Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

RESEARCH ARTICLE

Enhanced tree-based subgroup identification in meta-analysis

Xinru Li*[1] | Elise Dusseldorp*[2] | Juan Claramunt González[2] | Xiaogang Su[3] | Joyce van Megen[2] | Jacqueline J. Meulman[4,5]

[1]Mathematical Institute, Leiden University, 2300 RB Leiden, The Netherlands

[2]Methodology and Statistics unit, Institute of Psychology, Leiden University, 2300 RB Leiden, The Netherlands

[3]Department of Mathematical Sciences, University of Texas at El Paso, Austin, TX 78712, USA

[4]Luxs Data Science, 2333 BD Leiden, The Netherlands

[5]Department of Statistics, Stanford University, Stanford, CA 94305, USA

Correspondence
Corresponding author Elise Dusseldorp, Institute of Psychology, Leiden University, P.O. Box 9555, 2300 RB, Leiden, The Netherlands.
Email: elise.dusseldorp@fsw.leidenuniv.nl

Abstract

Meta-CART integrates regression trees into the framework of meta-analysis to perform exploratory moderator analysis. The meta-regression tree identifies, based on the study characteristics, homogeneous subgroups of studies with respect to their effect size. We propose three innovations for the random effects regression trees (RE0) in meta-CART to improve the stability of the tree construction process for subgroup identification, and to correct for potential bias in the statistical inference with respect to the identified subgroups. The first innovation is a smooth sigmoid surrogate strategy (instead of greedy search) combined with a look-ahead (RE+) to improve the splitting procedure. The second is a permutation test for the value of the Q–between (instead of the Chi-squared test). The third is a bootstrap procedure to correct for bias in the effect size estimates of the subgroups. Our simulation study shows that the first two innovations lead to satisfactory false discovery rates (< .05) even when less pruning is used. The recovery rate of the true tree-based models by RE+ is equal to or somewhat better than RE0. In addition, the results suggest that the bootstrap procedure improves the estimate of the residual heterogeneity. Therefore, the bias in the estimates of effect sizes in the identified subgroups decreases and the corresponding confidence intervals are wider.

KEYWORDS
meta-regression, moderator, interactions, meta-cart

1 | INTRODUCTION

With the rapid growth of the number of clinical trials in recent decades, the need for systematic reviews and meta-analyses to summarize the study findings has increased[1]. As an intervention may be applied to different populations with variation in implementation, heterogeneity may exist when evaluating the effectiveness of the intervention. For example, Michie et al.[2] investigated interventions designed to change health-related behaviors. These interventions generally include various behavior change techniques that can influence the effectiveness of the intervention. In such cases, it is of high interest to identify the behavior change techniques that explain the heterogeneity (i.e., moderators) and understand their influence on the effect sizes derived from the study results (e.g., the standardized mean difference between treatment and control group). However, in recent decades, statistical approaches in meta-analysis have focused mainly on confirmatory analyses, and method development aimed at explaining heterogeneity has been limited[3]. Conventionally, meta-regression is the most used method to examine the relationship between the study outcome (i.e., the effect size of each study) and the moderators (i.e., the predictor variables). Meta-regression uses study weights to take into account sampling and residual error. Two types of models can be used: a fixed-effect (FE) or a random-effects (RE) model. The main difference is the assumption of residual heterogeneity: the FE model assumes that there is zero residual heterogeneity (i.e., there is one underlying true effect size among the studies); in contrast, the RE model takes residual heterogeneity into account (i.e., each study has its own true effect size).

Meta-regression has several limitations. First, it assumes a linear relationship between moderators and effect size, which is not always warranted. Second, behavioral and medical studies often include multiple factors that may interact[4], and meta-regression has difficulty simultaneously investigating multiple moderators and examining interactions among them[5]. The main difficulty is

* **Shared first authorship:** These authors contributed equally to this work.

 © 2023 Copyright Holder Name

that examining interactions by (meta-)regression often requires multiple iterations of model selection, such as deciding whether to include higher-order terms or reducing the number of interactions based on prior knowledge, particularly when working with a limited sample size. These iterative decisions can introduce variability and reduce the robustness of the final results, as different choices can lead to different models and conclusions. Therefore, as an alternative, some recent work has begun to use tree-based methods to investigate sources of heterogeneity in meta-analysis[6,7,8,9].

Tree models are popular approaches for subgroup identification purposes, as they excel at dealing with interactions and handling nonlinear relationships. Furthermore, trees can automatically identify homogeneous subgroups that are characterized by moderators. A fitted tree model mimics a hierarchically nested set of decision rules[10]. So, the results of the analysis can be easily interpreted and used for answering research questions like "which combination of the treatment components results in the highest effectiveness?" and "which intervention is the most effective for a certain population?". Most tree-based methods are developed for single study settings (e.g., see[11,12,13,14,15,16,17]). The idea of applying a tree model in the framework of meta-analysis is relatively new.

Tree models in meta-analysis serve two purposes: 1) to identify subgroups with different pooled (summary) effect sizes and 2) to test statistical significance of the heterogeneity between the subgroups defined by the moderators. These subgroups are not predefined, but are derived from the data by the tree algorithm. Recently, Li et al.[18] developed a tree-based method, meta-CART, which serves these purposes. Assuming an FE model or RE model, this method uses new partitioning algorithms to fit fixed-effect or random-effects meta-regression trees, respectively[19]. It uses a between-subgroups test (based on the Q-between statistic) to test the moderator effect[18,20]. If the subgroup memberships are determined and evaluated using the same data set, which is frequently the case in practice because the number of studies included in a meta-analysis is usually limited, this poses post hoc concerns. Thus, the result of the significance test for the Q-between cannot be interpreted at its face value and only gives information when the Q-between is not significant, see details in Li et al.[18] Section 8.1.

To overcome the previously mentioned shortcoming and to improve the performance of subgroup identification, we propose new features for the existing version of meta-CART[18]. The zero residual heterogeneity assumption of the FE model frequently does not hold, and the majority of meta-CART applications employed RE meta-regression trees, for example, Reimer and Sengupta[9]. Therefore, we focus on the RE meta-regression trees. Our innovation includes three key elements: one for subgroup identification and two for statistical inference. First, trees have a greedy nature and may lead to results with instability. Instability means that small variations in the data can result in different trees. This problem is due to the greedy algorithm that optimizes the splitting criterion locally at each separate split. This process is "short-sighted" and may end up in a local optimum. If an error occurs in the top split, its influence will be propagated down to all the splits below because of the hierarchical structure, and this constitutes an inherent instability (see[21] p. 274). In addition, for numeric moderators a greedy search exhaustively evaluates all possible split points and selects the one resulting in the largest decrease of heterogeneity. This discrete process tends to produce large fluctuations with local spikes that indicate local minima or maxima[22]. To reduce this instability, ensemble learning methods such as Bagging[23], Boosting[24], and Random Forests[25] are often used in the field of data mining by combining or averaging trees built, for example, on resampled data. However, the resulting model, which is an ensemble of multiple trees, loses the interpretability of a single tree, and the influence of the predictor variables becomes unclear.

Therefore, instead of ensembling, we propose a novel approach that combines a look-ahead strategy, which looks one step further when choosing the splits[19], with the smooth sigmoid surrogate (SSS) strategy[22], which is an approximation method for greedy search that solves the problem of local spikes. Second, for pre-defined subgroup memberships the moderator effect can be tested by the Q-between statistic, which asymptotically follows a chi-squared distribution under the null hypothesis. However, the p-value from this standard Q-test is inappropriate in meta-CART analyses, since the subgroup membership results from the tree algorithm rather than a priori hypotheses. To solve this, we propose a permutation test based on the Q-between statistic to assess whether the effect of the identified moderators are spurious. Third, due to the adaptive nature of tree modeling, the confidence intervals of the subgroup summary effect estimates are usually over-optimistic, and the intervals may be too narrow to have the desired coverage. Therefore, we propose a bootstrap-based approach to correct the overoptimism.

The outline of this paper is as follows. First, we describe the subgroup identification procedure, which applies a combination of look-ahead and SSS in RE meta-regression trees. Next, we propose a permutation test and a bootstrap correction for the statistical inference of the identified subgroups. Third, we evaluate the proposed new features via an extensive simulation study. Then, we illustrate the application of the method by analyzing a real-world data set. Finally, we summarize and discuss the results.

2 | SUBGROUP IDENTIFICATION BY LOOK-AHEAD AND SSS

2.1 | RE Meta-Regression Trees Splitting Criterion

In a meta-CART analysis, the data consist of observations from K studies $\mathcal{D} = \{(g_k, \sigma^2_{\epsilon_k}, \mathbf{x}^T_k : k = 1, \ldots, K)\}$, where g_k and $\sigma^2_{\epsilon_k}$ are the estimated effect size (which is used as outcome) and the sampling variance in the k^{th} study, and $\mathbf{x}^T_k = [x_{k1}, \ldots, x_{kM}] \in \mathbb{R}^M$ is a row vector of length M with the scores of the k^{th} study on the M predictor variables (i.e., the potential moderators). RE meta-regression tree applies a sequential splitting algorithm in which heterogeneity between the study effect sizes in offspring nodes is maximized or, equivalently, the within-node heterogeneity is minimized. After each split is introduced, the global estimate of residual heterogeneity σ^2_τ, which is the between-study heterogeneity unexplained by the moderators, is refreshed. The "global" estimate means that σ^2_τ is estimated across all terminal nodes rather than only within the resulting left and right child nodes.

Note that the nature of the CART algorithm is recursive, which means that the optimization of a split in one node of the current tree does not depend on the other nodes of the tree; in contrast, the nature of the meta-CART algorithm is sequential, because the estimate of σ^2_τ is re-estimated after each split using all nodes t of the current tree. We explain the estimates in detail below.

The effect size within a node or subgroup is estimated as a weighted mean. Denote the final tree by $\mathcal{T}$, and let $\widetilde{\mathcal{T}}$ be the set of all the terminal nodes in the final tree $\mathcal{T}$. The within-node summary effect size (g^*_+), that is, the weighted mean effect size across the studies in a node, is computed as

$$g^*_{t+} = \frac{\sum_{k\in t} w^*_k g_k}{\sum_{k\in t} w^*_k}, \tag{1}$$

for each terminal node $t \in \widetilde{\mathcal{T}}$, and weights $w^*_k = 1/(\sigma^2_{\epsilon_k} + \sigma^2_\tau)$, where * refers to the RE model[19].

