SPECTRA: Revitalizing Image-Based Iterative Method Selection for Sparse Linear Systems

Kaiqi Zhang[a,b], Dali Chang[a,b], Mingguan Yang[b], Jing Zhao[a,*] and Wangdong Yang[c,d]

[a]School of Software, Dalian University of Technology, Dalian, Liaoning, 116620, China
[b]Greater Bay Area National Center of Technology Innovation, Guangzhou, Guangdong, 510535, China
[c]College of Computer Science and Electronic Engineering, Hunan University, Changsha, Hunan, 410082, China
[d]The National Supercomputing Center in Changsha, Changsha, Hunan, 410082, China

ARTICLE INFO

Keywords:
Sparse linear systems
Iterative method selection
Expanded-channel encoding
Right-hand-side vector
Diagonally-Enhanced Transformer

ABSTRACT

Iterative method selection for efficiently solving sparse linear systems has been a critical and longstanding challenge in science and engineering. Although image-based selection approaches showed initial promise, their prominence has been diminished by several fundamental limitations: (a) the reductionism of the encoding scheme collapsing underlying numerical fidelity, (b) the omission of the right-hand-side (RHS) vector precluding complete system characterization, and (c) the inductive bias of convolutions towards local patterns obscuring long-range dependencies. To address these limitations, we introduce SPECTRA (Systemic Problem Encoding and Contextual TRansformer Architecture), a novel framework designed to revitalize image-based iterative method selection through three key innovations: (a) a discriminative expanded-channel encoding scheme faithfully capturing numerical distributions, (b) the pioneering incorporation of RHS-awareness holistically representing the problem, and (c) a Diagonally-Enhanced Transformer module globally modeling dependencies guided by numerical priors. Comprehensive evaluations on the diverse SuiteSparse dataset demonstrate that SPECTRA establishes a new state-of-the-art. In the conventional fixed-RHS setting, SPECTRA surpasses leading selection approaches, improving top-1 selection accuracy, solution success rate, and computational speedup by 2.12%, 0.24%, and 0.46×, respectively. These gains are substantially amplified in more realistic variable-RHS scenarios, where the corresponding improvements reach 17.92%, 4.59%, and 2.78×. Notably, SPECTRA also exhibits superior generalization across diverse problem domains and system scales, underscoring its robustness and scalability.

1. Introduction

Solving sparse linear systems of the following form is ubiquitous in numerous scientific and engineering domains, with crucial applications in fields such as fluid dynamics [1], structural analysis [2], and electromagnetic simulation [3].

$$Ax = b. \tag{1}$$

Here, $A \in \mathbb{R}^{N\times N}$ is a sparse coefficient matrix, $b \in \mathbb{R}^N$ is the right-hand-side (RHS) vector, and $x \in \mathbb{R}^N$ is the unknown solution vector. Such systems are typically solved using either direct [4, 5] or iterative methods [6, 7]. The substantial memory and computational demands of direct methods limit their applicability, spurring the widespread adoption of iterative methods [8]. However, iterative methods exhibit inherently limited robustness, as their performance is closely related to the system's numerical properties. For a given system, a suitable method may converge rapidly, whereas an inappropriate choice can lead to slow convergence or even divergence [9]. Unfortunately, selecting the optimal iterative method for efficient solution remains a critical challenge, primarily due to the vast number of available methods and the frequent deviation of their practical performance from theoretical analysis, necessitating expert intuition and empirical evaluation [9–12]. Consequently, a long-standing objective has been to ease the reliance on such heuristics by developing an intelligent decision system that automates the selection of optimal iterative methods for efficiently solving sparse linear systems [13–16].

Automated iterative method selection initially employed machine learning using numerical features [11, 17–23], a paradigm subsequently surpassed by deep learning due to its superior representation learning capabilities. Image-based approaches emerged as a promising initial direction, demonstrating the potential to learn structural features by encoding the matrix as an image for Convolutional Neural Networks (CNNs) [24, 25]. However, the prominence of this paradigm was challenged by alternative strategies, such as scalar-based approaches with fully connected (FC) networks [26] and graph-based approaches using Graph Neural Networks (GNNs) [10], fostering a consensus that the image-based paradigm alone was insufficient. This consensus spurred the development of current state-of-the-art (SOTA) multi-modal approaches like MM [27] and RAF [28] that achieve superior performance by fusing numerical and structural features, relegating the once-pioneering image-based paradigm to a component within more complex frameworks.

However, we contend that this relegation stems not from an inherent weakness in the image-based approach, but from critical research gaps in prior implementations. These gaps span the entire selection pipeline, ranging from feature extraction (transforming linear systems into structured features) to pattern recognition (identifying predictive patterns for method selection), and are delineated as follows:

*Corresponding author
zhaoj9988@dlut.edu.cn (J. Zhao)

Gap (a): Numerical distortion from matrix encoding. Within feature extraction, conventional image-based approaches are undermined by the reductionist nature of their encoding schemes, which collapses the underlying numerical fidelity of the matrix. This flaw stems from a design that condenses the numerical distribution within a matrix block into a single scalar average, thereby discarding crucial information regarding its dynamic range and value distribution, which leaves numerically distinct blocks indiscernible. For instance, a block of uniform entries like $\left(\begin{smallmatrix}5&5\\5&5\end{smallmatrix}\right)$ and another containing a large outlier such as $\left(\begin{smallmatrix}1&1\\1&17\end{smallmatrix}\right)$ are rendered indistinguishable, as both are represented by their shared mean value of 5. This loss of numerical fidelity prevents the approach from discerning critical differences between distinct linear systems, on which the optimal selection depends, and fundamentally limits its achievable performance.

Gap (b): Characterization incompleteness from RHS omission. Another critical limitation of feature extraction in prevailing approaches is their omission of the RHS, which precludes a complete characterization of the linear system. Relying solely on features derived from the matrix introduces an ill-posed learning problem, as a single matrix can map to distinct optimal methods depending on the unseen RHS. For a given non-symmetric matrix, a smooth RHS may render GMRES the superior choice, whereas an oscillatory RHS can make BiCGSTAB more effective. This blindness to decisive performance factors renders matrix-only approaches fundamentally incapable of resolving these trade-offs, thus leading to frequent suboptimal selections.

