Appendix: Post-Hoc Refinement for Multitask Symbolic Regression via Consensus-Accelerated Shapley Analysis

Anonymous Submission
Anonymous affiliation

Appendix A. Background and Motivation

A.1. Detailed Analysis of Knowledge Loss in Prior GP Approaches

While traditional genetic programming (GP), an evolutionary computation method (Zhou, Yu and Qian 2019), and its multitask variants are powerful, they often fail to fully capitalize on the knowledge generated during evolution, particularly in complex, real-world scenarios.

Traditional GP
Traditional GP approaches encounter notable challenges when applied to certain demanding real-world scenarios. A primary difficulty arises in multitask symbolic regression (MTSR) problems characterized by multiple interconnected tasks, where the effective utilization and sharing of knowledge across these tasks remain problematic (Huang et al. 2022; Zhang et al. 2021; Zhong et al. 2020). Additionally, conventional GP implementations can prove inefficient when required to simultaneously construct multiple distinct yet high-performing expressions for an MTSR problem (Moyano et al. 2021; Zhong et al. 2020).

Cartesian GP (CGP) (Kushida, Hara and Takahama 2018; Miller 2020)
CGP can generate multiple outputs and shares knowledge by allowing different expressions to reference the same computational nodes in its graph-based representation. However, this link-dependent knowledge representation can make knowledge fragile. Minor changes may disrupt established knowledge structures since knowledge sharing relies on linking to the same nodes, hindering the accumulation and utilization of valuable cross-task knowledge.

Multi-gene GP
Frameworks like GEPMTR (Moyano et al. 2021) use a multi-genic chromosome structure where each gene can encode a separate expression or subexpression. While this allows for the learning of multiple expressions simultaneously, the final expressions are often constructed via ensemble aggregation or by selecting the best gene combination. This process risks overlooking beneficial insights present in suboptimal or related-task expressions, leading to the omission of valuable knowledge.

Multitask Genetic Programming (MTGP) (Zhang et al. 2022)
As outlined in the main text, the fundamental limitation of many MTGP methods is the practice of selecting the single best-performing expression for each task while discarding the rest of the population (Zhong et al. 2020). Although MTGP facilitates knowledge transfer during evolution, this final selection step creates a knowledge bottleneck. Valuable subexpressions, which are embedded with valuable knowledge or physically meaningful components, that exist within "suboptimal" or related-task expressions are permanently lost. Inaccuracies in the final expressions can weaken their predictive power and generalizability, thereby their effectiveness in real-world applications.

Our proposed method, an MTGP with bidirectional cooperation and Shapley analysis (MTGP-BS), directly addresses these limitations by creating a dedicated stage for systematically analyzing and synthesizing knowledge from the entire population of expressions, ensuring that valuable subexpressions are not prematurely discarded.

A.2. Challenges of Post-Hoc SHAP Analysis in MTSR

While Shapley additive explanations (SHAP) (Kumar et al. 2020) provides a powerful framework for feature attribution, its naive application to the raw pool of subexpressions generated by MTGP presents several significant challenges. Our framework is specifically engineered to overcome these issues.

High Dimensionality
A typical MTGP run can generate hundreds or thousands of unique subexpressions. This high-dimensional feature space, including massive redundant and noisy components, makes SHAP computation expensive (Peijin et al. 2023). Additionally, it makes SHAP values unstable and difficult to interpret, as the contribution of any single subexpression is easily obscured by this noise. Our **bidirectional subexpression co-**

operative extraction method directly addresses this by filtering redundant subexpressions and creating a compact, high-quality archive.

Incomplete Knowledge Source
Due to the stochastic nature of evolutionary computation, valuable knowledge, in the form of effective subexpressions, may not be fully captured within a single population of a task. This knowledge might be well-established in the population of a different but interconnected task. An analysis that only considers subexpressions from the target task would fail to leverage this distributed knowledge. Our **bidirectional extraction method** addresses this by explicitly creating a more complete knowledge base for analysis.

Sensitivity to Stochasticity
A single SHAP analysis is performed on a single surrogate model trained to predict the target variable using the subexpressions as inputs. The performance of this surrogate model, and thus the resulting SHAP values, can be sensitive to the random subsampling of data used for training or the inherent stochasticity of the learning algorithm. A single analysis might therefore produce misleading importance scores due to random error. Our **consensus-accelerated Shapley analysis** overcomes this issue by preserving the full distribution of SHAP values.

Ambiguity of Mean-Based Selection
Conventional ensemble approaches (Mendes-Moreira et al. 2012) often average SHAP values across multiple models to get a single importance score (Akbar et al. 2022; Wang et al. 2023). However, this can be misleading. A subexpression might have a high average importance because it is extremely important in a few models but irrelevant in others. Conversely, a subexpression that is moderately important across all models might have a similar average score. Averaging obscures this distributional information, which is significant for identifying consistently reliable components. Our **consensus-accelerated Shapley analysis** use a two-phase consensus-based subexpression selection mechanism instead of simple averaging to prevent the loss of important subexpressions.

Appendix B. Implementation Details of Bidirectional Cooperative Subexpression Extraction

B.1. Rationale for the Design of Each Stage

Each stage in our extraction method is designed to fulfill a specific role in refining the subexpression archive.

Intra-Task Frequency-Guided Archiving
The rationale for this stage is rooted in the “wisdom of the crowd” inherent in evolutionary computation (Lowrance, Abdelwahab and Yampolskiy 2015). In a well-converged MTGP run, subexpressions that form the building blocks of the true underlying model tend to survive across many individuals in the population. Consequently, their occurrence frequency becomes a strong heuristic indicator of their importance and relevance. By clustering subexpressions into “high-frequency” and “low-frequency” groups, we employ a data-driven mechanism to separate this strong signal from the noise of rare or incidental mutations, thereby creating a high-quality initial archive.

Conditional Inter-Task Supplementation
The conditional logic in the inter-task supplementation stage is crucial for balancing knowledge enrichment against the risk of introducing noise. Below, we provide a detailed rationale for each of the three cases.

- **Case 1: Sufficient diversity in current archive and the relevant-task best expression is correct.** When the archive is already diverse, our priority shifts from enrichment to reliability. A correct expression from a relevant task is a golden source of knowledge. By exclusively incorporating subexpressions from this high-confidence source, we enrich our strong archive with proven, valuable building subexpressions while minimizing the risk of adding noise.
- **Case 2: Sufficient diversity in current archive and the relevant-task best expression is incorrect.** In this scenario, we have a diverse set of potentially good building subexpressions from the current task. The best expression from the relevant task has not met our high bar for correctness. The risk of polluting our diverse archive with misleading or suboptimal structures from this unverified expression outweighs the potential benefit of adding more diversity. Therefore, the most prudent action is to skip the supplementation step.
- **Case 3: Insufficient diversity in archive.** This case signals potential stagnation, where the search is too narrow. In this context, the necessity for diversity and potential utility outweighs the risk associated with incorporating subexpressions from an incorrect expression. Even if the best expression from the relevant task is incorrect, it is still a highly-evolved structure and is more likely to contain useful components than random alternatives. We therefore prioritize the immediate need for diversity, accepting a calculated risk of introducing some noise in order to enrich the knowledge of the subexpression archive.

Intra-Task Best-Driven Injection
While frequency analysis captures collective wisdom, it is possible for a highly effective, yet complex and rare, combination of subexpressions to appear only in the single best-performing individual p_t^*. This final, unconditional injection serves as a crucial fail-safe to ensure that the subexpressions of the most successful solution discovered so far are guaranteed to be included in the archive, preventing the loss of potentially important knowledge.

Algorithm S1: Bidirectional subexpression cooperative extraction method

```
Input: Number of tasks $T$; optimized population $\{\mathbf{P}_t\}_{t=1}^{T}$; raw subexpression $\{\mathbf{R}_t\}_{t=1}^{T}$
Parameters: Diversity threshold $\theta_{div}$, performance threshold $\theta_{perf}$
Output: Subexpression archives $\mathbf{A} = \{\mathbf{A}_t\}_{t=1}^{T}$
1.   Set $\mathbf{A} = \emptyset$;
2.   For $t = 1$ to $T$ do
       // Stage 1: Intra-task frequency-guided archiving
3.       Set $\mathbf{A}_t^{(1)} = \text{cluster}_{\text{high}}(\mathbf{R}_t)$;
       // Stage 2: Conditional Inter-task Supplementation
4.       If $Div(\mathbf{A}_t^{(1)}) \leq \theta_{div}$ then            // Case 3
5.           Set $\mathbf{A}_t^{(2)} = \mathbf{A}_t^{(1)} \cup \mathcal{E}(p_{t'}^*)$;
6.       Else if $\text{Correct}(p_{t'}^*, \theta_{perf})$ then     // Case 1
7.           Set $\mathbf{A}_t^{(2)} = \mathbf{A}_t^{(1)} \cup \mathcal{E}(p_{t'}^*)$;
8.       Else                                               // Case 2
9.           Set $\mathbf{A}_t^{(2)} = \mathbf{A}_t^{(1)}$;
10.      End If
       // Stage 3: Intra-task best-driven injection
11.      Set $\mathbf{A}_t = \mathbf{A}_t^{(2)} \cup \mathcal{E}(p_t^*)$;
12.      $\mathbf{A} = \mathbf{A} \cup \mathbf{A}_t$;
13.  End For
14.  Return A.
```

B.2. Mitigating Negative Transfer through Adaptive Risk Management

The logic in conditional inter-task supplementation forms a strong safeguard against negative transfer. By evaluating not only the quality of the incoming knowledge but also its own internal state (i.e., the diversity of the archive), it can make a context-aware decision on whether to accept new subexpressions. It can carefully reject potentially noisy subexpressions from the interconnected tasks when its own knowledge base is strong and diverse, thereby preventing the pollution of high-quality solutions. This adaptive risk management ensures that knowledge is shared opportunistically and safely, harnessing the benefits of inter-task learning while inherently mitigating its potential detrimental effects.

B.3. Pseudocode and Complexity Analysis

To better understand the proposed bidirectional subexpression cooperative extraction method, the pseudocode is provided in Algorithm S1. For each task, the subexpression archive is constructed by selecting subexpressions from the current task in Line 3 and Line 11, and from relevant tasks in Lines 4-10. As the algorithm iterates over all tasks sequentially, and each iteration includes a constant number of subexpression operations, the overall time complexity is $O(T)$, where T denotes the number of tasks.

B.4. Definitions of Quantitative Criteria

- **Frequency Clustering** $\mathbf{cluster}_{\mathbf{high}}(\mathbf{R}_t)$: This function first calculates the occurrence frequency of each unique subexpression in $\mathbf{R}_t$. It then applies a k-means clustering algorithm to these frequency values, where the subexpressions are clustered into two groups. The function returns all subexpressions that belong to the cluster with the higher-frequency centroid.
- **Diversity** $\boldsymbol{Div}(\cdot)$: Defined as the number of unique subexpressions in an archive. The threshold θ_{div} is set to two.
- **Correctness check** $\mathbf{Correct}(\boldsymbol{p}_t^*)$: An expression p^* is correct if its mean squared error *mse* on a validation set is below a stringent threshold θ_{perf}, where the value of θ_{perf} is 10^{-16}.

Appendix C. Implementation Details for Consensus-Accelerated Shapley Analysis

C.1. Design Rationales

Motivation for an Ensemble-Based SHAP Approach

A standard SHAP analysis relies on a single surrogate model, resulting importance scores can be unstable, varying significantly due to the stochastic nature of model training (e.g., random weight initialization, data shuffling). A model that happens to overfit or underfit certain patterns can produce misleading SHAP values. By using an ensemble of models trained on different bootstrap samples, we can create a distribution of SHAP values for each subexpression, allowing for a much more reliable assessment of its true importance, mitigating the impact of any single outlier model.

Rationale for Multi-Model Attribution and Ranking Process

Within our ensemble framework, we made several key design choices. Instead of simplifying the output, all absolute SHAP values of each subexpression are preserved across all models. This ensures the completeness of contribution information, supporting a more comprehensive judgment of subexpression importance in the subsequent selection stage. Furthermore, the rankings of the subexpressions within each model are computed and stored. These rankings serve as a scale-invariant input for the consensus-based subexpression selection process, providing a stable foundation for identifying truly important subexpressions.

Rationale for the Consensus-Based Subexpression Selection

Simply averaging the SHAP values from the ensemble can be misleading. An average value fails to distinguish between a subexpression that is valuable in a few models but irrelevant in others, and one that is moderately important across all models. The functions of the approach including two phases are as follows:

- The voting-based selection is a strict, high-confidence filter designed to find strong consensus. It identifies subexpressions that are not just important, but whose importance is so stable that a majority of models agree on their exact high-ranking position. These are the undisputed the most important subexpressions.
- The enhanced roulette wheel selection is a more inclusive, probabilistic filter for weak consensus. It rescues valuable subexpressions that contribute consistently across many models but may not always achieve the same top rank. This ensures that reliable subexpressions are not discarded simply because they don't have the "star power" to pass the strict voting phase.

C.2. Bootstrap-Based Data Preparation

The data for training multiple surrogate models is prepared as follows. The structure of the data used for model training is first defined. Specifically, M subexpressions $\mathbf{A}_t = [a_{t,1}, a_{t,2}, \cdots a_{t,m}, \cdots, a_{t,M}]$ in the archive are regarded as input features, and the task-specific target is treated as the output feature. Based on this structure, a raw dataset $\mathbf{D} = [\mathbf{X}, \mathbf{Y}] = \{(\mathbf{X}_n, y_n)\}_{n=1}^{N}$ is generated for the task, where $\mathbf{X}_n = \{x_n^m\}_{m=1}^{M}$ and y_n refer to the values of the input features and the corresponding task-specific target for the *n*-th sample, respectively. x_n^m refers to the value of each input feature, calculated as the value of the *m*-th subexpression in the archive derived from the original sample inputs. *M* and *N* refer to the number of subexpressions in the archive and the number of samples in the dataset, respectively.

From this raw dataset, K bootstrap samples $\mathbf{D}' = \{\mathbf{X}'_k, \mathbf{Y}'_k\}_{k=1}^{K}$, each of size *N*, are sampled from $\mathbf{D}$ to train K different models. The $\mathbf{D}'_k = \{\mathbf{X}'_k, \boldsymbol{Y}'_k\}$ refers to the training input and output data for the k-th model, where $\mathbf{X}'_k = \{\mathbf{X}'_{k,n}\}_{n=1}^{N}$ and $\mathbf{Y}'_k = \{y'_{k,n}\}_{n=1}^{N}$. The $\{\mathbf{X}'_{k,n}, y'_{k,n}\}$ refers to the n-th sample in the k-th bootstrap. A crucial aspect of this preparation is reserving the out-of-bag samples $\mathbf{D}_k^{OOB} = \{\mathbf{X}_k^{OOB}, \mathbf{Y}_k^{OOB}\}_{k=1}^{K}$, those from $\mathbf{D}$ but not included in $\mathbf{D}'$, for subsequent SHAP analysis, enabling subexpression evaluation on unseen data and improving generalizability. The $\mathbf{X}_k^{OOB} = \{x_{k,n}^{OOB}\}_{n=1}^{N_{OOB}}$ and $\mathbf{Y}_k^{\boldsymbol{OOB}} = \{y_{k,n}^{OOB}\}_{n=1}^{N_{OOB}}$, where $\{x_{k,n}^{OOB}, y_{k,n}^{OOB}\}$ refers to the *n*-th out-of-bag sample in the *k*-th bootstrap sample, and N_{OOB} refers to the number of out-of-bag samples.

C.3. Surrogate Model Training and SHAP Definition in SHAP-Based Multi-Model Attribution

Based on bootstrapped datasets $\{\mathbf{X}'_k, \mathbf{Y}'_k\}_{k=1}^{K}$, K task-specific models are trained independently. Each model, using subexpressions in the archive as input features, is trained by minimizing the mean square error between its predicted values $\widehat{\mathbf{Y}}'_k = \{\widehat{\boldsymbol{y}}'_{k,n}\}_{n=1}^{N}$ and the actual targets $\mathbf{Y}'_k = \{y'_{k,n}\}_{n=1}^{N}$. The *mse* for the *k*-th model is calculated as

$$mse = \frac{1}{N}\sum_{n=1}^{N}\left(y'_{k,n} - \widehat{\boldsymbol{y}}'_{k,n}\right)^2.$$

Following independent training, SHAP values are computed for the k-th model f_k and a specific input sample x, the SHAP value $\varphi_{k,m}(f_k, x)$ for the m-th subexpression is defined as,

$$\varphi_{k,m}(f_k, x) = \sum_{\mathbf{S}\subseteq\mathbf{A}_t\setminus\{a_{t,m}\}} \frac{|\mathbf{S}|!(|\mathbf{A}_t|-|\mathbf{S}_t|-1)!}{|\mathbf{A}_t|!} \times \left(f_k(x^{\mathbf{S}_t\cup\{a_{t,m}\}}) - f_k(x^{\mathbf{S}_t})\right),$$

where $|\mathbf{A}_t|$ and $|\mathbf{S}_t|$ are cardinalities of the corresponding sets, a_m refers to the *m*-th subexpression in $\mathbf{A}_t$, $\mathbf{S}_t$ refers to subsets of subexpressions from $\mathbf{A}_t$ excluding $a_{t,m}$. Additionally, $f_k(\cdot)$ refers to the prediction function of the k-th model, $x^{\mathbf{S}_t\cup\{a_{t,m}\}}$ refers to an input sample where the values for features $\mathbf{S}_t \cup \{a_{t,m}\}$ are taken from x, and $x^{\mathbf{S}_t}$ refers to an input sample where the values for features $\mathbf{S}_t$ retained from x. Based on these equations, the subsequential quantitative measure of the contribution $\boldsymbol{\Phi}_t$ and the corresponding importance ranks $\boldsymbol{\Gamma}_t$ can be calculated as introduced in the main text.