We compute the residual heterogeneity σ^2_τ by using the DerSimonian and Laird method[26]:

$$\sigma^2_\tau = \frac{\sum_{t\in\widetilde{\mathcal{T}}} \sum_{k\in t} w_k (g_k - g_{t+})^2 - \sum_{t\in\widetilde{\mathcal{T}}} (K_t - 1)}{\sum_{t\in\widetilde{\mathcal{T}}} C_t}, \tag{2}$$

where $\mathcal{T}$ is the entire tree after the split, K_t is the number of studies in node t, and the components C_t are computed as

$$C_t = \sum_{k\in t} w_k - \frac{\sum_{k\in t} w^2_k}{\sum_{k\in t} w_k}, \tag{3}$$

where $w_k = 1/\sigma^2_{\epsilon_k}$.

Given the estimated σ^2_τ, the between-subgroups Q-statistic is computed as

$$Q^*_B = \sum_{t\in\mathcal{T}} \sum_{k\in t} w^*_k (g^*_{t+} - g^*_{++})^2, \tag{4}$$

where g^*_{++} is the grand weighted mean and the weights w^*_k are the same as in equation 1.

Using the maximization of the statistic Q^*_B as the splitting criterion, a meta-regression tree selects the split that maximizes the heterogeneity between subgroups and finds homogeneous subgroups as a result.

2.2 | Look ahead

The tree construction process described above has a greedy nature: it chooses each split without regard for future splits. Thus, for the full tree, we might obtain a local minimum solution. To alleviate this problem, a look-ahead strategy can be implemented that looks one step further when choosing the splits, as proposed by Li et al.[19].

Starting the algorithm at the root node, a look-ahead strategy searches for the optimal combination of a split of the root node and a split of one of its child nodes. The splitting criterion is evaluated at all possible combinations of two split points (i.e., single split points on any two moderators or any two split points on one moderator). The first split is chosen based on the combination that maximizes the splitting criterion. Then the look-ahead strategy can be repeated for the resulting child nodes. Because the computational burden grows exponentially as the number of splits to be examined increases[27], we recommend applying this look-ahead procedure only for a few initial splits (i.e., no more than two or three splits). After the look-ahead procedure, the resulting offspring nodes can be further split following a fully greedy procedure to maximize the splitting criterion at each split.

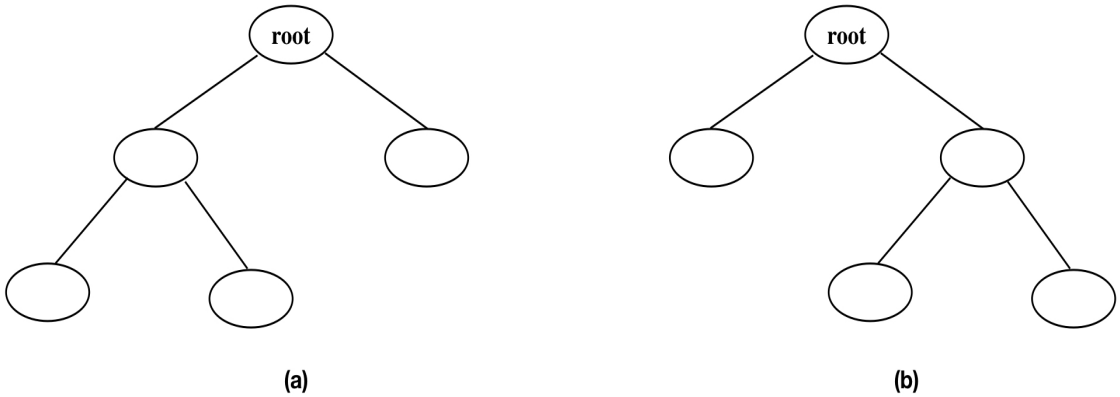

FIGURE 1 Two possible scenarios, (a) and (b), to grow a tree with two splits from the root node

2.3 | Look-ahead with SSS

When applying the look-ahead strategy in a tree, the computational burden can be high when moderators have many distinct values, which result in a large number of possible combinations of two splits. Thus, including numeric moderators into a meta-regression tree model may heavily increase the computational cost. Also, the splitting process that searches through all possible cutoff values in a discrete way tends to end up with a local optimum and introduces variable selection bias (i.e., variables with more values tend to be preferred in the splitting process to variables with fewer values). As a solution, we propose to combine the look-ahead with the smooth sigmoid surrogate (sss) strategy to identify the best split point for each moderator. This first innovation could help not only to find optimal split points, but also to avoid spurious splits due to variable selection bias.

Since a split is made on a selected moderator $\mathbf{x}_m$ at the corresponding split point c, we can introduce a binary variable $\delta_k = \delta(x_{km}; c) = I(x_{km} > c)$, and write the summary effect size in the two resulting left and right child nodes as

$$\begin{cases} g^*_{t_L+} = \frac{\sum_{k=1}^{K_t}(1-\delta_k)w_k^* g_k}{\sum_{k=1}^{K_t}(1-\delta_k)w_k^*}, \\ g^*_{t_R+} = \frac{\sum_{k=1}^{K_t}\delta_k w_k^* g_k}{\sum_{k=1}^{K_t}\delta_k w_k^*}. \end{cases} \tag{5}$$

We can use the logistic function to approximate the threshold indicator function

$$s(x_{km}; a, c) = \frac{\exp\{a(x_{km} - c)\}}{1 + \exp\{a(x_{km} - c)\}}, \tag{6}$$

with scale parameter $a > 0$. Then the summary effect sizes $g^*_{t_L+}$ and $g^*_{t_R+}$ can be approximated by

$$\begin{cases} \tilde{g}^*_{t_L+} = \frac{\sum_{k=1}^{K_t}(1-s_k)w_k^* g_k}{\sum_{k=1}^{K_t}(1-s_k)w_k^*}, \\ \tilde{g}^*_{t_R+} = \frac{\sum_{k=1}^{K_t}s_k w_k^* g_k}{\sum_{k=1}^{K_t}s_k w_k^*}. \end{cases} \tag{7}$$

Look-ahead strategy evaluates all possible combinations of two splits together, so there are two possible scenarios for each combination of two moderators: (a) the second split is from the left child node (Figure 1a), or (b) the second split is from the right child node (Figure 1b). Denote the three grandchild nodes by $t_{G_1}, t_{G_2}, t_{G_3}$. Let $\mathbf{x}_1$ be the first splitting moderator and c_1 its split point, and $\mathbf{x}_2$ and c_2 be the second splitting moderator and its split point. For scenario (a), the within-node (subgroup)

summary effect sizes are

$$\begin{cases} g^*_{t_{G_1}+} = \dfrac{\sum_{k=1}^K I(x_{k1} \le c_1; x_{k2} \le c_2) \cdot w_k^* g_k}{\sum_{k=1}^K I(x_{k1} \le c_1; x_{k2} \le c_2) \cdot w_k^*}, \\ g^*_{t_{G_2}+} = \dfrac{\sum_{k=1}^K I(x_{k1} \le c_1; x_{k2} > c_2) \cdot w_k^* g_k}{\sum_{k=1}^K I(x_{k1} \le c_1; x_{k2} > c_2) \cdot w_k^*}, \\ g^*_{t_{G_3}+} = \dfrac{\sum_{k=1}^K I(x_{k1} > c_1) \cdot w_k^* g_k}{\sum_{k=1}^K I(x_{k1} > c_1) \cdot w_k^*}. \end{cases} \tag{8}$$

These effect sizes can be approximated by

$$\begin{cases} \tilde{g}^*_{t_{G_1}+} = \dfrac{\sum_{k=1}^K (1 - s(x_{k1}; a, c_1))\,(1 - s(x_{k2}; a, c_2)) \cdot w_k^* g_k}{\sum_{k=1}^K (1 - s(x_{k1}; a, c_1))\,(1 - s(x_{k2}; a, c_2)) \cdot w_k^*}, \\ \tilde{g}^*_{t_{G_2}+} = \dfrac{\sum_{k=1}^K (1 - s(x_{k1}; a, c_1))\, s(x_{k2}; a, c_2) \cdot w_k^* g_k}{\sum_{k=1}^K (1 - s(x_{k1}; a, c_1))\, s(x_{k2}; a, c_2) \cdot w_k^*}, \\ \tilde{g}^*_{t_{G_3}+} = \dfrac{\sum_{k=1}^K s(x_{k1}; a, c_1) \cdot w_k^* g_k}{\sum_{k=1}^K s(x_{k1}; a, c_1) \cdot w_k^*}. \end{cases} \tag{9}$$

It follows that the splitting criterion for two splits can be approximated by

$$\begin{aligned} \tilde{Q}_B^* = {} & (\tilde{g}^*_{t_{G_1}+} - g^*_{++})^2 \sum_{k=1}^K \Big(1 - s(x_{k1}; a, c_1)\Big)\Big(1 - s(x_{k2}; a, c_2)\Big) \cdot w_k^* \\ & + (\tilde{g}^*_{t_{G_2}+} - g^*_{++})^2 \sum_{k=1}^K \Big(1 - s(x_{k1}; a, c_1)\Big) s(x_{k2}; a, c_2) \cdot w_k^* \\ & + (\tilde{g}^*_{t_{G_3}+} - g^*_{++})^2 \sum_{k=1}^K s(x_{k1}; a, c_1) \cdot w_k^*. \end{aligned} \tag{10}$$

Similarly, the splitting criterion of scenario (b) can be approximated by

$$\begin{aligned} \tilde{Q}_B^* = {} & (\tilde{g}^*_{t_{G_1}+} - g^*_{++})^2 \sum_{k=1}^K \Big(1 - s(x_{k1}; a, c_1)\Big) \cdot w_k^* \\ & + (\tilde{g}^*_{t_{G_2}+} - g^*_{++})^2 \sum_{k=1}^K s(x_{k1}; a, c_1)\Big(1 - s(x_{k2}; a, c_2)\Big) \cdot w_k^* \\ & + (\tilde{g}^*_{t_{G_3}+} - g^*_{++})^2 \sum_{k=1}^K s(x_{k1}; a, c_1) s(x_{k2}; a, c_2) \cdot w_k^*. \end{aligned} \tag{11}$$

The value of a can be fixed a priori. Su et al.[22] recommend to fix a at a value in the range $[10, 50]$ Note that fixing a entails standardization of each covariate. Increasing the value of a makes the sigmoid approach a step function. For each scenario, the split points $\hat{c}_1$ and $\hat{c}_2$ are obtained by maximizing $\tilde{Q}_B^*$. We solve this two-dimensional optimization problem using the limited memory Broyden-Fletcher-Goldfarb-Shanno box constraints (L-BFGS-B) method[28]. The L-BFGS-B method belongs to the quasi-Newton methods and can solve large nonlinear optimization problems with simple box constraints. The method is implemented in the `R` software environment by the function `optim()`, and options for multiple starts and constraining the search range are available. Note that moderator $\mathbf{x}_m$ needs to be standardized to use this function. After solving the optimization problem, the identified $\hat{c}$ can be transformed back to the original scale for interpretability.