Gap (c): Long-range obscurity from convolutional bias. For pattern recognition, standard image-based approaches are constrained by the inherent inductive bias of convolutions toward local patterns, which obscures crucial long-range dependencies. The predisposition on local kernels hinders the efficient perception of the global structures decisive for method selection. For example, though convolutions can recognize the local stencil of a convection-diffusion matrix, they fail to capture its global advection-domina ted nature, a critical property that dictates whether a symmetric method like CG suffices or a non-symmetric one like GMRES is required. This architectural myopia regarding the global context ultimately yields selections lacking the robustness and generalization for diverse and complex linear systems.

To resolve these limitations in both feature extraction and pattern recognition, we introduce SPECTRA, a novel framework illustrated in Fig. 1, to revitalize image-based iterative method selection through several key contributions:

- We propose a discriminative expanded-channel encoding scheme to mitigate the numerical distortion in conventional RGB image encoding. This scheme augments the mean-based red channel with two additional channels, designated peak and valley, which respectively encode the maximum and minimum values within each matrix block. For instance, although blocks $\left(\begin{smallmatrix}5&5\\5&5\end{smallmatrix}\right)$ and $\left(\begin{smallmatrix}1&1\\1&17\end{smallmatrix}\right)$ share an identical mean, SPECTRA differentiates them by extracting distinct peak and valley pairs of (5, 5) and (17, 1), respectively. Through complementing the mean-based red channel with peak and valley channels, this expanded scheme captures the numerical distribution faithfully, thereby ensuring more fine-grained selections.

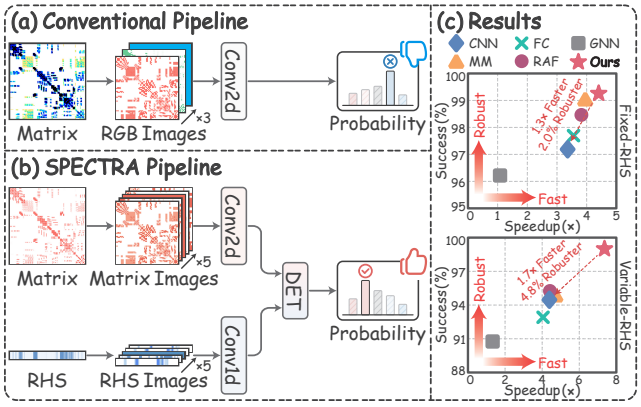

Figure 1: Comparison of SPECTRA with conventional image-based iterative method selection. (a) Conventional approaches are limited by reductionist RGB image encoding, matrix-only characterization, and locality-biased CNNs. (b) SPECTRA introduces faithful expanded five-channel encoding, holistic RHS-awareness, and a global Diagonally-Enhanced Transformer (DET). (c) SPECTRA establishes SOTA performance, with its superiority substantially amplified in realistic variable-RHS scenarios compared to the standard fixed-RHS setting.

- To the best of our knowledge, we pioneer incorporating RHS-awareness to address the characterization incompleteness of prior approaches. RHS-awareness is realized by encoding the RHS into a five-channel image analogous to the matrix, thereby unifying both feature sets for joint analysis. For a given non-symmetric matrix, SPECTRA informs the trade-off between methods like GMRES and BiCGSTAB by distinguishing the visual patterns of a gentle, gradient-like image of a smooth RHS from the sharp, periodic texture of an oscillatory one. Through leveraging previously inaccessible RHS information, this RHS-awareness represents the problem holistically, thus enabling more system-aware selections.

- We introduce a Diagonally-Enhanced Transformer (DET) module to overcome the long-range obscurity in standard image-based pattern recognition. DET captures global structures via self-attention, while integrating crucial numerical priors by reinforcing the significance of the matrix diagonal. Leveraging explicit diagonal tokens to guide its self-attention, SPECTRA discerns the global advection-dominated nature of a convection-diffusion system, determining the selection between CG and GMRES. Through synthesizing self-attention with numerical priors, DET models dependencies globally, in turn leading to more structurally-informed selections.

- Comprehensive evaluations on the diverse SuiteSparse dataset [29] substantiate SPECTRA's superior performance, demonstrating that it not only revitalizes the image-based iterative method selection but also establishes a new SOTA. In the conventional fixed-RHS setting, SPECTRA significantly outperforms the standard image-based approach, improving top-1 selection accuracy by 6.73%, solution success rate by 2.01%, and computational speedup by 1.05×. These performance gains are dramatically magnified in the more challenging and realistic variable-RHS scenarios, where the corresponding improvements reach 18.7%, 4.81%, and 2.97×, respectively. Remarkably, SPECTRA exhibits superior generalization across diverse problem domains and out-of-distribution system scales, underscoring its robustness and scalability.

2. Preliminaries and motivations

2.1. Iterative methods for sparse linear systems

Iterative methods solve sparse linear systems by starting with an initial guess and progressively refining the solution until a termination criterion (e.g., achieving a desired tolerance or exceeding a maximum number of iterations) is met [9]. In practice, these methods are typically accelerated through preconditioning, which transforms the original system into a better-conditioned, equivalent form [6, 7]:

$$M^{-1}Ax = M^{-1}b, \tag{2}$$

where M is the preconditioner. Algorithm 1 outlines the general framework for such preconditioned iterative methods, highlighting two distinct yet synergistic components:

- **Preconditioner** (Line 5) generates the preconditioned residual z from the linear system $Mz = r$.
- **Solver** (Line 6) computes the solution update Δx from the preconditioned residual z.

The landscape of available solvers and preconditioners is vast, with a variety of common options illustrated in Table 1.

The performance of an iterative method hinges on the chosen solver-preconditioner pair, whose effectiveness is so highly problem-dependent that no single combination is universally superior [9]. This inherent lack of robustness necessitates selecting a suitable pair to achieve efficient solution, a process that presents a formidable challenge. The combinatorial complexity and theoretical unreliability collectively create a pressing need for automated selection.