C.4. Consensus-Based Subexpression Selection Process

To further clarify the procedure of the consensus-based subexpression selection mechanism, we provide a detailed prose description for each phase.

Phase 1: Voting-Based Selection

This process iterates through each subexpression $a_{t,m}$. For a given $a_{t,m}$, the method examines its rank vector across all K models, $\{\boldsymbol{\Gamma}_{t,k,m}\}_{m=1}^{M}$. It then checks if there exists any specific high-ranking integer γ^*, $1 \le \gamma^* \le \lfloor M/2 \rfloor$, that appears more than $K/2$ times in this vector. For example, if $K = 10$, the algorithm checks if subexpression $a_{t,m}$ is ranked exactly first by more than five models, then checks if it is ranked exactly second by more than five models, and so on. If such a strong consensus on a specific high rank is found, $a_{t,m}$ is immediately added to the set of the most important subexpressions $\overline{\mathbf{A}}_t$.

Phase 2: Enhanced Roulette Wheel Selection

This phase operates on the set of subexpressions $\mathbf{U}$ that are not selected by the voting mechanism. The process consists of K independent trials, one corresponding to each model f_k. In the k-th trial, the mechanism performs $|\mathbf{U}|$ rounds of roulette wheel selection (sampling with replacement), where $|\mathbf{U}|$ is the number of subexpressions in the set $\mathbf{U}$. For each of these $|\mathbf{U}|$ spins, a single subexpression is chosen from $\mathbf{U}$, with the probability for each subexpression $a_{t,m} \in \mathbf{U}$ being proportional to its absolute SHAP value $\boldsymbol{\Phi}_{t,k,m}$ in the context of that specific model. This generates a multiset of $|\mathbf{U}|$ selected subexpressions for the k-th trial. After all K trials are complete, the algorithm proceeds to the final check. For each subexpression $a_{t,m} \in \mathbf{U}$, it counts in how many of the K trials it appeared at least once. If this count exceeds the

```
Algorithm S2: Consensus-accelerated Shapley analysis
Input: the $t$-th subexpressions archive $\mathbf{A}_t = \{a_{t,m}\}_{m=1}^{M}$; number of models K; task number t; data D for the t-th task
Output: the most important subexpressions $\overline{\mathbf{A}}_t$ for the t-th task
1.  Set $\overline{\mathbf{A}}_t = \emptyset$;
    // Stage a) Bootstrap-based data preparation
2.  Generate $K$ bootstrap samples $\{\mathbf{D}'_k\}_{k=1}^{K}$ and out-of-bag samples $\{\mathbf{D}_k^{OOB}\}_{k=1}^{K}$ from D;
    // Stage b) SHAP-based multi-model attribution
3.  Set $\mathbf{\Phi}_t = \emptyset, \mathbf{\Gamma}_t = \emptyset$; // $\mathbf{\Phi}_t$ and $\mathbf{\Gamma}_t$ are contributions and
                                  // rankings of subexpressions from the
                                  // K models, respectively
4.  For k = 1 to K do
5.      Train model $f_k$ on $\mathbf{D}'_k$ by minimizing mse;
6.      Set $\mathbf{\Phi}_{t,k,\cdot} = \{|\phi_{k,m}|\}_{m=1}^{M}$ via $f_k$ and $\mathbf{D}_k^{OOB}$;
7.      Set $\mathbf{\Gamma}_{t,k,\cdot} = \{\Gamma_{t,k,m}\}_{m=1}^{M} = \text{Rank}(\mathbf{\Phi}_{t,k,\cdot})$;
8.      Set $\mathbf{\Phi}_t = \mathbf{\Phi}_t \cup \mathbf{\Phi}_{t,k,\cdot}$
9.      Set $\mathbf{\Gamma}_t = \mathbf{\Gamma}_t \cup \mathbf{\Gamma}_{t,k,\cdot}$;
10. End For
    // Stage c) Consensus-based subexpression selection
    // Phase 1: Voting-based selection
11. For each $a_{t,m} \in \mathbf{A}_t$ do
12.     If $\exists \gamma^* \in \left\{1, \cdots \left\lfloor \frac{M}{2} \right\rfloor\right\}$ s.t. $\sum_{k=1}^{K} \mathbb{I}(\Gamma_{t,k,m} = \gamma^*) > \frac{K}{2}$ then
13.         $\overline{\mathbf{A}}_t = \overline{\mathbf{A}}_t \cup \{a_{t,m}\}$;
14.     End If
15. End For
    // Phase 2: Enhanced roulette wheel selection
16. Set $\mathbf{U} = \mathbf{A}_t \backslash \overline{\mathbf{A}}_t$;
17. Set $counts[a_{t,m}] = 0$ for all $a_{t,m} \in \mathbf{U}$;
18. For k = 1 to K do
19.     Set $\mathbf{A}_{temp} = \emptyset$; // $\mathbf{A}_{temp}$ is a temporary set
20.     Calculate $\{\Pr(a_{t,m} \mid f_k)\}_{a_{t,m} \in U}$ using $Pr(a_{t,m}|f_k) = \frac{|\phi_{k,m}|}{\sum_{a_{t,m} \in \mathbf{U}} |\phi_{k,m}|}$;
21.     For spin = 1 to |U| do
22.         Set $a_{selected}$ = select one subexpression from U
            via roulette wheel;
23.         $\mathbf{A}_{temp} = \mathbf{A}_{temp} \cup \{a_{selected}\}$;
24.     End For
25.     For each unique $a_{t,m} \in \mathbf{A}_{temp}$ do
26.         $counts[a_{t,m}] = counts[a_{t,m}] + 1$;
27.     End For
28. End For
29. For each $a_{t,m} \in \mathbf{U}$ do
30.     If $counts[a_{t,m}] > K/2$ then
31.         $\overline{\mathbf{A}}_t = \overline{\mathbf{A}}_t \cup \{a_{t,m}\}$;
32.     End If
33. End For
34. Return $\overline{\mathbf{A}}_t$.
```

majority threshold of $K/2$, the subexpression is deemed to consistent support and is added to the final set $\overline{A}_t$.

C.5. Pseudocode and Complexity Analysis

The procedure of consensus-accelerated Shapley analysis is provided in Algorithm S2, with a detailed complexity analysis for each stage. Let K be the number of models, N be the number of data samples, and M be the number of subexpressions in the archive.

First, the initialization in handled in Lines 1-2, where K diverse training datasets $\mathbf{D}'_k$ and corresponding evaluation sets $\mathbf{D}_k^{OOB}$ are generated using the standard bootstrap method. The time complexity of this stage is $O(K \times N)$. Second, the SHAP-based multi-model attribution, a key part of our framework, is implements in Lines 3-10. The function of this stage is to iterate through the K models, train each one using $\mathbf{D}'_k$, and compute the importance scores $\mathbf{\Phi}_t$ and ranks $\mathbf{\Gamma}_t$ for all M subexpressions with respect to each model using $\mathbf{D}_k^{OOB}$. Thus, the complexity is $O(K \cdot (C_{\text{train}}(N, M) + C_{\text{SHAP}}(N \times M)))$ when using a small-scale model. Here, $C_{\text{train}}(N, M)$ represents the time complexity of training a single surrogate model, which is approximately $O(N, M)$ as the number of training epoch is a constant. $C_{\text{SHAP}}(N \times M)$ is the complexity of computing the SHAP values, which is approximately $O(N \times M)$ using an efficient explainer on $\mathbf{D}_k^{OOB}$,as the size of $\mathbf{D}_k^{OOB}$ is proportional to N. Thus, the total complexity for this block is $O(K \times N \times M)$. Third, the voting-based selection phase is executed in Lines 11-15., which iterates through each of the M subexpressions. In each iteration, it must check for the existence of a specific rank γ^* in the top half for each model. In the worst case, this requires an inner loop over $O(M \times K)$ possible ranks. Thus, the time complexity of this process is approximately $O(K \times M^2)$. Forth, the enhanced roulette wheel selection phase is executed in Lines 16-33. The complexity in this phase is dominated by the nested loops. It first runs K trials with each trial performing $|\mathbf{U}|$ roulette wheel selections, where $|\mathbf{U}|$ is at most M. As a single roulette wheel selection requires scanning the probabilities of all $|\mathbf{U}|$ subexpressions, the nested loops for gathering the selection counts in Lines 16-28 have a complexity of $O(K \cdot M^2)$. The final loops for checking the counts are less complex. Thus, the complexity of this phase is $O(K \cdot M^2)$.

The total complexity is the sum of all stages. Therefore, the overall time complexity of the algorithm can be concisely expressed as $O(K \times M \times \max(N \times M))$.

Appendix D. Implementation Details for Final Expression Synthesis

The final stage of the MTGP-BS constructs the refined expression e_t for the t-th task by treating the set of the most meaningful subexpressions $\overline{\mathbf{A}}_t$ identified by the consensus-accelerated Shapley analysis as a set of basic components, where $\overline{\mathbf{A}}_t = \{\overline{a}_{t,1}, \overline{a}_{t,2}, \cdots, \overline{a}_{t,m'}, \cdots, \overline{a}_{t,M'}\}$, M' refers to the number of the selected subexpressions. The final expression is formulated as a linear model,

$$e_t(\mathbf{z}) = w_0 + \sum_{m'=1}^{M'} w_{m'} \overline{a}_{t,m'}(\mathbf{z}),$$

where $\mathbf{z}$ is the vector of input variables, $\mathbf{w} = [w_0, w_1, \cdots, w_{m'}, \cdots, w_{M'}]$ is the vector of coefficients to be determined, and $\overline{a}_{t,m'}(\boldsymbol{z})$ is the output of the m'-th subexpression.

To find the optimal coefficients, we use the training dataset for the t-th task, denoted as $\{(\mathbf{z}_n, y_n)\}_{n=1}^{N}$, where $\mathbf{z}_n$ is the vector of original input variables for the n-th sample, y_n is its corresponding target value, and N is the number of samples. First, a design matrix $\mathbf{B}$ of size $N \times (M' + 1)$ is created. The first column of $\mathbf{B}$ is a vector of ones for the intercept term w_0, and each subsequent column $m' + 1$ contains the values of the subexpression $\overline{a}_{t,m'}$ evaluated for each of the N samples. Specifically, the element at the n-th row and the $(m' + 1)$-th column is

$$\mathbf{B}_{n,(m'+1)} = \overline{a}_{t,m'}(\mathbf{z}_n).$$

The optimal coefficient vector $\mathbf{w}^*$ is found not in a single step, but through an iterative pruning process. Initially, the standard least squares problem,

$$\mathbf{w}^* = arg\ \underset{\boldsymbol{w}}{min} \|\mathbf{Y} - \mathbf{B}\mathbf{w}\|_2^2,$$

is solved using all M' subexpressions, where $\mathbf{Y} = \{y_n\}_{n=1}^{N}$. After this regression, any subexpression whose corresponding coefficient $|w_{m'}|$ is below a small threshold (e.g., 10^{-16}) is considered to have a negligible contribution and is removed, where $|\cdot|$ refers to calculating the absolute value. This process of regression and pruning is repeated on the reduced set of subexpressions until an iteration occurs where no coefficients are pruned. This iterative procedure ensures that the final, synthesized expression $e_t(\mathbf{z})$ is constructed only from components with a meaningful impact.

Appendix E. Overall Time Complexity Analysis

The overall time complexity of the MTGP-BS is the sum of its two main phases, including the MTGP evolution phase and the post-hoc refinement phase. Let T be the number of tasks, I be the population size per task, G be the number of generations, N be the number of data samples, K be the number of ensemble models, and M be the number of subexpressions in an archive.

The MTGP evolution is completed in Lines 1-2. As the complexity of a standard MTGP evolution process is primarily driven by the fitness evaluation of all individuals over all generations, its time complexity is $O(T \times I \times G \times N)$. The post-hoc refinement is executed in Lines 3-10, which is composed of bidirectional subexpression cooperative extraction in Line 4 with negligible complexity $O(T)$, the consensus-accelerated Shapley analysis in Line 6 with time complexity $O(T \times K \times M \times \max(N \times M))$, and the final synthesis in Line 7 with negligible time complexity $O(N \times M^2)$. Thus, the overall time complexity of MTGP-BS is $O(T \times I \times G \times N + T \times K \times M \times \max(N \times M))$.

In practice, the computational cost of the evolutionary process is typically the bottleneck, as the number of generations G and the population size I are often large. Therefore, the additional cost of our post-hoc refinement does not fundamentally change the overall complexity $O(T \times I \times G \times N)$ of a standard, large-scale MTGP application.

Appendix F. Theoretical Analysis

Here, we provide a formal theoretical analysis for the statistical properties of our voting-based selection mechanism.

F.1. Formal Setup and Assumptions

Let our K models $\{f_1, f_2, \dots, f_K\}$ be independent and identically distributed (i.i.d.). For any subexpression a_m and model f_k, let $\Gamma_{t,k,m} \in \{1, \dots, M\}$ be the importance rank assigned by the model.

The voting-based selection rule in our consensus-based subexpression selection mechanism states that a subexpression a_m is selected if a majority of models agree on a specific high rank. Formally, we define the selection event ζ_m for subexpression a_m as:

$$\zeta_m\colon \exists\gamma^* \in \left\{1, \dots, \left\lfloor\frac{M}{2}\right\rfloor\right\} \text{ s.t. } \textstyle\sum_{k=1}^{K} \mathbb{I}\big(\Gamma_{t,k,m} = \gamma^*\big) > \frac{K}{2}.$$

Our goal is to demonstrate that the occurrence of event ζ_m is a statistically significant indicator of genuine importance and is highly unlikely to occur by random chance.

F.2. Proof of Statistical Significance of the Consensus

Proposition 1. *Under a null hypothesis where importance ranks are assigned randomly within the high-rank pool, the probability of a subexpression being selected by our voting-based selection is exceedingly low, demonstrating that a selection is a statistically significant event.*

Proof. The proof proceeds by calculating the probability of the selection event ζ_m occurring under a null hypothesis (H_0).