To summarize, a look-ahead strategy with SSS consists of three sub-procedures. The first one finds all possible combinations of two moderators [‡]. Within each combination, the second sub-procedure finds the best split points that correspond to the two possible scenarios, and selects the scenario with higher values for the Q_B^*-statistic. Finally, the third sub-procedure compares the selected scenario and split points across all combinations, and selects the combination that results in the highest value for the Q_B^*-statistic.

[‡] Note that two splits on the same moderator are allowed.

2.4 | Pruning to avoid overfitting

The splitting process of a meta-regression tree continues until all terminal nodes contain only a few studies (e.g., three studies). The result is a large initial tree $\mathcal{T}_{init}$ with $|\mathcal{T}_{init}|$ terminal nodes, where $|\mathcal{T}_{init}|$ denotes the number of elements in $\mathcal{T}_{init}$. To avoid overfitting, $\mathcal{T}_{init}$ is pruned back to a smaller size using cross-validation. This procedure consists of two steps. First, tenfold cross-validation is performed to estimate the prediction errors for each subtree of size $1, \ldots, |\mathcal{T}_{init}|$. Then, the final tree $\mathcal{T}$ is selected as the smallest tree with a cross-validation error within a $c_{pruning} \times$ the standard error distance from the minimum cross-validation error. When $c_{pruning} = 0$, the subtree with the minimum cross-validation error is selected. As the value of $c_{pruning}$ increases, the final tree tends to be smaller, resulting in a more conservative selection of moderators and fewer terminal nodes. Based on the results of their extensive simulation study, Li et al.[18] recommended $c_{pruning} = 0.5$ for the previous version of the algorithm. In the simulation study with the new meta-regression tree algorithm, we will reexamine the choice of the amount of pruning (see Section 4).

3 | STATISTICAL INFERENCE FOR IDENTIFIED SUBGROUPS

3.1 | Permutation Test

Our second innovation is the use of a permutation test for the statistical significance of identified moderators. After the tree model has been pruned via cross-validation, a subgroup meta-analysis is performed to test the significance of the between-subgroups heterogeneity (i.e., whether the identified subgroups account for a significant proportion of the variance in study effect sizes). In a standard meta-analysis, this heterogeneity test is based on the between-subgroups Q-statistic, which asymptotically follows a chi-squared distribution under the null hypothesis. However, this standard Q_B^*-test is inappropriate in meta-CART analysis, since the subgroup membership results from the tree model that maximizes the same Q_B^*-statistic.

As a solution, we propose to use a permutation test to correct for the overoptimism in the Q_B^*-test. A permutation test was proposed by Higgins et al.[29] to appropriately temper the statistical significance of meta-regression findings. Within the survival tree setting, LeBlanc et al.[30] used a permutation test to assess the performance of each split in the final survival tree. For meta-regression trees, we adapted this approach to provide a stricter way to determine whether moderator effects, represented by the tree model, are spurious. This permutation test is shown in Algorithm 1. We permute the original meta-analytic data in

Algorithm 1 Permutation significance test for between-subgroups heterogeneity

input: original meta-analytic data of K studies in matrix form: $\mathcal{D} = \mathbf{D}_0 = \{\mathbf{g}, \sigma_\epsilon^2, \mathbf{X}\}$
initialize total number of permutations J
fit a pruned meta-regression tree $\mathcal{T}_0$ for $\mathbf{D}_0$
let $|\mathcal{T}_0|$ denote the total number of terminal nodes of $\mathcal{T}_0$
compute Q_B^* between the $|\mathcal{T}_0|$ terminal nodes
set $\gamma \leftarrow 1$
for $j \leftarrow 1$ to J **do**
  keep columns $\mathbf{g}$ and σ_ϵ^2 fixed in $\mathbf{D}_0$
  permute row-wise $\mathbf{X}$ to obtain $\mathbf{D}_j = \{\mathbf{g}, \sigma_\epsilon^2, \mathbf{X}'\}$, where $\mathbf{X}'$ denotes the permuted matrix
  fit a meta-regression tree $\mathcal{T}_j$ with fixed tree size $|\mathcal{T}_0|$ for $\mathbf{D}_j$
  compute $Q_{B,j}^*$ between the terminal nodes of $\mathcal{T}_j$
  if $Q_{B,j}^* \geq Q_B^*$ **then**
    $\gamma \leftarrow \gamma + 1$
  else
    $\gamma \leftarrow \gamma$
  end if
end for
compute permuted p-value for the Q_B^* as $p = \gamma/(J+1)$

such a way that the correlation between the study effect sizes and the moderators is disrupted, while the correlations among the moderators themselves remain unchanged.

3.2 | Bias Correction Bootstrap

Our third innovation is bias correction with a bootstrap procedure. As addressed in the previous sections, the subgroups obtained by a meta-CART analysis are not predefined but found by the tree algorithm. As a result, the within-subgroup heterogeneity is underestimated, and the estimated standard errors of the subgroup summary effect sizes are overly optimistic. To quantify the uncertainty in the estimates for the subgroup summary effect sizes in an appropriate way, the bias in the within-node heterogeneity needs to be estimated. For each terminal node t of tree $\mathcal{T}$, the within-node heterogeneity μ_t is computed as the average weighted sum of squares

$$\mu_t = \sum_{k\in t} w_k^*(g_k - g_{t+}^*)^2 / \sum_{k\in t} w_k^*. \tag{12}$$

One common method for bias correction is the bootstrap[31]. In this approach, the estimates from the bootstrap samples are averaged and compared to the original estimate to give an estimator of the bias. However, one major obstacle to applying bootstrap in tree-based methods is the instability of trees. Trees are unstable in the sense that a small perturbation of the data often results in a substantially different tree model. As a result, the tree models obtained by using bootstrap samples are different from each other and from the final tree model constructed from the original sample. To address this problem, we developed a bootstrapping strategy that was inspired by the bias correction method of Loh et al.[16]. Their method was proposed for individual-level data to construct confidence intervals for treatment effects by distributing the observations from the original data set to the trees obtained from bootstrap samples. We made extensive modifications to this method so that it could be used within meta-CART (see Algorithm 2).

Note that a tree model forms a natural grouping of the entire data defined by the terminal nodes of a tree. With two tree structures, observations in a terminal node of one tree can be distributed into different terminal nodes of the other tree. Specifically, let $\mathbf{m}_0 \in \mathbb{R}^K$ denote the node membership vector that assigns a terminal node of $\mathcal{T}$ to each observation in $\mathcal{D}$. This node membership vector $\mathbf{m}_0$ is categorical with levels $\{1, 2, \ldots, |\widetilde{\mathcal{T}}|\}$. For a bootstrap sample $\mathcal{D}_b$, a best-sized tree $\mathcal{T}_b$ can be constructed and the average weighted sum of squares for $\mathcal{D}_b$, denoted by $\{\mu_{bt'} : t' \in \widetilde{\mathcal{T}}_b\}$ can be obtained, where t' are the terminal nodes of tree $\mathcal{T}_b$. By sending $\mathcal{D}$ down to $\mathcal{T}_b$, the average weighted sum of squares for $\mathcal{D}$ given $\mathcal{T}_b$ $\{\mu_{0t'} : t' \in \widetilde{\mathcal{T}}_b\}$ can be computed. In each terminal node $\{t' \in \widetilde{\mathcal{T}}_b\}$ the bias can then be computed as $b_{bt'} = \mu_{0t'} - \mu_{bt'}$.

In order to obtain the bias estimate for μ_{0t} in the original tree $\mathcal{T}$, we can look at how observations in $t \in \widetilde{\mathcal{T}}$ are distributed over $\mathcal{T}_b$. Let $\mathbf{m}_b \in \mathbb{R}^K$ denote the node membership vector that assigns a terminal node of $\mathcal{T}_b$ to each observation in $\mathcal{D}$. The two categorical vectors $\mathbf{m}_0$ and $\mathbf{m}_b$ form a $|\widetilde{\mathcal{T}}| \times |\widetilde{\mathcal{T}}_b|$ two-way contingency table of weights $\{m_{tt'}^* = \sum_{k\in t \text{ and } k\in t'} w_k^* : t \in \widetilde{\mathcal{T}} \text{ and } t' \in \widetilde{\mathcal{T}}_b\}$. Let $p_{tt'}^* = m_{tt'}^*/m_{t\cdot}^*$ be the weighted marginal proportions of the rows, where $m_{t\cdot}^* = \sum_{t'} m_{tt'}^*$ is the total of the t-th row. Then the bias estimate for the average weighted sum of squares in the original tree $\mathcal{T}$ is given by $\sum_{t'\in\widetilde{\mathcal{T}}_b} p_{tt'}^* b_{bt'}$.

