Relative-Absolute Fusion: Rethinking Feature Extraction in Image-Based Iterative Method Selection for Solving Sparse Linear Systems

Kaiqi Zhang[1,2], Mingguan Yang[2], Dali Chang[1,2], Chun Chen[2], Yuxiang Zhang[2], Kexun He[1,3], Jing Zhao[1]

***Abstract*— Iterative method selection is crucial for solving sparse linear systems because these methods inherently lack robustness. Though image-based selection approaches have shown promise, their feature extraction techniques might encode distinct matrices into identical image representations, leading to the same selection and suboptimal method. In this paper, we introduce RAF (Relative-Absolute Fusion), an efficient feature extraction technique to enhance image-based selection approaches. By simultaneously extracting and fusing image representations as relative features with corresponding numerical values as absolute features, RAF achieves comprehensive matrix representations that prevent feature ambiguity across distinct matrices, thus improving selection accuracy and unlocking the potential of image-based selection approaches. We conducted comprehensive evaluations of RAF on SuiteSparse and our developed BMCMat (Balanced Multi-Classification Matrix dataset), demonstrating solution time reductions of 0.08s-0.29s for sparse linear systems, which is 5.86%-11.50% faster than conventional image-based selection approaches and achieves state-of-the-art (SOTA) performance. BMCMat is available at `https://github.com/zkqq/BMCMat`.**

I. Introduction

Solving sparse linear systems, as shown in Eq. 1, is a fundamental task in scientific computing.

$$\mathcal{A}x = b \tag{1}$$

where $\mathcal{A} \in \mathbb{R}^{n\times n}$ denotes a sparse coefficient matrix, $b \in \mathbb{R}^n$ represents the right-hand side vector, and $x \in \mathbb{R}^n$ is the unknown solution vector [1]. Such systems are typically solved through either direct methods [2] or iterative methods [3]. For large-scale systems, solving Eq. 1 is computationally intensive, especially accounting for over 70% of the total computation time in reservoir engineering [4]. This computational burden has prompted the widespread adoption of iterative methods, which inherently exhibit lower computational complexity. However, iterative methods demonstrate limited robustness. An appropriate iterative method can solve the system efficiently, whereas an unsuitable one may result in slow convergence or divergence. Unfortunately, selecting an optimal (fastest convergence) iterative method for a given linear system remains challenging, often relying on trial and error or expert intuition [5].

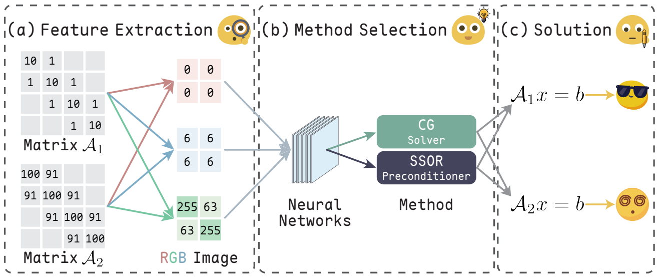


Fig. 1: Motivation for RAF. Conventional feature extraction techniques in image-based selection approaches encode distinct matrices $\mathcal{A}_1$ and $\mathcal{A}_2$ into identical RGB image representations, yielding the same selection method (*CG & SSOR*). Although the *CG & SSOR* method efficiently solves $\mathcal{A}_1x = b$, it exhibits poor performance for $\mathcal{A}_2x = b$.

Deep learning advances have driven research into selecting optimal iterative methods for sparse linear systems based on the features of matrix $\mathcal{A}$. Early studies utilized Fully Connected (FC) networks to select optimal methods based on manually identified numerical features from $\mathcal{A}$ [6]. Subsequent approaches modeled $\mathcal{A}$ as a topological graph and employed Graph Neural Network (GNN) for selection [7]. The SOTA work involves image-based selection, which encodes $\mathcal{A}$ as an RGB image and applies Convolutional Neural Network (CNN) for efficient selection [8], [9].