- **Null hypothesis (H_0):** Assume that the subexpression a_m is not genuinely important. Under this assumption, the rank $\Gamma_{t,k,m}$ assigned by any model f_k is a random variable. Let assume that the ranks are uniformly distributed among the top $\Upsilon = \lfloor M/2 \rfloor$ positions. The probability that a single model assigns a specific high rank γ^* to a_m is

  $$Pr_{\gamma^*} = 1/\Upsilon.$$
- **Binomial probability:** For any fixed rank γ^*, the number of models C_{γ^*} that assign this exact rank to a_m follows a binomial distribution,

  $$C_{\gamma^*} \sim B(K, Pr_{\gamma^*}).$$

  The probability that a majority of models agree on this specific rank γ^* by chance is given by the sum of the binomial probabilities,

  $$P(C_{\gamma^*} > K/2) = \textstyle\sum_{K'=\lfloor K/2\rfloor+1}^{K} \binom{K}{K'} Pr_{\gamma^*}^{i} (1 - Pr_{\gamma^*})^{K-K'},$$

  where $\binom{K}{K'}$ is the binomial coefficient, representing the total number of ways for exactly K' out of K models to reach consensus, Pr_{γ^*} is the probability of K' models all

Set	Expression	# of Vars	Operators	Range
Set 1	$x_1^2+2x_2^2+3x_3^2$	3	+, -, ×, ^2	Train, test: [0,10]
	$3x_1^2+4x_1x_2+2x_1x_3+2x_2^2+2x_2x_3+x_3^2$	3		
Set 2	$x_1^2+2x_2^2$	2	+, -, ×, ^2	Train, test: [0,10]
	$100x_1^4-200x_1^2x_2+x_1^2-2x_1+100x_2^2+1$	2		
Set 3	$x_1^2+2x_2^2+3x_3^2$	3	+, -, ×, ^2, *cos*	Train, test: [0,10]
	$x_1^2+x_2^2+x_3^2-10cos\,(x_1)-10cos\,(x_2)-10cos\,(x_3)+30$	3		
Set 4	$x_1^2+2x_2^2$	2	+, -, ×, ^2, *cos*	Train, test: [0,10]
	$x_1^2/4000+x_2^2/4000-cos\,(x_1)cos\,(x_2/1.41)+1$	2		
Set 5	$x_1^2+2x_2^2$	2	+, -, ×, ^2, *cos*	Train, test: [0,10]
	$(x_2-0.13x_1^2+1.59x_1-6)^2+9.6cos\,(x_1)+10$	2		
Set 6	$2x_1^2+2x_1x_2+x_2^2$	2	+, -, ×, ^2	Train, test: [0,10]
	$100x_1^4-200x_1^2x_2+x_1^2-2x_1+100x_2^2+1$	2		
Set 7	$3x_1^2+4x_1x_2+2x_1x_3+2x_2^2+2x_2x_3+x_3^2$	3	+, -, ×, ^2, *cos*	Train, test: [0,10]
	$x_1^2+x_2^2+x_3^2-10cos\,(x_1)-10cos\,(x_2)-10cos\,(x_3)+30$	3		
Set 8	$2x_1^2+2x_1x_2+x_2^2$	2	+, -, ×, ^2, *cos*	Train, test: [0,10]
	$x_1^2/4000+x_2^2/4000-cos\,(x_1)cos\,(x_2/1.41)+1$	2		
Set 9	$2x_1^2+2x_1x_2+x_2^2$	2	+, -, ×, ^2, *cos*	Train, test: [0,10]
	$(x_2-0.13x_1^2+1.59x_1-6)^2+9.6cos\,(x_1)+10$	2		
Set 10	$100x_1^4-200x_1^2x_2+x_1^2-2x_1+100x_2^2+1$	2	+, -, ×, ^2, *cos*	Train, test: [0,10]
	$x_1^2+x_2^2-10cos\,(x_1)-10cos\,(x_2)+20$	2		
Set 11	$100x_1^4-200x_1^2x_2+x_1^2-2x_1+100x_2^2+1$	2	+, -, ×, ^2, *cos*	Train, test: [0,10]
	$x_1^2/4000+x_2^2/4000-cos\,(x_1)cos\,(x_2/1.41)+1$	2		
Set 12	$100x_1^4-200x_1^2x_2+x_1^2-2x_1+100x_2^2+1$	2	+, -, ×, ^2, *cos*	Train, test: [0,10]
	$(x_2-0.13x_1^2+1.59x_1-6)^2+9.6cos\,(x_1)+10$	2		
Set 13	$x_1^2+x_2^2-10cos\,(x_1)-10cos\,(x_2)+20$	2	+, -, ×, ^2, *cos*	Train, test: [0,10]
	$x_1^2/4000+x_2^2/4000-cos\,(x_1)cos\,(x_2/1.41)+1$	2		
Set 14	$x_1^2+x_2^2-10cos\,(x_1)-10cos\,(x_2)+20$	2	+, -, ×, ^2, *cos*	Train, test: [0,10]
	$(x_2-0.13x_1^2+1.59x_1-6)^2+9.6cos\,(x_1)+10$	2		
Set 15	$x_1^2/4000+x_2^2/4000-cos\,(x_1)cos\,(x_2/1.41)+1$	2	+, -, ×, ^2, *cos*	Train, test: [0,10]
	$(x_2-0.13x_1^2+1.59x_1-6)^2+9.6cos\,(x_1)+10$	2		
Set 16	$x_1^2-10cos\,(x_1)+10$	1	+, -, ×, ^2, *cos*	Train, test: [0,10]
	$x_1^2/4000-cos\,(x_1)+1$	1		

Table S1: Benchmark problems for the main experiment and their properties

giving exactly the rank γ^*, and $(1-Pr_{\gamma^*})^{K-K'}$ is the probability of the remaining $K-K'$ models not giving the rank γ^*.

- **Union bound for total probability:** The event ζ_m occurs if a consensus is reached on any rank from 1 to Υ. We can use the union bound, also known as the Boole's inequality (Shehzadi et al. 2025), to find an upper bound for the total probability of a false positive under H_0,

$$\begin{aligned} P(\zeta_m|H_0) &= P\left(\bigcup_{\gamma^*=1}^{\Upsilon}\left\{C_{\gamma^*}>\frac{K}{2}\right\}\right) \\ &\le \sum_{\gamma^*=1}^{\Upsilon} P\left(C_{\gamma^*}>\frac{K}{2}\right) \end{aligned}.$$

  This simplifies to

$$P(\zeta_m|H_0) \le \Upsilon \cdot \left(\sum_{k'=\lfloor K/2\rfloor+1}^{K} \binom{K}{k'} p^{k'}(1-p)^{K-k'}\right).$$

- **Statistical significance:** Given that the number of top ranks $\Upsilon=\lfloor M/2\rfloor$ is typically much greater than one, the probability $p_{\gamma^*}=1/\Upsilon$ is a small value. For a small p_{γ^*}, the probability of achieving a majority consensus (more

Algorithms	Core idea
MO-MFEA (Gupta et al. 2017)	Implements automatic knowledge transfer via multifactorial optimization.
MFEA-DGD (Liu et al. 2024)	Employs diffusion gradient descent to accelerate task convergence.
BLKT-DE (Jiang et al. 2024)	Prevents negative knowledge transfer by sharing knowledge between similar dimensions.
MTDE-MKTA (Li and Gong 2025)	Uses adaptive schemes to manage multiple types of knowledge and their transfer.
MTES-KG (Li, Gong and Li 2024)	Incorporates external knowledge to guide the evolution of each task.

Table S2: Description of baseline algorithms

Algorithms	Parameter settings
MTGP-BS	*K* = 5
MO-MFEA	*RMP* = 0.3; *MuC* = 20; *MuM* = 15
MFEA-DGD	*RMP* = 0.3; *Gamma* = 0.1
BLKT-DE	*F* = 0.5; *CR* = 0.7
MTDE-MKTA	Tau_1 = 0.2; Tau_2= 0.1
MTES-KG	tau_0 = 2; *alpha* = 0.5; *adjGap* = 50; $sigma_0$ = 0.3

Table S3: List of parameters for all algorithms

than $K/2$ models) for any single rank is extremely low. Summing these extremely small probabilities over the Υ possible ranks still results in a very small upper bound for $P(\zeta_m|H_0)$.

- **Conclusion:** The probability of our selection rule being met by random chance is exceedingly low. Therefore, when a subexpression is selected, we can reject the null hypothesis and conclude with high confidence that this consensus is due to a genuine, strong signal of importance shared across the independent models.

F.3. Implications

This proof provides a rigorous statistical justification for our specific voting-based selection mechanism. It demonstrates that our criterion is not an arbitrary heuristic but acts as a powerful statistical filter against random noise.

The strength of this approach lies in its demand for a very strong form of agreement. Specifically, multiple independent models must not only concur that a subexpression is important in a vague sense, but they must agree on its precise relative standing among its peers. A consensus of this nature is highly improbable to occur by chance, ensuring that only subexpressions with the most robust and consistent signals of importance are selected. This provides a solid theoretical foundation for the subsequent synthesis of superior symbolic expressions.

Appendix G. Detailed Experimental Settings

This section provides the comprehensive setup details for the experiments conducted in the main paper.

Set	Task	MTGP-BS	BLKT-DE	MFEA-DGD	MO-MFEA	MTDE-MKTA	MTES-KG
Set 1	Task 1	**1.8E-3(±4.7E-3)**	4.0E-2(±1.2E-2)+	6.3E-2(±2.0E-2)+	3.9E-2(±1.0E-2)+	3.5E-2(±1.1E-2)+	4.2E-2(±1.1E-2)+
	Task 2	**1.3E-2(±1.2E-2)**	2.4E-2(±1.0E-2)+	4.9E-2(±2.6E-2)+	2.4E-2(±1.3E-2)+	1.9E-2(±8.4E-3)≈	3.7E-2(±2.3E-2)+
Set 2	Task 1	**0.0E+0(±0.0E+0)**	4.2E-3(±1.0E-2)+	3.6E-2(±2.2E-2)+	7.5E-3(±1.2E-2)+	2.0E-3(±7.9E-3)+	1.3E-2(±1.3E-2)+
	Task 2	2.9E-3(±6.0E-3)	2.7E-5(±2.5E-5)-	2.3E-2(±2.4E-2)+	2.1E-3(±5.5E-3)-	**2.3E-5(±6.4E-6)-**	1.6E-2(±1.6E-2)+
Set 3	Task 1	**6.5E-3(±1.0E-2)**	4.2E-2(±1.6E-2)+	5.3E-2(±1.8E-2)+	4.1E-2(±1.0E-2)+	3.5E-2(±1.4E-2)+	4.2E-2(±9.2E-3)+
	Task 2	**3.0E-2(±1.0E-2)**	5.4E-2(±1.0E-2)+	7.4E-2(±2.5E-2)+	5.2E-2(±1.2E-2)+	5.4E-2(±1.4E-2)+	5.9E-2(±9.3E-3)+
Set 4	Task 1	**0.0E+0(±0.0E+0)**	1.2E-2(±1.2E-2)+	4.1E-2(±2.9E-2)+	9.1E-3(±1.3E-2)+	5.6E-4(±2.4E-3)+	2.1E-2(±1.4E-2)+
	Task 2	**1.1E-1(±5.5E-2)**	1.5E-1(±4.3E-2)+	2.1E-1(±3.1E-2)+	1.7E-1(±2.8E-2)+	1.5E-1(±3.3E-2)+	2.0E-1(±3.1E-2)+
Set 5	Task 1	**0.0E+0(±0.0E+0)**	1.4E-2(±1.3E-2)+	3.1E-2(±2.4E-2)+	1.0E-2(±1.4E-2)+	3.4E-3(±8.1E-3)+	1.2E-2(±1.2E-2)+
	Task 2	**5.1E-2(±1.6E-2)**	8.6E-2(±1.5E-2)+	1.0E-1(±2.1E-2)+	9.4E-2(±1.8E-2)+	8.4E-2(±2.2E-2)+	9.6E-2(±1.8E-2)+
Set 6	Task 1	**2.9E-17(±9.4E-18)**	5.8E-3(±6.8E-3)+	2.6E-2(±2.0E-2)+	5.8E-3(±6.1E-3)+	2.7E-3(±5.1E-3)+	3.4E-3(±3.8E-3)+
	Task 2	1.5E-3(±2.6E-3)	**2.2E-5(±6.1E-6)≈**	2.3E-2(±2.6E-2)+	1.0E-3(±2.7E-3)-	1.1E-4(±4.7E-4)≈	1.6E-2(±2.5E-2)+
Set 7	Task 1	**1.3E-2(±1.3E-2)**	2.9E-2(±1.3E-2)+	5.3E-2(±2.6E-2)+	2.1E-2(±1.1E-2)+	2.1E-2(±1.4E-2)+	2.0E-2(±1.6E-2)+
	Task 2	**3.9E-2(±2.3E-2)**	5.4E-2(±9.9E-3)+	7.1E-2(±2.9E-2)+	5.4E-2(±9.6E-3)+	5.3E-2(±1.1E-2)+	5.5E-2(±1.0E-2)+
Set 8	Task 1	**3.0E-17(±1.0E-17)**	5.5E-3(±7.2E-3)+	3.3E-2(±2.2E-2)+	7.4E-3(±7.8E-3)+	2.8E-3(±4.1E-3)+	2.1E-3(±2.8E-3)+
	Task 2	**1.0E-1(±6.7E-2)**	1.4E-1(±5.1E-2)+	2.1E-1(±3.5E-2)+	1.6E-1(±3.7E-2)+	1.6E-1(±5.3E-2)+	1.8E-1(±3.0E-2)+
Set 9	Task 1	**2.9E-17(±9.6E-18)**	6.6E-3(±8.6E-3)+	2.7E-2(±2.0E-2)+	4.2E-3(±6.8E-3)+	1.3E-3(±2.5E-3)+	3.8E-3(±4.5E-3)+
	Task 2	**5.2E-2(±1.5E-2)**	8.6E-2(±1.4E-2)+	1.0E-1(±2.3E-2)+	8.8E-2(±1.8E-2)+	7.8E-2(±1.3E-2)+	1.0E-1(±1.8E-2)+
Set 10	Task 1	7.4E-3(±1.9E-2)	1.2E-3(±3.9E-3)-	1.7E-2(±2.1E-2)+	4.7E-4(±1.0E-3)-	**3.1E-5(±3.8E-5)≈**	1.8E-2(±1.9E-2)+
	Task 2	**1.8E-2(±1.9E-2)**	3.7E-2(±5.3E-3)+	4.5E-2(±1.5E-2)+	3.8E-2(±8.2E-3)+	3.5E-2(±7.5E-3)+	4.1E-2(±7.2E-3)+
Set 11	Task 1	2.1E-2(±6.4E-2)	**8.2E-5(±2.1E-4)≈**	1.3E-2(±1.5E-2)-	1.3E-4(±4.4E-4)≈	2.1E-4(±7.0E-4)≈	8.3E-3(±1.0E-2)-
	Task 2	**9.1E-2(±6.3E-2)**	1.4E-1(±5.6E-2)+	2.1E-1(±4.6E-2)+	1.7E-1(±3.0E-2)+	1.4E-1(±4.8E-2)+	1.8E-1(±3.6E-2)+
Set 12	Task 1	3.7E-3(±1.2E-2)	5.0E-4(±2.4E-3)-	1.6E-2(±2.1E-2)+	1.3E-3(±3.2E-3)-	**5.2E-5(±1.2E-4)-**	8.7E-3(±1.6E-2)+
	Task 2	**6.0E-2(±1.8E-2)**	8.4E-2(±1.3E-2)+	1.0E-1(±2.7E-2)+	8.7E-2(±1.8E-2)+	8.1E-2(±2.2E-2)+	1.0E-1(±1.8E-2)+
Set 13	Task 1	**1.6E-2(±2.8E-2)**	3.9E-2(±6.5E-3)+	4.8E-2(±1.6E-2)+	3.8E-2(±6.8E-3)+	3.3E-2(±6.3E-3)+	4.0E-2(±7.9E-3)+
	Task 2	**1.0E-1(±6.3E-2)**	1.3E-1(±6.2E-2)+	2.0E-1(±3.0E-2)+	1.8E-1(±4.0E-2)+	1.4E-1(±6.1E-2)+	1.8E-1(±2.4E-2)+
Set 14	Task 1	**3.1E-2(±2.8E-2)**	3.5E-2(±5.4E-3)+	4.6E-2(±1.4E-2)+	3.6E-2(±7.4E-3)+	3.7E-2(±6.9E-3)+	3.9E-2(±7.8E-3)+
	Task 2	**6.0E-2(±2.2E-2)**	8.5E-2(±2.0E-2)+	9.7E-2(±2.0E-2)+	8.8E-2(±2.0E-2)+	7.6E-2(±1.6E-2)+	9.7E-2(±1.8E-2)+
Set 15	Task 1	**1.4E-1(±1.1E-1)**	1.4E-1(±5.9E-2)≈	2.0E-1(±3.6E-2)+	1.7E-1(±2.1E-2)+	1.4E-1(±6.0E-2)≈	1.8E-1(±2.2E-2)+
	Task 2	**7.4E-2(±4.7E-2)**	7.9E-2(±1.5E-2)+	1.0E-1(±2.5E-2)+	8.2E-2(±1.5E-2)+	8.3E-2(±1.4E-2)+	9.6E-2(±1.3E-2)+
Set 16	Task 1	**2.6E-3(±1.1E-2)**	1.3E-2(±4.2E-3)+	3.6E-2(±1.5E-2)+	1.5E-2(±7.6E-3)+	1.1E-2(±4.6E-3)+	2.1E-2(±8.3E-3)+
	Task 2	**2.6E-3(±5.5E-3)**	3.0E-3(±8.0E-4)+	2.1E-2(±4.7E-2)+	3.3E-3(±3.7E-4)+	3.3E-3(±4.3E-4)+	3.3E-3(±3.0E-4)+
+/≈/-			26/3/3	31/0/1	27/1/4	25/5/2	31/0/1

Table S4: The mean and standard deviation of the *mse* for MTGP-BS compared with other five competing algorithms

G.1. Benchmark Problems

The 16 benchmark problems (Zhang et al. 2024), each comprising two tasks, are detailed in Table S1. These problems cover a variety of operators and variable counts. For each task, 100 data points are randomly sampled from the ground-truth expression for training and testing purposes.

G.2. Baseline Algorithms

The five state-of-the-art multitask algorithms used for comparison are described in Table S2. The core innovation and general parameter setting of each algorithm are summarized to provide context for the performance comparison.

G.3. Parameter Settings

All other algorithm-specific parameters are set to the values recommended in their original respective publications as shown in Table S3.

G.4. Computing Infrastructure

The experiments are conducted on a desktop computer with a 12th Gen Intel(R) Core (TM) i7-12700 CPU (2.10 GHz), and 32.0 GB of 3200 MHz RAM. The system ran on Windows 11 Pro (version 24H2). The software environment included MATLAB R2024b and Python 3.9.

Appendix H. Detailed Analysis of Learning Performance

H.1. Full Numerical Results

Table S4 presents the mean and standard deviation of the *mse* for all algorithms over 30 independent runs on the 16 MTSR problems. The best-performing algorithm on each task is marked in bold.