The whole procedure is defined in Algorithm 2 with more details. Averaging over B bootstrap samples leads to the bias estimate for μ_t. Put together, the bias-corrected within-node heterogeneity Q_t^* is given by

$$Q_t^* = \left(\mu_t - \frac{1}{B}\sum_{i=1}^{B}\sum_{t'\in\widetilde{\mathcal{T}}_b} p_{tt'}^*(\mu_{0t'} - \mu_{bt'})\right) \cdot \sum_{k\in t} w_k^*. \tag{13}$$

With the bias-corrected Q_t^*, the residual heterogeneity σ_τ^2 is re-estimated using (2). To correct the confidence intervals of subgroup summary effect sizes, the standard errors need be re-computed as

$$SE = \frac{1}{\sqrt{\sum_k^K 1/(\sigma_{\epsilon_k}^2 + \sigma_\tau^2)}}. \tag{14}$$

4 | SIMULATION STUDIES

In the following simulation studies, our aim is to evaluate the performance of the proposed innovations for RE meta-regression trees. We compare the ability to retrieve the true model underlying the generated data between the standard RE meta-regression

Algorithm 2 Bias correction for within-node heterogeneity in tree modeling.

Input: data $\mathcal{D} = \{g_k, \sigma_{\epsilon_k}, \mathbf{x}_k^T, k = 1, 2, ..., K\}$.
initialize B = number of bootstrap samples
construct a best-sized tree $\mathcal{T}$ from $\mathcal{D}$ via pruning using cross validation
obtain node membership vector $\mathbf{m}_0 \in \mathbb{R}^K$ for all studies in $\mathcal{D}$ w.r.t. $\mathcal{T}$
compute μ_t for each $t \in \widetilde{\mathcal{T}}$ based on $\mathcal{D}$
set bias $b_t = 0$ for $t \in \widetilde{\mathcal{T}}$
for $b \leftarrow 1$ to B **do**
  draw a bootstrap sample $\mathcal{D}_b$
  construct a best-sized tree $\mathcal{T}_b$ from $\mathcal{D}_b$ via pruning using cross validation
  compute $\mu_{bt'} = \sum_{k \in t'} w_k^* (g_k - g_{t'+}^*)^2 / \sum_{k \in t'} w_k^* : t' \in \widetilde{\mathcal{T}}_b$
  send $\mathcal{D}$ down to $\mathcal{T}_b$ and recompute $\{\mu_{0t'} : t' \in \widetilde{\mathcal{T}}_b\}$ on basis of $\mathcal{D}$
  compute bias $b_{bt'} = \mu_{0t'} - \mu_{bt'}$ for $t' \in \widetilde{\mathcal{T}}_b$
  see how studies in $t \in \widetilde{\mathcal{T}}$ are distributed over $\widetilde{\mathcal{T}}_b$.
  obtain node membership vector $\mathbf{m}_b \in \mathbb{R}^n$ for all studies in $\mathcal{D}$ w.r.t. $\mathcal{T}_b$
  form contingency table $\{m_{tt'}^* = \sum_{k \in t \text{ and } k \in t'} w_k^* : t \in \widetilde{\mathcal{T}}$ and $t' \in \widetilde{\mathcal{T}}_b\}$ with $\mathbf{m}_0$ and $\mathbf{m}_b$
  compute weighted row proportions $p_{tt'}^* = m_{tt'}^* / m_{t\cdot}^*$
  for $t \in \widetilde{\mathcal{T}}$ **do**
    update $b_t = b_t + \sum_{t' \in \widetilde{\mathcal{T}}_b} p_{tt'}^* b_{bt'}$
  end for
end for
compute average bias $b_t = b_t / B$ for each $t \in \widetilde{\mathcal{T}}$
compute bias corrected within-node $Q_t^* = (\mu_t + b_t) \cdot \sum_{k \in t} w_k^*$ for each node $t \in \widetilde{\mathcal{T}}$.

tree without look-ahead and SSS (RE0) and RE meta-regression tree applying both look-ahead and SSS (RE+). In addition, we reexamine the pruning rule and we apply the permutation test for both RE0 and RE+. Furthermore, in a separate simulation study, we investigate the bootstrap correction procedure using the RE+ algorithm, and evaluate whether the overoptimism in the estimate of the residual heterogeneity σ_τ^2 is properly reduced.

The simulation studies are performed in the R language[32]. The meta-CART analyses are performed using R-package `metacart`, which is available via Comprehensive R Archive Network (CRAN) at https://cran.r-project.org/package=metacart. We used version 3.0.0.

4.1 | Design

We focus on three design factors: the number of studies $K = \{40, 120\}$, the residual heterogeneity $\sigma_\tau^2 = \{0, 0.025, 0.05\}$, and the five models (A-E) to generate the data differing in complexity as shown in Table 1. For each cell of the design, 1000 data sets were generated as training sets with fixed values for average within-study sample size ($\bar{n} = 80$), number of candidate moderators ($M = 10$), and correlation between the moderators ($\rho = 0.3$).

TABLE 1 The five models used to generate data sets. The values of the coefficients were fixed.

Model	Form
A	$g(x) = 0.5$
B	$g(x) = 0.5 \cdot I(x_1 > 0.5) I(x_2 > 0.5)$
C	$g(x) = 0.5 \cdot I(x_1 > 0.5) I(x_2 \leq 0.5) + 0.5 \cdot I(x_1 \leq 0.5) I(x_2 > 0.5)$
D	$g(x) = -0.5 + 0.5x_1 + 0.5x_2$
E	$g(x) = -1.5 + \sin(\pi x_1 x_2) + 2(x_3 - 0.5)^2 + x_4 + 0.5x_5$

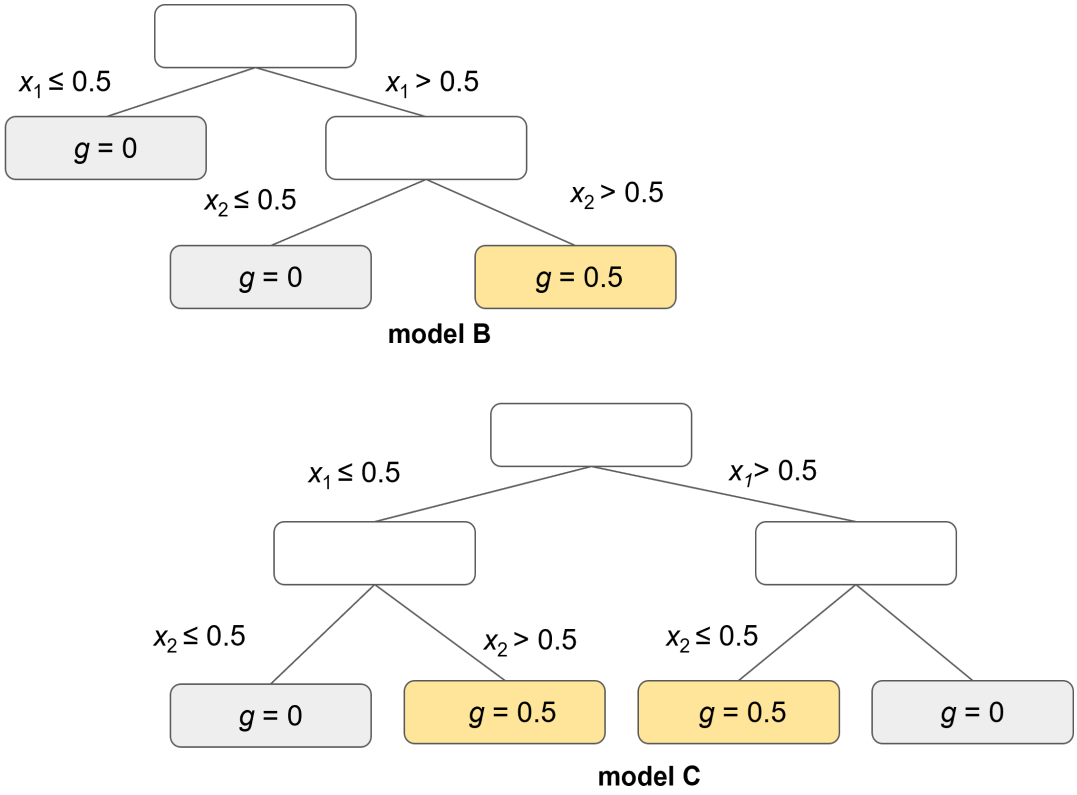

FIGURE 2 The tree representations of true models B and C used in the simulation study. The population summary effect sizes (g) are shown in the terminal nodes.

The five models, as shown in Table 1, were used to generate true effect sizes as a function of x ($g(x)$) given a selection of moderators defined by the model. The models vary in the number of moderators involved and the type of relationship between the moderators and the effect size. Note that we generated ten candidate moderators in each data set. Therefore, the moderators that were not involved in the true model were just noise variables. All moderators were continuous variables, uniformly distributed between 0 and 1. The details about moderator and effect size generation can be found in Appendix 7.

Model A is a null model to assess the false positive rate (meta-CART falsely identifies significant moderator effects when the true model does not have a moderator effect). Models B and C (see Figure 2) are used to evaluate the ability of meta-CART to identify the interaction effect(s) when the underlying structure can be fully expressed by a binary tree. Model B is a two-way interaction effect between two moderators. Model C includes two balanced two-way interactions, which is similar to the simulated example used by Tibshirani et al.[33]. This model is shown to be difficult for greedy search procedures such as CART because there is no information on where to split at the top level. Thus, this model represents a case where the tree is likely to suffer a local optimum problem. Models D and E are used to evaluate the ability of a meta-regression tree to identify influential moderators when the underlying structure cannot be fully captured by a binary tree. Model D is a linear model with the linear main effects of two moderators. Model E is a nonlinear model derived from Friedman et al.[34]. The coefficients in model E are rescaled so that the resulting effect size ranges between –1.2 and 1.2, which is commonly encountered in practice.

In total, $2 \times 3 \times 5 \times 1000 = 30,000$ training sets were generated. One test set with $K = 2000$ was generated for each design cell to estimate the overoptimism in the subgroup analysis results.

4.2 | Meta-CART analysis and Evaluation Criteria

Both the standard RE meta-regression tree algorithm (RE0) and the enhanced RE meta-regression tree algorithm (RE+) are applied to the training sets that were generated. Due to computational costs, the look-ahead strategy of RE+ is applied only to

the root node, that is, for the first two splits. For each meta-regression tree algorithm, two types of rules are applied to prune the initial tree back: the half-standard-error rule ($c_{pruning} = 0.5$) and the minimum cross-validated error rule ($c_{pruning} = 0$).