Automated iterative method selection has conventionally focused on matrix properties, such as dimension [26], trace [11], and sparsity pattern [24, 25]. However, the role of the RHS is equally critical yet overlooked [6, 7]. For Krylov solvers, the RHS fundamentally shapes the search space, as the solution is constructed within the Krylov subspace $\mathcal{K}_k(M^{-1}A, M^{-1}r^0) = span\{M^{-1}r^0, (M^{-1}A)(M^{-1}r^0), \dots, (M^{-1}A)^{k-1}(M^{-1}r^0)\}$, which is defined by the initial residual $r^0 = b - Ax^0$ [30–35]. This sensitivity also extends to the preconditioning step, where subsequent residuals $r^{(k-1)}$, derived from the RHS, influence the transient effectiveness of stationary methods (e.g., Jacobi) and even the smoothing operations within advanced preconditioners (e.g., AMG) [6]. Further theoretical analysis of the RHS and residual with respect to the Krylov subspace and convergence behavior is provided in Section A. Therefore, developing a truly accurate and robust selection approach requires transcending a matrix-only perspective to adopt a holistic system representation that accounts for the intricate interplay between the matrix and the RHS.

Algorithm 1: General framework of preconditioned iterative methods for sparse linear systems with a relative residual stopping criterion.

Require: Matrix $A \in \mathbb{R}^{N\times N}$, RHS $b \in \mathbb{R}^N$, initial guess $x^{(0)} \in \mathbb{R}^N$, preconditioner M, relative tolerance $\tau > 0$, maximum number of iterations $S \in \mathbb{N}$.
Ensure : Approximate solution x to $Ax = b$.

1 ▷ Compute initial residual
2 $r^{(0)} \leftarrow b - Ax^{(0)}$;
3 ▷ Main iteration loop
4 **for** $s \leftarrow 1$ **to** S **do**
5 Solve residual $z^{(s-1)}$ via $Mz^{(s-1)} = r^{(s-1)}$;
6 Compute correction $\Delta x^{(s)}$ using $z^{(s-1)}$;
7 ▷ Update solution
8 $x^{(s)} \leftarrow x^{(s-1)} + \Delta x^{(s)}$;
9 ▷ Update residual
10 $r^{(s)} \leftarrow b - Ax^{(s)}$;
11 ▷ Check convergence
12 **if** $\|r^{(s)}\|_2 \le \tau\|r^{(0)}\|_2$ **then**
13 **return** $x^{(s)}$;
14 ▷ Maximum iterations reached without convergence
15 **return** $x^{(S)}$;

Table 1
Common solvers and preconditioners for constructing iterative methods in the PETSc library [9, 36–38].

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

2.2. Image-based iterative method selection

Image-based iterative method selection, which encodes a matrix into an image for feature extraction and applies CNNs for pattern recognition, initially showed promise but has since been challenged by alternative approaches [24, 25].

Conventional feature extraction encodes the matrix into an RGB image through three steps: (a) defining the image resolution Q, which dictates the representation's granularity, (b) partitioning the matrix $A \in \mathbb{R}^{N\times N}$ into a $Q \times Q$ grid of

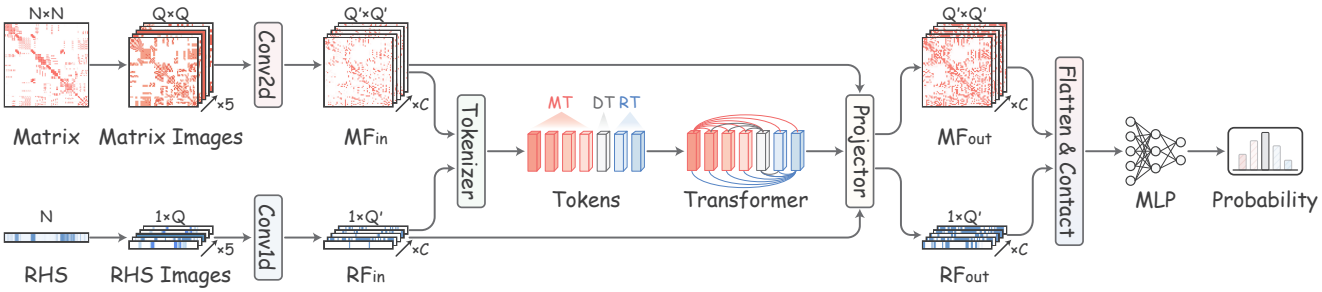

Figure 2: Pipeline for SPECTRA. The matrix and RHS are encoded into expanded five-channel images and processed with 2D/1D convolutions to yield initial C-channel feature maps (MF_{in}, RF_{in}) at resolution Q'. These maps are then fed into the DET module, which models global dependencies and consists of: a tokenizer converting the maps into matrix tokens (MT), RHS tokens (RT), and a diagonal token (DT) as numerical priors; a transformer capturing global dependencies among these tokens; and a projector mapping the processed tokens back into the feature space and fusing them with the initial maps to yield refined features (MF_{out}, RF_{out}). Finally, these refined features are flattened, concatenated, and passed through an MLP to predict the optimal method.

blocks, where each block $\mathcal{A}_{ij}$ corresponds to a pixel in the resulting image, and (c) computing the RGB channels for each pixel to extract distinct matrix features. Specifically, the RGB channels are defined as follows:

- **Red Channel (R)** captures the average magnitude of non-zero elements. For each pixel, R_{ij} is computed by normalizing r_{ij}, where r_{ij} is the average of the biased value $\eta(a)$ over the NNZ_{ij} non-zero elements a within the corresponding block $\mathcal{A}_{ij}$:

$$\begin{cases} R_{ij} = \left\lfloor \dfrac{r_{ij} - \min(r)}{\max(r) - \min(r)} \times 255 \right\rfloor, \\ r_{ij} = \begin{cases} \dfrac{\sum_{a \in \mathcal{A}_{ij}} \eta(a)}{\mathrm{NNZ}_{ij}}, & \delta \le 255 \\ \dfrac{\sum_{a \in \mathcal{A}_{ij}} \log_2 \eta(a)}{\mathrm{NNZ}_{ij}}, & \delta > 255 \end{cases}, \\ \eta(a) = a - \min(A) + 1, \\ \delta = \max(A) - \min(A). \end{cases} \tag{3}$$