Although image-based selection approaches achieve SOTA performance, their feature extraction techniques may encode distinct matrices into identical RGB image representations, potentially leading to suboptimal method selection. As illustrated in Fig. 1, although *CG & SSOR* is deemed optimal for both matrices $\mathcal{A}_1$ and $\mathcal{A}_2$, it is only truly appropriate for the strongly diagonally dominant matrix $\mathcal{A}_1$, whereas the weakly diagonally dominant matrix $\mathcal{A}_2$ is solved more efficiently using *CG & ω-Jacobi*. This limitation stems from feature extraction techniques that compute RGB image representations, incorporating only relative matrix features while omitting absolute ones. For instance, the red channel computation biases non-zero elements by subtracting the matrix's minimum value (Eq. 3c) and subsequently normalizes the results (Eq. 3a), thus capturing only the relative magnitude relationships between matrix elements. The exclusion of absolute features may allow matrices with varying magnitudes to yield identical red channels, leading

[1] School of Software Technology, Dalian University of Technology, Dalian, China. {zhangkq, 569377793}@mail.dlut.edu.cn, zhaoj9988@dlut.edu.cn
[2] Greater Bay Area National Center of Technology Innovation, Guangzhou, China. {yangmingguan, chenchun, zhangyuxiang}@ncti-gba.cn
[3] CATARC Automotive Test Center (Tianjin) Co., Ltd, Tianjin, China. hekexun@catarc.ac.cn

to feature ambiguity and limiting the effectiveness of image-based selection approaches.

Rethinking feature extraction techniques in image-based selection approaches, we propose RAF, an efficient feature extraction technique that addresses existing limitations. RAF extracts both image representations as relative features and corresponding numerical values as absolute features, subsequently fusing these complementary features to achieve complete matrix representations and eliminate feature ambiguity across distinct matrices. For instance, when computing the red channel, RAF simultaneously extracts the matrix's minimum value as a bias reference, which characterizes matrix magnitude as an absolute feature, ensuring comprehensive representation of non-zero element magnitudes after fusion, preventing matrices with varying magnitudes from yielding identical features. Our proposed RAF mitigates information loss typically associated with purely relative features and enhances the efficiency of image-based selection approaches.

Our contributions can be summarized as follows:

- We developed BMCMat based on Partial Differential Equation (PDE) discretization to mitigate label imbalance in the widely used SuiteSparse dataset [10], facilitating research on iterative method selection for sparse linear systems (§ II-B).
- We introduce RAF, an efficient feature extraction technique that extracts and fuses relative and absolute matrix features to eliminate feature ambiguity across matrices, unlocking the potential of image-based method selection. (§ II-C, II-D)
- We comprehensively evaluated RAF on SuiteSparse and BMCMat, demonstrating selection accuracy improvements of 0.022-0.039 and solution time reductions of 0.08s-0.29s for linear systems, which is 5.86%-11.50% faster than conventional image-based selection approaches. To our best knowledge, RAF achieves SOTA performance in method selection (§ III-B).

II. Methodology

The efficacy of deep learning fundamentally depends on two critical factors: high-quality training data and effective algorithm design. This section enhances iterative method selection through dual innovations: improved dataset and optimized model.

A. Problem Formulation

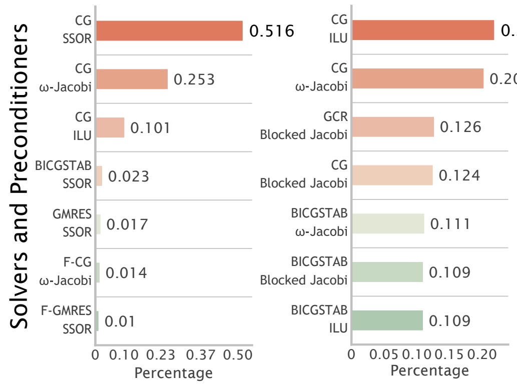
Iterative method selection fundamentally establishes a mapping f from a given matrix $\mathcal{A}$ to the optimal method, as shown in Eq. 2.

$$y = f(\mathcal{A}) \tag{2}$$

where y denotes an iterative method, comprising a solver and a preconditioner. Consequently, method selection can be formulated as a multi-class classification problem. For a set of linear systems with k available iterative methods, each matrix $\mathcal{A}$ is assigned an k-dimensional label vector $\{0,1\}^k$, where only the element corresponding to the optimal method is 1, and the remaining $k-1$ entries, corresponding to suboptimal methods, are 0.

TABLE I: Common iterative methods comprising various solvers and preconditioners [1], [11].

Iterative Methods			
Solvers		Preconditioners	
CG	F-CG	ω-Jacobi	Blocked Jacobi
GMRES	F-GMRES	G-S	SSOR
L-GMRES	BICG	GMG	AMG
GCR	BICGSTAB	DDM	ILU



(a) SuiteSparse. (b) BMCMat.

Fig. 2: Distribution of optimal iterative methods across datasets. For clarity, only seven methods with the highest percentages are listed for the SuiteSparse dataset.

B. BMCMat

Effective k-class classification in deep learning relies on high-quality training data due to its inherently data-driven nature.