Four criteria are used to evaluate the performance of the meta-CART algorithms. The first three criteria evaluate how well the algorithms are able to recover the true underlying relationship between pooled effect sizes and moderators and the fourth criterion is to evaluate their predictive performance.

Criterion 1: false discovery rate. Meta-CART falsely detects the presence of (a) moderator effect(s) in the data sets generated from model A (false positive rate). A false positive is found when meta-CART results in a pruned non-trivial tree (i.e., a pruned regression tree that is larger than the root node) and the heterogeneity test (Q_B^*) is significant ($p < 0.05$), according to the asymptotic p-values and the p-values from permutation.

Criterion 2: full recovery rate. Meta-CART successfully selects all moderators used in the true model (full recovery rate). This criterion examines whether the true model is fully recovered with all the true moderators and if no spurious moderators are selected. It is calculated as the percentage of the true moderators correctly identified with significant Q_B^* ($p < 0.05$), according to the asymptotic p-values and the p-values from permutation.

Criterion 3: partial recovery rate. Meta-CART selects part of the moderators used in the true model (partial recovery rate), and does not select any spurious moderators. Criterion 3 is calculated as the percentage of the true moderators partially identified with significant Q_B^* ($p < 0.05$), according to the asymptotic p-values and the p-values from permutation.

Criterion 4: prediction error. The trained meta-regression trees are applied to predict the study effect sizes for the test set, and the prediction errors, measured by the root mean squared error (RMSE), are computed.

4.3 | Results

4.3.1 | False positive rates

The false positive rates of the RE+ algorithm are consistently lower than those of RE0 (see Table 2), regardless of which Q_B^*-test is applied. When we prune the tree by the half-standard-error rule ($c_{pruning} = 0.5$), the standard Q_B^*-test results in a satisfactory false discovery rate (i.e., < .05) for most cells of the RE+ algorithm. When we apply the permutation Q_B^*-test, all cells of the RE+ algorithm show a satisfactory false discovery rate (i.e., < .05), and most cells of RE0. Ideally, the false positive rates in 2 are close to .05, because a significance level of .05 for the Q_B^*-test was used to detect a false positive result. The permutation seems to correct well for the overoptimism in the standard Q_B-test, but in combination with a more conservative pruning rule ($c_{pruning} = 0$) it corrects too strongly.

TABLE 2 A comparison of standard RE meta-regression tree (RE0), and the new RE meta-regression applying SSS with look-ahead (RE+) using different amounts of pruning ($c_{pruning}$) applied to the 1000 data sets generated from Model A differing in the number of studies (K) and the residual heterogeneity (σ_τ^2). In each cell, the false positive rates are displayed.

$c_{pruning}$	K	σ_τ^2	standard Q_B^* test		permutation Q_B^* test	
			RE0	RE+	RE0	RE+
0.5	40	0	.092	.050	.035	.020
		0.025	.077	.049	.036	.020
		0.05	.070	.051	.028	.022
	120	0	.035	.018	.020	.011
		0.025	.048	.018	.032	.012
		0.05	.066	.023	.032	.013
0	40	0	.195	.146	.049	.042
		0.025	.182	.156	.055	.035
		0.05	.159	.130	.048	.037
	120	0	.153	.120	.042	.034
		0.025	.176	.119	.047	.029
		0.05	.175	.131	.056	.040

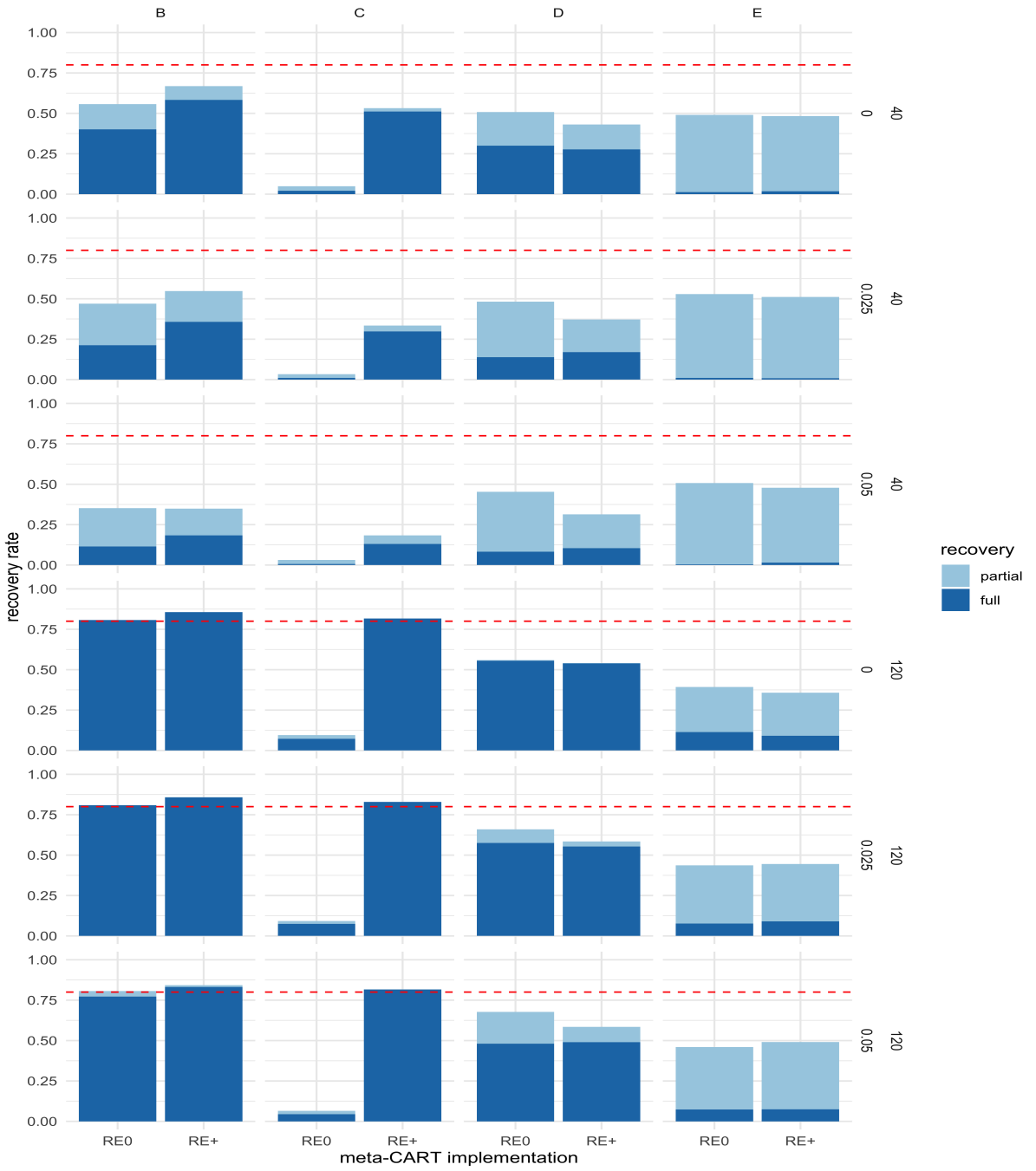

FIGURE 3 The recovery rate of the standard RE meta-CART (RE0), and the RE meta-CART applying SSS with look-ahead (RE+). The top three rows correspond to datasets simulated with $K = 40$, with $\sigma^2_\tau = 0, 0.025$, and 0.05, respectively, and the bottom three rows correspond to those with $K = 120$, with $\sigma^2_\tau = 0, 0.025$, and 0.05. The red dashed line is at a recovery rate of .80.

When we focus on the results of the RE+ algorithm, applying a more liberal pruning rule ($c_{pruning} = 0$) in combination with the permutation test results in the best false discovery rates, that is, just below .05, even in the more difficult situations with $K = 40$ and $\sigma^2_\tau = 0.05$. Therefore, for the new RE+ algorithm, we recommend using $c_{pruning} = 0$.

4.3.2 | Recovery rates and prediction error

Figure 3 presents the full recovery rates and the partial recovery rates for the two algorithms (evaluation criteria 2 and 3). Based on the recommended setting indicated by the false positive rate results, we show only the results with $c_{pruning} = 0$ in combination with the permutation Q_B^*–test, using 500 permutations. For Models B and C, RE+ shows satisfactory full recovery rates (i.e., > .80) when $K = 120$ (no matter the values of σ_τ^2), while RE0 never achieves satisfactory recovery rates. The performance difference between the two algorithms is most visible for model C, where RE0 suffers from the local minimum problem.

For models D and E, where the underlying structures are difficult to fully capture by tree models, none of the recovery rates of both methods are > .80. For model D the full recovery rates of both methods are around .50 or higher for cells with a large sample size ($K = 120$), while the partial recovery rates of RE0 are a bit higher than those of RE+ for all cells. For model E the full recovery rates of both methods are small no matter which cell of the design, and the partial recovery rates of both methods are only around .50 for the cells with a small sample size ($K = 40$).

Figure 4 presents parallel box plots of the prediction errors (criterion 4) of the two algorithms compared to the null model (i.e., predicting the study effect sizes as the overall effect size in the training set). For models B and C, the RE meta-regression tree with look-ahead and SSS performs better than RE0, especially when the number of studies is large ($K = 120$), that is, the bottom row of Figure 4. For models D and E, the difference in the prediction error between both algorithms is very small.

4.3.3 | Bias Correction Bootstrap for Residual Heterogeneity

After the true underlying model is successfully retrieved, we applied bias correction by the bootstrap for the estimated residual heterogeneity. New test sets were simulated with the same underlying model to assess how much the residual heterogeneity was underestimated and whether the bias correction bootstrap effectively corrected for this underestimation.