- **Green Channel (G)** quantifies the density of non-zero elements. For each pixel, G_{ij} is determined by the ratio of the number of non-zero elements NNZ_{ij} within a block to its total capacity γ^2, where $\gamma \approx N/Q$:

$$G_{ij} = \left\lfloor \frac{\mathrm{NNZ}_{ij}}{\gamma^2} \times 255 \right\rfloor. \tag{4}$$

- **Blue Channel (B)** encodes the scale of the matrix. For each matrix, B_{ij} is uniform across all pixels and is calculated by normalizing its scale N against the minimum N_{min} and maximum N_{max} scales observed across the entire dataset:

$$B_{ij} = \left\lfloor \frac{N - N_{min}}{N_{max} - N_{min}} \times 255 \right\rfloor. \tag{5}$$

This encoding scheme transforms matrix features into a structured format amenable to pattern recognition, which conventional approaches employ CNNs to predict the optimal probability distribution over the available solver-preconditioner pairs. However, CNNs are founded on the convolution, which applies learnable kernels to compute weighted sums of pixels within local receptive fields, creating a strong inductive bias towards local patterns [39–43]. Although this bias is effective for natural image tasks that rely on local texture and shape, it becomes a bottleneck when long-range dependencies are crucial for understanding the system's global structure decisive for method selection.

3. SPECTRA

Fig. 2 illustrates the pipeline of SPECTRA, founded on three core innovations, each engineered to resolve a fundamental limitation of conventional image-based approaches:

- A discriminative expanded-channel encoding scheme captures the numerical distribution faithfully, thereby mitigating the numerical distortion in conventional RGB image encoding (Section 3.2).
- The pioneering incorporation of RHS-awareness representing the problem holistically, thus addressing the characterization incompleteness of prior matrix-only approaches (Section 3.3).
- A DET module, guided by numerical priors, models dependencies globally, in turn overcoming the long-range obscurity in standard image-based pattern recognition (Section 3.4).

3.1. Problem formulation

The primary goal of iterative method selection is to establish a mapping f from a given sparse linear system to its optimal iterative method [9, 10], formulated as follows:

$$e^* = f(A, b), \tag{6}$$

where the input comprises the matrix A and the RHS b of the system. Notably, in contrast to prior work that focuses exclusively on the matrix, our incorporation of the RHS facilitates

a more holistic system representation [10, 11, 24–28]. The output e^* is the optimal method selected from a predefined set of K candidate methods $E = \{e_1, \ldots, e_K\}$, where each method e_k corresponds to a unique solver-preconditioner pair from Table 1 and achieves optimal performance for at least one system in the dataset. Consequently, we formulate the selection task as a joint K-way classification problem over the set E, rather than predicting the solver and preconditioner independently. We emphasize that no solver-preconditioner pair is universally optimal across all systems. Instead, the objective is to learn an instance-wise and system-dependent optimum within E.

The optimal method e^* is formally defined as the method that minimizes the wall-clock solution time $\mathcal{T}_S(A, b, e)$ under prescribed stopping criteria determined by a relative tolerance and a maximum iteration limit:

$$e^* = \arg\min_{e \in E} \mathcal{T}_S(A, b, e). \tag{7}$$

Here, $\mathcal{T}_S(A, b, e)$ denotes the end-to-end runtime and explicitly encompasses both the preconditioner setup and iterative solving times. We assign $\mathcal{T}_S(A, b, e) = \infty$ for methods that fail to converge within the limit, thereby ensuring their exclusion from the selection process.

Since the objective is to select a single method e^* from the finite set E, this task is formulated as a multi-class classification problem. For each system, the ground-truth label is a one-hot vector $y \in \{0, 1\}^K$, where the element for the optimal method is 1, while all others are 0:

$$y_i = \begin{cases} 1, & \text{if } i = \arg\min\limits_{k \in \{1,\ldots,K\}} \mathcal{T}_S(A, b, e_k) \\ 0, & \text{otherwise} \end{cases}. \tag{8}$$

The mapping f is then learned by training a selection approach to minimize a loss function between its predicted probability distribution $\hat{y}$ and the ground-truth label y.

3.2. Expanded-channel encoding scheme

To mitigate the numerical distortion in conventional RGB image encoding, we propose a discriminative expanded-channel encoding scheme that augments the standard RGB channels with two additional Peak (P) and Valley (V) channels to capture numerical distributions faithfully, as illustrated in the top row of Fig. 3 and defined as follows:

- **Peak Channel (P)** identifies the maximum magnitude of non-zero elements. For each pixel, P_{ij} is obtained by normalizing p_{ij}, which represents the maximum value among non-zero elements a within $\mathcal{A}_{ij}$:

$$\begin{cases} P_{ij} = \left\lfloor \dfrac{p_{ij} - \min(p)}{\max(p) - \min(p)} \times 255 \right\rfloor, \\ p_{ij} = \max_{a \in \mathcal{A}_{ij}}(a). \end{cases} \tag{9}$$

- **Valley Channel (V)** captures the minimum magnitude of non-zero elements. Similarly, V_{ij} is derived by normalizing the minimum non-zero element v_{ij}:

$$\begin{cases} V_{ij} = \left\lfloor \dfrac{v_{ij} - \min(v)}{\max(v) - \min(v)} \times 255 \right\rfloor, \\ v_{ij} = \min_{a \in \mathcal{A}_{ij}}(a). \end{cases} \tag{10}$$

Figure 3: Expanded-channel encoding for the matrix (top) and the RHS (bottom). For the matrix, the five channels encode: (R) the average magnitude of non-zero elements, (G) the density of non-zero elements, (B) the scale of the matrix, along with our proposed (P) peak and (V) valley magnitude of non-zero elements. An analogous five-channel encoding scheme is applied to the RHS for feature extraction.

The detailed computational procedure for the expanded-channel matrix encoding scheme is provided in Appendix B.

3.3. RHS encoding scheme

To address the characterization incompleteness of matrix-only approaches, we encode the RHS into a five-channel 1D image via a matrix-analogous scheme, thereby establishing the foundation for RHS-awareness and representing the system holistically, as depicted in the bottom row of Fig. 3.