We initially examined the widely adopted SuiteSparse dataset, from which we extracted 576 matrices $\mathcal{A} \in \mathbb{R}^{n\times n}$ with $1000 \leq n \leq 10000$ [7]. Using common iterative methods from the PETSc library [1], [11] listed in Table I, we determined the optimal method for each matrix. Notably, although 64 (8×8) theoretical solver-preconditioner combinations are possible, only a subset was computationally feasible, and an even smaller fraction demonstrated optimal performance. Finally, the method selection was formulated as a 25-class classification task, as illustrated in Fig. 2a, revealing significant class imbalance within the SuiteSparse. The *CG & SSOR* method constitutes over 50% of all cases, while 19 other methods individually represent less than 1% of the data. This imbalance introduces a selection bias toward dominant classes (e.g., *CG & SSOR*). Although such bias may yield high selection accuracy, it significantly compromises performance on minority classes, leading to suboptimal method selection for specific linear systems.

To address this limitation, we developed BMCMat, a relatively balanced multi-class classification Matrix dataset. Specifically, we generated 50,000 linear systems $\mathcal{A}x = b$ with $1000 \leq n \leq 10000$ using OpenMat [12], a parallel sparse matrix generator based on PDE discretization, and applied iterative methods in Table I to identify the optimal method for each system. We then extracted 3,819 matrices from these systems to construct BMCMat, which contains seven distinct method classes, as illustrated in Fig. 2b. Compared to SuiteSparse, BMCMat exhibits a more balanced class distribution, with a maximum class ratio of only 2:1. This balanced distribution improves feature learning across method classes, thereby enhancing the model's ability to accurately select optimal methods for diverse linear systems in practical applications.

C. RAF

In addition to high-quality training data, an efficient algorithm is crucial for k-class classification in deep learning.

Conventional feature extraction techniques in image-based selection approaches involve three sequential steps: (1) defining image resolution m, (2) partitioning matrix $\mathcal{A}$ into m^2 blocks, and (3) computing RGB channels for each pixel, representing its corresponding block. Each RGB channel captures distinct matrix features: the red channel represents non-zero element magnitudes (Eqs. 3a, 3b, 3c, 3d), the blue channel encodes matrix dimensions (Eq. 3e), and the green channel quantifies non-zero element density (Eq. 3f) [8], [9].

$$\left\{\begin{aligned}
R_{ij} &= \left\lfloor \frac{\gamma_{ij} - \min(\gamma)}{\max(\gamma) - \min(\gamma)} \times 255 \right\rfloor && \text{(3a)}\\
\gamma_{ij} &= \begin{cases} \dfrac{\sum_{a\in A_{ij}} v(a)}{NNZ_{ij}}, & \delta \le 255 \\ \dfrac{\sum_{a\in A_{ij}} \log_2 v(a)}{NNZ_{ij}}, & \delta > 255 \end{cases} && \text{(3b)}\\
v(a) &= a - \min(\mathcal{A}) + 1 && \text{(3c)}\\
\delta &= \max(\mathcal{A}) - \min(\mathcal{A}) && \text{(3d)}\\
B_{ij} &= \left\lfloor \frac{N_{\mathcal{A}} - N_{\min}}{N_{\max} - N_{\min}} \times 255 \right\rfloor && \text{(3e)}\\
G_{ij} &= \left\lfloor \frac{NNZ_{ij}}{{N_b}^2} \times 255 \right\rfloor && \text{(3f)}
\end{aligned}\right.$$

Here, indices $i, j \in \{1, \dots, m\}$ identify specific blocks within the partitioned matrix. δ represents the matrix value range, with $\min(\mathcal{A})$ and $\max(\mathcal{A})$ denoting the minimum and maximum values, respectively. $v(a)$ denotes the biased value of matrix element a. NNZ_{ij} counts the number of non-zero elements in block $\mathcal{A}_{ij}$, while γ_{ij} represents their biased average. $N_{\mathcal{A}}$ indicates the matrix order, while $N_{\min}$ and $N_{\max}$ denote the minimum and maximum matrix orders in the dataset, respectively. The order of each block, N_b, is approximated by $N_b \approx N_{\mathcal{A}}/m$.