The bias correction bootstrap was applied only to RE meta-regression trees with SSS and the look-ahead strategy because it has better recovery performance than the standard one. In addition, bootstrapping was performed for a smaller subset of the training data due to the computational cost. Among the training sets that resulted in nontrivial trees, we randomly selected 100 nontrivial solutions (i.e., tree solutions having at least one moderator) to estimate the residual heterogeneity. We also use the corresponding

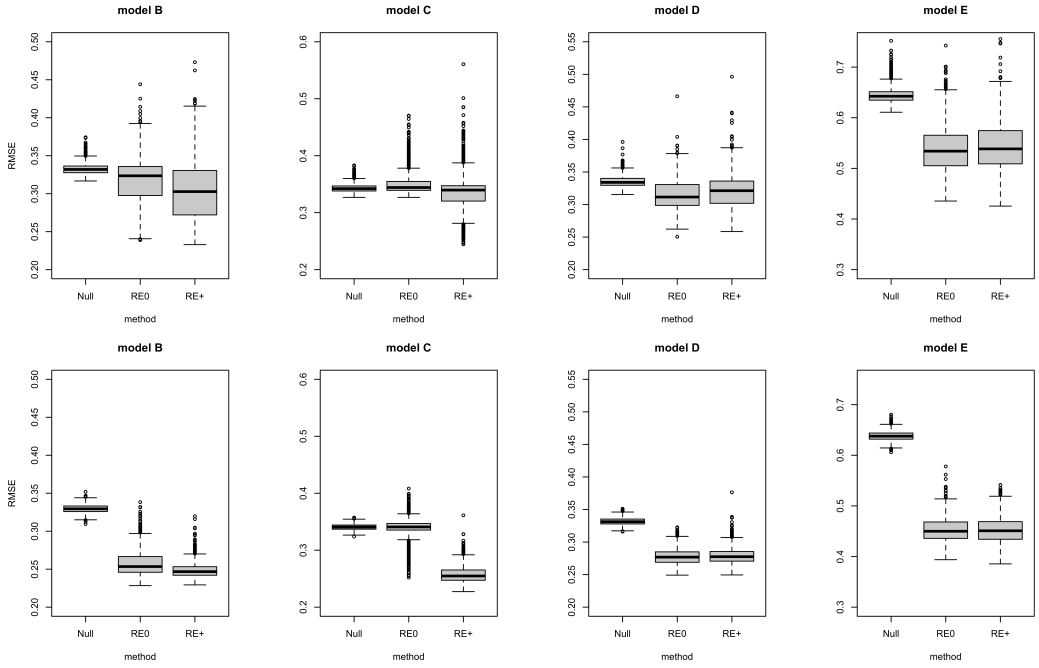

FIGURE 4 The prediction errors (RMSE) of the null model (Null) and the meta-CART algorithms: the standard RE meta-CART (RE0), and the RE meta-CART applying SSS with look-ahead (RE+). Top row displays the results of $K = 40$, bottom row of $K = 120$. The y-axes of the plots for the various models differ due to the difference in the range of the prediction errors.

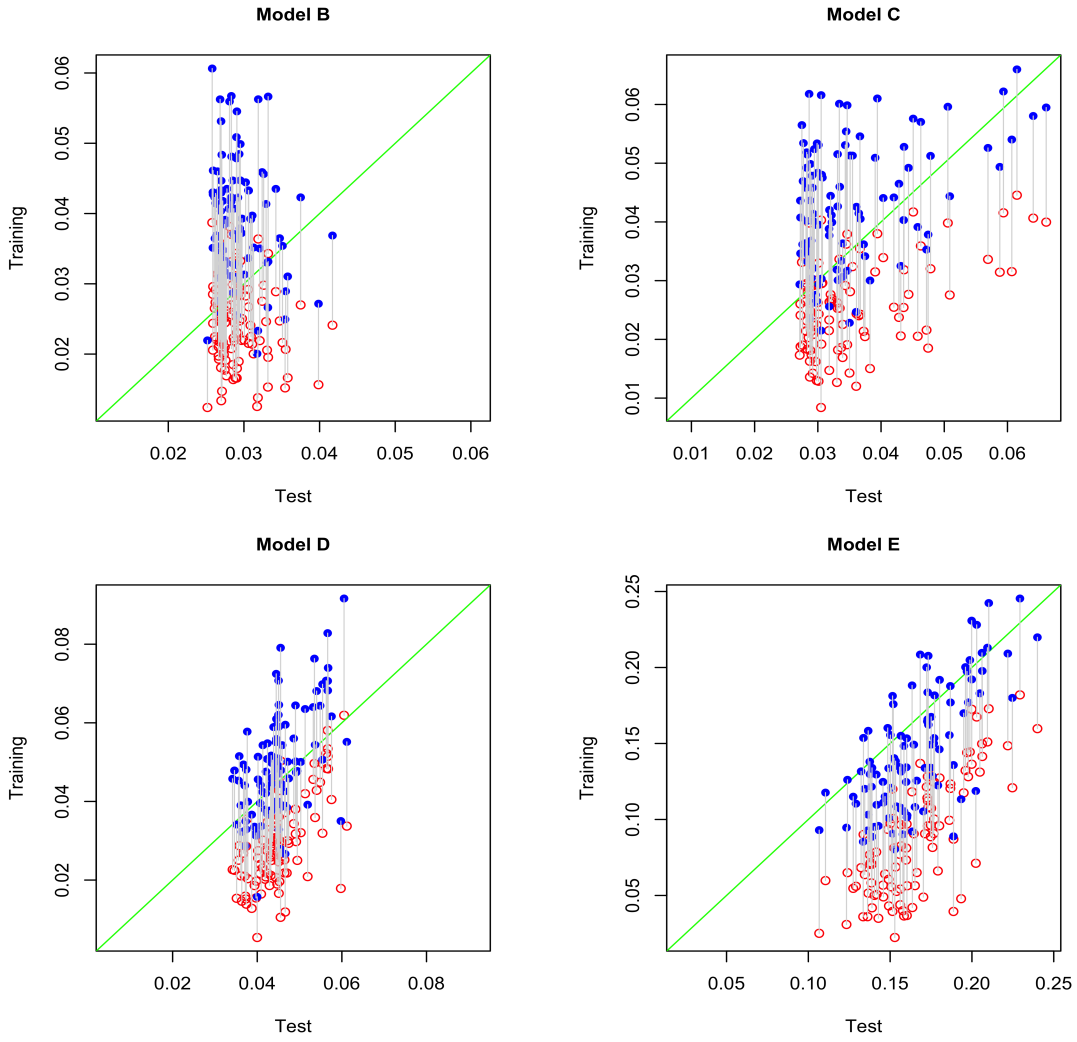

FIGURE 5 Plots of the residual heterogeneity estimates in the training samples (vertical axis) versus the test samples (horizontal axis). The axes of the various models differ due to the difference in the estimated residual heterogeneity. The bias-corrected (blue dots) and uncorrected (red circle) estimates of the residual heterogeneity for the same study are connected by a gray line. The reference line (in green) is $y = x$.

test sets that are generated from the same underlying structure to estimate to residual heterogeneity with pre-defined subgroup membership. Then, we applied Algorithm 2 to each selected training set $\mathcal{D}$, and the resulting tree model $\mathcal{T}$, with the number of bootstrap samples $B = 50$, as Le Blanc et al.[30] suggest using $25 \leq B \leq 100$ [§]. As a result, for the designed underlying model B-E, we obtain 100 estimates of residual heterogeneity in the training sets, 100 estimates of residual heterogeneity in the test sets, and 100 estimates of residual heterogeneity after bootstrap correction in the training sets. If the residual heterogeneity is properly estimated, the latter estimates in the training sets should be close to their corresponding estimates in the test sets.

Figure 5 shows the estimated residual heterogeneity in the test sets versus the estimated residual heterogeneity in the training sets when $K = 120, \sigma_\tau^2 = 0.025$. The uncorrected residual heterogeneity estimates (i.e., the red circles) are shown to be systematically underestimated in the training sets (i.e., the circles are all below the green line). After bias correction, the residual heterogeneity estimates (i.e., the blue dots) become reasonably close in most cases to the estimated values in the test sets (i.e., closer to the green line). The results are similar for the other cells of the simulation design.

[§] The choice of B suggested by Le Blanc et al.[30] served for slightly difference purpose, but the main purpose was also bias correction.

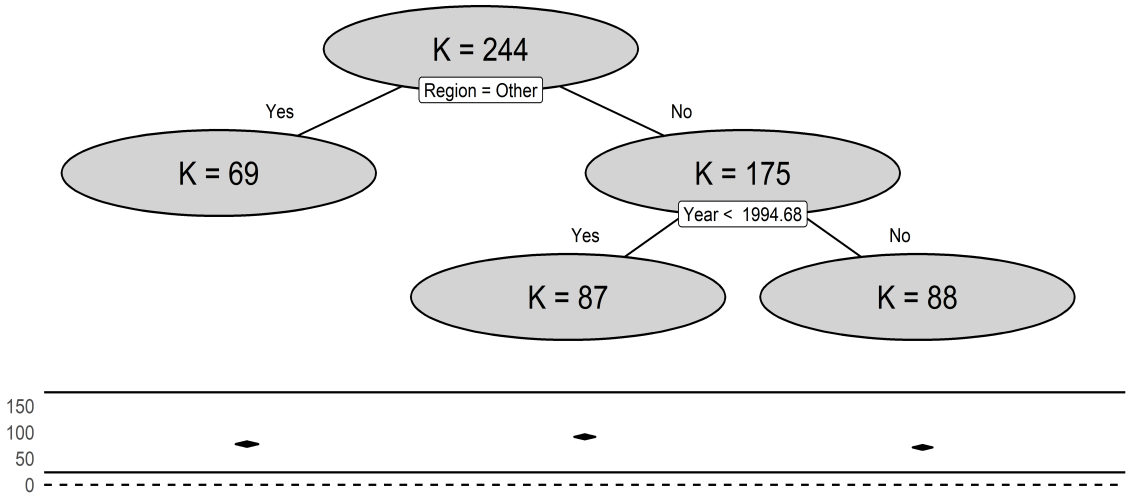

FIGURE 6 The resulting RE+ meta-regression tree for the meta-analytic data collected by Levine et al.[35]. Three subgroups are defined by two splits: 1) whether the sample is collected from Western countries or other regions, and 2) whether the midpoint of the data collection period is 1994.68 or earlier. The height of the diamonds between the solid lines represents the 95% bootstrap confidence intervals of the weighted subgroup means of sperm concentration.

.

