

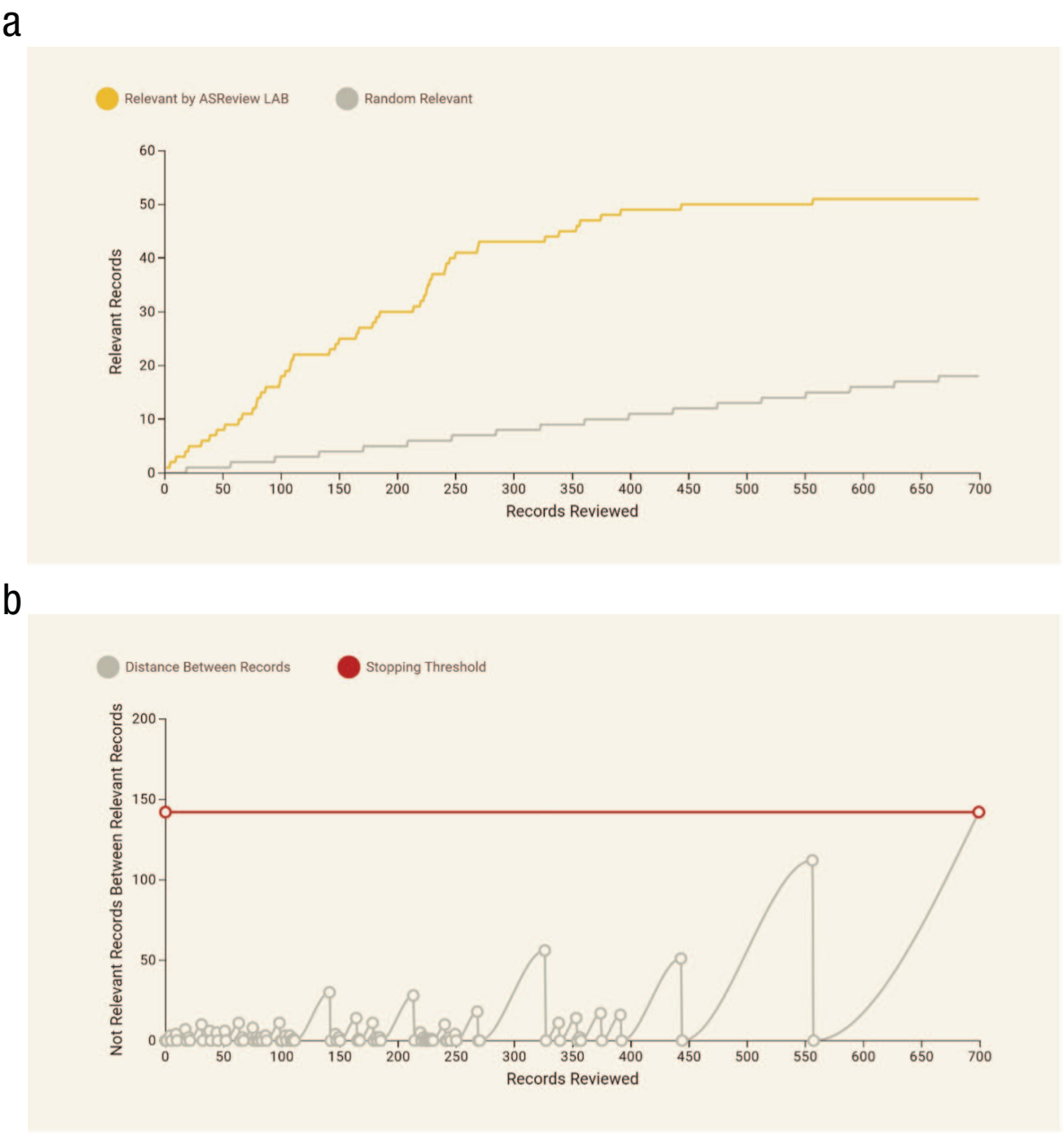

Fig. 17. Dashboard: (a) "Recall" and (b) "Wave" visualizations.

semiautomated-screening tools has been discussed in the literature as a valuable strategy to reduce the time and resources required for abstract screening while maintaining the same level of controllability as semiautomated screening alone (Bron et al., 2024). For example, users may verify all abstracts classified as relevant by the LLM and then use these to train an ML algorithm to detect potentially misclassified abstracts among those labeled as irrelevant.

Because in this tutorial we focus on ASReview, which currently supports only semiautomated abstract screening, we limit our guidance to controlling the results of an automated-screening procedure. Users seeking recommendations on conducting fully automated screening are referred to other sources (e.g., Bron et al., 2024; Li et al., 2024).

In the following section, we present our recommended procedure for combining automated with semiautomated screening while ensuring consistency with standard semiautomated approaches. This procedure is designed to align seamlessly with both the combined- and the purely semiautomated-screening workflows.