Such feature extraction techniques that focus solely on relative relationships cannot completely represent matrices, leading to identical relative feature extraction for distinct matrices due to the absence of critical absolute features (Fig. 1), thus limiting the effectiveness of image-based selection approaches. This limitation arises primarily because matrices with varying magnitudes but identical ranges may yield identical red channel values owing to the bias (Eq. 3c) and normalization (Eq. 3a). Furthermore, linear and logarithmic block-wise averaging can generate identical γ values despite different biased inputs $v(a)$, further contributing to feature

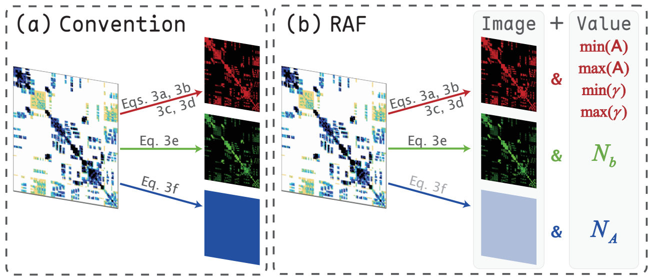


Fig. 3: Comparison between conventional feature extraction techniques (left) and RAF (right) in image-based selection approaches. RAF simultaneously extracts relative image representations and corresponding absolute numerical values, yielding comprehensive matrix features for enhanced characterization. Notably, because $N_{\mathcal{A}}$ demonstrates higher effectiveness than the blue channel, RAF retains only the red and green channels in the image representation.

ambiguity (Eqs. 3b, 3d). Moreover, the reliance of the blue and green channels on the values $N_{\min}$, $N_{\max}$, and N_b limits the accuracy of matrix representation.

To address these limitations, we propose RAF, an efficient feature extraction technique for enhancing image-based selection approaches. As illustrated in Fig. 3, RAF employs a dual-feature strategy that simultaneously extracts image representations as relative features along with corresponding numerical values as absolute features to create comprehensive matrix representations. For non-zero element magnitudes, RAF extracts both the red channel and the absolute boundary values $\min(\mathcal{A})$, $\max(\mathcal{A})$, $\min(\gamma)$, and $\max(\gamma)$ (Eqs. 3a, 3b) for complete characterization. For dimensionality, RAF uses the matrix order $N_{\mathcal{A}}$ (Eq. 3e) to fully characterize the matrix size. Interestingly, since $N_{\mathcal{A}}$ provides complete dimensional information, RAF eliminates the blue channel to reduce feature redundancy. For non-zero element density, RAF extracts both the green channel and the block order N_b (Eq. 3f) to characterize the sparsity pattern. For the matrices $\mathcal{A}_1$ and $\mathcal{A}_2$ shown in Fig. 1, when extracting the red and green channels, RAF also extracts $\min(\mathcal{A}_1) = 1$, $\max(\mathcal{A}_1) = 10$, $\min(\mathcal{A}_2) = 91$, and $\max(\mathcal{A}_2) = 100$ to distinguish between them and avoid feature ambiguity. Additionally, RAF extracts $\min(\gamma) = 1$, $\max(\gamma) = 5.5$, $N_b = 2$, and $N_{\mathcal{A}} = 4$ for a more precise characterization of the matrices. Finally, relative and absolute features are fused to select the optimal iterative method for linear systems (§ II-D). By extracting and fusing relative image representations with absolute numerical values, RAF achieves more complete matrix representations, enhancing image-based selection approaches to better differentiate matrices and improve selection accuracy.

D. Image-Based Method Selection with RAF

After introducing the efficient RAF, the next step is to consider its integration into the image-based method selection architecture.

As illustrated in Fig. 4, the image-based method selection with RAF comprises three components: (a) extracting