H.2. Detailed Analysis

As shown in Table S4, MTGP-BS demonstrates better performance on both tasks for 11 problems, where the expressions learned for these tasks are with the best accuracy compared to those learned by other algorithms. Moreover, 27 expressions from different problems learned by MTGP-BS are with the minimum *mse*, while 2, 0, 0, 3 and 0 expressions learned by BLKT-DE, MFEA-DGD, MO-MFEA, MTDE-MKTA and MTES-KG are with the minimum *mse*, respectively. Specifically, the performance of MTGP-BS is significantly better than BLKT-DE, MFEA-DGD, MO-MFEA, MTDE-MKTA and MTES-KG on 26, 31, 27, 25 and 31 tasks, respectively, highlighting the effectiveness of MTGP-BS. While MTGP-BS demonstrated overall superiority, an in-depth analysis of the few cases where it is outperformed, offering insights into the mechanisms of competing algorithms.

With BLKT-DE

MTGP-BS is outperformed by BLKT-DE on five tasks. This performance difference stems from the core design of BLKT-DE, which implements a conservative strategy specifically to prevent negative transfer when tasks are dissimilar. The tasks in Set 10 serve as an excellent example. The target expression for Task 1 is a polynomial expression $100x_1^4 - 200x_1^2x_2 + x_1^2 - 2x_1 + 100x_2^2 + 1$, while Task 2 is trigonometric expressions $x_1^2 + x_2^2 - 10cos\,(x_1) - 10cos\,(x_2) + 20$. In this case, transferring trigonometric components like cos (x_1) from the second task to the first is detrimental. BLKT-DE effectively avoids this pitfall by restricting knowledge transfer, thereby protecting the solution. In contrast, the comprehensive nature of the transfer mechanism in our post-hoc refinement framework, while beneficial in most scenarios, led to the integration of these incompatible subexpressions, resulting in a compromised final expression.

Compared with MTDE-MKTA

MTGP-BS is outperformed by MTDE-MKTA on two tasks, such as in Set 2. This problem involves two structurally related polynomials, but one is far more complex. Task 1 is a simple convex function of the form $x_1^2 + 2x_2^2$, while Task 2 is a highly complex polynomial given by $100x_1^4 - 200x_1^2x_2 + x_1^2 - 2x_1 + 100x_2^2 + 1$.The adaptive schemes in MTDE-MKTA allow it to identify foundational terms like x_1^2 and x_2^2 from the simpler task and integrate them during the evolutionary search for the more complex task. This focuses the search effort effectively. In contrast, our post-hoc framework applies knowledge transfer after the optimization is complete. In some specific cases, when more complex structures like$x_1^2x_2$ are not learned for this task, there is no opportunity to explore them that build upon these foundational terms during the evolution itself with the information from the relevant tasks. This failure represents an exception arising under specific structural conditions where early-stage knowledge integration is crucial.

Compare with Other Baselines

MTGP-BS underperforms MFEA-DGD and MTES-KG on only one task each, and MO-MFEA on four tasks. These minor victories for the baselines are attributed to their specific mechanisms, such as accelerated convergence via diffusion gradient descent (MFEA-DGD) or the application of highly specific external knowledge (MTES-KG). However, these algorithms only achieve superior accuracy on a very small subset of tasks, whereas MTGP-BS, by effectively integrating knowledge from the entire population, discovers better expressions across the vast majority of problems, confirming its broad effectiveness.

H.3. Conclusion

We compare the performance of MTGP-BS against five competing multitask algorithms. It can be observed that

MTGP-BS achieves statistically significant performance against all competitors on the vast majority of tasks, demonstrating its superior accuracy.

Appendix I. Detailed Qualitative Analysis of Discovered Expressions

This section provides a detailed, case-by-case qualitative analysis of the expressions discovered by each algorithm on the three representative problems, Sets 2, 3, and 16.

I.1. Rationale for the Selection of Representative Problems

To provide a comprehensive qualitative evaluation, we select three problem sets, each with distinct characteristics of the MTSR problems. These tasks exhibit a clear diversity in structure and complexity, forming a solid foundation for evaluating SR performance.

- **Set 2** consists of purely polynomial expressions of varying complexity, ranging from simple quadratic forms to high-order polynomials with coupled variables.
- **Set 3** extends the challenge to higher dimensions, involving three-variable expressions that integrate both polynomial and trigonometric components.
- **Set 16** features expressions that combine polynomial terms with trigonometric components, and notably includes a term with an extremely small coefficient.

I.2. Case-by-Case Analysis of Discovered Expressions

Analysis on problem Set 2 (polynomials)

For the purely polynomial problem Set 2, MFEA-DGD and MTES-KG learned some erroneous substructures in the first task. While BLKT-DE discovered the correct expression structure, it includes a redundant constant term. In contrast, MTGP-BS, along with MO-MFEA and MTDE-MKTA, successfully learn the correct quadratic expression $x_1^2 + 2x_2^2$. This indicates comparable performance among these three algorithms for simpler structures. However, in the second task, which involves higher-order terms like x_1^4, more complex cross-terms such as $x_1^2x_2$, and other substructures, the expression $0.26x_2 - 200x_1^2x_2 + 0.67x_1^2 + 100x_2^2 + 100x_1^4 - 1.7$ learned by MTGP-BS exhibits the closest structure to the true expression $100x_1^4 - 200x_1^2x_2 + x_1^2 - 2x_1 + 100x_2^2 + 1$. This is achieved with only a minor error of mistakenly identifying x_1 as x_2 in one term. For the other five comparative algorithms, two or more substructures are lacked in their learned expression compared to the true expression. This result suggests that the consensus-accelerated Shapley analysis designed in MTGP-BS enables it to identify and integrate structures embedding valuable knowledge, thereby reducing the loss of valuable subexpressions effectively.

Set	Task	MTGP-BS	MTGP-S
Set 1	Task 1	**1.8E-3(±4.7E-3)**	1.8E-2(±1.8E-2)+
	Task 2	**1.3E-2(±1.2E-2)**	1.9E-2(±1.0E-2)≈
Set 2	Task 1	**0.0E+0(±0.0E+0)**	0.0E+0(±0.0E+0)≈
	Task 2	**2.9E-3(±6.0E-3)**	3.2E-3(±5.4E-3)+
Set 3	Task 1	**6.5E-3(±1.0E-2)**	8.1E-3(±9.3E-3)≈
	Task 2	**3.0E-2(±1.0E-2)**	3.7E-2(±1.9E-2)+
Set 4	Task 1	**0.0E+0(±0.0E+0)**	0.0E+0(±0.0E+0)≈
	Task 2	**1.1E-1(±5.5E-2)**	4.0E-1(±6.9E-1)+
Set 5	Task 1	**0.0E+0(±0.0E+0)**	0.0E+0(±0.0E+0)≈
	Task 2	**5.1E-2(±1.6E-2)**	1.0E-1(±8.5E-2)+
Set 6	Task 1	**2.9E-17(±9.4E-18)**	9.7E-4(±5.3E-3)≈
	Task 2	**1.5E-3(±2.6E-3)**	5.7E-3(±7.4E-3)+
Set 7	Task 1	**1.3E-2(±1.3E-2)**	1.8E-2(±2.7E-2)≈
	Task 2	**3.9E-2(±2.3E-2)**	6.4E-2(±7.1E-2)+
Set 8	Task 1	**3.0E-17(±1.0E-17)**	3.2E-17(±1.0E-17)≈
	Task 2	**1.0E-1(±6.7E-2)**	9.0E-1(±2.1E+0)+
Set 9	Task 1	**2.9E-17(±9.6E-18)**	2.9E-17(±1.1E-17)≈
	Task 2	**5.2E-2(±1.5E-2)**	1.2E-1(±1.7E-1)+
Set 10	Task 1	**7.4E-3(±1.9E-2)**	1.1E-2(±2.1E-2)+
	Task 2	**1.8E-2(±1.9E-2)**	1.8E-2(±1.5E-2)≈
Set 11	Task 1	**2.1E-2(±6.4E-2)**	1.9E-1(±8.7E-1)≈
	Task 2	**9.1E-2(±6.3E-2)**	7.3E-1(±2.6E+0)+
Set 12	Task 1	**3.7E-3(±1.2E-2)**	6.5E-3(±1.6E-2)≈
	Task 2	**6.0E-2(±1.8E-2)**	8.5E-2(±2.8E-2)+
Set 13	Task 1	1.6E-2(±2.8E-2)	**1.1E-2(±3.4E-2)-**
	Task 2	**1.0E-1(±6.3E-2)**	2.5E-1(±1.7E-1)+
Set 14	Task 1	3.1E-2(±2.8E-2)	**2.3E-2(±2.6E-2)≈**
	Task 2	**6.0E-2(±2.2E-2)**	1.0E-1(±8.3E-2)+
Set 15	Task 1	**1.4E-1(±1.1E-1)**	8.9E-1(±1.8E+0)+
	Task 2	**7.4E-2(±4.7E-2)**	1.6E-1(±3.8E-1)≈
Set 16	Task 1	**2.6E-3(±1.1E-2)**	1.1E-2(±6.1E-2)≈
	Task 2	**2.6E-3(±5.5E-3)**	7.4E-1(±2.7E+0)+
+/≈/-			16/15/1

Table S5: Results of *mse* for MTGP-BS and MTGP-S

Analysis on problem Set 3 (three variables)

MTGP-BS distinctly outperforms other comparative algorithms when addressing problem Set 3, which involves three variables. In the first task, MTGP-BS successfully learns the correct three-variable quadratic expression $x_1^2 + 2x_2^2 + 3x_3^2$, whereas all five comparative algorithms produce structurally incorrect expressions. For the second task, involving a hybrid expression with both polynomial and trigonometric subexpressions, no algorithm completely learns the correct expression. The difficulty in learning this expression stems primarily from two factors. First, the search space is expanded caused by the presence of three variables, increasing the difficulty of finding the correct expression. Second, the challenging term $-10cos\ (x_1) - 10cos\ (x_2) - 10cos\ (x_3)$ in the second task is not part of the shared knowledge between the two tasks. Thus, the information-sharing mechanisms designed in conventional multitask evolutionary algorithms have limited ability to enhance the learning of these

Set	Task	MTGP-BS	MTGP-B1	MTGP-B2	MTGP-B3
Set 1	Task 1	**1.8E-3(±4.7E-3)**	9.2E-2(±5.5E-2)+	7.7E-2(±5.2E-2)+	4.8E-2(±4.9E-2)+
	Task 2	**1.3E-2(±1.2E-2)**	4.7E-2(±2.7E-2)+	5.1E-2(±3.1E-2)+	4.1E-2(±3.7E-2)+
Set 2	Task 1	**0.0E+0(±0.0E+0)**	0.0E+0(±0.0E+0)≈	0.0E+0(±0.0E+0)≈	0.0E+0(±0.0E+0)≈
	Task 2	**2.9E-3(±6.0E-3)**	2.2E-2(±2.0E-2)+	2.3E-2(±2.7E-2)+	1.2E-2(±2.0E-2)+
Set 3	Task 1	**6.5E-3(±1.0E-2)**	8.0E-2(±6.3E-2)+	7.3E-2(±6.0E-2)+	4.9E-2(±5.6E-2)+
	Task 2	**3.0E-2(±1.0E-2)**	9.8E-2(±3.8E-2)+	9.8E-2(±3.8E-2)+	6.7E-2(±5.0E-2)+
Set 4	Task 1	**0.0E+0(±0.0E+0)**	0.0E+0(±0.0E+0)≈	0.0E+0(±0.0E+0)≈	0.0E+0(±0.0E+0)≈
	Task 2	**1.1E-1(±5.5E-2)**	1.5E-1(±4.6E-2)+	1.5E-1(±7.6E-2)+	1.5E-1(±5.1E-2)+
Set 5	Task 1	**0.0E+0(±0.0E+0)**	6.5E-3(±2.5E-2)≈	6.6E-3(±2.5E-2)≈	0.0E+0(±0.0E+0)≈
	Task 2	**5.1E-2(±1.6E-2)**	8.2E-2(±3.0E-2)+	8.8E-2(±3.6E-2)+	1.0E-1(±5.2E-2)+
Set 6	Task 1	**2.9E-17(±9.4E-18)**	2.1E-3(±8.3E-3)≈	8.4E-3(±3.7E-2)≈	4.3E-3(±1.7E-2)≈
	Task 2	**1.5E-3(±2.6E-3)**	2.8E-2(±3.4E-2)+	2.0E-2(±2.3E-2)+	1.9E-2(±4.0E-2)+
Set 7	Task 1	**1.3E-2(±1.3E-2)**	5.4E-2(±4.2E-2)+	4.8E-2(±3.3E-2)+	4.9E-2(±3.9E-2)+
	Task 2	**3.9E-2(±2.3E-2)**	8.9E-2(±3.7E-2)+	9.2E-2(±3.8E-2)+	7.2E-2(±4.6E-2)+
Set 8	Task 1	**3.0E-17(±1.0E-17)**	3.0E-17(±1.0E-17)≈	3.0E-17(±1.0E-17)≈	3.0E-17(±1.0E-17)≈
	Task 2	**1.0E-1(±6.7E-2)**	1.4E-1(±5.9E-2)+	1.5E-1(±6.8E-2)+	1.5E-1(±5.5E-2)+
Set 9	Task 1	**2.9E-17(±9.6E-18)**	7.8E-3(±4.3E-2)≈	2.9E-17(±9.7E-18)≈	9.8E-4(±5.3E-3)≈
	Task 2	**5.2E-2(±1.5E-2)**	8.8E-2(±3.6E-2)+	9.2E-2(±3.1E-2)+	1.0E-1(±4.2E-2)+
Set 10	Task 1	**7.4E-3(±1.9E-2)**	2.7E-2(±3.1E-2)+	1.3E-2(±9.2E-3)+	4.8E-2(±6.8E-2)+
	Task 2	**1.8E-2(±1.9E-2)**	7.3E-2(±4.5E-2)+	8.6E-2(±5.1E-2)+	5.3E-2(±4.9E-2)+
Set 11	Task 1	**2.1E-2(±6.4E-2)**	4.0E-2(±6.6E-2)+	2.3E-1(±1.1E+0)+	2.9E-2(±6.2E-2)+
	Task 2	**9.1E-2(±6.3E-2)**	1.2E-1(±6.0E-2)+	1.2E-1(±6.1E-2)+	1.2E-1(±5.6E-2)+
Set 12	Task 1	**3.7E-3(±1.2E-2)**	2.7E-2(±4.0E-2)+	1.6E-2(±2.9E-2)+	3.4E-2(±5.5E-2)+
	Task 2	**6.0E-2(±1.8E-2)**	8.6E-2(±2.0E-2)+	8.6E-2(±2.1E-2)+	1.0E-1(±4.2E-2)+
Set 13	Task 1	**1.6E-2(±2.8E-2)**	6.5E-2(±5.0E-2)+	6.3E-2(±4.4E-2)+	9.5E-2(±3.0E-1)+
	Task 2	**1.0E-1(±6.3E-2)**	1.5E-1(±7.5E-2)+	1.9E-1(±1.8E-1)+	1.5E-1(±6.1E-2)+
Set 14	Task 1	**3.1E-2(±2.8E-2)**	8.1E-2(±5.2E-2)+	8.3E-2(±4.8E-2)+	6.3E-2(±5.0E-2)+
	Task 2	**6.0E-2(±2.2E-2)**	9.1E-2(±2.5E-2)+	9.7E-2(±3.7E-2)+	1.0E-1(±5.0E-2)+
Set 15	Task 1	**1.4E-1(±1.1E-1)**	2.2E-1(±2.6E-1)+	1.7E-1(±1.6E-1)≈	1.6E-1(±8.7E-2)≈
	Task 2	**7.4E-2(±4.7E-2)**	9.5E-2(±4.5E-2)+	1.0E-1(±4.1E-2)+	1.1E-1(±5.4E-2)+
Set 16	Task 1	**2.6E-3(±1.1E-2)**	2.7E-1(±1.3E+0)+	3.8E-2(±3.6E-2)+	5.6E-2(±7.3E-2)+
	Task 2	**2.6E-3(±5.5E-3)**	1.0E-1(±4.9E-1)+	3.0E-2(±1.3E-1)+	3.1E-2(±8.3E-2)≈
+/≈/-			26/6/0	25/7/0	24/8/0

Table S6: Results of *mse* for MTGP-BS compared with MTGP-B1, MTGP-B2, and MTGP-B3

non-shared components. Despite these challenges, expressions learned by MTGP-BS and BLKT-DE are with the optimal *mse* for this task. A closer analysis of the expression structures reveals that MTGP-BS learns more valid substructures compared to BLKT-DE, whose learned substructures are notably incorrect. In addition, when compared to expressions with lower precision, the expression structure learned by MTGP-BS is indeed the closest to the true expression. This highlights the advantage of MTGP-BS in multi-variable cooperative optimization and valuable structural component identification, allowing it to manage the complexities arising more effectively from increased variable dimensionality.