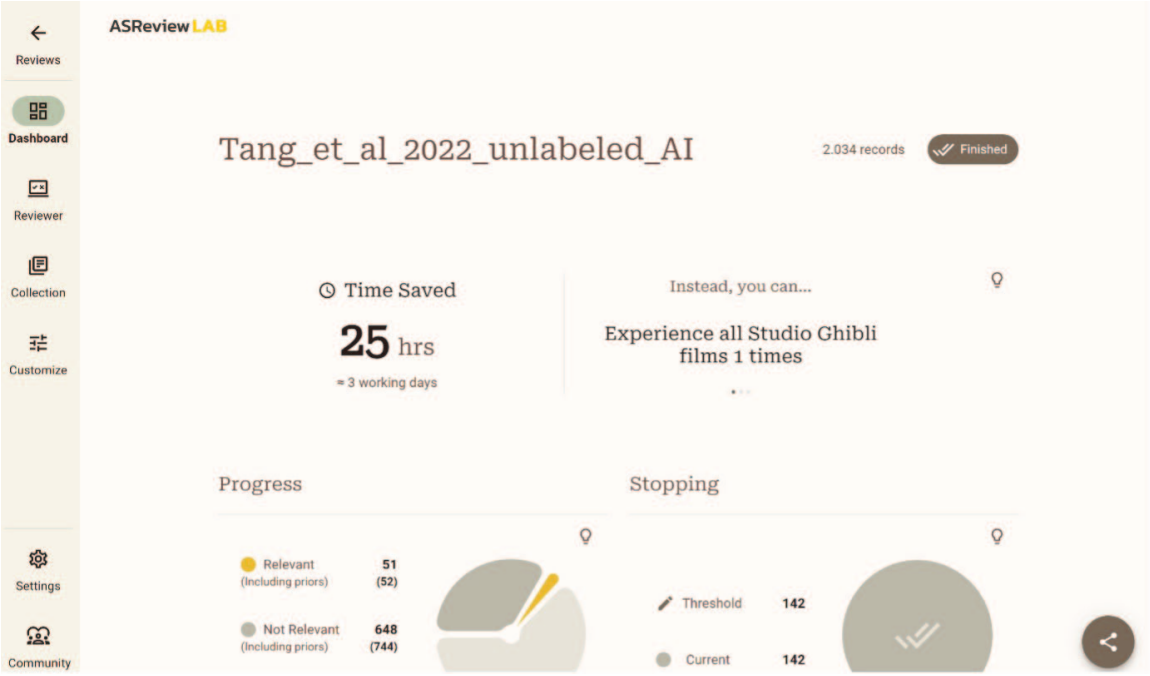

Fig. 18. Estimated time saved.

Table 3 provides a checklist of key information that should be documented at each step to ensure transparency.

Step 1: prescreening.
When combining fully automated with semiautomated screening, it is essential to first evaluate the classifications produced by the automated system. We recommend reviewing all abstracts identified as relevant by the automated screening and 100 of those identified as irrelevant. This approach ensures that the ML algorithms used in the AI-assisted screening phase are trained on accurately classified abstracts. In addition, the selected subset allows users to estimate the sensitivity and specificity of the automated-screening tool, thereby assessing its classification performance. These metrics can also be used to calculate the proportion of missed relevant abstracts and the prevalence of relevant abstracts in the data set. This information ensures that the process integrates seamlessly into semiautomated workflows, which have undergone more extensive evaluation. For example, when many relevant abstracts have potentially been misclassified by the automated-screening tool, the semiautomatic screening has less training data, and therefore, more conservative stopping rules for the AI-aided screening might be beneficial.

ASReview includes a feature for evaluating data sets with prelabeled abstracts by displaying the existing label directly below each abstract during review. Since Version 2.0, this functionality has been integrated into the standard review mode, with the label appearing above the buttons where users indicate whether the abstract is relevant or irrelevant (see Fig. 19). To activate the feature, the column containing these labels must be named "label_included." For example, when an abstract was previously labeled as irrelevant, that label will be visible during review. ASReview also stores both the original screening decisions and the reviewer's current decisions in separate columns ("label_included" and "asreview_label"). This setup enables the calculation of interrater reliability and supports the rescreening of abstracts previously labeled by human reviewers or automated tools.

A limitation of ASReview is its inability to exclusively screen abstracts labeled as relevant or irrelevant. In this procedure, the user would select all relevant identifiers and abstracts and a random sample of 100 irrelevant abstracts to create a new data set for prescreening. Thus, we recommend separating the data by the labels assigned by an automated-screening tool before conducting the prescreening. The subset will then be merged with the full data set when AI-aided screening is applied so that the prescreened data can be used to train the algorithm. However, users can use the following R code to prepare the data for prescreening. The required data set, "Tang_et_al_2022_prelabeled_full.csv," is available on our OSF project page (https://osf.io/xrb9z/overview).

Table 3. ASReview Checklist for Combining Automated and Semiautomatic Screening

Phases	Description	Examples
Prescreening	During the random screening of prelabeled abstracts, it is recommended that researchers document the data-set name, the number of abstracts labeled by the automated-screening tool, the number of abstracts included in the prescreening sample, the number of identified relevant and irrelevant abstracts, the estimated prevalence, and the sensitivity and specificity of the automated screening. The IDs of the key studies and training data are not relevant at this stage because all studies in the data set are screened and subsequently used as training data during the AI-aided screening phase.	• We screened all relevant and the same number of irrelevant labeled abstracts. • Data-set name: Tang_et_al_2022_PreScreeningPrelabeled_completed.csv • N_abstracts = 2,034 • N_dataset = 338 • n_screened = 338 • n_irr = 298 • n_rel = 40 • Sensitivity = 17% • Specificity = 100% • Prevalence = 2.5%
Stopping-rule selection	Explain and define stopping-rule and cutoff-value selection.	• Based on the prevalence of 2.5%, we decided to stop screening when 7% of the abstracts were irrelevant in a row (data-driven heuristic).
AI-aided screening	Specify details of the screening process.	• One reviewer; ambiguous cases are discussed by the team.
	Document the model details.	• Feature extractor: SBERT • Classifier: LR • Query strategy: certainty-based (maximum) • Balance strategy: balanced
	Document final results.	• Data-set name: Tang_et_al_2022_PreScreeningPrelabeled_completed • N_abstracts = 2,035 • n_screened = 825 • n_irr = 772 • n_rel = 53 • Prevalence = 2.6%

Note: AI = artificial intelligence; SBERT = Sentence-BERT; LR = logistic regression.

```
df <- read.csv("./Tang_et_al_2022_
  prelabeld_full.csv")
set.seed(435)
sub_r <- df[df$label_included == 1,
  "data_id"]
sub_i <- sample(df[df$label_included
  == 0, "data_id"], 100]
df_sub <- rbind(df[df$data_id %in%
  c(sub_r, sub_i), ])
write.csv(df_sub, "./Tang_et_al_2022_
  prelabeled_pre.csv", row.names =
  FALSE)
```

As demonstrated, we recommend setting a seed using the `set.seed()` function before sampling to ensure reproducibility when selecting irrelevant abstracts. For this step, only the "data_ids" of the article entries are needed. When the data set does not already contain a unique identifier, users can generate one, as shown in the example code. All references labeled as relevant and the 100 irrelevant references sampled can then be extracted using indexing in R (see the example code below). This subset can be saved and subsequently used for prescreening in ASReview.

Screening. To screen the subset of prelabeled abstracts, the model does not matter because all abstracts of the subset require screening. The rest of the project-setup process follows the same procedure described in the prescreening for unlabeled data (see above). During screening, users can monitor whether their screening decisions align with the preassigned labels, enabling the calculation of true positives, false positives, true negatives, and false negatives. In addition, they can define the cutoff of the time-based heuristic for the breakout strategy and combined strategy, respectively. However, manually recording each decision can be time-consuming, and ASReview does not provide built-in functionality to directly extract this information. Thus, we recommend exporting the data set after all abstracts have been screened and using R to prepare the data for AI-aided screening and to estimate the sensitivity, specificity, and prevalence of relevant abstracts.