Specifically, this encoding process consists of three sequential steps: (a) adopting the same image resolution Q as the matrix to ensure representational consistency, (b) partitioning the RHS into Q 1D segments, where each segment corresponds to a pixel in the resulting $1 \times Q$ image, and (c) computing the five channels (R, G, B, P, V) for each pixel to extract distinct RHS features by adapting the operations defined in Eqs. 3-5, 9 and 10 from 2D matrix blocks to the 1D vector segments.

By encoding both the matrix and the RHS into analogous images, a unified feature set is established, which is crucial for enabling joint analysis in subsequent pattern recognition.

3.4. DET

To overcome the long-range obscurity in standard image-based pattern recognition, we introduce the DET module, which complements preceding convolutions by employing self-attention [44] to capture the global dependencies across the matrix, the RHS, and their intricate interplay. Our insight is to leverage the synergistic strengths of both convolutions and transformers: (a) early in the backbone, CNNs are employed to capture densely-distributed, low-level patterns, and (b) later in the network, DET module is utilized to model the global relationships among more sparsely-distributed, higher-order semantic concepts. Similar paradigms can be

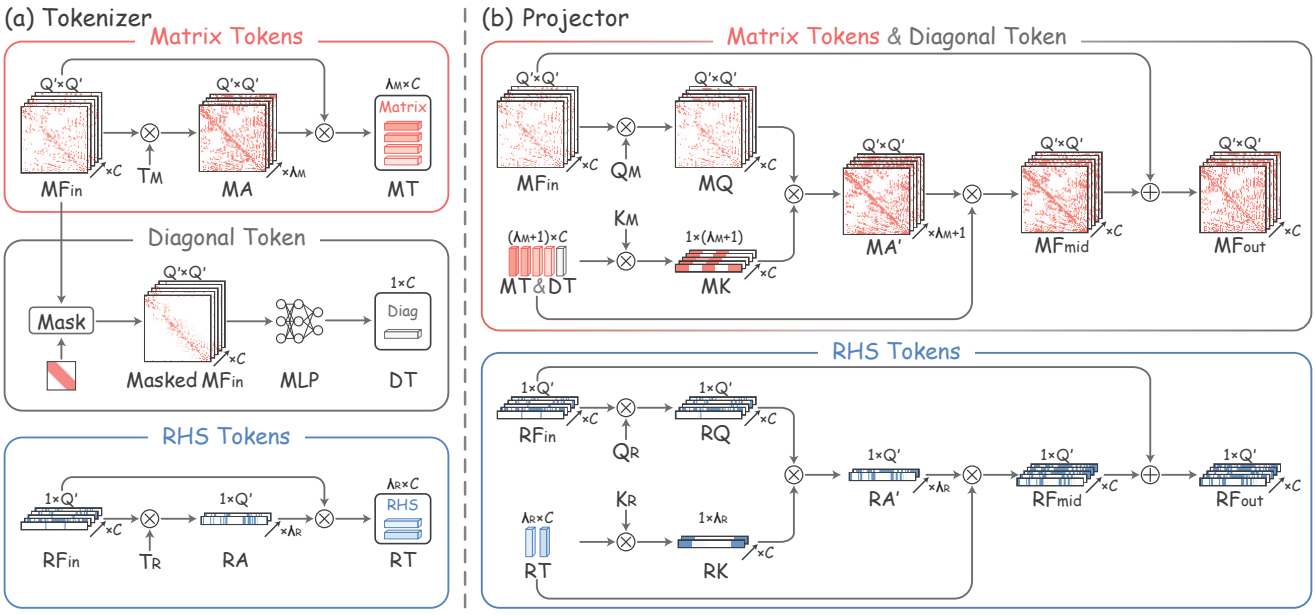

Figure 4: (a) Pipeline for diagonally-enhanced tokenizer. Tokenizer converts initial feature maps (MF_{in}, RF_{in}) into matrix tokens (MT), a diagonal token (DT), and RHS tokens (RT). MT and RT are generated via an attention mechanism that distills features from MF_{in} and RF_{in}, respectively. DT is constructed by applying a diagonal mask to MF_{in} and processing with an MLP. **(b) Pipeline for projector.** Projector maps output tokens from the Transformer back into feature space via a reverse attention mechanism, where MF_{in} and RF_{in} serve as the queries and the contextually-enriched tokens are the keys and values. In the matrix stream, MT and DT are utilized to compute an intermediate feature map (MF_{mid}), which is subsequently fused with MF_{in} via a residual connection to produce the refined MF_{out}. An analogous process is applied to the RHS stream to generate refined RF_{out}.

found in [45–50] but with one critical difference: prior works focus on standard image classification, whereas DET is specifically designed for iterative method selection by integrating the crucial numerical prior regarding the significance of the matrix diagonal to guide attention.

DET processes the initial, locally-aware feature maps MF_{in} and RF_{in} from the convolution layers, yielding refined, globally-aware counterparts MF_{out} and RF_{out} through the synergistic interplay of three core components:

- **Diagonally-enhanced tokenizer** converts the initial feature maps MF_{in} and RF_{in} into a sequence of tokens while embedding the diagonal prior (Section 3.4.1).
- **Transformer** processes these tokens via self-attention to model global dependencies of the matrix, the RHS, and their intricate interplay (Section 3.4.2).
- **Projector** maps the processed tokens back into feature space and fuses them with the initial maps to yield refined feature maps MF_{out} and RF_{out} (Section 3.4.3).

3.4.1. Diagonally-enhanced tokenizer

To convert the initial feature maps into a token sequence amenable to the Transformer architecture, SPECTRA employs a diagonally-enhanced tokenizer, as illustrated in Fig. 4(a). Unlike standard tokenizers that treat all pixels uniformly [45, 46], our tokenizer is specifically engineered to embed critical numerical priors by constructing a specialized diagonal token, guiding the subsequent attention towards structurally significant information, which is crucial to understand the decisive properties of the system.