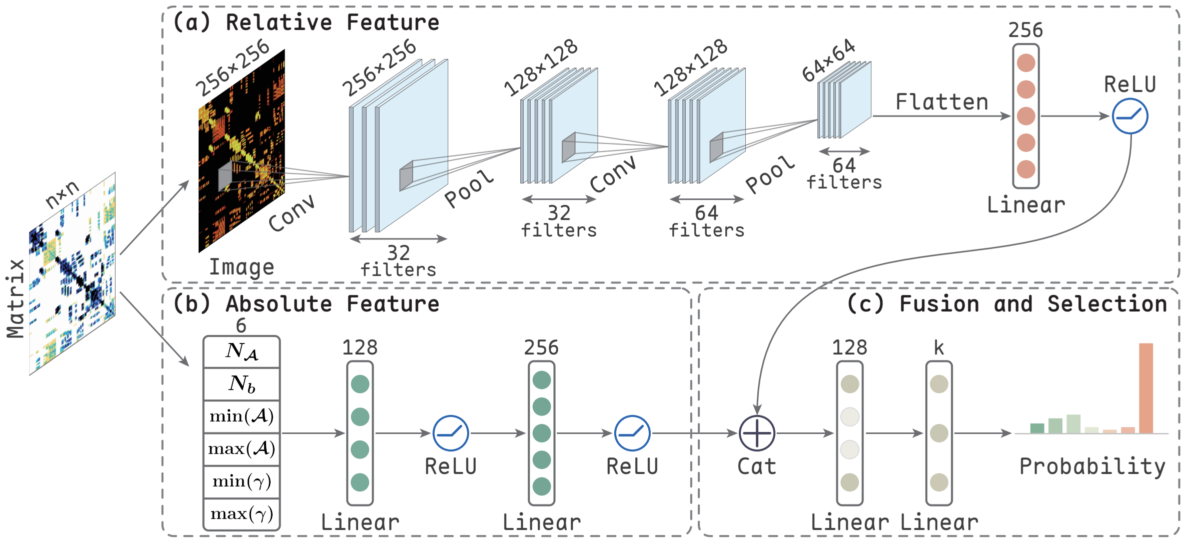


Fig. 4: Pipeline for image-based iterative method selection with RAF. Component (a) extracts the red and green channels as relative matrix features, which are subsequently learned via convolution to capture spatial patterns within the matrix. Component (b) simultaneously extracts six corresponding numerical values as absolute matrix features and learns them through linear layers. Finally, component (c) fuses these features to select the optimal iterative method for the given matrix.

and learning the matrix's relative features, (b) extracting and learning the matrix's absolute features, and (c) fusing features to select the optimal iterative method for the input matrix. As RAF, component (a) exclusively extracts red and green channel image representations as the matrix's relative features. The red and green channels of the 256×256 pixel image undergo transformed into a $64 \times 64 \times 64$ tensor via sequential operations: a 3×3 convolution (Conv) with 32 filters, 2×2 max pooling (Pool), followed by a 3×3 convolution with 64 filters, and a final 2×2 max pooling. To match the dimension of the absolute feature vector, this tensor is subsequently flattened and reduced to a 256-dimensional vector through a linear transformation with ReLU activation. Simultaneously, component (b) extracts six corresponding numerical values representing the matrix's absolute features, expanding this feature vector to 256 dimensions via two sequential linear transformation layers, each followed by ReLU activation. Finally, in component (c), the two 256-dimensional vectors (relative and absolute features) are concatenated (Cat) to form a unified 512-dimensional feature vector. This combined vector passes through two linear layers, with a dropout rate of 0.5 applied to the first layer to mitigate overfitting. The second layer contains k neurons, where k corresponds to the number of candidate methods in the dataset (SuiteSparse: $k = 25$; BMCMat: $k = 7$). The output represents probability distributions across all candidate methods, wherein the method exhibiting the highest probability is selected as optimal for the given matrix.

III. Experiment

A. Experimental Setup

Models and parameters. We evaluated four iterative method selection models: existing FC [6], GNN [7], CNN [8], [9], and our proposed image-based approach with RAF (simplified as RAF). All models were trained on an A6000 GPU using SuiteSparse [10] and BMCMat. For RAF, we employed a learning rate of 8e-4, a batch size of 64, and early stopping with a maximum of 100 epochs. All linear systems were solved using the PETSc library on an Intel Xeon 8352V CPU with 256 GB RAM.

Metrics. We evaluated model performance using the following quantitative metrics:

- Solution time (s): The walltime for solving linear systems using model-selected iterative methods.
- Slowdown: The ratio of optimal method computation time to the model-selected method time. Values closer to 1 indicate the selected method optimal performance.
- Selection Accuracy: The probability that the model correctly selects the optimal method.
- Top-n Selection Accuracy: The probability that the optimal method is included in the model's top n selections. Higher values indicate the model's ability to rank the optimal method among its top choices, demonstrating practical effectiveness.

TABLE II: Solution time and slowdown of existing models and **RAF** on SuiteSparse and BMCMat.

Model	SuiteSparse		BMCMat	
	Time↓	Slowdown↑	Time↓	Slowdown↑
FC	0.70	0.31	5.18	0.77
GNN	0.70	0.31	5.45	0.73
CNN	0.68 (Base)	0.32 (Base)	4.95 (Base)	0.81 (Base)
RAF	**0.61** (-0.07)	**0.36** (+0.04)	**4.66** (-0.29)	**0.86** (+0.05)

B. Effectiveness

Table II compares the solution times of four iterative method selection approaches on the SuiteSparse and BMCMat datasets. RAF demonstrates SOTA performance, achieving the fastest solution times among all evaluated approaches.