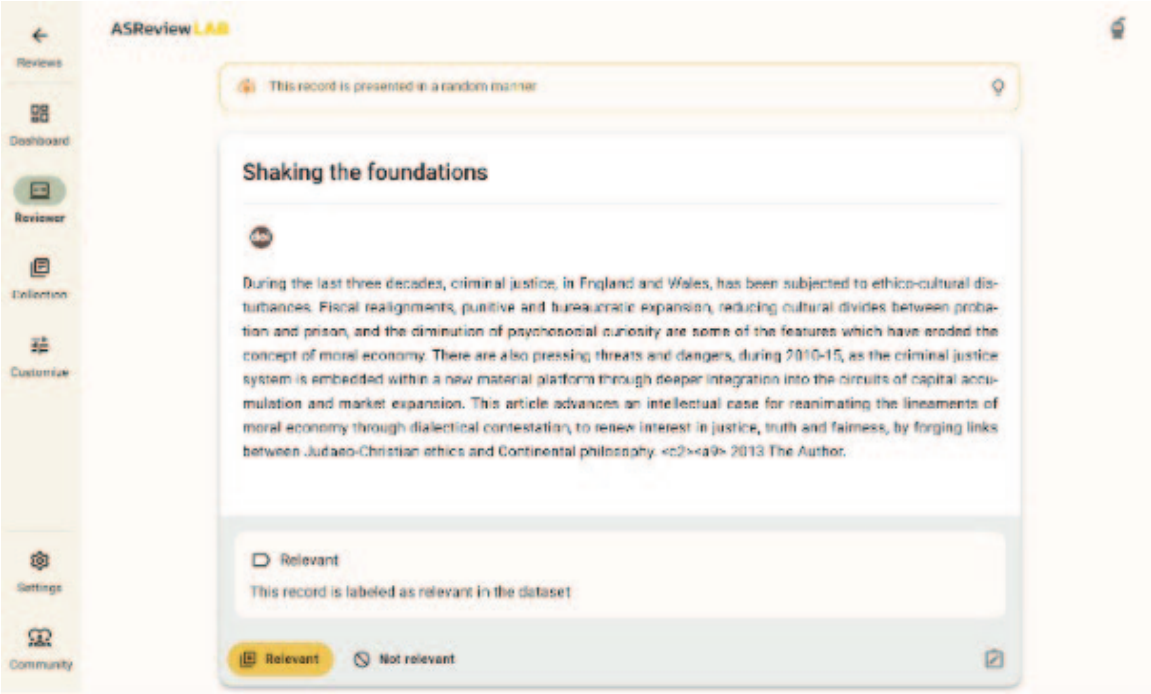

Fig. 19. Reviewing with prior labels displayed.

To retrieve the necessary information—frequencies of true positives, true negatives, false positives, and false negatives—the `table()` function in *R* can be used:

```
table(df_sub$label_included,
  df_sub$asreview_label)
```

Output:

```
     0   1
0  100 198
1    0  40
```

In this function, the rows correspond to the first variable entered, and the columns correspond to the second. As shown in the output, the automatic screening correctly labeled 40 relevant abstracts and 100 irrelevant abstracts from the subset. However, it also misclassified 198 irrelevant abstracts as relevant. This screening would have resulted in an observed interrater agreement (Equation 4) of 41% and a specificity of 100%. Although these estimates might not be entirely accurate because they are based on only a small sample, they inform users whether they need to correct the prevalence estimate. When specificity is below 100%, users can adjust the prevalence estimate by accounting for relevant abstracts that the ML algorithm labeled as irrelevant. One way to do this is to estimate how often abstracts classified by the ML algorithm as irrelevant are relevant in the checked subset and then apply this rate to the remaining abstracts that have not been evaluated by the human reviewer (see Equation 5). However, because stopping rules for lower prevalences are more conservative, ignoring potentially relevant abstracts that might have been missed in prevalence estimation also yields conservative stopping rules. In our example, we estimated a specificity of 100%. Thus, we would estimate the prevalence from the 40 relevant abstracts, yielding a prevalence of roughly 2%. This information should be documented (see Table 3) for use when selecting the stopping strategy.

Before proceeding, users should merge the prescreening data with the data produced by the automated-screening method that was not used for prescreening. ASReview automatically adds or modifies some columns in the data set, such as the new "asreview_label" column storing the user's decision, columns for all tags, "asreview_time," and "asreview_note."

Because both data sets use the previously created "data_id" column to store the same identifier for each abstract, this column can now be used to remove all abstracts screened during prescreening from the unscreened data set. This process ensures that no duplicates are introduced when merging the prescreened and unscreened data sets to create the final data set for AI-aided screening. To remove the prescreened data entries from the unscreened data set, we used indexing

in R. To merge the data sets, we recommend using the `bind_rows()` function from the *dplyr* package (Wickham et al., 2020), which allows for row-wise merging of data frames even when they contain different columns. Once these steps are completed, the data set for AI-aided screening can be merged and used for AI-aided screening:

```
library(dplyr)
setwd("Path/To/Your/Data")
df_LLM <- read.csv("./Tang_et_al_2022_
  prelabeled_full.csv")
df_sub <- read.csv("./Tang_et_al_2022_
  prelabeled_pre_completed.csv")
names(df_LLM)[names(df_LLM) == "label_
  included"] <- "LLM_label"
names(df_sub)[names(df_sub) == "label_
  included"] <- "LLM_label"
df <- df_LLM[!df_LLM$data_id %in% df_
  sub$data_id, ]
df <- bind_rows(df, df_sub)
write.csv(df, "./Tang_et_al_2022_
  prelabeled_AI.csv", row.names = FALSE)
```

Step 2: stopping-rule selection. In the following section, we outline the underlying rationale that guided our recommendations for selecting stopping rules and illustrate their implementation with the example data set.

Evidence. Although empirical evaluations of combined semiautomated- and automated-screening approaches have shown high rates of correctly classified abstracts, generalizable evidence for their combination remains limited. Consequently, reliable recommendations for stopping rules are also lacking. Nonetheless, our proposed procedure allows existing evidence from semiautomated screening to serve as a guideline. For instance, recent findings suggest that screening approximately 30% of abstracts often yields 80% of the relevant studies (Campos et al., 2024). Thus, applying a time-based heuristic may no longer be necessary when the automated tool has already identified more than 80% of relevant abstracts. In such cases, the cutoff values for the time-based heuristic can be aligned with the estimated percentage of already identified relevant abstracts. This proportion can be calculated using the sensitivity and specificity values obtained during prescreening. Likewise, these values can inform prevalence estimates, which can then be used to align the cutoff values for the data-driven heuristic, as described for semiautomated screening. However, given the lack of robust evidence, we propose adopting more conservative estimates regardless of prevalence.