Analysis on Problem Set 16 (Subtle Features)

Problem Set 16 further validates the capability of MTGP-BS in handling expressions with subtle features. Among all algorithms selected in this work, MTGP-BS is the only one that successfully reconstructs the correct expressions for both tasks. Particularly in the second task, a quadratic term $x_1^2/4000$ with an extremely small coefficient contained is contained in the correct expression. This minute quadratic term is successfully identified and included by MTGP-BS, whereas most comparative algorithms failed to capture it, typically only fitting the trigonometric component. Concurrently, other algorithms learned very complex incorrect expressions for the first task, while the correct expression can by learned by MTGP-BS. This indicates that the consensus-accelerated Shapley analysis and bidirectional subexpression cooperative extraction method in MTGP-BS allows it to outperform other comparative algorithms in terms of precision, structural integrity, and the ability to capture subtle valuable knowledge, thereby demonstrating strong reliability, especially when tackling complex and diverse symbolic regression tasks.

I.3. Conclusion

To inspect the structural integrity of solutions, we select three representative problems as introduced in Subsection I.1. We compare the median-run expressions from each al-

Set	Task	MTGP-BS	MTGP	BLKT-DE-BS	BLKT-DE	MFEA-DGD-BS	MFEA-DGD
Set 1	Task 1	**1.8E-3(±4.7E-3)**	2.2E-2(±3.7E-3)+	**2.6E-3(±7.1E-3)**	4.0E-2(±1.2E-2)+	**2.3E-2(±2.4E-2)**	6.3E-2(±2.0E-2)+
	Task 2	1.3E-2(±1.2E-2)	**1.0E-2(±3.2E-3)≈**	**1.0E-2(±8.4E-3)**	2.4E-2(±1.0E-2)+	**1.7E-2(±1.2E-2)**	4.9E-2(±2.6E-2)+
Set 2	Task 1	**0.0E+0(±0.0E+0)**	0.0E+0(±0.0E+0)≈	**0.0E+0(±0.0E+0)**	4.2E-3(±1.0E-2)+	**7.7E-3(±1.3E-2)**	3.6E-2(±2.2E-2)+
	Task 2	2.9E-3(±6.0E-3)	**2.4E-5(±2.8E-6)-**	**3.1E-6(±1.6E-6)**	2.7E-5(±2.5E-5)+	**8.5E-3(±1.3E-2)**	2.3E-2(±2.4E-2)+
Set 3	Task 1	**6.5E-3(±1.0E-2)**	2.2E-2(±6.9E-3)+	**4.9E-3(±7.2E-3)**	4.2E-2(±1.6E-2)+	**1.3E-2(±1.5E-2)**	5.3E-2(±1.8E-2)+
	Task 2	**3.0E-2(±1.0E-2)**	4.1E-2(±8.4E-3)+	**3.9E-2(±8.7E-3)**	5.4E-2(±1.0E-2)+	**4.6E-2(±1.3E-2)**	7.4E-2(±2.5E-2)+
Set 4	Task 1	**0.0E+0(±0.0E+0)**	0.0E+0(±0.0E+0)≈	**3.2E-3(±9.0E-3)**	1.2E-2(±1.2E-2)+	**8.8E-3(±1.6E-2)**	4.1E-2(±2.9E-2)+
	Task 2	**1.1E-1(±5.5E-2)**	1.7E-1(±1.9E-2)+	**1.3E-1(±4.4E-2)**	1.5E-1(±4.3E-2)+	**2.1E-1(±2.2E-2)**	2.1E-1(±3.1E-2)≈
Set 5	Task 1	**0.0E+0(±0.0E+0)**	8.9E-5(±4.9E-4)≈	**4.7E-4(±2.5E-3)**	1.4E-2(±1.3E-2)+	**5.7E-3(±1.0E-2)**	3.1E-2(±2.4E-2)+
	Task 2	**5.1E-2(±1.6E-2)**	7.2E-2(±1.0E-2)+	**5.6E-2(±1.4E-2)**	8.6E-2(±1.5E-2)+	**9.0E-2(±1.8E-2)**	1.0E-1(±2.1E-2)+
Set 6	Task 1	**2.9E-17(±9.4E-18)**	8.2E-4(±2.1E-3)≈	**2.8E-17(±7.8E-18)**	5.8E-3(±6.8E-3)+	**1.6E-2(±1.7E-2)**	2.6E-2(±2.0E-2)≈
	Task 2	1.5E-3(±2.6E-3)	**2.8E-5(±1.2E-5)≈**	**3.1E-6(±2.5E-6)**	2.2E-5(±6.1E-6)+	**1.2E-2(±1.8E-2)**	2.3E-2(±2.6E-2)+
Set 7	Task 1	1.3E-2(±1.3E-2)	**1.0E-2(±2.8E-3)≈**	**8.3E-3(±8.8E-3)**	2.9E-2(±1.3E-2)+	**1.9E-2(±1.2E-2)**	5.3E-2(±2.6E-2)+
	Task 2	**3.9E-2(±2.3E-2)**	4.2E-2(±5.7E-3)+	**3.8E-2(±1.0E-2)**	5.4E-2(±9.9E-3)+	**5.0E-2(±9.8E-3)**	7.1E-2(±2.9E-2)+
Set 8	Task 1	**3.0E-17(±1.0E-17)**	3.0E-17(±1.0E-17)≈	**1.3E-3(±5.0E-3)**	5.5E-3(±7.2E-3)+	**5.5E-3(±1.2E-2)**	3.3E-2(±2.2E-2)+
	Task 2	**1.0E-1(±6.7E-2)**	1.6E-1(±2.0E-2)+	**1.2E-1(±3.0E-2)**	1.4E-1(±5.1E-2)+	**2.0E-1(±1.8E-2)**	2.1E-1(±3.5E-2)≈
Set 9	Task 1	**2.9E-17(±9.6E-18)**	3.2E-4(±1.2E-3)≈	**1.2E-4(±6.6E-4)**	6.6E-3(±8.6E-3)+	**3.6E-3(±7.9E-3)**	2.7E-2(±2.0E-2)+
	Task 2	**5.2E-2(±1.5E-2)**	7.0E-2(±1.4E-2)+	**5.3E-2(±1.1E-2)**	8.6E-2(±1.4E-2)+	**9.1E-2(±2.7E-2)**	1.0E-1(±2.3E-2)≈
Set 10	Task 1	7.4E-3(±1.9E-2)	**1.7E-5(±9.6E-6)≈**	**5.4E-4(±2.0E-3)**	1.2E-3(±3.9E-3)+	**1.6E-2(±2.0E-2)**	1.7E-2(±2.1E-2)≈
	Task 2	**1.8E-2(±1.9E-2)**	3.3E-2(±4.9E-3)+	**2.6E-2(±9.6E-3)**	3.7E-2(±5.3E-3)+	5.1E-2(±2.2E-2)	**4.5E-2(±1.5E-2)≈**
Set 11	Task 1	2.1E-2(±6.4E-2)	**1.9E-5(±9.3E-6)-**	1.3E-4(±4.8E-4)	**8.2E-5(±2.1E-4)≈**	**9.7E-3(±9.4E-3)**	1.3E-2(±1.5E-2)≈
	Task 2	**9.1E-2(±6.3E-2)**	1.7E-1(±1.8E-2)+	**1.2E-1(±3.0E-2)**	1.4E-1(±5.6E-2)+	**2.0E-1(±1.7E-2)**	2.1E-1(±4.6E-2)≈
Set 12	Task 1	3.7E-3(±1.2E-2)	**8.4E-5(±2.5E-4)≈**	**4.3E-4(±1.3E-3)**	5.0E-4(±2.4E-3)≈	**5.0E-3(±7.0E-3)**	1.6E-2(±2.1E-2)+
	Task 2	**6.0E-2(±1.8E-2)**	7.9E-2(±7.0E-3)+	**5.7E-2(±1.7E-2)**	8.4E-2(±1.3E-2)+	**9.3E-2(±1.9E-2)**	1.0E-1(±2.7E-2)≈
Set 13	Task 1	**1.6E-2(±2.8E-2)**	2.8E-2(±6.5E-3)+	**3.1E-2(±1.1E-2)**	3.9E-2(±6.5E-3)+	**3.0E-2(±1.8E-2)**	4.8E-2(±1.6E-2)+
	Task 2	**1.0E-1(±6.3E-2)**	1.6E-1(±1.8E-2)+	**1.3E-1(±3.8E-2)**	1.3E-1(±6.2E-2)≈	**2.0E-1(±1.8E-2)**	2.0E-1(±3.0E-2)≈
Set 14	Task 1	**3.1E-2(±2.8E-2)**	3.3E-2(±8.3E-3)+	**2.6E-2(±1.1E-2)**	3.5E-2(±5.4E-3)+	**2.6E-2(±1.6E-2)**	4.6E-2(±1.4E-2)+
	Task 2	**6.0E-2(±2.2E-2)**	7.1E-2(±1.0E-2)+	**6.0E-2(±1.8E-2)**	8.5E-2(±2.0E-2)+	**9.0E-2(±2.2E-2)**	9.7E-2(±2.0E-2)≈
Set 15	Task 1	**1.4E-1(±1.1E-1)**	1.7E-1(±2.6E-2)+	**1.1E-1(±5.0E-2)**	1.4E-1(±5.9E-2)+	**1.9E-1(±3.6E-2)**	2.0E-1(±3.6E-2)+
	Task 2	7.4E-2(±4.7E-2)	**6.9E-2(±2.2E-2)≈**	**6.7E-2(±2.2E-2)**	7.9E-2(±1.5E-2)+	**8.3E-2(±3.1E-2)**	1.0E-1(±2.5E-2)+
Set 16	Task 1	**2.6E-3(±1.1E-2)**	1.2E-2(±2.2E-3)+	**0.0E+0(±0.0E+0)**	1.3E-2(±4.2E-3)+	1.7E-1(±6.8E-2)	**3.6E-2(±1.5E-2)-**
	Task 2	**2.6E-3(±5.5E-3)**	3.5E-3(±7.2E-4)+	**0.0E+0(±0.0E+0)**	3.0E-3(±8.0E-4)+	**1.7E-2(±4.3E-2)**	2.1E-2(±4.7E-2)≈
+/≈/-			18/12/2		29/3/0		19/12/1

Table S7: Results of *mse* for MTGP-BS, BLKT-DE and MFEA-DGD compared with their variants with post-hoc refinement framework

gorithm, revealing the superiority of MTGP-BS in discovering expressions that are both highly accurate and structurally sound.

Appendix J. Detailed Ablation Study for Bidirectional Extraction

J.1. Full Numerical Results

Table S5 presents the mean and standard deviation of the *mse* for MTGP-BS and its ablated variant, MTGP-S, over 30 independent runs.

J.2. Detailed Case-Study Analysis

While the overall results demonstrate the superiority of MTGP-BS, a detailed analysis provides deeper insights into why the framework succeeds and where its limitations lie. To provide a more intuitive comparison, the second task in problem Set 5 is selected as an illustrative example. As shown in the function mapping curves in Figure 4(a) of the main text, the curves produced by MTGP-BS are significantly closer to the true function. In contrast, MTGP-S, which lacks the extraction method, is more susceptible to be influenced by a large, redundant set of subexpressions. This leads to a less effective SHAP analysis and ultimately, the construction of expressions that deviate more significantly from the ground truth. This case highlights the effectiveness of the bidirectional subexpression cooperative extraction method in creating a clean, high-quality archive for subsequent analysis.

Worse average *mse* values are obtained by MTGP-BS than by MTGP-S on Sets 13 and 14. This is likely caused by the failure of the extraction method to eliminate certain redundant subexpressions, compounded by the introduction of excessive irrelevant knowledge from the interconnected task. In such scenarios, the challenge for the subsequent SHAP analysis is inadvertently increased, leading to a higher probability of mistakenly identifying noisy or unhelpful subexpressions as important. In contrast, MTGP-S, by forgoing

Set	Task	MO-MFEA-BS	MO-MFEA	MTDE-MKTA-BS	MTDE-MKTA	MTES-KG-BS	MTES-KG
Set 1	Task 1	**7.2E-4(±3.9E-3)**	3.9E-2(±1.0E-2)+	**8.3E-4(±3.1E-3)**	3.5E-2(±1.1E-2)+	**1.4E-3(±5.6E-3)**	4.2E-2(±1.1E-2)+
	Task 2	**1.0E-2(±9.2E-3)**	2.4E-2(±1.3E-2)+	**7.0E-3(±6.4E-3)**	1.9E-2(±8.4E-3)+	**7.7E-3(±7.0E-3)**	3.7E-2(±2.3E-2)+
Set 2	Task 1	**0.0E+0(±0.0E+0)**	7.5E-3(±1.2E-2)+	**0.0E+0(±0.0E+0)**	2.0E-3(±7.9E-3)+	**0.0E+0(±0.0E+0)**	1.3E-2(±1.3E-2)+
	Task 2	**3.2E-6(±2.0E-6)**	2.1E-3(±5.5E-3)+	**4.0E-6(±9.9E-7)**	2.3E-5(±6.4E-6)+	**1.6E-3(±2.6E-3)**	1.6E-2(±1.6E-2)+
Set 3	Task 1	**2.4E-3(±5.5E-3)**	4.1E-2(±1.0E-2)+	**4.6E-3(±6.7E-3)**	3.5E-2(±1.4E-2)+	**2.2E-3(±5.3E-3)**	4.2E-2(±9.2E-3)+
	Task 2	**3.8E-2(±8.8E-3)**	5.2E-2(±1.2E-2)+	**3.7E-2(±7.9E-3)**	5.4E-2(±1.4E-2)+	**4.0E-2(±7.7E-3)**	5.9E-2(±9.3E-3)+
Set 4	Task 1	**5.4E-3(±1.1E-2)**	9.1E-3(±1.3E-2)+	2.0E-3(±6.2E-3)	**5.6E-4(±2.4E-3)-**	**2.5E-3(±7.8E-3)**	2.1E-2(±1.4E-2)+
	Task 2	**1.4E-1(±4.2E-2)**	1.7E-1(±2.8E-2)+	**1.1E-1(±4.0E-2)**	1.5E-1(±3.3E-2)+	**1.6E-1(±5.2E-2)**	2.0E-1(±3.1E-2)+
Set 5	Task 1	**4.6E-4(±2.5E-3)**	1.0E-2(±1.4E-2)+	**0.0E+0(±0.0E+0)**	3.4E-3(±8.1E-3)+	**3.1E-3(±8.2E-3)**	1.2E-2(±1.2E-2)+
	Task 2	**6.8E-2(±1.8E-2)**	9.4E-2(±1.8E-2)+	**5.3E-2(±1.2E-2)**	8.4E-2(±2.2E-2)+	**7.2E-2(±2.1E-2)**	9.6E-2(±1.8E-2)+
Set 6	Task 1	**3.0E-17(±9.0E-18)**	5.8E-3(±6.1E-3)+	**2.8E-17(±8.4E-18)**	2.7E-3(±5.1E-3)+	**1.7E-4(±6.6E-4)**	3.4E-3(±3.8E-3)+
	Task 2	**3.1E-5(±1.4E-4)**	1.0E-3(±2.7E-3)+	**7.4E-5(±2.9E-4)**	1.1E-4(±4.7E-4)+	**1.4E-3(±1.6E-3)**	1.6E-2(±2.5E-2)+
Set 7	Task 1	**1.0E-2(±7.0E-3)**	2.1E-2(±1.1E-2)+	**7.9E-3(±6.0E-3)**	2.1E-2(±1.4E-2)+	**9.3E-3(±8.5E-3)**	2.0E-2(±1.6E-2)+
	Task 2	**4.0E-2(±6.6E-3)**	5.4E-2(±9.6E-3)+	**3.8E-2(±8.5E-3)**	5.3E-2(±1.1E-2)+	**3.8E-2(±1.2E-2)**	5.5E-2(±1.0E-2)+
Set 8	Task 1	**4.7E-4(±2.5E-3)**	7.4E-3(±7.8E-3)+	**3.0E-17(±8.5E-18)**	2.8E-3(±4.1E-3)+	2.5E-3(±8.8E-3)	**2.1E-3(±2.8E-3)-**
	Task 2	**1.4E-1(±3.8E-2)**	1.6E-1(±3.7E-2)+	**1.3E-1(±3.1E-2)**	1.6E-1(±5.3E-2)+	**1.4E-1(±2.9E-2)**	1.8E-1(±3.0E-2)+
Set 9	Task 1	**8.5E-4(±3.6E-3)**	4.2E-3(±6.8E-3)+	**5.1E-4(±2.8E-3)**	1.3E-3(±2.5E-3)+	**1.5E-3(±5.5E-3)**	3.8E-3(±4.5E-3)+
	Task 2	**5.9E-2(±1.5E-2)**	8.8E-2(±1.8E-2)+	**5.3E-2(±1.4E-2)**	7.8E-2(±1.3E-2)+	**7.8E-2(±1.6E-2)**	1.0E-1(±1.8E-2)+
Set 10	Task 1	1.2E-3(±4.3E-3)	**4.7E-4(±1.0E-3)-**	2.5E-4(±1.0E-3)	**3.1E-5(±3.8E-5)-**	**3.1E-3(±6.2E-3)**	1.8E-2(±1.9E-2)+
	Task 2	**2.4E-2(±1.2E-2)**	3.8E-2(±8.2E-3)+	**2.1E-2(±1.0E-2)**	3.5E-2(±7.5E-3)+	**3.4E-2(±1.0E-2)**	4.1E-2(±7.2E-3)+
Set 11	Task 1	1.6E-3(±4.2E-3)	**1.3E-4(±4.4E-4)-**	1.3E-3(±3.9E-3)	**2.1E-4(±7.0E-4)≈**	**8.1E-3(±1.1E-2)**	8.3E-3(±1.0E-2)≈
	Task 2	**1.5E-1(±4.3E-2)**	1.7E-1(±3.0E-2)+	**1.4E-1(±4.0E-2)**	1.4E-1(±4.8E-2)≈	**1.3E-1(±3.3E-2)**	1.8E-1(±3.6E-2)+
Set 12	Task 1	**1.8E-4(±7.3E-4)**	1.3E-3(±3.2E-3)≈	1.7E-3(±6.8E-3)	**5.2E-5(±1.2E-4)≈**	**4.1E-3(±6.1E-3)**	8.7E-3(±1.6E-2)≈
	Task 2	**5.8E-2(±1.3E-2)**	8.7E-2(±1.8E-2)+	**5.5E-2(±2.0E-2)**	8.1E-2(±2.2E-2)+	**8.0E-2(±1.7E-2)**	1.0E-1(±1.8E-2)+
Set 13	Task 1	**2.8E-2(±1.7E-2)**	3.8E-2(±6.8E-3)+	**2.3E-2(±1.1E-2)**	3.3E-2(±6.3E-3)+	**3.3E-2(±1.3E-2)**	4.0E-2(±7.9E-3)+
	Task 2	**1.2E-1(±3.4E-2)**	1.8E-1(±4.0E-2)+	**1.3E-1(±3.1E-2)**	1.4E-1(±6.1E-2)+	**1.4E-1(±4.4E-2)**	1.8E-1(±2.4E-2)+
Set 14	Task 1	**2.4E-2(±1.2E-2)**	3.6E-2(±7.4E-3)+	**2.0E-2(±1.4E-2)**	3.7E-2(±6.9E-3)+	**3.8E-2(±6.3E-3)**	3.9E-2(±7.8E-3)≈
	Task 2	**6.0E-2(±1.5E-2)**	8.8E-2(±2.0E-2)+	**5.5E-2(±1.7E-2)**	7.6E-2(±1.6E-2)+	**7.6E-2(±2.2E-2)**	9.7E-2(±1.8E-2)+
Set 15	Task 1	**1.3E-1(±4.2E-2)**	1.7E-1(±2.1E-2)+	**1.1E-1(±4.5E-2)**	1.4E-1(±6.0E-2)+	**1.5E-1(±4.7E-2)**	1.8E-1(±2.2E-2)+
	Task 2	**6.1E-2(±1.8E-2)**	8.2E-2(±1.5E-2)+	**6.9E-2(±1.6E-2)**	8.3E-2(±1.4E-2)+	**7.1E-2(±2.0E-2)**	9.6E-2(±1.3E-2)+
Set 16	Task 1	**0.0E+0(±0.0E+0)**	1.5E-2(±7.6E-3)+	**0.0E+0(±0.0E+0)**	1.1E-2(±4.6E-3)+	**0.0E+0(±0.0E+0)**	2.1E-2(±8.3E-3)+
	Task 2	**0.0E+0(±0.0E+0)**	3.3E-3(±3.7E-4)+	6.1E-2(±3.3E-19)	**3.3E-3(±4.3E-4)+**	**4.9E-19(±1.6E-18)**	3.3E-3(±3.0E-4)+
+/≈/-			29/1/2		27/3/2		28/3/1