The tokenizer generates three distinct sets of tokens via parallel pathways:

- **Matrix tokens (MT)** are designed to distill features from $MF_{in} \in \mathbb{R}^{Q'^2 \times C}$ into a set of representative clusters. Unlike fixed patching [45, 51, 52], we employ an attention-based mechanism from [46] for data-driven feature grouping. An attention map $MA \in \mathbb{R}^{Q'^2 \times \lambda_M}$ is first computed via a learnable projection matrix $T_M \in \mathbb{R}^{C \times \lambda_M}$ to group the pixels of MF_{in} into λ_M clusters. The resulting λ_M matrix tokens $MT \in \mathbb{R}^{\lambda_M \times C}$ are then generated as the weighted average of MF_{in}, with weights provided by MA. Formally,

$$MT = \underbrace{\text{Softmax}\left(MF_{in} \cdot T_M\right)^T}_{MA \in \mathbb{R}^{Q'^2 \times \lambda_M}} MF_{in}. \tag{11}$$

- **RHS tokens (RT)** are generated analogously to summarize features from $RF_{in} \in \mathbb{R}^{Q' \times C}$. A learnable projection matrix $T_R \in \mathbb{R}^{C \times \lambda_R}$ is applied to compute an attention map $RA \in \mathbb{R}^{Q' \times \lambda_R}$, which in turn yields λ_R RHS tokens $RT \in \mathbb{R}^{\lambda_R \times C}$:

$$RT = \underbrace{\text{Softmax}\left(RF_{in} \cdot T_R\right)^T}_{RA \in \mathbb{R}^{Q' \times \lambda_R}} RF_{in}. \tag{12}$$

- **Diagonal token (*DT*)** embodies numerical priors as a strong inductive bias, crafted to guide attention towards structurally vital features in the matrix diagonal for enhanced predictive performance. *DT* explicitly encodes significant matrix diagonal features by applying a mask that isolates a predefined diagonal band, grounded in a twofold theoretical rationale:
  - The convergence rate of iterative methods is profoundly influenced by the matrix's spectral properties, specifically the distribution of its eigenvalues [6], and the Gershgorin Circle Theorem establishes the diagonal elements as the fundamental anchors for these eigenvalues [53, 54].
  - The image encoding scheme and convolutional layers jointly establish a spatial correspondence, mapping the crucial diagonal elements of the original matrix to a diagonal band within MF_{in}.

  Inspired by these, we highlight features within a predefined diagonal band by constructing a mask $Mask \in \mathbb{R}^{\mathcal{Q}' \times \mathcal{Q}'}$, where the value of each element $Mask_{ij}$ is determined by its distance from the main diagonal:

$$Mask_{ij} = \begin{cases} 1, & \text{if } |i-j| \le d \\ 0, & \text{otherwise} \end{cases}, \tag{13}$$

  where the half-width d is a hyperparameter determined by the architecture of the preceding CNNs. For a CNN with Γ layers, where each layer comprises a convolutional operation (kernel size τ, stride 1) and a pooling operation (kernel size and stride ϱ), a principled value for d is formulated as $\lceil \varrho^{-\Gamma} + (\tau-1)(\varrho^{\Gamma}-1)/(\varrho^{\Gamma}(\varrho-1)) \rceil$ to capture the projection field of the diagonal elements. A detailed, step-by-step derivation for d is provided in Appendix C.

  This mask is then broadcast across the channel dimension and applied element-wise to the matrix feature map, $Mask\ MF_{in} = Mask \odot MF_{in}$, to isolate the diagonal features. The resulting tensor is subsequently flattened and processed by a MLP to generate a single, consolidated diagonal token $DT \in \mathbb{R}^{1\times C}$, which encapsulates this vital structural information.

Finally, these three token sets are concatenated into a comprehensive sequence, $T_0 = [MT; DT; RT]$, which encapsulates matrix-wide semantics, RHS characteristics, and the critical numerical prior, serving as the enriched input for the subsequent Transformer.

3.4.2. Transformer

To model the global dependencies among the matrix, the RHS, and their intricate interplay, SPECTRA processes tokens T_0 from the tokenizer using a Transformer encoder [44, 45]. The encoder creates a fully connected computational graph over these tokens, enabling the model to explicitly reason about the interplay between any pair of features, which is critical for a comprehensive understanding of the system. Specifically, it enables three types of interactions:

- **Intra-feature interaction** allows *MT* and *RT* tokens to self-attend within their respective sets, enabling the model to form a global understanding of the internal structure of the matrix and RHS in isolation.
- **Cross-feature interaction** ensures *MT* and *RT* tokens to attend to one another, explicitly modeling their crucial interplay, which is a key determinant of the convergence rate for many iterative methods.
- **Feature-prior interaction** is established via the specialized *DT*, which acts as an anchor for the numerical prior. This interaction is twofold: the *DT* attends to *MT* and *RT* to aggregate global information from the diagonal's perspective, while *MT* and *RT* attend to the *DT* to contextualize themselves relative to this critical structural information. This interaction introduces a theoretically-grounded inductive bias, which guides the model toward more structurally-informed selections, enhancing its predictive performance.

These three interaction types are realized within a standard Transformer encoder composed of L identical layers, each comprising a Multi-Head Attention (MHA) block and a Feed-Forward Network (FFN) block. Layer normalization (LN) is applied before each block, and residual connections are used after each block [55, 56]. The computation for the l-th Transformer encoder layer is formulated as:

$$\begin{cases} T_l' = \mathrm{MHA}\big(\mathrm{LN}\big(T_{l-1}\big)\big) + T_{l-1}, \\ T_l = \mathrm{FFN}\big(\mathrm{LN}\big(T_l'\big)\big) + T_l', \end{cases} \tag{14}$$

where T_{l-1}, T_l', and $T_l \in \mathbb{R}^{(\lambda_M+1+\lambda_R)\times C}$ are the input, intermediate, and output tokens of the l-th layer, respectively.