Recommendation. Based on these considerations, we suggest the following strategies while emphasizing that they lack direct empirical support beyond what is available for semiautomated screening:

- Strategy A: When the estimated specificity of the automated-screening tool exceeds 80%, apply only the data-driven heuristic, with a suggested cutoff value of 7% regardless of the estimated prevalence.
- Strategy B: When the estimated specificity of the automated-screening tool is between 50% and 80%, combine the data-driven heuristic with a time-based heuristic. We suggest a 10% cutoff for the time-based heuristic and a 7% cutoff for the data-driven heuristic regardless of the estimated prevalence.
- Strategy C: When the estimated specificity of the automated-screening tool is below 50%, combine the data-driven heuristic with a time-based heuristic, aligning the cutoff values with the estimated prevalence. Specifically, when prevalence is below 5%, we suggest a 20% cutoff for the time-based heuristic and a 7% cutoff for the data-driven heuristic. When prevalence is above 5%, we suggest a 10% cutoff for the time-based heuristic and conservatively, a 7% cutoff for the data-driven heuristic.

Implementation. In our example, we observed a specificity of 100% and a prevalence of 2%. Thus, we would use Strategy A, applying only the data-driven heuristic with a cutoff of 7%. However, because we screened 338 abstracts to evaluate the automated screening, we had already screened about 15% of the abstracts. Thus, a 15% cutoff for the time-based heuristic would have been fulfilled regardless. However, when accuracy was below 50%, we would stop at 15% with an additional 5% and then apply the data-driven heuristic, which corresponds to Strategy B.

Because ASReview does not support selecting a cutoff of 7%, we would need to calculate the corresponding number of abstracts, which is 142.

Step 3: AI-aided screening

Project setup. After completing the prescreening and selecting a stopping-rule procedure, the AI-aided screening process can be initialized. However, in contrast to prescreening, the AI-aided screening process is the same for both unlabeled and prelabeled references, and the model is essential. For the example presented here, we use the data set "Tang_et_al_2022_prelabeled_AI.csv," available on our OSF project page (https://osf.io/xrb9z/overview).

Screening. For the AI-aided screening of prelabeled references, we recommend the same model configuration as for semiautomated screening (see Fig. 14). The model may take longer to initialize because of the larger volume of training data. However, when users want to start screening while the model is trained, they can click on

"I can't wait" to start screening randomly. Before screening, we recommend creating tags for the inclusion and exclusion criteria and defining the cutoff for the data-driven heuristic in the "Dashboard" section before screening the first abstract.

In this example, we met the stopping threshold after screening 825 abstracts (see Fig. A2 in Appendix A), which corresponds to 41% of the abstracts. As a result, all relevant articles were identified. Thus, we would have saved 59% of the screening time without any loss. A cutoff value of 5% for the time-based heuristic would have resulted in missing one relevant article. The semiautomated screening took 5% longer to achieve the same identification rate of relevant articles. However, we want to emphasize that our example is not generalizable, and we simulated a worse-performing automated-screening method than state-of-the-art research suggests (e.g., Li et al., 2024; Vembye et al., 2024). However, before concluding the screening, users should export the data (e.g., "Tang_et_al_2022_prelabeled_AI_completed.csv") and document their results. Moreover, they can take additional steps to improve the quality of their screening, as summarized below.

Additional screening measures

After the two screening phases are complete, users can take additional steps to further enhance their screening measures and improve transparency and screening performance. For example, researchers can enhance screening quality by following the SAFE procedure's recommendations (Boetje & van de Schoot, 2024). This approach suggests activating a different AI model and continuing screening until 50 consecutive irrelevant abstracts have been seen. Research has shown that different ML algorithms may rank abstracts differently, with some models performing better in identifying the final relevant studies because of their complexity. However, when using the LR+SBERT algorithm, switching to a different model did not improve performance (Teijema et al., 2023). Nonetheless, in coordinated projects involving multiple reviewers, using different ML algorithms or training sets may be beneficial because these variations can affect the screening order. When applying alternative ML models, users may also want to adjust the cutoff values of the stopping rules in accordance with the respective algorithm estimates (see Campos et al., 2024; König, Zitzmann, Fütterer, et al., 2024).

Working in research groups also enables interrater reliability (agreement) to be estimated, as demonstrated by the agreement between the machine and the human reviewer in the example above. However, the reviewers will not necessarily see the same abstracts because they may use different algorithms and prior knowledge. As a result, they will likely screen different abstracts, especially those labeled as irrelevant. Even so, an initial estimate of agreement is still possible when reviewers screen different numbers of abstracts or different sets altogether. This estimate should be interpreted cautiously and should not be treated as true interrater reliability.

Another recommendation of the SAFE procedure is to rescreen all abstracts initially classified as irrelevant using all identified relevant abstracts as training data. This screening phase can serve as an error-control step and be concluded once 50 consecutive irrelevant abstracts have been screened (Boetje & van de Schoot, 2024).

Regardless of whether additional measures are undertaken, we emphasize that users of AI-aided screening tools should document all screening steps and results, as illustrated in Table 1. In addition, all data sets should be stored to ensure transparency and reproducibility. This practice can support future research efforts, such as updating a meta-analysis (Neeleman et al., 2024). We also highlight that the first two steps of the recommended screening procedure can be preregistered. Moreover, the preregistration can be updated before conducting the AI-aided screening to include specific stopping strategies. To further support transparency and reproducibility, we provide text examples for preregistering AI-aided abstract screening and for reporting the procedure in a research paper on our OSF project page (https://osf.io/xrb9z/overview). These templates are intended to facilitate the documentation of screening workflows and decision rules in line with open-science practices.

Conclusion

AI-supported tools such as ASReview offer clear advantages for conducting systematic reviews and meta-analyses. By applying evidence-based methods for semiautomated abstract screening, researchers can achieve substantial time savings without compromising methodological rigor. In this tutorial, we provided a practical guide to help researchers make optimal use of ASReview, grounded in the latest scientific evidence. We outlined empirically supported recommendations for appropriate stopping rules to ensure the efficient and accurate identification of relevant studies.