Table S8: Results of *mse* for MO-MFEA, MTDE-MKTA and MTES-KG compared with their variants with post-hoc refinement framework

inter-task knowledge transfer entirely, avoids this specific pitfall. Nonetheless, the superior performance of MTGP-BS on the vast majority of other problem sets demonstrates that the bidirectional subexpression cooperative extraction method remains a crucial and highly beneficial component of the overall framework.

J.3. Conclusion

To verify our bidirectional extraction method, we compare MTGP-BS against a variant, MTGP-S, which omits this module. It can be found that MTGP-BS consistently achieves significantly lower *mse*, demonstrating the crucial advantage of our strategic subexpression extraction. The inset plot in Figure 4(a) of the main text provides a qualitative example of this improved alignment with the ground truth.

Appendix K. Detailed Ablation Study for Consensus-Accelerated Shapley Analysis

K.1. Full Numerical Results

Table S6 presents the mean and standard deviation of the *mse* for our MTGP-BS and the three ablated variants over 30 independent runs.

K.2. Description of Ablated Algorithm Variants

To comprehensively evaluate the contributions of the two key stages within the consensus-accelerated Shapley analysis, SHAP-based multi-model attribution and consensus-based subexpression selection, three distinct ablation algorithms are designed.

Algorithm	MTGP			BLKT-DE			MFEA-DGD		
Time type Set	Optimization time	Post-processing time	Selection time	Optimization time	Post-processing time	Selection time	Optimization time	Post-processing time	Selection time
Set 1	1.10E+02	7.10E+00	5.86E-05	8.26E+01	7.07E+00	5.56E-05	1.09E+02	7.51E+00	5.32E-05
Set 2	1.10E+02	3.69E+00	5.97E-05	8.37E+01	5.73E+00	6.21E-05	1.10E+02	6.15E+00	4.71E-05
Set 3	1.07E+02	7.19E+00	5.11E-05	8.13E+01	7.03E+00	5.64E-05	1.08E+02	7.64E+00	6.16E-05
Set 4	1.11E+02	5.13E+00	5.62E-05	7.93E+01	7.20E+00	6.22E-05	1.22E+02	6.62E+00	5.54E-05
Set 5	1.05E+02	3.86E+00	1.71E-04	8.44E+01	6.00E+00	1.61E-04	1.20E+02	6.34E+00	1.65E-04
Set 6	1.10E+02	3.76E+00	3.02E-05	7.25E+01	6.54E+00	3.11E-05	1.27E+02	6.62E+00	2.85E-05
Set 7	1.04E+02	6.70E+00	3.31E-05	7.78E+01	7.12E+00	3.08E-05	1.18E+02	7.26E+00	3.10E-05
Set 8	1.03E+02	3.87E+00	3.18E-05	8.01E+01	7.17E+00	2.99E-05	1.24E+02	6.64E+00	2.89E-05
Set 9	1.05E+02	3.79E+00	4.40E-05	7.97E+01	7.58E+00	3.14E-05	1.27E+02	4.37E+00	2.94E-05
Set 10	9.99E+01	7.14E+00	3.07E-05	1.31E+02	7.12E+00	3.21E-05	1.08E+02	6.06E+00	2.81E-05
Set 11	9.97E+01	7.34E+00	3.14E-05	8.07E+01	7.20E+00	3.14E-05	1.29E+02	7.43E+00	3.04E-05
Set 12	1.01E+02	7.37E+00	2.83E-05	8.93E+01	7.21E+00	3.06E-05	1.10E+02	7.27E+00	2.84E-05
Set 13	1.02E+02	7.58E+00	3.04E-05	7.83E+01	7.07E+00	3.11E-05	1.20E+02	7.38E+00	3.34E-05
Set 14	1.02E+02	7.20E+00	3.02E-05	8.51E+01	7.22E+00	3.17E-05	1.16E+02	7.94E+00	3.20E-05
Set 15	1.05E+02	7.25E+00	3.05E-05	8.11E+01	7.20E+00	3.15E-05	1.50E+02	7.64E+00	3.01E-05
Set 16	1.05E+02	7.11E+00	2.89E-05	7.90E+01	7.05E+00	3.25E-05	1.25E+02	1.71E+00	3.58E-05
Total time	**1.68E+03**	**9.61E+01**	**7.46E-04**	**1.35E+03**	**1.12E+02**	**7.41E-04**	**1.92E+03**	**1.05E+02**	**7.19E-04**

Table S9: Running time of optimization, post-hoc refinement, and direct selection

MTGP-B1 (No Ensemble)
This variant is designed to evaluate the effectiveness of the multi-model attribution. The bidirectional subexpression co-operative extraction method is followed by an analysis based on only a single surrogate model. Since the consensus-based selection mechanism relies on multi-model outputs, important subexpressions are instead selected using a simple threshold, where any subexpression with an absolute SHAP value exceeding the average is chosen.

MTGP-B2 (No Distributional Information)
This variant assesses the benefit of retaining the full distribution of SHAP values. While it uses an ensemble of models, the final importance of each subexpression is determined by averaging its absolute SHAP values across all models into a single score. The same mean-based thresholding procedure as in MTGP-B1 is then used for selection.

MTGP-B3 (No Enhanced Roulette Wheel)
This variant evaluates the effectiveness of the consensus-based selection mechanism. It retains the full multi-model attribution stage, but for selection, it applies only the strict,

Baseline	MO-MFEA			MTDE-MKTA			MTES-KG		
Time type Set	Optimization time	Post-processing time	Selection time	Optimization time	Post-processing time	Selection time	Optimization time	Post-processing time	Selection time
Set 1	8.37E+01	7.07E+00	6.35E-05	9.52E+01	7.01E+00	5.35E-05	1.38E+02	6.98E+00	2.37E-04
Set 2	8.15E+01	5.12E+00	6.26E-05	8.38E+01	4.96E+00	5.23E-05	1.36E+02	7.06E+00	2.57E-04
Set 3	8.40E+01	7.29E+00	5.28E-05	8.82E+01	6.95E+00	5.08E-05	1.40E+02	7.10E+00	2.39E-04
Set 4	8.54E+01	4.48E+00	5.71E-05	9.02E+01	4.57E+00	5.32E-05	1.45E+02	6.50E+00	2.43E-04
Set 5	8.73E+01	5.01E+00	1.74E-04	8.87E+01	5.63E+00	1.51E-04	1.42E+02	7.24E+00	4.07E-04
Set 6	8.38E+01	6.34E+00	3.03E-05	8.32E+01	6.28E+00	2.84E-05	1.30E+02	7.00E+00	2.14E-04
Set 7	7.87E+01	6.96E+00	3.06E-05	8.90E+01	7.08E+00	2.88E-05	1.34E+02	7.09E+00	2.33E-04
Set 8	8.30E+01	5.78E+00	3.05E-05	9.45E+01	5.13E+00	2.89E-05	1.41E+02	5.33E+00	2.02E-04
Set 9	7.90E+01	7.12E+00	2.85E-05	9.41E+01	5.73E+00	2.83E-05	1.40E+02	6.72E+00	1.78E-04
Set 10	8.27E+01	7.09E+00	3.00E-05	8.73E+01	7.07E+00	3.45E-05	1.36E+02	7.17E+00	2.13E-04
Set 11	8.32E+01	7.05E+00	3.34E-05	8.82E+01	7.19E+00	2.85E-05	1.35E+02	7.26E+00	2.05E-04
Set 12	8.86E+01	7.10E+00	3.19E-05	8.86E+01	7.05E+00	2.82E-05	1.31E+02	7.21E+00	2.38E-04
Set 13	8.60E+01	7.16E+00	2.98E-05	9.32E+01	7.15E+00	3.03E-05	1.36E+02	1.05E+01	2.09E-04
Set 14	8.60E+01	7.17E+00	3.08E-05	9.16E+01	7.05E+00	2.84E-05	1.43E+02	7.06E+00	1.95E-04
Set 15	8.45E+01	7.44E+00	3.11E-05	9.23E+01	7.10E+00	2.94E-05	1.47E+02	7.33E+00	1.96E-04
Set 16	8.51E+01	7.18E+00	3.15E-05	9.19E+01	7.89E+00	2.95E-05	1.43E+02	7.32E+00	1.88E-04
Total time	**1.34E+03**	**1.05E+02**	**7.49E-04**	**1.44E+03**	**1.04E+02**	**6.84E-04**	**2.22E+03**	**1.15E+02**	**3.65E-03**

Table S10: Running time of optimization, post-hoc refinement, and direct selection.

Test	Name	Expression	# of Vars	Operators	Range	# of sample data
Set 1	Jin-2	$8x_1^2 + 8x_2^3 - 15$	2	+, -, ×, ^2, ^3	Train, test: $\{x_1, x_2\} \in [-3,3]$	Train, test: [100,30]
	Jin-3	$0.2x_1^3 + 0.5x_2^3 - 1.2x_2 + 0.5x_1$	2			
Set 2	Jin-4	$1.5\exp(x_1) + 5\cos(x_2)$	2	+, -, ×, *sin*, *cos*, *exp*	Train, test: $\{x_1, x_2\} \in [-3,3]$	Train, test: [100,30]
	Jin-5	$6\sin(x_1)\cos(x_2)$	2			
Set 3	Nguyen-2	$x_1^4 + x_1^3 + x_1^2 + x_1$	1	+, -, ×	Train, test: $\{x_1\} \in [-1,1]$	Train, test: [20,10]
	Nguyen-3	$x_1^5 + x_1^4 + x_1^3 + x_1^2 + x_1$	1			
Set 4	Nguyen-2	$x_1^4 + x_1^3 + x_1^2 + x_1$	1	+, -, ×	Train, test: $\{x_1\} \in [-1,1]$	Train, test: [100,30]
	Nguyen-3	$x_1^5 + x_1^4 + x_1^3 + x_1^2 + x_1$	1			
Set 5	Nguyen-9	$\sin(x_1) + \sin(x_2^2)$	2	+, -, ×, *sin*, *cos*	Train, test: $\{x_1, x_2\} \in [0,1]$	Train, test: [20,10]
	Nguyen-10	$2\sin(x_1)\cos(x_2)$	2			
Set 6	Nguyen-9	$\sin(x_1) + \sin(x_2^2)$	2	+, -, ×, *sin*, *cos*	Train, test: $\{x_1, x_2\} \in [0,1]$	Train, test: [100,30]
	Nguyen-10	$2\sin(x_1)\cos(x_2)$	2			

Table S11: MTSR problem sets are constructed by pairing well-known single-task SR problems

voting-based component. The enhanced roulette wheel phase is omitted.

K.3. Conclusion

To validate our consensus-accelerated Shapley analysis, we create three ablated variants, MTGP-B1 (single-model analysis), MTGP-B2 (simple SHAP averaging), and MTGP-B3 (voting-only consensus). The results show that removing any core components leads to a significant degradation in performance.

Appendix L. Detailed Analysis for Generality and Effectiveness of the Post-Hoc Refinement

L.1. Full Numerical Results

Table S7 and Table S8 presents the mean and standard deviation of the *mse* for all algorithms and their variants with post-hoc refinement framework over 30 independent runs on the 16 MTSR problems.

L.2. In-Depth Analysis of MTGP-BS with Traditional MTGP

The MTGP-BS is proposed to significantly enhance the traditional MTGP. As shown in Table S7, MTGP-BS achieves significantly better performance on 18 tasks, highlighting the effectiveness of synthesizing knowledge from the entire population instead of relying on a single winner expression.