The MHA block comprises U parallel attention "heads", and the computation for the u-th head, H^u, is defined as:

$$H^u = \mathrm{Softmax}\left(\frac{\left(T_{l-1}W_Q^u\right)\left(T_{l-1}W_K^u\right)^T}{\sqrt{D_h}}\right) T_{l-1}W_V^u, \tag{15}$$

where $W_Q^u, W_K^u, W_V^u \in \mathbb{R}^{C\times D_h}$ are the learnable weight matrices for the query, key, and value projections, respectively, and $D_h = C/U$ is the dimension of each head. The outputs of all U heads are then concatenated and passed through a final linear projection layer, which is governed by the weight matrix $W_O \in \mathbb{R}^{C\times C}$, to produce the final MHA output:

$$\mathrm{MHA}\left(T_{l-1}\right) = \left[H^1, H^2, \ldots, H^U\right] W_O. \tag{16}$$

Following the MHA block and its subsequent residual connection, the T_l' is then layer-normalized and processed by the FFN, which consists of two linear layers with an intermediate GELU activation function, expressed as:

$$\mathrm{FFN}\left(T_l'\right) = \mathrm{GELU}\left(T_l'W_{F_1}\right) W_{F_2}, \tag{17}$$

where $W_{F_1} \in \mathbb{R}^{C\times 4C}$ and $W_{F_2} \in \mathbb{R}^{4C\times C}$ are the learnable weight matrices.

Through the Transformer encoder, the initial tokens T_0 are progressively evolved into the final sequence $T_L = [MT_L; DT_L; RT_L]$, where each token encapsulates not only its original features but also rich contextual information derived from its global interactions within the system.

3.4.3. Projector

While the tokenizer abstracts the feature map into a condensed token sequence to enable efficient global modeling, this process inherently sacrifices fine-grained, pixel-level details. To bridge this gap, SPECTRA introduces a projector module, inspired by [46], which maps tokens back to feature space and fuses them with the initial feature maps, ensuring the final features benefits from both local precision and global context, as illustrated in Fig. 4(b).

The projector constructs the refined feature maps via two parallel pathways:

- **Matrix feature refinement** augments the initial feature map MF_{in} by fusing it with enriched tokens MT_L and DT_L. This process is implemented via an attention mechanism, where each pixel in MF_{in} serves as a query to retrieve contextual information from the enriched tokens, which function as the key. The query MQ and key MK are generated using learnable projection matrices $Q_M, K_M \in \mathbb{R}^{C\times C}$, where $MQ = MF_{in} \cdot Q_M$ and $MK = [MT_L; DT_L] \cdot K_M$. Subsequently, the attention map MA' is computed to quantify the relevance of each token to every pixel and is then used to form a weighted token sum, producing the intermediate feature map MF_{mid} that distills the global context for each pixel location:

$$MF_{mid} = \underbrace{\text{Softmax}\left(MQ \cdot MK^T\right)}_{MA' \in \mathbb{R}^{Q'^2 \times (\lambda_M + 1)}} [MT_L; DT_L]. \quad (18)$$

  Finally, MF_{mid} is residually added to the MF_{in} to yield the refined feature map, $MF_{out} \in \mathbb{R}^{Q'^2 \times C}$.

- **RHS feature refinement** follows an analogous procedure to augment the initial RHS feature map RF_{in}, where each pixel queries the enriched tokens RT_L to aggregate relevant global context. Similarly, the query RQ and key RK are generated using learnable projection matrices $Q_R, K_R \in \mathbb{R}^{C\times C}$ ($RQ = RF_{in} \cdot Q_R$ and $RK = RT_L \cdot K_R$), which in turn are used to compute the intermediate feature map RF_{mid}:

$$RF_{mid} = \underbrace{\text{Softmax}\left(RQ \cdot RK^T\right)}_{RA' \in \mathbb{R}^{Q' \times \lambda_R}} RT_L. \quad (19)$$

  The final RHS feature map $RF_{out} \in \mathbb{R}^{Q' \times C}$ is then produced via residual connection.

Through the projector, SPECTRA constructs refined feature maps (MF_{out} and RF_{out}) that integrate pixel-level details from the CNNs with long-range dependencies captured by the Transformer, which are subsequently fed to a final MLP to predict the optimal iterative methods.

4. Experiment

4.1. Experimental setup

4.1.1. Datasets

To comprehensively evaluate SPECTRA and ensure fair comparison with existing approaches, we construct two sparse linear system datasets using matrices sampled from SuiteSparse [29], a diverse matrix collection derived from practical engineering and scientific applications. These datasets assess model performance under both conventional and realistic scenarios:

- **Fixed-RHS dataset** is designed to provide fair comparison with existing approaches. Since prior works rely solely on the matrix and completely omit the RHS, a de facto standard evaluation protocol has emerged wherein each matrix is paired with a single predefined RHS [10, 24–28]. Following this convention, we construct this dataset by sampling matrices with scales $10^3 \le N \le 10^4$ [10, 28] from SuiteSparse and pair each with a fixed RHS of all ones.

- **Variable-RHS dataset** is established to evaluate performance under more realistic and challenging conditions where the RHS varies. This design reflects practical scenarios where the same matrix is solved with multiple RHS (e.g., in time-stepping simulations), which critically influence optimal method selection. To generate diverse RHS patterns that capture this variability, we reuse the matrices from the fixed-RHS dataset and generate ten distinct RHS per matrix by generating random solutions $x \in [0, 1)^N$ and computing the corresponding RHS as $b = Ax$.

To prepare for model training and evaluation, we partition the sampled matrices into training (80%) and test (20%) sets using random seeds, with all systems derived from a single matrix assigned to the same subset to prevent data leakage. This distribution yields 562 training and 141 testing systems for the fixed-RHS dataset, while the variable-RHS dataset comprises 5620 training and 1410 testing systems.

4.1.2. Implementation details

To obtain ground-truth labels for model training, we solve all sparse linear systems using the PETSc library on an AMD 9950X CPU with relative tolerance of $\tau = 10^{-6}$ [11] and a maximum number of iterations of $S = 1000$ [26]. We comprehensively evaluate all solver-preconditioner pairs generated by the Cartesian product of the components listed in Table 1. Pairs that fail to converge or are inexecutable incur a value of $\mathcal{T}_S(A, b, e) = \infty$. We restrict the final set E to pairs that are optimal for at least one system to preclude empty classes. This definition yields $K = 23$ classes for the fixed-RHS case and $K = 36$ classes for the variable-RHS case. The image resolution is set to $Q \in \{64, 128, 256\}$ [27], and complete model details for SPECTRA are provided in Appendix D. All models are trained on two Nvidia 4090 GPUs with a learning rate of 5×10^{-4}, a batch size of 64,

and 400 epochs, employing the cross-entropy loss:

$$loss = \sum_{i=1}^{\zeta} \text{CrossEntropy}\left(y_i, \hat{y}_i\right), \tag{20}$$

where ζ is the number of training samples, y_i is the ground-truth label, and $\hat{y}_i$ is the predicted probability distribution. The training procedure requires less than two hours of wall-clock time, equivalent to under four GPU hours, even at the highest resolution. A detailed cost-benefit analysis quantifying the trade-off between training overhead and runtime savings is provided in Section E.