Although we derived these suggestions (e.g., regarding stopping-rule selection) from empirical evidence and evaluated them in a simulation study, we note that they have not yet been evaluated in a way that allows generalizability. For example, a large part of the findings that guided the recommendations is based on the same data we used to test them in the simulation study,

essentially introducing data leakage. However, they also align with findings of studies using different data (see above).

It is crucial to acknowledge that strict adherence to these recommendations does not guarantee optimal identification of the relevant literature. In fact, they are designed to identify at least 95% of the relevant literature, not 100%, because reaching full recall would substantially diminish the time gains of AI-aided screening. However, once the percentage of potential missed studies is known, meta-analysts can assess the robustness of their results. For example, the remaining 5% of potentially missed studies could be treated similarly to the fail-safe *N* method (Orwin, 1983), which examines the robustness of meta-analytic findings under assumptions about missing evidence. Specifically, one could augment the data set with a small number of hypothetical studies that represent this 5% and assign them more conservative effect sizes (e.g., closer to the null or even in the opposite direction of the observed effect). Controlling the robustness of meta-analytical findings under such scenarios would allow researchers to assess whether their conclusions remain stable when assuming that a small fraction of relevant studies was not identified during screening.

However, researchers must still critically evaluate various factors when selecting a stopping criterion, including the specificity of their inclusion criteria and the characteristics of their research domain. Specifically, the adoption of broad inclusion criteria or imprecise terminology can adversely affect the performance of ML algorithms. Future research should assess the performance of ML algorithms and stopping strategies using large-scale, labeled training data sets, as would be expected when evaluating screening decisions produced by automated-screening methods.

Finally, despite promising developments, an important question remains: How will reviewers and journal editors respond to the use of automated procedures? Whereas anecdotal evidence suggests both skepticism and growing acceptance of AI-aided screening, more systematic evaluation is needed. In this article, we aim to contribute to the conversation by demonstrating that ASReview can be used responsibly and methodologically soundly, based on the best available evidence. Future research and academic discussions should continue to explore how AI-supported screening tools can be effectively integrated and standardized in the scientific publication process.

Appendix A

List of abbreviations

AI = artificial intelligence.
API = application programming interface.
ASReview = Active Screening Review.
CSV = comma-separated values.
DOI = digital object identifier.
ELAS = efficient learning for ASReview.
FN = false negative.
FP = false positive.
LLM = large language model.
LR = logistic regression.
ML = machine learning.
OSF = Open Science Framework.
PICOS = population, intervention, comparison, outcome, study design.
UI = user interface.
SAFE = The SAFE procedure consists of four phases: screen a random set for training data, apply active learning, find more relevant abstracts with a different model, evaluate quality.
SBERT = Sentence-BERT (Sentence Bidirectional Encoder Representations from Transformers).
SVM = support vector machine.
TF-IDF = term frequency-inverse document frequency.
TN = true negative.
TP = true positive.

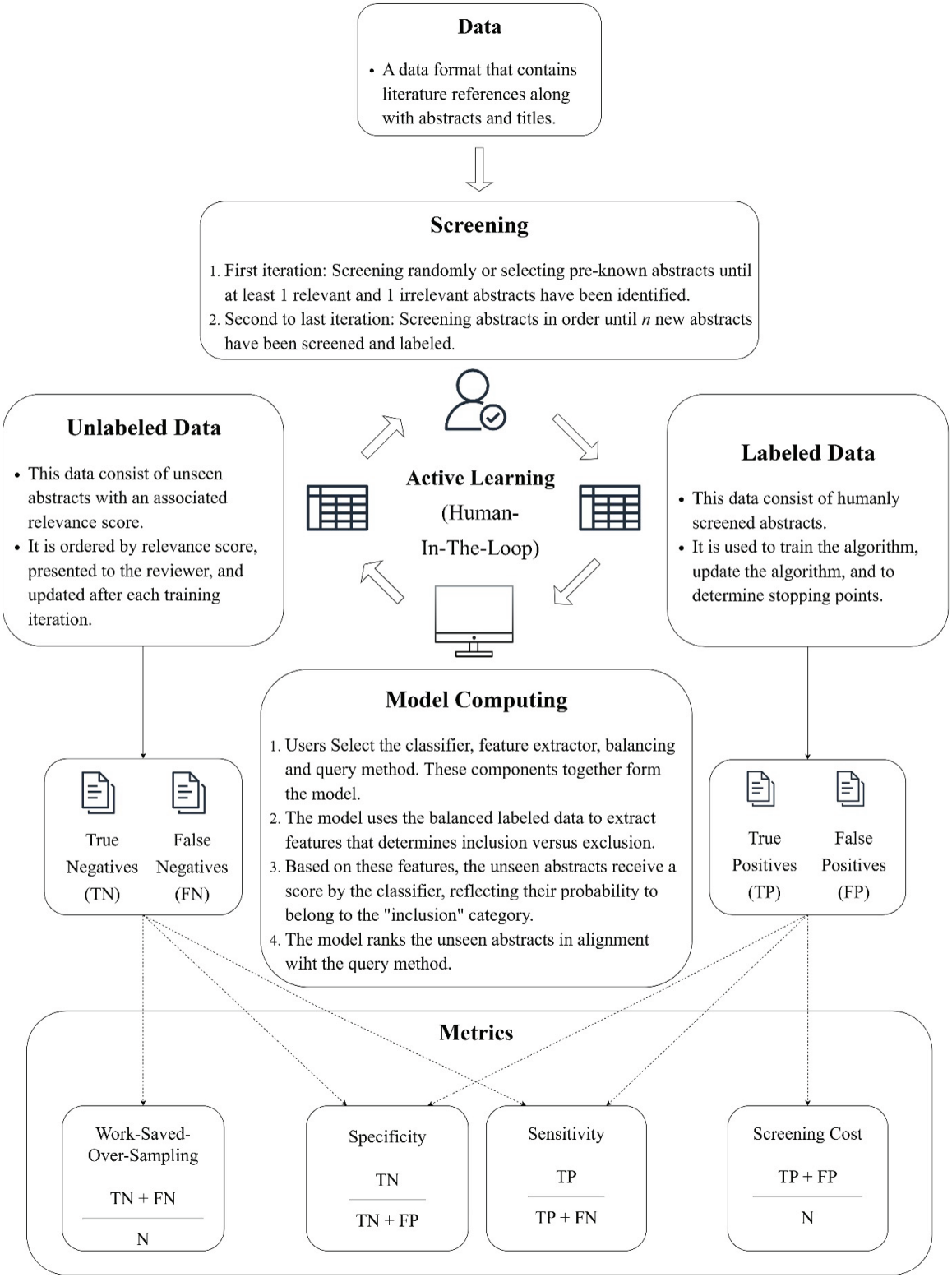

Fig. A1. Active learning in the realm of artificial-intelligence-aided abstract screening. Adapted from König, Zitzmann, and Hecht (2024).