However, MTGP-BS performs worse than MTGP on two problems. To analyze this, the first task from Set 11 is considered as an example, where the expression to be learned is $100x_1^4 - 200x_1^2x_2 + x_1^2 - 2x_1 + 100x_2^2 + 1$. While that to be learned in the second task is $x_1^2/4000 + x_2^2/4000 - cos\,(x_1)cos\,(x_2/1.41) + 1$. While the shared knowledge of x_1^2 and x_2^2 could be beneficial, the trigonometric components from the second task are also transferred to the first. This introduces redundant, interfering knowledge that complicates the subsequent SHAP analysis and can lead to the construction of a suboptimal final expression. In contrast, the simpler selection of traditional MTGP avoids this specific negative transfer by chance, but at the cost of failing to

Set	Task	MTGP-BS	BLKT-DE	MFEA-DGD	MO-MFEA	MTDE-MKTA	MTES-KG
Set 1	Task 1	**0.0E+0(±0.0E+0)**	1.1E-2(±2.0E-2)+	2.9E-2(±2.1E-2)+	1.1E-2(±1.8E-2)+	4.0E-3(±1.3E-2)+	3.7E-2(±1.0E-2)+
	Task 2	**2.9E-3(±5.6E-3)**	3.6E-2(±9.1E-3)+	4.4E-2(±1.3E-2)+	3.5E-2(±9.3E-3)+	3.4E-2(±7.6E-3)+	4.2E-2(±1.0E-2)+
Set 2	Task 1	**0.0E+0(±0.0E+0)**	4.4E-2(±2.8E-2)+	8.3E-2(±2.7E-2)+	3.8E-2(±1.9E-2)+	3.6E-2(±2.4E-2)+	6.4E-2(±2.8E-2)+
	Task 2	**1.3E-17(±6.3E-18)**	3.0E-2(±4.4E-2)+	1.3E-1(±6.4E-2)+	4.6E-2(±5.1E-2)+	1.3E-2(±3.3E-2)+	9.6E-2(±5.8E-2)+
Set 3	Task 1	**6.9E-18(±6.8E-18)**	6.1E-3(±1.0E-2)+	2.3E-2(±2.5E-2)+	1.8E-2(±4.6E-2)+	6.9E-3(±1.1E-2)+	9.2E-3(±7.4E-3)+
	Task 2	**1.1E-17(±1.2E-17)**	1.2E-2(±9.9E-3)+	2.3E-2(±2.6E-2)+	1.8E-2(±2.5E-2)+	1.2E-2(±7.0E-3)+	1.2E-2(±6.6E-3)+
Set 4	Task 1	**9.1E-18(±5.2E-18)**	2.8E-3(±4.6E-3)+	1.6E-2(±1.1E-2)+	7.4E-3(±6.2E-3)+	1.9E-3(±5.1E-3)+	7.0E-3(±4.7E-3)+
	Task 2	**5.0E-4(±1.5E-3)**	9.2E-3(±3.1E-3)+	1.7E-2(±1.2E-2)+	9.8E-3(±3.5E-3)+	8.5E-3(±3.2E-3)+	8.9E-3(±2.6E-3)+
Set 5	Task 1	**4.9E-3(±4.4E-3)**	1.8E-2(±7.1E-3)+	2.7E-2(±1.6E-2)+	1.6E-2(±8.7E-3)+	1.5E-2(±1.0E-2)+	1.5E-2(±7.0E-3)+
	Task 2	**3.2E-4(±1.4E-3)**	8.1E-3(±8.0E-3)+	4.4E-2(±3.5E-2)+	1.1E-2(±8.9E-3)+	7.8E-3(±7.6E-3)+	1.3E-2(±8.9E-3)+
Set 6	Task 1	**5.9E-3(±4.8E-3)**	1.1E-2(±4.8E-3)+	2.5E-2(±1.7E-2)+	1.0E-2(±5.3E-3)+	1.0E-2(±3.8E-3)+	1.2E-2(±3.4E-3)+
	Task 2	**1.8E-3(±3.9E-3)**	8.1E-3(±7.7E-3)+	3.4E-2(±1.9E-2)+	1.1E-2(±7.2E-3)+	5.9E-3(±6.9E-3)+	1.5E-2(±9.0E-3)+
+/≈/-			12/0/0	12/0/0	12/0/0	12/0/0	12/0/0

Table S12: Results of *mse* for MTGP-BS compared with BLKT-DE, MFEA-DGD, MO-MFEA, MTDE-MKTA and MTES-KG.

Set	Algorithm	Task 1	*mse*	Task 2	*mse*
Set 1	MTGP-BS	$8.0x_1^2 + 8.0x_2^3 - 15.0$	0.00E+00	$0.2x_1^3 - 1.2x_2 - 0.5x_1 + 0.5x_2^3$	1.39E-17
	BLKT-DE	$8.0x_1^2 + 8.0x_2^3 - 15.0$	0.00E+00	$0.47x_1 - 0.94x_2 + 0.47x_2^3 + 0.004$	4.05E-02
	MFEA-DGD	$8.44x_2^3 - 2.11x_2 + 7.28$	4.07E-02	$0.47x_2^3 - 0.93x_2 + 0.075$	3.99E-02
	MO-MFEA	$8.0x_1^2 + 8.0x_2^3 - 15.0$	0.00E+00	$0.05x_1^3 - 0.4x_2 + 0.4x_2^3 + 0.002$	3.63E-02
	MTDE-MKTA	$8.0x_1^2 + 8.0x_2^3 - 15.0$	0.00E+00	$0.47x_1 - 0.93x_2 + 0.47x_2^3 - 0.002$	3.61E-02
	MTES-KG	$8.34x_2^3 - 1.19x_2^2 - 1.19x_2 + 15.43$	3.49E-02	$0.045x_1x_2^2 + 0.045x_2^5 + 0.12$	4.20E-02
Set 2	MTGP-BS	$5.0\cos(x_2) + 1.5\exp(x_1)$	0.00E+00	$6.0\cos(x_2)\sin(x_1)$	1.43E-17
	BLKT-DE	$1.51\cos(x_2) + 1.51\exp(x_1) - 0.054$	7.19E-02	$6.0\cos(x_2)\sin(x_1) - 6.754\mathrm{e}{-}17$	1.76E-17
	MFEA-DGD	$1.73x_1 - 1.63x_2^2 + 9.47$	1.13E-01	$1.11x_1 - 0.49x_1x_2^2 + 0.25x_2^2 - 0.79$	1.13E-01
	MO-MFEA	$3.12\cos(x_2) - 1.56\cos(\cos(\exp(x_1)))$ $+1.56\exp(x_1) + 0.87$	3.13E-02	$3.89x_1\sin(\exp(\cos(x_1)))\cos(x_2) - 0.058$	5.06E-02
	MTDE-MKTA	$1.45\exp(x_1) - 1.45x_2^2 + 4.83$	2.26E-02	$6.56\sin(\cos(x_2)\sin(x_1)) - 0.0055$	1.10E-02
	MTES-KG	$1.39\exp(x_1) - 1.39\sin(\sin(x_2 - 0.48))$ $-1.39x_2^2 + 4.79$	3.18E-02	$0.96x_2^2\cos(x_2)\sin(x_1) - 0.26$	1.42E-01
Set 3	MTGP-BS	$x_1 + x_1^2 + x_1^3 + x_1^4$	6.88E-18	$x_1 + x_1^2 + x_1^3 + x_1^4 + x_1^5$	5.55E-18
	BLKT-DE	$x_1 + x_1^2 + x_1^3 + x_1^4 + 9.564\mathrm{e}{-}18$	2.08E-18	$0.89x_1 + 0.44x_1^2 + 1.78x_1^3 + 1.78x_1^4 + 0.073$	1.31E-02
	MFEA-DGD	$1.02x_1 + 1.78x_1^2 + 1.02x_1^3 - 0.063$	1.60E-02	$1.01x_1 + 1.6x_1^2 + 1.69x_1^3 + 0.34x_1^4 - 0.067$	1.38E-02
	MO-MFEA	$0.91x_1 + 1.52x_1^2 + 1.21x_1^3 - 0.027$	1.49E-02	$0.83x_1 + 0.74x_1^2 + 1.85x_1^3 + 1.48x_1^4 + 0.02$	6.97E-03
	MTDE-MKTA	$x_1 + x_1^2 + x_1^3 + x_1^4 - 2.975\mathrm{e}{-}17$	1.28E-17	$0.97x_1 + 1.67x_1^2 + 1.67x_1^3 - 0.035$	1.35E-02
	MTES-KG	$1.01x_1 + 1.01x_1^2 + 1.01x_1^3 + 0.81x_1^4 + 0.037$	1.12E-02	$0.8x_1 + 1.2x_1^2 + 2x_1^3 + 0.8x_1^4 - 0.04$	8.62E-03
Set 4	MTGP-BS	$x_1 + x_1^2 + x_1^3 + x_1^4$	8.56E-18	$x_1 + x_1^2 + x_1^3 + x_1^4 + x_1^5$	5.55E-18
	BLKT-DE	$x_1 + x_1^2 + x_1^3 + x_1^4 - 1.239\mathrm{e}{-}16$	2.87E-17	$0.95x_1 + 1.9x_1^2 + 1.9x_1^3 - 0.095$	1.12E-02
	MFEA-DGD	$0.94x_1 + 1.89x_1^2 + 1.08x_1^3 - 0.096$	1.30E-02	$0.33x_1 + 1.99x_1^2 + 2.66x_1^3 - 0.13$	1.92E-02
	MO-MFEA	$0.92x_1 + 1.38x_1^2 + 1.15x_1^3 + 0.46x_1^4 - 0.019$	9.36E-03	$0.72x_1 + 0.72x_1^2 + 2.15x_1^3 + 1.43x_1^4 + 0.008$	7.29E-03
	MTDE-MKTA	$x_1 + x_1^2 + x_1^3 + x_1^4 + 8.979\mathrm{e}{-}17$	2.55E-17	$0.712x_1 + 0.71x_1^2 + 2.14x_1^3 + 1.43x_1^4 + 0.012$	7.71E-03
	MTES-KG	$0.94x_1 + 1.48x_1^2 + 1.07x_1^3 + 0.54x_1^4 - 0.078$	9.21E-03	$0.92x_1 + 1.15x_1^2 + 1.83x_1^3 + 0.92x_1^4 - 0.038$	6.42E-03
Set 5	MTGP-BS	$\sin(x_1) - 0.02x_1 + 0.19\sin(x_2) + 0.73x_2^2$ $- 7.4\mathrm{e}{-}3\cos(x_2)\sin(x_1) - 0.017$	5.88E-03	$2.0\cos(x_2)\sin(x_1)$	0.00E+00
	BLKT-DE	$0.91x_1 + 0.91x_2^2 + 0.025$	1.83E-02	$1.92\cos(0.5x_1) + 0.49\sin(0.5x_1)$ $+ 1.7x_1\cos(x_2) - 1.91$	1.09E-02
	MFEA-DGD	$0.95x_1 + 0.95\sin(x_2^2) + 0.008$	3.03E-02	$1.38x_1 - 0.46x_2 + 0.34$	5.15E-02
	MO-MFEA	$0.9x_1 - 0.9\cos(x_2)\sin(\cos(0.5x_2))$ $+0.9\cos(\cos(0.5x_2))\sin(x_2) + 0.68$	1.92E-02	$1.92x_1\cos(x_2) + 0.24x_1^2 - 0.48x_1^3 + 0.0059$	6.80E-03
	MTDE-MKTA	$1.05x_1 + 0.42\cos(x_1) + 0.42x_1x_2^2$ $+0.84x_2^2\cos(x_1) - 0.41$	6.98E-03	$0.98x_1 - 0.98x_2 + 0.98\cos(x_1)\sin(x_2)$ $+0.98\cos(x_2)\sin(x_1) + 0.013$	1.09E-02
	MTES-KG	$0.88x_1 + 0.88x_2^2 + 0.037$	1.76E-02	$1.96\sin(x_1\cos(x_2)) + 0.0043$	9.28E-03
Set 6	MTGP-BS	$\sin(x_1) - 1.7\cos(x_2) + 0.11\sin(x_2) + 1.7$	8.26E-03	$2\cos(x_2)\sin(x_1)$	0.00E+00
	BLKT-DE	$0.87x_1 + 0.87x_2^2 + 0.045$	1.64E-02	$2\cos(x_2)\sin(x_1) - 6.876\mathrm{e}{-}16$	4.08E-16
	MFEA-DGD	$0.92x_1 - 0.92\cos(x_2) + 0.46\sin(x_2) + 0.87$	2.42E-02	$1.32x_1 - 1.32\sin(x_2\sin(\sin(x_2 - 0.5))) + 0.2$	4.59E-02
	MO-MFEA	$0.87x_1 + 0.44\sin(x_2 + 0.5)(x_2 + x_2^2)$ $+0.0017$	1.03E-02	$1.995x_1\cos(0.5x_1 + x_2 - 0.5x_1x_2) + 0.0048$	1.26E-02
	MTDE-MKTA	$0.87x_1 + 0.87x_2^2 + 0.047$	1.50E-02	$2\cos(x_2)\sin(x_1) + 9.179\mathrm{e}{-}16$	6.13E-16
	MTES-KG	$0.6x_1 + 0.3x_2 + 0.3\sin(x_1) + 0.6x_2^2 - 0.017$	9.24E-03	$0.93x_1 + 0.62\cos(x_2) + 0.62x_1\cos(x_2) - 0.47$	1.93E-02

Table S13: Expressions obtained from the median-performing run of each algorithm for all six problems

utilize beneficial knowledge in the majority of other problems. This detailed analysis confirms the superiority of MTGP-BS while transparently identifying its limitations. However, MTGP can still hinder effective knowledge learning for most other problems due to the direct discarding of valid knowledge in poor performance expressions.

L.3. Analysis of Generality on Other Baselines

The post-hoc refinement framework consistently enhanced the performance of each distinct baseline algorithm, confirming its versatility as a plug-and-play module.

The framework achieved a superior result on 29 tasks with no losses for BLKT-DE, as shown in Table S7. This indicates that our post-hoc knowledge synthesis is highly compatible with, and significantly improves upon, the conservative strategy of BLKT-DE for preventing negative transfer. Similar significant improvements are observed across the remaining baselines MFEA-DGD, MO-MFEA, MTDE-MKTA, and MTES-KG, with win counts of 19, 29, 27, and 28 respectively. This consistent success across algorithms that employ diverse strategies, from diffusion gradients to external knowledge, underscores that our method effectively captures and synthesizes valuable knowledge that is complementary to the specific mechanisms of the underlying evolutionary algorithm.

L.4. Conclusion

The results demonstrate that the general effectiveness of our post-hoc module, which consistently and substantially improves performance across six multitask algorithms.

Appendix M. Detailed Analysis for Running Time

M.1. Full Numerical Results

Table S9 and Table S10 presents the runtime in seconds for each of the three main phases across. “Optimization time” refers to the time taken by the baseline multitask algorithms to complete its evolutionary search. “Post-processing time” refers to the additional time required by our post-hoc refinement framework to perform extraction, analysis, and synthesis. “Selection time” refers to the negligible time taken by the traditional method to simply select the best individual.

M.2. Conclusion

The results reveal that the initial evolution overwhelmingly dominates the runtime. The overhead of our framework is a small fraction of this cost, demonstrating that its significant performance gains are achieved with modest computational overhead.

Appendix N. Results on Constructed Multitask Benchmarks

This section provides the detailed benchmark descriptions and full numerical results for the experiment constructed multitask benchmarks.

N.1. Benchmark Construction

The six MTSR problem sets are constructed by pairing well-known single-task SR problems (Jin et al. 2019; Uy et al. 2011). The specific expressions used for each set are detailed in Table S11. The original reference for the Nguyen benchmark problems utilizes a training set size of 20. However, as the majority of experiments in this work are conducted with a training set size of 100, we evaluated our algorithm on the selected Nguyen problems under both conditions. This ensures a fair and direct comparison against the baseline results.

N.2. Full Numerical Results

Table S12 presents the mean and standard deviation of the *mse* for all algorithms on the six constructed multitask problems.

N.3. Discovered Median-Run Expressions

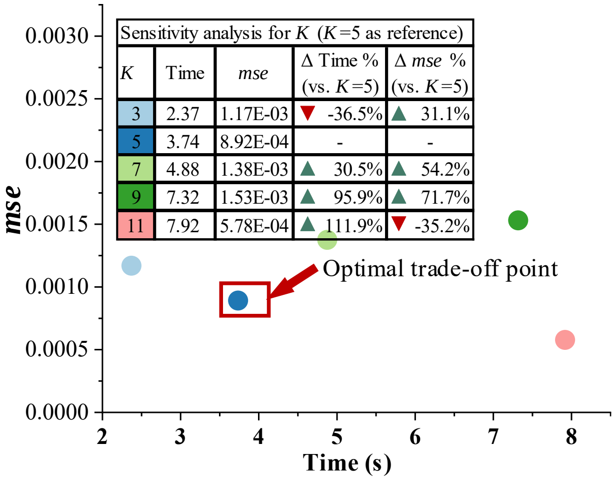
Table S13 presents the symbolic expressions obtained from the median-performing run of each algorithm on the six constructed multitask problems.

N.4. Conclusion

We test MTGP-BS on six new MTSR problems derived from established SR benchmarks. It can be observed that, compared with other algorithms, MTGP-BS achieves the best *mse* on all these problems, demonstrating its superior performance.

Appendix O. Detailed Sensitivity Analysis on the Number of Models

This appendix provides a comprehensive sensitivity analysis for the number of models, K, used in MTGP-BS. The analysis validates our selection of $K = 5$ as the optimal parameter, which balances predictive performance (measured by *mse*) and computational cost (measured by runtime in seconds). We teste K on the set [3, 5, 7, 9, 11] across three representative problems, Set 2, Set 3, and Set 16. The detailed results and discussion below support our choice of $K = 5$ as the most efficient configuration for general application.