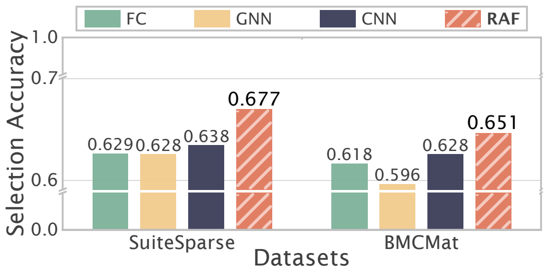


Fig. 5: Selection accuracy of FC, CNN, GNN, and **RAF** on SuiteSparse and BMCMat.

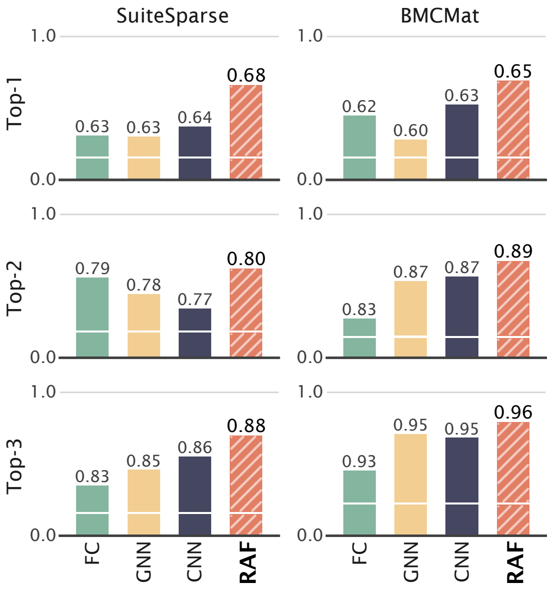


Fig. 6: Top-n selection accuracy of FC, CNN, GNN, and **RAF** on SuiteSparse and BMCMat.

On SuiteSparse, RAF reduces solution times by 0.08s-0.10s compared to competing models, yielding speedups of 1.13x-1.16x. For BMCMat, RAF decreases solution times by 0.29s-0.79s relative to alternative approaches, achieving speedups of 1.06x-1.17x. RAF's superior performance can be attributed to its ability to more effectively extract matrix features compared to conventional image-based approaches, thus charactering matrix completely and avoiding feature ambiguity. Table II further validates these findings by comparing the slowdown across all four approaches on both datasets. RAF achieves optimal slowdown of 0.36 and 0.86 on SuiteSparse and BMCMat, respectively, outperforming competitors by margins of 0.04-0.12, confirming its selection of near-optimal methods and further demonstrating its superiority.

Figs. 5 and 6 present the selection accuracy and top-n accuracy, where RAF achieves SOTA performance on both SuiteSparse and BMCMat. For selection accuracy, RAF outperforms existing models by margins of 0.039-0.048 on SuiteSparse and 0.022-0.055 on BMCMat with advanced feature extraction and learning mechanisms, directly translating to reduced solution times in practical applications. As further illustrated in Fig. 6, RAF consistently maintains its advantage in top-n selection accuracy, indicating superior selection quality and enhanced reliability for method selection.

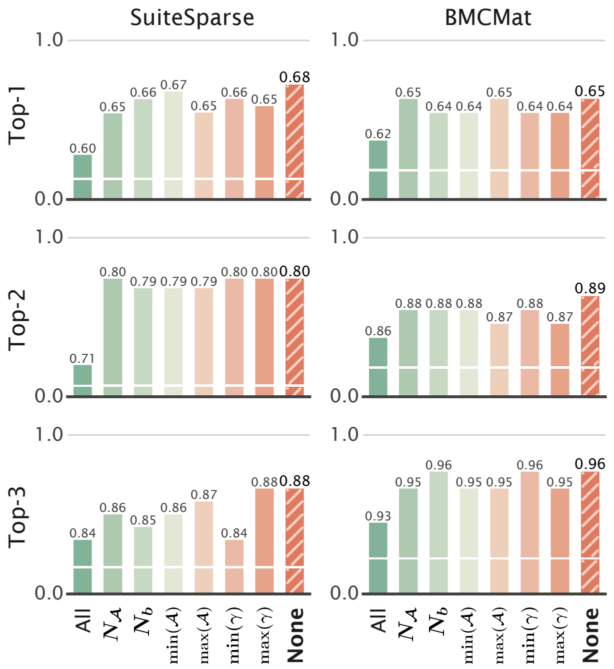


Fig. 7: Top-n selection accuracy of various **RAF** variants on SuiteSparse and BMCMat. The horizontal axis represents different configurations of numerical value exclusions. "All" refers to the **RAF** variant with all numerical values removed and "**None**" represents the complete **RAF**.