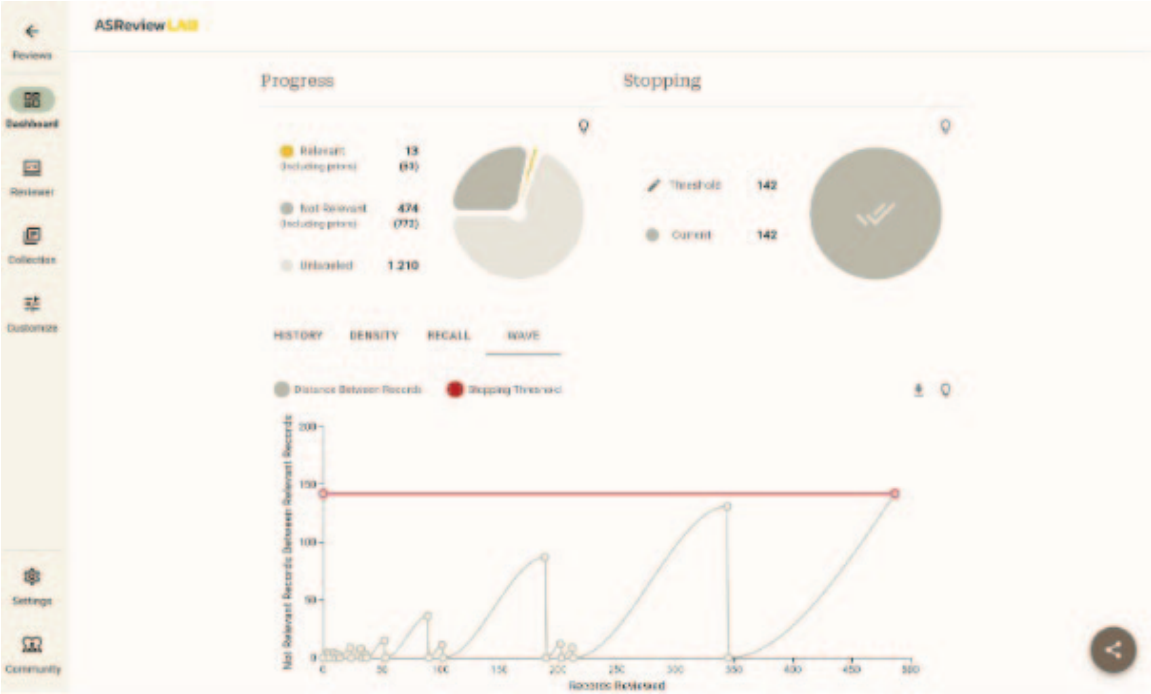

Fig. A2. Analytics after stopping combined artificial-intelligence-aided screening.

Transparency

Action Editor: David A. Sbarra
Editor: David A. Sbarra
Author Contributions

Tim Fütterer: Conceptualization; Data curation; Formal analysis; Investigation; Methodology; Project administration; Software; Visualization; Writing – original draft.

Lars König: Conceptualization; Data curation; Formal analysis; Methodology; Software; Visualization; Writing – original draft; Writing – review & editing.

Diego G. Campos: Formal analysis; Methodology; Writing – review & editing.

Ronny Scherer: Writing – review & editing.

Steffen Zitzmann: Writing – review & editing.

Martin Hecht: Project administration; Supervision; Writing – review & editing.

Tim Fütterer and Lars König shared first authorship.

Declaration of Conflicting Interests

The author(s) declared that there were no conflicts of interest with respect to the authorship or the publication of this article.

Funding

This research was supported in part by LEAD Intramural Research Funding from the University of Tubingen and the Postdoctoral Academy of Education Sciences and Psychology of the Hector Research Institute of Education Sciences and Psychology, Tübingen, funded by the Baden-Württemberg Ministry of Science, Research, and the Arts. This work was partially supported by the Research Council of Norway through its Centres of Excellence scheme, Project No. 331640. This project has received funding from the European Union's Horizon Europe research and innovation program under Grant Agreement No. 101132474. The simulation study used the HPC-cluster HSUper, which has been provided by the project hpc.bw, funded by dtec.bw—Digitalization and Technology Research Center of the Bundeswehr. dtec.bw is funded by the European Union–NextGenerationEU.

Open Practices

This article has received the badges for Open Data and Open Materials. More information about the Open Practices badges can be found at http://www.psychologicalscience.org/publications/badges.

ORCID iDs

Tim Fütterer https://orcid.org/0000-0001-5399-9557
Diego G. Campos https://orcid.org/0000-0002-8820-5881
Ronny Scherer https://orcid.org/0000-0003-3630-0710
Martin Hecht https://orcid.org/0000-0002-5168-4911

Acknowledgments

We want to thank Rebekka Steinhäuser for her support in this work. The authors are responsible for the content of this publication. We declare that we used Grammarly (Version 1.105.0.0) and ChatGPT (GPT-4o, Version 1.2025.105) to review and refine the text for spelling, grammar, and coherence of phrasing. Views and opinions expressed are those of

the author(s) only and do not necessarily reflect those of the European Union or the Agency. Neither the European Union nor the granting authority can be held responsible for them.

Note

1. In our OSF project, we also provide a short video tutorial that illustrates the basic functionalities and workflow of ASReview based on this written tutorial. Moreover, we provide a short version of all necessary steps for applied researchers. However, this written tutorial encompasses additional background information and details.

References

ASReview. (n.d.). *Seven ways to integrate ASReview in your systematic review workflow*. https://asreview.nl/blog/seven-ways-to-integrate-asreview/

ASReview LAB Developers. (2022). *ASReview LAB software documentation*. Zenodo. https://doi.org/10.5281/ZENODO.7319090

Boetje, J., & van de Schoot, R. (2024). The SAFE procedure: A practical stopping heuristic for active learning-based screening in systematic reviews and meta-analyses. *Systematic Reviews*, *13*(1), Article 81. https://doi.org/10.1186/s13643-024-02502-7

Borah, R., Brown, A. W., Capers, P. L., & Kaiser, K. A. (2017). Analysis of the time and workers needed to conduct systematic reviews of medical interventions using data from the PROSPERO registry. *BMJ Open*, *7*(2), Article e012545. https://doi.org/10.1136/bmjopen-2016-012545

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (2017). *Classification and regression trees*. Routledge. https://doi.org/10.1201/9781315139470

Bron, M. P., Greijn, B., Coimbra, B. M., van de Schoot, R., & Bagheri, A. (2024). Combining large language model classifications and active learning for improved technology-assisted review. *CEUR Workshop Proceedings*, *3770*, 77–95.