Sensitivity analysis for K (K=5 as reference)				
K	Time	*mse*	Δ Time % (vs. K=5)	Δ *mse* % (vs. K=5)
3	2.37	1.17E-03	▼ -36.5%	▲ 31.1%
5	3.74	8.92E-04	-	-
7	4.88	1.38E-03	▲ 30.5%	▲ 54.2%
9	7.32	1.53E-03	▲ 95.9%	▲ 71.7%
11	7.92	5.78E-04	▲ 111.9%	▼ -35.2%

Figure S1: Sensitivity analysis for K on Set 2

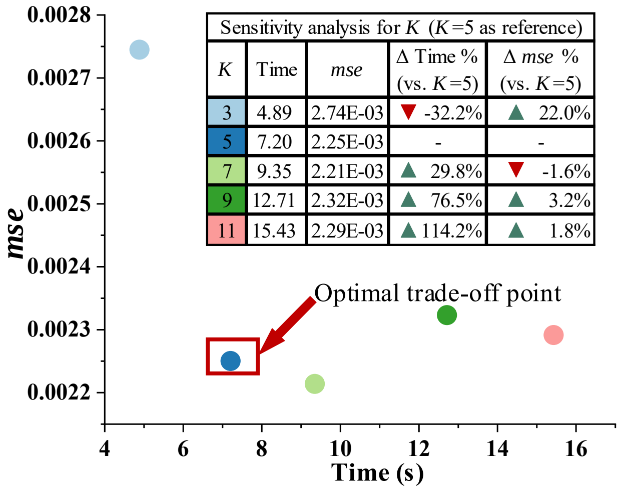
O.1. Detailed Analysis

Analysis on Set 3

The results for Set 3 (main paper, Figure 9) present a clear case for selecting $K = 5$. The parameter $K = 5$ achieves the lowest *mse* of 1.82E-02, establishing it as the most accurate configuration for this problem. While decreasing K to three reduces runtime by 33.3%, it comes at the cost of a 3.4% worse in accuracy, representing a sacrifice in performance. Increasing K beyond five leads to a clear degradation in both metrics. For $K = [7, 9, 11]$, the runtime and *mse* both increase significantly. This demonstrates strong diminishing returns, where a larger number of models not only increases computational cost but also harms predictive performance. Given these observations, $K = 5$ is unambiguously the optimal choice for Set 3.

Analysis on Set 2

The sensitivity analysis on Set 2 (Figure S1) reveals that the overall conclusion regarding the optimal trade-off remains consistent. The configuration with $K = 11$ achieves the best accuracy with an *mse* of 5.78E-04. However, this superior performance comes at an excessive computational cost, requiring a 111.9% increase in runtime compared to $K = 5$ (from 3.74s to 7.92s). The $K = 5$ setting provides a highly competitive *mse* of 8.92E-04, which is significantly better than the results for $K = [3, 7, 9]$. For a generalized framework where both performance and efficiency are crucial, the extreme trade-off offered by $K = 11$ is impractical. $K = 5$ presents a much more balanced choice, avoiding the poor accuracy of $K = 3$ and the excessive computational demands of $K = 11$.



Sensitivity analysis for K (K=5 as reference)				
K	Time	*mse*	Δ Time % (vs. K=5)	Δ *mse* % (vs. K=5)
3	4.89	2.74E-03	▼ -32.2%	▲ 22.0%
5	7.20	2.25E-03	-	-
7	9.35	2.21E-03	▲ 29.8%	▼ -1.6%
9	12.71	2.32E-03	▲ 76.5%	▲ 3.2%
11	15.43	2.29E-03	▲ 114.2%	▲ 1.8%

Figure S2: Sensitivity analysis for K on Set 16

Analysis on Set 16

The results for Set 16 in Figure S2 highlight a classic performance-efficiency trade-off. In this case, $K = 7$ obtains the absolute lowest *mse* (2.21E-03). However, this represents only a marginal 1.6% improvement over $K = 5$ (2.25E-03), while incurring a substantial 29.8% increase in runtime. Conversely, selecting $K = 3$, while 32.2% faster than $K = 5$, results in a significant 22.0% degradation in *mse*, making it an undesirable choice due to the large performance drop. This analysis validates the selection of $K = 5$ as the optimal trade-off point, as it provides near-optimal accuracy with considerably greater computational efficiency.

In summary, across all three distinct problems, $K = 5$ consistently provides the most efficient balance between predictive accuracy and runtime. While other values of K may yield marginally better accuracy in isolated cases (e.g., $K = 7$ on Set 16 or $K = 11$ on Set 2), they do so at a disproportionately high computational cost. Therefore, our selection of $K = 5$ is well-justified as the default parameter for our study.

O.2. Theoretical Analysis of the Rationality for *K*=5

Our proposition in Appendix F demonstrates that the voting-based selection becomes a more statistically stringent filter as the number of models K increases. Our empirical sensitivity analysis, however, reveals that stricter is not necessarily better, as the end-to-end performance of the framework is optimized at a finite $K = 5$. This indicates that the overall system performance is not solely determined by the theoretical power of the filter, but by achieving an optimal balance between our two complementary selection strategies, the voting-based selection and the enhanced roulette wheel selection.

The performance degradation at high K values stems from a two-fold problem. First, the statistical significance proof is built on the foundational assumption that the K models are independent. As K grows on a finite dataset, bootstrap sampling inevitably generates highly similar data subsets, leading to correlated models that violate this core assumption. This risks a false consensus, where a group of models sharing the same biases confidently agrees on a suboptimal subexpression, thereby invalidating the statistical promise. Second, an overly stringent filter at a high K value upsets the delicate synergy between our selection mechanisms. The voting-based selection is designed to act as a high-precision filter to identify a core set of indisputable candidates, while the enhanced roulette wheel selection is designed to ensure exploration diversity by considering other consistently important subexpressions. If the voting filter is too strict, it selects very few candidates, causing the final outcome to be disproportionately determined by the exploratory nature of the roulette wheel, rather than a balanced combination of both strategies.

Year	Gross domestic production	Unit energy consumption	Human population	Carbon emissions
2010	413.8387	568.804088	78693400	56360.05184
2011	459.5265	584.5152741	80229900	65193.34223
2012	506.602	552.6866872	81198100	67502.61337
2013	555.8011	507.4316023	81924400	66749.3757
2014	603.5943	466.7125877	82810900	64853.27604
2015	655.52	442.909569	83151100	66074.80995
2016	706.6570683	423.7978782	83814700	68526.12467
2017	757.5220149	404.8712229	84235000	70451.55739
2018	808.2771193	388.1481474	84461900	71502.00286
2019	855.5613387	376.6825817	84690900	74096.33108
2020	886.8321463	354.4977219	84772600	72633.32425

Table S14: Data used in case study

The choice of *K*=5 represents the optimal operating point that achieves the most effective synergy between these competing factors. At this value, the ensemble is large enough for the consensus rule to be a statistically significant filter against random noise, but small enough that the models remain sufficiently independent. This allows the voting mechanism to identify a meaningful core of high-confidence candidates, which is then perfectly complemented by the exploratory pressure of the roulette wheel selection. Therefore, the selection of *K*=5 is the direct result of a system-level design that achieves the optimal balance between a theoretically-grounded, high-precision filter and a principled heuristic for exploration, maximizing the performance of the entire framework under most MTSR problems in this work.

Appendix P. Detailed Analysis of the Real-World Case Study

P.1. Background and Motivation

In recent years, there has been increasing research on the balance between economic development and environmental sustainability (Kolasa-Wiecek et al. 2025; Malatyinszki, Zeman and Kalman 2025). Exploring the relationship between these factors helps policymakers understand economic development models and consider how to balance economic growth and environmental protection by improving energy efficiency, which is crucial for achieving sustainable development.

To validate the value of the proposed MTGP-BS in a real-world application, this case study on population and carbon emission prediction is introduced. Specifically, gross domestic product and unit energy consumption are key indicators for measuring economic development. These factors not only influence population dynamics but also directly relate to energy efficiency and its environmental impact. Analyzing these factors contributes to a deeper understanding of population growth trends and the importance of improving energy efficiency in reducing environmental pollution. To formulate effective policies, it is necessary to comprehensively understand the interaction mechanisms between these variables.

P.2. Training Dataset

Dataset used in this case study is listed in Table S14.

P.3 Problem Formulation

The problem is formulated as a two-task MTSR problem, aiming to find the mapping functions E_1 and E_2,

$$\begin{cases} H = E_1(D, \varepsilon) \\ Q = E_2(D, \varepsilon) \end{cases}\text{’}$$

where H refers to the human population, Q refers to the carbon emissions, D refers to the gross domestic production, ε refers to the unit energy consumption, and E_1 and E_2 refer to the mapping function from $[D, \varepsilon]$ to H and Q, respectively. The expressions E_1 and E_2 will be learned using MTGP-BS and the comparative algorithms MO-MFEA, MFEA-DGD, BLKT-DE, MTDE-MKTA, and MTES-KG. The learning process will be conducted based on data collected over the period from 2010 to 2020.

P.4. Detailed Qualitative Comparison of Discovered Expressions

Table 4 in the main text lists the median-run expressions discovered by each algorithm. A detailed structural analysis is provided below.

Task 1 (Human Population Prediction)

The expression learned by MTGP-BS is $43.62D\varepsilon - 24.92D - 12.46\varepsilon^2 + 7.28^7$. Compared to the expression $11.54D + 46.17D\varepsilon - 11.54\varepsilon^2 + 7.18^7$ learned by BLKT-DE, the internal patterns of data are better captured by the composite structure $D\varepsilon - D - \varepsilon^2$ learned by MTGP-BS, rather than $D\varepsilon + D - \varepsilon^2$ structure from BLKT-DE. This comparison strongly suggests that a more effective structure describing the change in population is captured by MTGP-BS. Regarding the other comparative algorithms, MFEA-

DGD, MO-MFEA, MTDE-MKTA, and MTES-KG, an explicit association between D and H is not captured, which may explain why their learned expressions exhibited lower accuracy.

Task 2 (Carbon Emission Prediction)

The expression learned by MTGP-BS is $3.9D - 62.0\varepsilon + 0.25D\varepsilon + 0.096\varepsilon^2 - 0.2$. Compared to the expression $0.07D + 0.28D\varepsilon + 0.04\varepsilon^2 - 22280$ learned by MTDE-MKTA, a crucial difference is the presence of the $-\varepsilon$ term our solution. The structure $-\varepsilon + D\varepsilon + \varepsilon^2$, provides higher predictive accuracy than the $D\varepsilon + \varepsilon^2$ structure learned by MTDE-MKTA. Regarding the expressions learned by other comparative algorithms, structural simplicity is observed in those learned by MFEA-DGD and MTES-KG, where a direct association with D is lacking. Additionally, expressions learned by MO-MFEA and BLKT-DE contained a negative coefficient for D, significantly reducing the interpretability of these expressions.

P.5. Conclusion

As discussed above, a good capability is demonstrated by MTGP-BS in addressing a case study on population and carbon emission prediction, allowing for the regression of expressions with higher accuracy. In conclusion, the application of MTGP-BS to this case study can provide a scientific basis for decision-makers, supporting the achievement of sustainable development goals by effectively reducing environmental pollution while promoting economic growth.

References

Akbar, S.; Ali, F.; Hayat, M.; Ahmad, A.; Khan, S.; and Gul, S. 2022. Prediction of Antiviral peptides using transform evolutionary & SHAP analysis based descriptors by incorporation with ensemble learning strategy, *Chemometrics and Intelligent Laboratory Systems,* 230104682. doi.org/10.1016/j.chemolab.2022.104682.

Gupta, A.; Ong, Y. S.; Feng, L.; and Tan, K. C. 2017. Multiobjective Multifactorial Optimization in Evolutionary Multitasking, *IEEE Transactions on Cybernetics,* 47(7): 1652-1665. 10.1109/TCYB.2016.2554622.

Huang, Z.; Zhang, F.; Mei, Y.; and Zhang, M. 2022. An Investigation of Multitask Linear Genetic Programming for Dynamic Job Shop Scheduling. In proceedings of the 25th European Conference on Genetic Programming (EuroGP) Held as Part of EvoStar Conference, Complutense Univ Madrid, Madrid, Spain. 10.1007/978-3-031-02056-8_11

Jiang, Y.; Zhan, Z. H.; Tan, K. C.; and Zhang, J. 2024. Block-Level Knowledge Transfer for Evolutionary Multitask Optimization, *IEEE Transactions on Cybernetics,* 54(1): 558-571. 10.1109/TCYB.2023.3273625.

Jin, Y.; Fu, W.; Kang, J.; Guo, J.; and Guo, J. 2019. Bayesian Symbolic Regression. arXiv:1910.08892.

Kolasa-Wiecek, A.; Steinberga, I.; Pilarska, A. A.; Suszanowicz, D.; and Wzorek, M. 2025. Study of the Environmental Kuznets Curve in the EU27 Countries Taking into Account Socio-Economic Factors and GHG and PM Emissions, *Energies,* 18(1). 10.3390/en18010068.

Kumar, C. S.; Choudary, M. N. S.; Bommineni, V. B.; Tarun, G.; and Anjali, T. 2020. Dimensionality Reduction based on SHAP Analysis: A Simple and Trustworthy Approach. In proceedings of the 2020 International Conference on Communication and Signal Processing (ICCSP). 10.1109/ICCSP48568.2020.9182109

Kushida, J. I.; Hara, A.; and Takahama, T. 2018. Cartesian Genetic Programming with Module Mutation for Symbolic Regression. In proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC). 10.1109/SMC.2018.00038

Li, Y., and Gong, W. 2025. Multiobjective Multitask Optimization With Multiple Knowledge Types and Transfer Adaptation, *IEEE Transactions on Evolutionary Computation,* 29(1): 205-216. 10.1109/TEVC.2024.3353319.

Li, Y.; Gong, W.; and Li, S. 2024. Multitask Evolution Strategy With Knowledge-Guided External Sampling, *IEEE Transactions on Evolutionary Computation,* 28(6): 1733-1745. 10.1109/TEVC.2023.3330265.

Liu, Z.; Li, G.; Zhang, H.; Liang, Z.; and Zhu, Z. 2024. Multifactorial Evolutionary Algorithm Based on Diffusion Gradient Descent, *IEEE Transactions on Cybernetics,* 54(7): 4267-4279. 10.1109/TCYB.2023.3270904.

Lowrance, C. J.; Abdelwahab, O.; and Yampolskiy, R. V. 2015. Evolution of a Metaheuristic for Aggregating Wisdom from Artificial Crowds. In proceedings of the 17th Portuguese Conference on Artificial Intelligence (EPIA), Univ Coimbra, Coimbra, Portugal. 10.1007/978-3-319-23485-4_24

Malatyinszki, S.; Zeman, Z.; and Kalman, B. G. 2025. Resource productivity and sustainability-a comparison of two European countries, *Humanities & Social Sciences Communications,* 12(1). 10.1057/s41599-025-04428-4.

Mendes-Moreira, J.; Soares, C.; Jorge, A. M.; and De Sousa, J. F. 2012. Ensemble Approaches for Regression: A Survey, *Acm Computing Surveys,* 45(1). 10.1145/2379776.2379786.

Miller, J. F. 2020. Cartesian genetic programming: its status and future, *Genetic Programming and Evolvable Machines,* 21(1): 129-168. 10.1007/s10710-019-09360-6.

Moyano, J. M.; Reyes, O.; Fardoun, H. M.; and Ventura, S. 2021. Performing multi-target regression via gene expression programming-based ensemble models, *Neurocomputing,* 432275-287. 10.1016/j.neucom.2020.12.060.

Peijin, G.; Wengiang, H.; Qinmiao, Z.; and Yuhui, W. 2023. Prediction algorithm of nugget diameter in resistance spot welding based on cascade forest, *Computer Integrated Manufacturing Systems,* 29(07): 2267-2276. 10.13196/j.cims.2023.07.012.

Shehzadi, A.; Budak, H.; Haider, W.; Mateen, A.; and Chen, H. 2025. Fractional Boole's inequalities for twice differentiable functions for Riemann-Liouville fractional integrals, *Journal of Applied Mathematics and Computing,* (71): 5781–5800. 10.1007/s12190-025-02465-5.

Uy, N. Q.; Hoai, N. X.; O'Neill, M.; McKay, R. I.; and Galván-López, E. 2011. Semantically-based crossover in genetic programming: application to real-valued symbolic regression, *Genetic Programming and Evolvable Machines,* 12(2): 91-119. 10.1007/s10710-010-9121-2.

Wang, J.; Xu, P.; Ji, X.; Li, M.; and Lu, W. 2023. MIC-SHAP: An ensemble feature selection method for materials machine learning, *Materials Today Communications,* 37. 10.1016/j.mtcomm.2023.106910.

Zhang, F.; Mei, Y.; Nguyen, S.; Zhang, M.; and Tan, K. C. 2021. Surrogate-Assisted Evolutionary Multitask Genetic Programming for Dynamic Flexible Job Shop Scheduling, *IEEE Transactions on Evolutionary Computation,* 25(4): 651-665. 10.1109/tevc.2021.3065707.

Zhang, F.; Mei, Y.; Su, N.; Tan, K. C.; and Zhang, M. 2022. Multitask Genetic Programming-Based Generative Hyperheuristics: A Case Study in Dynamic Scheduling, *IEEE Transactions on Cybernetics,* 52(10): 10515-10528. 10.1109/tcyb.2021.3065340.

Zhang, Y.; Li, X.; Hu, W.; and Yen, G. G. 2024. Multiexpression Symbolic Regression and Its Circuit Design Case, *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* 1-14. 10.1109/TSMC.2024.3519675.

Zhong, J.; Feng, L.; Cai, W.; and Ong, Y. S. 2020. Multifactorial Genetic Programming for Symbolic Regression Problems, *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* 50(11): 4492-4505. 10.1109/TSMC.2018.2853719.

Zhou, Z.; Yu, Y.; and Qian, C. 2019. *Evolutionary Learning: Advances in Theories and Algorithms*. Singapore : Springer Singapore. doi.org/10.1007/978-981-13-5956-9