C. Ablation Study

Compared to conventional image-based method selection, the key innovation of RAF lies in the extraction and fusion of relative image representations with six absolute numerical values, enabling more precise matrix characterization and improved selection accuracy. We conducted an ablation study to quantify the contribution of each numerical value to RAF performance. Experimental results evaluated on SuiteSparse and BMCMat are presented in Table III and Fig. 7. Overall, removing any single numerical value degraded model performance, albeit to varying degrees. Compared to the complete RAF, removing a single numerical value decreased selection accuracy (top-1) by 0.006-0.030, increased solution time by 0.01s-0.19s, reduced computational efficiency by 0.98%-7.07%, and decreased slowdown by 0.004-0.034 across both datasets. These observations confirm that each numerical value in RAF contributes positively to model performance, collectively forming a complementary feature representation system. Furthermore, removing all numerical values significantly degraded model performance, resulting in performance below that of a conventional CNN. This underperformance occurs because, without numerical values, RAF is reduced to a conventional CNN lacking the blue channel features, further validating the contribution of matrix dimension ($N_{\mathcal{A}}$) features to method selection.

IV. Related Work

Early method selection predominantly utilized machine learning. Initially, Alternating Decision Trees were applied for method selection leveraging matrix features [13]. A sub-

TABLE III: Ablation study results showing selection accuracy, speedup, and slowdown for **RAF** variants on SuiteSparse and BMCMat. Numerical values are represented by symbols, with "✓" signifying presence and "✗" signifying absence. The best and worst performance are illustrated with corresponding colors. Results of complete **RAF** are displayed in **bold**.

	Numerical Value						Metric		
Dataset	$N_{\mathcal{A}}$	N_b	$\min(\mathcal{A})$	$\max(\mathcal{A})$	$\min(\gamma)$	$\max(\gamma)$	Selection accuracy ↑	Solution time ↓	Slowdown ↑
SuiteSparse	✓	✓	✓	✓	✓	✓	**0.677** (Base)	**0.605** (Base)	**0.364** (Base)
	✗	✓	✓	✓	✓	✓	0.647 (-0.030)	0.651 (+0.046)	0.338 (-0.026)
	✓	✗	✓	✓	✓	✓	0.661 (-0.016)	0.622 (+0.017)	0.354 (-0.010)
	✓	✓	✗	✓	✓	✓	0.669 (-0.008)	0.611 (+0.006)	0.360 (-0.004)
	✓	✓	✓	✗	✓	✓	0.648 (-0.029)	0.645 (+0.040)	0.341 (-0.023)
	✓	✓	✓	✓	✗	✓	0.662 (-0.015)	0.615 (+0.010)	0.358 (-0.006)
	✓	✓	✓	✓	✓	✗	0.654 (-0.023)	0.635 (+0.030)	0.346 (-0.018)
	✗	✗	✗	✗	✗	✗	0.603 (-0.074)	0.730 (+0.125)	0.301 (-0.063)
BMCMat	✓	✓	✓	✓	✓	✓	**0.651** (Base)	**4.660** (Base)	**0.858** (Base)
	✗	✓	✓	✓	✓	✓	0.645 (-0.006)	4.751 (+0.091)	0.842 (-0.016)
	✓	✗	✓	✓	✓	✓	0.642 (-0.009)	4.806 (+0.146)	0.832 (-0.026)
	✓	✓	✗	✓	✓	✓	0.643 (-0.008)	4.788 (+0.128)	0.835 (-0.023)
	✓	✓	✓	✗	✓	✓	0.645 (-0.006)	4.752 (+0.092)	0.842 (-0.016)
	✓	✓	✓	✓	✗	✓	0.644 (-0.007)	4.764 (+0.104)	0.840 (-0.018)
	✓	✓	✓	✓	✓	✗	0.638 (-0.013)	4.852 (+0.192)	0.824 (-0.034)
	✗	✗	✗	✗	✗	✗	0.617 (-0.034)	5.302 (+0.642)	0.754 (-0.104)

sequent study employed Decision Trees and proposed three strategies based on composite methods [14]. For transient simulations, WEKA and MULAN were used, implementing K-Nearest Neighbors, Random Forests, and Decision Trees [15]. The Lighthouse framework integrated several machine learning algorithms, including BayesNet and Random Forest, to select methods from the PETSc and Trilinos libraries [16].