Bühler, B., Fütterer, T., Von Keyserlingk, L., Bozkir, E., Kasneci, E., Gerjets, P., & Trautwein, U. (2025). Mapping mind wandering to the "self-regulated learning process, multimodal data, and analysis grid": A systematic review. *Educational Psychology Review*, *37*(3), Article 76. https://doi.org/10.1007/s10648-025-10041-3

Burgard, T., & Bittermann, A. (2023). Reducing literature screening workload with machine learning: A systematic review of tools and their performance. *Zeitschrift Für Psychologie*, *231*(1), 3–15. https://doi.org/10.1027/2151-2604/a000509

Callaghan, M. W., & Müller-Hansen, F. (2020). Statistical stopping criteria for automated screening in systematic reviews. *Systematic Reviews*, *9*(1), Article 1. https://doi.org/10.1186/s13643-020-01521-4

Campos, D. G., Fütterer, T., Gfrörer, T., Lavelle-Hill, R., Murayama, K., König, L., Hecht, M., Zitzmann, S., & Scherer, R. (2024). Screening smarter, not harder: A comparative analysis of machine learning screening algorithms and heuristic stopping criteria for systematic reviews in educational research. *Educational Psychology Review*, *36*(1), Article 19. https://doi.org/10.1007/s10648-024-09862-5

Carrasco-Labra, A., Urquhart, O., & Spallek, H. (2021). Machine learning in evidence synthesis research. In C.-C. Ko, D. Shen, & L. Wang (Eds.), *Machine learning in dentistry* (pp. 147–161). Springer International Publishing. https://doi.org/10.1007/978-3-030-71881-7_12

Chai, K. E. K., Lines, R. L. J., Gucciardi, D. F., & Ng, L. (2021). Research Screener: A machine learning tool to semi-automate abstract screening for systematic reviews. *Systematic Reviews*, *10*(1), Article 93. https://doi.org/10.1186/s13643-021-01635-3

Chernikova, O., Stadler, M., Melev, I., & Fischer, F. (2024). Using machine learning for continuous updating of meta-analysis in educational context. *Computers in Human Behavior*, *156*, Article 108215. https://doi.org/10.1016/j.chb.2024.108215

Cooper, H. M., Hedges, L. V., & Valentine, J. C. (Eds.). (2019). *Handbook of research synthesis and meta-analysis* (3rd ed.). Russell Sage Foundation.

Cormack, G. V., & Grossman, M. R. (2016). Engineering quality and reliability in technology-assisted review. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 75–84). Association for Computing Machinery. https://doi.org/10.1145/2911451.2911510

Dai, Z.-Y., Shen, C., Ji, Y.-L., Li, Z.-Y., Wang, Y., & Wang, F.-Q. (2024). *Accuracy of large language models for literature screening in systematic reviews and meta-analyses*. SSRN. https://doi.org/10.2139/ssrn.4943759

de Bruin, J., Lombaers, P., Kaandorp, C., Teijema, J., van der Kuil, T., Yazan, B., Dong, A., & van de Schoot, R. (2025). ASReview LAB v.2: Open-source text screening with multiple agents and a crowd of experts. *Patterns*, *6*(7), Article 101318. https://doi.org/10.1016/j.patter.2025.101318

Elicit. (2025). *Elicit: The AI Research Assistant [AI tool]* [Computer software]. https://elicit.com

Ferdinands, G., Schram, R., De Bruin, J., Bagheri, A., Oberski, D. L., Tummers, L., & van de Schoot, R. (2020). *Active learning for screening prioritization in systematic reviews—A simulation study*. OSF. https://doi.org/10.31219/osf.io/w6qbg

Fütterer, T., Campos, D. G., Gfrörer, T., Lavelle-Hill, R., Murayama, K., & Scherer, R. (2026). AI tools for systematic literature reviews and meta-analyses in educational psychology: An overview and a practical guide. *Learning and Individual Differences*, *126*, Article 102849. https://doi.org/10.1016/j.lindif.2025.102849

Gates, A., Guitard, S., Pillay, J., Elliott, S. A., Dyson, M. P., Newton, A. S., & Hartling, L. (2019). Performance and usability of machine learning for screening in systematic reviews: A comparative evaluation of three tools. *Systematic Reviews*, *8*(1), Article 278. https://doi.org/10.1186/s13643-019-1222-2

Gates, A., Johnson, C., & Hartling, L. (2018). Technology-assisted title and abstract screening for systematic reviews: A retrospective evaluation of the Abstrackr machine learning tool. *Systematic Reviews*, *7*(1), Article 45. https://doi.org/10.1186/s13643-018-0707-8

Harrison, H., Griffin, S. J., Kuhn, I., & Usher-Smith, J. A. (2020). Software tools to support title and abstract screening for

systematic reviews in healthcare: An evaluation. *BMC Medical Research Methodology*, *20*(1), Article 7. https://doi.org/10.1186/s12874-020-0897-3

König, L., Zitzmann, S., Fütterer, T., Campos, D. G., Scherer, R., & Hecht, M. (2024). An evaluation of the performance of stopping rules in AI-aided screening for psychological meta-analytical research. *Research Synthesis Methods*, *15*(6), 1120–1146. https://doi.org/10.1002/jrsm.1762

König, L., Zitzmann, S., & Hecht, M. (2024). *Strategizing AI utilization for psychological literature screening: A comparative analysis of machine learning algorithms and key factors to consider*. PsyArXiv. https://doi.org/10.31234/osf.io/nc8hs

Li, M., Sun, J., & Tan, X. (2024). Evaluating the effectiveness of large language models in abstract screening: A comparative analysis. *Systematic Reviews*, *13*(1), Article 219. https://doi.org/10.1186/s13643-024-02609-x

Manning, C. D., Raghavan, P., & Schütze, H. (2009). *An introduction to information retrieval*. Cambridge University Press. https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf

Methley, A. M., Campbell, S., Chew-Graham, C., McNally, R., & Cheraghi-Sohi, S. (2014). PICO, PICOS and SPIDER: A comparison study of specificity and sensitivity in three search tools for qualitative systematic reviews. *BMC Health Services Research*, *14*(1), Article 579. https://doi.org/10.1186/s12913-014-0579-0

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient estimation of word representations in vector space* (Version 3). arXiv. https://doi.org/10.48550/ARXIV.1301.3781

Moreau, D., & Gamble, B. (2022). Conducting a meta-analysis in the age of open science: Tools, tips, and practical recommendations. *Psychological Methods*, *27*(3), 426–432. https://doi.org/10.1037/met0000351

Neeleman, R., Leenaars, C. H. C., Oud, M., Weijdema, F., & Van De Schoot, R. (2024). Addressing the challenges of reconstructing systematic reviews datasets: A case study and a noisy label filter procedure. *Systematic Reviews*, *13*(1), Article 69. https://doi.org/10.1186/s13643-024-02472-w

OpenAI. (2025). *ChatGPT Deep Search* (Version April 13 [Large language model]) [Computer software]. https://chat.openai.com

Orwin, R. G. (1983). A fail-safe *N* for effect size in meta-analysis. *Journal of Educational Statistics*, *8*(2), 157–159. https://doi.org/10.3102/10769986008002157

Oude Wolcherink, M. J., Pouwels, X. G. L. V., Van Dijk, S. H. B., Doggen, C. J. M., & Koffijberg, H. (2023). Can artificial intelligence separate the wheat from the chaff in systematic reviews of health economic articles? *Expert Review of Pharmacoeconomics & Outcomes Research*, *23*(9), 1049–1056. https://doi.org/10.1080/14737167.2023.2234639

Pigott, T. D., & Polanin, J. R. (2020). Methodological guidance paper: High-quality meta-analysis in a systematic review. *Review of Educational Research*, *90*(1), 24–46. https://doi.org/10.3102/0034654319877153

Polanin, J. R., Pigott, T. D., Espelage, D. L., & Grotpeter, J. K. (2019). Best practice guidelines for abstract screening large-evidence systematic reviews and meta-analyses. *Research Synthesis Methods*, *10*(3), 330–342. https://doi.org/10.1002/jrsm.1354

Priem, J., Piwowar, H., & Orr, R. (2022). *OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts* (Version 2). arXiv. https://doi.org/10.48550/ARXIV.2205.01833

R Core Team. (2023). *R: A language and environment for statistical computing* [Software]. R Foundation for Statistical Computing. https://www.R-project.org

Reimers, N., & Gurevych, I. (2019). *Sentence-BERT: Sentence embeddings using siamese BERT-networks*. arXiv. https://doi.org/10.48550/ARXIV.1908.10084

Ros, R., Bjarnason, E., & Runeson, P. (2017). A machine learning approach for semi-automated search and selection in literature studies. In *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering* (pp. 118–127). Association for Computing Machinery. https://doi.org/10.1145/3084226.3084243

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, *24*(5), 513–523. https://doi.org/10.1016/0306-4573(88)90021-0

Scherhag, J., & Burgard, T. (2023). *Performance of semi-automated screening using Rayyan and ASReview: A retrospective analysis of potential work reduction and different stopping rules*. PsychArchives. https://doi.org/10.23668/PSYCHARCHIVES.12843

Scott, A. M., Forbes, C., Clark, J., Carter, M., Glasziou, P., & Munn, Z. (2021). Systematic review automation tools improve efficiency but lack of knowledge impedes their adoption: A survey. *Journal of Clinical Epidemiology*, *138*, 80–94. https://doi.org/10.1016/j.jclinepi.2021.06.030

Smith, V., Devane, D., Begley, C. M., & Clarke, M. (2011). Methodology in conducting a systematic review of systematic reviews of healthcare interventions. *BMC Medical Research Methodology*, *11*(1), Article 15. https://doi.org/10.1186/1471-2288-11-15

Tang, X., Renninger, K. A., Hidi, S. E., Murayama, K., Lavonen, J., & Salmela-Aro, K. (2022). The differences and similarities between curiosity and interest: Meta-analysis and network analyses. *Learning and Instruction*, *80*, Article 101628. https://doi.org/10.1016/j.learninstruc.2022.101628

Teijema, J. J., Hofstee, L., Brouwer, M., De Bruin, J., Ferdinands, G., De Boer, J., Vizan, P., Van Den Brand, S., Bockting, C., Van De Schoot, R., & Bagheri, A. (2023). Active learning-based systematic reviewing using switching classification models: The case of the onset, maintenance, and relapse of depressive disorders. *Frontiers in Research Metrics and Analytics*, *8*, Article 1178181. https://doi.org/10.3389/frma.2023.1178181

van de Schoot, R., de Bruin, J., Schram, R., Zahedi, P., de Boer, J., Weijdema, F., Kramer, B., Huijts, M., Hoogerwerf, M., Ferdinands, G., Harkema, A., Willemsen, J., Ma, Y., Fang, Q., Hindriks, S., Tummers, L., & Oberski, D. L. (2021). An open source machine learning framework for efficient and transparent systematic reviews. *Nature Machine Intelligence*, *3*(2), 125–133. https://doi.org/10.1038/s42256-020-00287-7

Vembye, M. H., Christensen, J., Bondebjerg, A., & Schytt, F. L. W. (2024). *GPT API models can function as highly reliable second screeners of titles and abstracts in systematic reviews*. OSF. https://doi.org/10.31219/osf.io/yrhzm

Wallace, B. C., Trikalinos, T. A., Lau, J., Brodley, C., & Schmid, C. H. (2010). Semi-automated screening of biomedical citations for systematic reviews. *BMC Bioinformatics*, *11*(1), Article 55. https://doi.org/10.1186/1471-2105-11-55

Wang, Z., Nayfeh, T., Tetzlaff, J., O'Blenis, P., & Murad, M. H. (2020). Error rates of human reviewers during abstract screening in systematic reviews. *PLOS ONE*, *15*(1), Article e0227742. https://doi.org/10.1371/journal.pone.0227742

Wickham, H., François, R., Henry, L., & Müller, K. (2020). *Dplyr: A grammar of data manipulation* (R package Version 1.0.0). https://CRAN.R-project.org/package=dplyr

Zhang, Q., & Neitzel, A. (2024). Choosing the right tool for the job: Screening tools for systematic reviews in education. *Journal of Research on Educational Effectiveness*, *17*(3), 513–539. https://doi.org/10.1080/19345747.2023.2209079