5 | AN ILLUSTRATIVE APPLICATION

We applied the RE+ algorithm to the meta-analytic data collected by Levine et al.[35]. The R script of our analysis is available online as supporting information (data can be obtained from Levine et al.). The data set consists of 244 samples from 185 peer-reviewed studies that reported (in English-language) primary data on human sperm count. As some studies reported more than one sample, the number of samples $K = 244$ is larger than the number of studies. The effect size of interest was sperm concentration, that is, the number of sperm cells per ml in a semen sample. The overall summary estimate of the sperm concentration was 81 million/ml according to Levine et al.[35]. The data included sixteen potential moderators, such as the midpoint of the sample collection period (referred to as Year), fertility status, and geographic group (i.e., Region). The question of interest was: "which combinations of study characteristics can identify subgroups with more homogeneous summary estimates of sperm concentration? In meta-CART analysis, we used the look-ahead and SSS options (thus RE+), $c_{pruning} = 0$, the minimum terminal node size was 3, and the maximum number of splits was 10. In addition, we performed the analysis without and with a permutation test of the Q_B^*-statistic (with $J = 100$) and without and with bias correction of the within-node heterogeneity ($B = 50$).

The RE+ analysis identified three subgroups, see Figure 6. The first split was on Region and the second split on Year (with splitpoint 1994.68). The test of the moderator effect was significant without and with permutation ($Q_B^* = 61.37, df = 2$, p-value $< .001$; permutation p-value $= .010$). The estimate of residual heterogeneity was 420.1 after applying the bootstrap bias correction and 227.9 before that.

The first subgroup consists of 69 samples collected from non-western countries with a subgroup summary estimate of sperm concentration $\bar{g} = 77.4$ million/ml, bootstrap CI: [72.2; 82.5]. The second subgroup consists of 87 samples that were collected from Western countries, and the midpoint of the sample collection period is 1994.68 or earlier. This subgroup has the highest summary estimate of sperm concentration ($\bar{g} = 91.2$ million/ml, bootstrap CI: [86.4, 95.9]). The third subgroup consists of 88 samples that were collected from Western countries, and the midpoints of the sample collection period are later than 1994.68. This subgroup has the lowest summary estimate of sperm concentration ($\bar{g} = 71.0$ million/ml, bootstrap CI: [66.5, 75.5]). These bootstrap CIs were wider than the non-bootstrap CIs. It is interesting to note that the CI of the first subgroup includes the overall summary estimate of sperm concentration, which is 81 million/ml, but the second subgroup is obviously above this average and the third is below it.

To summarize the results of the analyses, sperm concentration is influenced by two moderators: the region where the samples were collected and the time period during which the samples were collected. The interaction effect between region and time period reveals that there is a decline in sperm concentration over time in the Western countries, but not in the non-western

countries. Furthermore, the highest mean sperm concentration was found for samples in the Western countries collected in 1994.68 or earlier. These findings are in line with the results of the meta-regression analyses reported in Levine et al.[35], who also found, after testing multiple possible two-way interactions, a significant interaction between region and time period. The benefit of the RE meta-CART approach for this application is that it more efficiently pinpoints the most crucial interaction between the 16 moderators than meta-regression does, and it ensures that a two-way interaction is sufficient (i.e., the pruning resulted in a two-way interaction, indicating that there is no higher order interaction present).

6 | DISCUSSION

We propose three new enhancements for random effect meta-regression trees to achieve good performance in identifying homogeneous subgroups in meta-analysis, and to make appropriate statistical inference for the analysis results. We provide a novel approach to construct random-effects meta-regression trees (RE+) that combines look-ahead and smooth sigmoid surrogate strategy to reduce the local optimum issue that tree models frequently face. For statistical inference on the identified subgroups, we propose a permutation test based on between-subgroups heterogeneity; the p-value of this permutation test is interpretable, while the p-value of the previous meta-regression trees is over-optimistic at face value[18]. For the pooled effect sizes, we proposed using bootstrap for bias correction to correct overoptimism in confidence intervals. This approach was inspired by the bootstrap method used by Loh et al.[16], and we adapted this approach for the first time in the meta-analysis framework.

In a simulation study, we compared these enhancements to the standard meta-regression tree algorithm (RE0), and we assessed whether a more liberal pruning rule (i.e., the minimum standard error rule) could be used. Our results show that for the scenario without a moderator effect, the false discovery rates for the new algorithm RE+ are smaller than RE0 and the permutation test works well, especially with the more liberal pruning rule. Furthermore, for scenarios with an underlying relationship between pooled effect size and moderators, RE+ outperformed RE0 in recovering the true moderators, especially for the scenario where ordinary tree algorithms are prone to suffer a local optimum problem. The RE+ algorithm can achieve recovery rates > 0.80 for moderator effects that can be fully explained by trees. Furthermore, for moderator effects that cannot be fully captured by trees, the recovery rates are around .50 when the sample size is large enough (i.e., $\geq$ 120 studies). In addition, the results show that the prediction error of RE+ was smaller than RE0 in most scenarios. In a separate simulation study it was shown that the bootstrap correction was able to reduce the overoptimism in the estimate of residual heterogeneity.

Look-ahead has not often been used in data mining, because it is not suitable for large scaled, "big", data due to the expensive computational cost. But this should not be a problem for meta-analyses, which typically have a limited number of studies and moderators (i.e., < 1000). For example, the computation time for the analysis (with look-ahead and SSS) in the illustrative example was 41 seconds (we used a general windows computer with an Intel Core i7 CPU, and 16GB RAM); the computation time for the same analysis with 100 permutations was 8.1 minutes, and for the 50 bootstrap samples was 2.9 minutes. In the future, it would be interesting to explore the possibility of building a tree model that guarantees a global optimum to completely solve the instability problem. For individual-level studies, Meulman et al.[10] have shown that the Exact Tree algorithm can find (guaranteed) optimal solutions for moderately-sized data with about 500 observations, 15 predictors and a maximum tree size with depth of seven. Given that meta-analytical data are typically moderately-sized, it can be promising to integrate the Exact Tree algorithm into meta-regression trees in future work.

False discovery rates of RE meta-regression trees are controlled by both the tree model identification and the between-subgroups heterogeneity test. Therefore, implementing look-ahead and SSS alone reduced the false discovery rate to ≤ 0.05, as the algorithm has better performance to retrieve the true underlying structure of the data. In other words, when there are no moderator effects, applying look-ahead and SSS reduces the risk of finding spurious moderators as a local optimal solution. Although applying look-ahead with SSS controls the Type I error rate of RE meta-regression trees close to the nominated value, we believe that using the permutation test is still relevant for two reasons. First, the p-value without correction by the permutation test is overoptimistic and loses its interpretability. Second, discrepancies have been observed between meta-analyses and large randomized controlled trials[36]. As the latter is considered to be the gold standard in evaluating the efficacy of an intervention, such discrepancies suggest that meta-analyses tend to provide liberal results[37]. Therefore, it is recommended to perform permutation tests to further reduce the risk of finding spurious moderator effects and making misleading conclusions based on those.

This paper has several strengths. One strength is that the performance of the proposed algorithm and features were examined through an extensive simulation study. The data were generated based on both linear and non-linear models, including cases where trees are good at, where trees are likely to get stuck at a local optimum, and where trees are not able to fully retrieve the relationship. The simulation study also takes sample size and residual heterogeneity into account, as these two factors influence

the Type I error and the Power[5]. Another strength of this paper is that all the proposed features are implemented in the R package `metacart`, which is published in the "Comprehensive R Archive Network". The analyses in the illustrative application section can be replicated by using the package.

There are several limitations that remain to be addressed in future research. One limitation of this paper is that it assumes only one effect size reported in each study in a meta-analysis. However, in practice, a single study may report multiple effect sizes (as in the example data in Section 5). In such cases, meta-regression trees ignore the dependency between effect sizes from the same study and may underestimate the heterogeneity between studies. Therefore, it can be interesting to extend the algorithm for multi-level meta-analytic data in future work.

Another limitation is that this paper did not examine the effect of missing values in moderators. Since meta-analytic data are usually collected from multiple independent studies, the interested moderators may not have been reported in all studies. Although tree-based methods can easily deal with missing values by treating the missing values as a new class "missing" and randomly assigning them into children nodes, a large number of missing moderators can deteriorate the performance of meta-regression trees. The performance of the algorithm on data with various missing rates should be further explored in future simulation studies.

Furthermore, this study did not investigate how publication bias affects the analysis findings. Publication bias refers to the phenomenon that studies with non-significant effect sizes have less probability to get published than those with significant effect sizes. Publication bias exists in many fields[38], and such bias in the primary studies will lead to biased results in meta-analysis[20]. For a standard meta-analysis, Van Assen et al.[39] and Van Aert et al.[40] proposed to use the *p*-uniform and *p*-curve methods to correct for publication bias. It is worthwhile to implement publication bias correction in meta-regression tree analysis in future research.

Last but not least, our current hypothesis testing framework is performed at the level of the entire tree and does not assess the significance of individual moderators. However, recent developments in selective inference for CART[41] allow for more granular testing at the level of individual splits, offering insights which specific subgroups differ significantly in their summary effect sizes. This framework can also be used to construct confidence intervals for effect sizes in terminal nodes. An important direction for future research would be to incorporate selective inference into meta-CART to evaluate the significance of individual splits and to compare the performance and reliability of confidence intervals derived from selective inference versus those obtained using bootstrap methods.

7 | CONCLUDING REMARKS

This paper has demonstrated that random effects meta-regression trees with look-ahead and SSS can be a powerful tool to identify homogeneous subgroups in meta-analysis with satisfactory control of Type I error and good recovery rate to retrieve the moderators underlying the data. Our proposal outperforms the existing meta-regression tree algorithm and provides more appropriate statistical inference for the identified subgroups by using a permutation test and bootstrap bias corrections.

For applications, we recommend always implementing the look-ahead strategy for higher stability of the results. When dealing with numeric moderators, smooth sigmoid surrogate approximation can further reduce the risk of getting stuck in a local optimum and speed up the computation. However, when only binary or categorical moderators are available, SSS is not needed since it does not improve the model performance. If subgroups are identified, we recommend performing the permutation test and bootstrap bias correction to obtain a lower risk of spurious findings. Tree models are known to be susceptible to instability, which can lead to unstable results. To address this issue, we propose to perform the analysis on the same dataset multiple times (e.g., fit meta-regression trees 20 times). This will generate a set of candidate models from which the most stable model can be selected on the basis of the frequency of occurrence. If two or more candidate models have the same frequency of occurrence, we recommend using permutation test results to break the tie.