Advances in deep learning have spurred research into its application for method selection, surpassing traditional machine learning due to its superior feature learning and handling of non-linear relationships. Early studies employed FC for solver selection with 18 matrix features [6]. Subsequent research modeled matrices as topological graphs with five node and ten graph features, using GNN for method selection [7]. Current SOTA approaches encode matrices as images, employing CNN to capture spatial patterns for method selection [8], [9]. In contrast, RAF enhances image-based approaches by extracting and fusing relative image representations with absolute numerical values, thereby comprehensively characterizing matrices and unlocking the potential of image-based iterative method selection.

V. Conclusion

In this paper, we introduce RAF, an efficient feature extraction technique that enhances image-based iterative method selection for solving sparse linear systems. RAF simultaneously extracts image representations as relative features and corresponding numerical values as absolute features, fusing them to enhance matrix representations, thereby improving selection accuracy and accelerating linear system solutions. Additionally, we developed BMCMat, a balanced matrix dataset constructed through PDE discretization to facilitate method selection research. We comprehensively evaluated RAF on both SuiteSparse and BMCMat, demonstrating its SOTA performance, with improved selection accuracy by 0.02-0.06 and reduced solution times by 0.08s-0.79s, yielding 1.06x-1.17x higher computational efficiency compared to existing method selection approaches.

References

[1] H. Zou, X. Xu, and C.-S. Zhang, "A survey on intelligent iterative methods for solving sparse linear algebraic equations," *arXiv preprint arXiv:2310.06630*, 2023.

[2] T. A. Davis, S. Rajamanickam, and W. M. Sid-Lakhdar, "A survey of direct methods for sparse linear systems," *Acta Numerica*, vol. 25, pp. 383–566, 2016.

[3] Y. Saad, *Iterative methods for sparse linear systems*. SIAM, 2003.

[4] L. Gasparini, J. R. Rodrigues, D. A. Augusto, L. M. Carvalho, C. Conopoima, P. Goldfeld, J. Panetta, J. P. Ramirez, M. Souza, M. O. Figueiredo *et al.*, "Hybrid parallel iterative sparse linear solver framework for reservoir geomechanical and flow simulation," *Journal of Computational Science*, vol. 51, p. 101330, 2021.

[5] J. Scott and M. Tůma, *Algorithms for sparse linear systems*. Springer Nature, 2023.

[6] Y. Funk, M. Götz, and H. Anzt, "Prediction of optimal solvers for sparse linear systems using deep learning," in *Proceedings of the 2022 SIAM Conference on Parallel Processing for Scientific Computing*. SIAM, 2022, pp. 14–24.

[7] Z. Tang, H. Zhang, and J. Chen, "Graph neural networks for selection of preconditioners and krylov solvers," in *NeurIPS 2022 Workshop: New Frontiers in Graph Learning*, 2022.

[8] K. Yamada, T. Katagiri, H. Takizawa, K. Minami, M. Yokokawa, T. Nagai, and M. Ogino, "Preconditioner auto-tuning using deep learning for sparse iterative algorithms," in *2018 Sixth International Symposium on Computing and Networking Workshops (CANDARW)*. IEEE, 2018, pp. 257–262.

[9] M. Souza, L. M. Carvalho, D. Augusto, J. Panetta, P. Goldfeld, and J. R. Rodrigues, "A comparison of image and scalar-based approaches in preconditioner selection," *arXiv preprint arXiv:2312.15747*, 2023.

[10] T. A. Davis and Y. Hu, "The university of florida sparse matrix collection," *ACM Transactions on Mathematical Software (TOMS)*, vol. 38, no. 1, pp. 1–25, 2011.

[11] PETSc, "Summary of sparse linear solvers available in PETSc," https://petsc.org/main/overview/linear_solve_table, 2025.

[12] H. Zhang and C. Zhang, "OpenMat," https://github.com/zhf-0/OpenMat, 2024.

[13] S. Bhowmick, V. Eijkhout, Y. Freund, E. Fuentes, and D. Keyes, "Application of machine learning to the selection of sparse linear solvers," *Int. J. High Perf. Comput. Appl*, 2006.

[14] V. Eijkhout and E. Fuentes, "Machine learning for multi-stage selection of numerical methods," *New Advances in Machine Learning. INTECH*, pp. 117–136, 2010.

[15] P. R. Eller, J.-R. C. Cheng, and R. S. Maier, "Dynamic linear solver selection for transient simulations using multi-label classifiers," *Procedia Computer Science*, vol. 9, pp. 1523–1532, 2012.

[16] P. Motter, K. Sood, E. Jessup, and B. Norris, "Lighthouse: an automated solver selection tool," in *Proceedings of the 3rd International Workshop on Software Engineering for High Performance Computing in Computational Science and Engineering*, 2015, pp. 16–24.