AUTHOR CONTRIBUTIONS

Li, Dusseldorp, and Meulman initiated the innovations of meta-CART. Li and Su developed the look-ahead with the SSS strategy, the permutation, and the bootstrap correction algorithm. The remaining authors contributed to the improvement of these algorithms. Li, Dusseldorp, and Claramunt designed and performed the simulation studies. Li and Dusseldorp performed the analysis of the application. All authors contributed to the writing of the manuscript.

ACKNOWLEDGMENTS

We would like to acknowledge Levine et al.[35] for providing us the data for the application, and Mark de Rooij and Rüya Koçer for their comments on an earlier draft of the paper and suggestions for the permutation procedure.

FINANCIAL DISCLOSURE

None reported.

CONFLICT OF INTEREST

The authors declare that they have no potential conflict of interest.

References

1. Bastian H, Glasziou P, Chalmers I. Seventy-five trials and eleven systematic reviews a day: how will we ever keep up?. *PLoS Medicine.* 2010;7(9):e1000326.

2. Michie S, Abraham C, Whittington C, McAteer J, Gupta S. Effective techniques in healthy eating and physical activity interventions: a meta-regression.. *Health Psychology.* 2009;28(6):690 – 701. doi: 10.1037/a0016136

3. Tipton E, Pustejovsky JE, Ahmadi H. A History of Meta-Regression: Technical, Conceptual, and Practical Developments between 1974 and 2018. *Research Synthesis Methods.* 2018.

4. Zhang J, Wiecaszek P, Sami S, Meiser-Stedman R. Association between panic disorder and childhood adversities: a systematic review and meta-analysis. *Psychological Medicine.* 2021:1–11.

5. Li X, Dusseldorp E, Meulman JJ. Meta-CART: A tool to identify interactions between moderators in meta-analysis.. *British Journal of Mathematical and Statistical Psychology.* 2017;70(1):118–136. doi: 10.1111/bmsp.12088

6. Van Genugten L, Dusseldorp E, Webb TL, Van Empelen P. Which combinations of techniques and modes of delivery in internet-based interventions effectively change health behavior? A meta-analysis. *Journal of Medical Internet Research.* 2016;18(6).

7. Bull ER, McCleary N, Li X, Dombrowski SU, Dusseldorp E, Johnston M. Interventions to Promote Healthy Eating, Physical Activity and Smoking in Low-Income Groups: a Systematic Review with Meta-Analysis of Behavior Change Techniques and Delivery/Context. *International Journal of Behavioral Medicine.* 2018;25(6):605–616.

8. Davis C, Kim J. Bilingual lexical representation: We have some words to say. *Journal of Second Language Studies.* 2021;4(2):353–374.

9. Reimer NK, Sengupta NK. Meta-analysis of the "ironic" effects of intergroup contact. *Journal of Personality and Social Psychology.* 2023;124(2):362.

10. Meulman JJ, Dusseldorp E, Van Os BJ. *Jan Karel Lenstra, the Traveling Science Man: Liber Amicorum:*ch. An exact dynamic programming algorithm for regression trees; CWI . 2011.

11. Breiman L, Friedman J, Stone CJ, Olshen RA. *Classification and regression trees*. Wadsworth, Belmont, CA: CRC press, 1984.

12. Su X, Tsai CL, Wang H, Nickerson DM, Li B. Subgroup analysis via recursive partitioning. *Journal of Machine Learning Research.* 2009;10(Feb):141–158.

13. Foster JC, Taylor JM, Ruberg SJ. Subgroup identification from randomized clinical trial data. *Statistics in Medicine.* 2011;30(24):2867–2880.

14. Lipkovich I, Dmitrienko A, Denne J, Enas G. Subgroup identification based on differential effect search—a recursive partitioning method for establishing response to treatment in patient subpopulations. *Statistics in Medicine.* 2011;30(21):2601–2621.

15. Dusseldorp E, Van Mechelen I. Qualitative interaction trees: a tool to identify qualitative treatment–subgroup interactions. *Statistics in Medicine.* 2014;33(2):219–237.

16. Loh WY, He X, Man M. A regression tree approach to identifying subgroups with differential treatment effects. *Statistics in Medicine.* 2015;34(11):1818–1833.

17. Fokkema M, Smits N, Zeileis A, Hothorn T, Kelderman H. Detecting treatment-subgroup interactions in clustered data with generalized linear mixed-effects model trees. *Behavior Research Methods.* 2018;50:2016–2034.

18. Li X, Dusseldorp E, Meulman JJ. A flexible approach to identify interaction effects between moderators in meta-analysis. *Research Synthesis Methods.* 2019;10(1):134–152. doi: 10.1002/jrsm.1334

19. Li X, Dusseldorp E, Su X, Meulman JJ. Multiple moderator meta-analysis using the R-package Meta-CART. *Behavior Research Methods.* 2020;52:2657–2673.

20. Borenstein M, Hedges LV, Higgins JP, Rothstein HR. *Introduction to Meta-analysis*. Hoboken, NJ: Wiley, 2009

21. Friedman J, Hastie T, Tibshirani R. *The Elements of Statistical Learning*. 1. Springer series in statistics New York, NY, USA:, 2001.

22. Su X, Peña AT, Liu L, Levine RA. Random forests of interaction trees for estimating individualized treatment effects in randomized trials. *Statistics in Medicine.* 2018;37(17):2547–2560.

23. Breiman L. Bagging Predictors. *Machine Learning.* 1996;24(2):123–140.

24. Friedman J. Greedy function approximation: a gradient boosting machine. *Annals of Statistics.* 2001:1189–1232.

25. Breiman L. Random Forests. *Machine Learning.* 2001;45(1):5–32.

26. DerSimonian R, Laird N. Meta-analysis in clinical trials. *Controlled Clinical Trials.* 1986;7(3):177–188.

27. Esmeir S, Markovitch S. Anytime learning of decision trees. *Journal of Machine Learning Research.* 2007;8(May):891–933.

28. Byrd RH, Lu P, Nocedal J, Zhu C. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing.* 1995;16(5):1190–1208.

29. Higgins J, Thompson SG. Controlling the risk of spurious findings from meta-regression. *Statistics in Medicine.* 2004;23(11):1663–1682. doi: 10.1002/sim.1752

30. LeBlanc M, Crowley J. Survival trees by goodness of split. *Journal of the American Statistical Association.* 1993;88(422):457–467.

31. Efron B, Tibshirani RJ. *An Introduction to the Bootstrap*. CRC press, 1994.

32. R Core Team . *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing; Vienna, Austria: 2021.

33. Tibshirani R, Knight K. Model search by bootstrap “bumping". *Journal of Computational and Graphical Statistics.* 1999;8(4):671–686.

34. Friedman J. Multivariate adaptive regression splines. *The Annals of Statistics.* 1991;19(1):1–67.

35. Levine H, Jørgensen N, Martino-Andrade A, et al. Temporal trends in sperm count: a systematic review and meta-regression analysis. *Human reproduction update.* 2017;23(6):646–659.

36. LeLorier J, Gregoire G, Benhaddad A, Lapierre J, Derderian F. Discrepancies between meta-analyses and subsequent large randomized, controlled trials. *New England Journal of Medicine.* 1997;337(8):536–542.

37. Noma H, Nagashima K, Maruo K, Gosho M, Furukawa TA. Bartlett-type corrections and bootstrap adjustments of likelihood-based inference methods for network meta-analysis. *Statistics in Medicine.* 2018;37(7):1178–1190.

38. Rothstein HR, Sutton AJ, Borenstein M. Publication bias in meta-analysis. *Publication bias in meta-analysis: Prevention, assessment and adjustments.* 2005:1–7.

39. Van Assen MALM, Van Aert RMC, Wicherts JM. Meta-analysis using effect size distributions of only statistically significant studies.. *Psychological methods.* 2015;20(3):293.

40. Van Aert RCM, Wicherts JM, Van Assen MALM. Conducting meta-analyses based on p values: Reservations and recommendations for applying p-uniform and p-curve. *Perspectives on Psychological Science.* 2016;11(5):713–729.

41. Neufeld AC, Gao LL, Witten D. Tree-values: selective inference for regression trees. *Journal of Machine Learning Research.* 2022;23(305):1–43.

42. Gilli M, Maringer D, Schumann E. *Numerical methods and optimization in finance*. Academic Press, 2011.

43. Viechtbauer W. Accounting for heterogeneity via random-effects models and moderator analyses in meta-analysis. *Zeitschrift für Psychologie/Journal of Psychology.* 2007;215(2):104–121. doi: 10.1027/0044-3409.215.2.104

SUPPORTING INFORMATION

Additional supporting information (the R script for the analysis described in section 5) can be found in the online version of the article at the publisher’s website.

□

APPENDIX

A GENERATION OF MODERATORS AND EFFECT SIZES

For each data set, 10 moderators are generated from the uniform distribution ($x_m \sim$ uniform[0,1] for $m = 1, \ldots, 10$) with a population correlation $\rho = 0.3$. To achieve this, first, the multivariate normal vectors are simulated with correlation ρ. Then the probability integral transform is applied to the simulated vectors in the same way as in Gilli et al.[42].

For a single study, the effect size was sampled from a normal distribution with mean $\bar{g}(x_k)$ computed as in Table 1, and variance as predefined σ^2_τ. We used the same method as in Viechtbauer[43] to generate the within-study sample size n_k from a normal distribution with mean $\bar{n}$ and standard deviation $\bar{n}/3$. Finally, given the values of n_k and $\bar{g}(x_k)$, the observed effect size g_k was sampled from a non-central t-distribution

$$c(m_k)^{-1}(\tilde{n_k})^{1/2} g_k \sim t_{m_k} \bar{g}(x_k) \sqrt{\tilde{n_k}}), \tag{A1}$$

where $\tilde{n_k} = (n_k^2)/(n_k + n_k)$, and

$$c(m_k) \approx 1 - \frac{3}{4(2n_k - 2) - 1}. \tag{A2}$$