To mitigate randomness, we conduct each experiment with five different random seeds and report the mean performance across all runs.

4.1.3. Evaluation metrics

We comprehensively evaluate method selection approaches using four distinct metrics from two critical perspectives: the first assesses classification accuracy, while the remaining three quantify practical solution effectiveness.

- **Top-n selection accuracy ($ACC@n$)** assesses the model's classification performance by measuring the frequency with which the optimal method appears within the top-n predictions. While $ACC@1$ is the primary metric for automated selection since only the top-1 method is deployed, we also report $ACC@n \geq 2$, which evaluates the model's utility as a decision-support tool that generates a shortlist of candidate methods for expert review and final selection.

- **Solution success rate (*Success*)** measures the robustness of the model's top-1 selections, defined as the percentage of systems for which the selected method successfully converges under the relative residual stopping criterion $\|r^{(s)}\|_2 \leq \tau \|r^{(0)}\|_2$ with $\tau = 10^{-6}$ and a maximum iteration limit of $S = 1000$.

- **Solution efficiency (*Efficiency*)** quantifies how closely the model's selections approximate the optimal. For the i-th system (A_i, b_i), the *efficiency* is the ratio of the optimal solution time $\mathcal{T}_S^i(e^*)$ to that of the selected method $\mathcal{T}_S^i(e_m)$, where e_m denotes the model's top-1 prediction, with 0 assigned to non-convergent selections to penalize non-robustness. The average *efficiency* across the ϕ test systems is formulated as:

$$Efficiency = \frac{1}{\phi} \sum_{i=1}^{\phi} \begin{cases} \dfrac{\mathcal{T}_S^i(e^*)}{\mathcal{T}_S^i(e_m)}, & \text{if } (A_i, b_i, e_m) \text{ converges} \\ 0, & \text{otherwise} \end{cases}. \tag{21}$$

- **Computational speedup (*Speedup*)** evaluates the solution acceleration achieved by the selection approach relative to the baseline representing the expected solution time of random choices. For the i-th system, the baseline time $\mathcal{T}_B^i$ is computed as the average solution time across all candidate methods, with non-convergent methods assigned a penalty time equal to the maximum convergent time (Eq. 22). The final *speedup* is the ratio of $\mathcal{T}_B$ to the model's end-to-end time including feature extraction $\mathcal{T}_F$, model inference $\mathcal{T}_M$, and the selected method solution time, with 0 assigned for non-convergent selections (Eq. 23).

$$\mathcal{T}_B^i = \frac{1}{K} \sum_{e \in E} \begin{cases} \mathcal{T}_S^i(e), & \text{if } (A_i, b_i, e) \text{ converges} \\ \max\limits_{\substack{e' \in E \\ (A_i, b_i, e') \text{ converges}}} \mathcal{T}_S^i(e'), & \text{otherwise} \end{cases}, \tag{22}$$

$$Speedup = \frac{1}{\phi} \sum_{i=1}^{\phi} \begin{cases} \dfrac{\mathcal{T}_B^i}{\mathcal{T}_F^i + \mathcal{T}_M^i + \mathcal{T}_S^i(e_m)}, & \text{if } (A_i, b_i, e_m) \text{ converges} \\ 0, & \text{otherwise} \end{cases}. \tag{23}$$

4.1.4. Baselines

To rigorously evaluate SPECTRA and its gains relative to engineering practice, we benchmark it against a rule-based heuristic baseline and five representative deep learning-based approaches spanning the primary paradigms for iterative method selection. For learning-based baselines, we focus on deep learning methods, as their superiority over traditional machine learning methods has been well established in prior works [10, 25, 27].

- **HR** (Heuristic Rule) is a rule-based approach that mirrors engineering selection strategies according to system properties. It employs CG with SSOR for Symmetric Positive Definite (SPD) systems while applying GMRES with ILU for non-SPD instances. This choice aligns with established conventions as CG is tailored for SPD systems and GMRES is utilized for general non-SPD systems, while SSOR and ILU function as computationally efficient preconditioners. For fair comparison, SPD verification costs via Cholesky factorization are added to the feature extraction time $\mathcal{T}_F$ and the inference time $\mathcal{T}_M$ is set to zero. As deterministic HR yields a single selection rather than a probability distribution, $ACC@n \geq 2$ are omitted.

- **FC** [26] is a scalar-based approach that employs a fully connected network to predict optimal methods based on numerical features extracted from the matrix.

- **GNN** [10] is a graph-based approach that models the matrix as a graph and employs a graph neural network to learn its topological features.

- **CNN** [24, 25] is the conventional image-based approach that encodes the matrix into an RGB image and employs CNNs for classification.

- **MM** [27] and **RAF** [28] are current SOTA multimodal approaches that extract and fuse numerical and structural features extracted from the matrix.

Table 2
Comparison with SOTA approaches under *fixed-RHS* (top) and *variable-RHS* (bottom) settings. All metrics are reported as mean ± standard deviation across five runs. SPECTRA achieves SOTA performance in both settings, with gains substantially amplified in variable-RHS scenario compared to fixed-RHS setting. The best results for each metric are highlighted in **blue**.

Model	*ACC@1* (%)↑	*ACC@2* (%)↑	*ACC@3* (%)↑	*ACC@4* (%)↑	*ACC@5* (%)↑	*Success* (%)↑	*Efficiency* (%)↑	*Speedup* (×)↑
Fixed-RHS								
HR	33.19±2.86	–	–	–	–	89.02±1.03	5.24±2.12	1.68±0.66
FC [26]	71.28±2.38	86.52±1.69	90.60±0.99	91.55±0.94	91.73±1.27	97.70±0.59	16.19±1.67	3.56±0.48
GNN [10]	70.93±0.94	86.05±1.06	90.54±1.03	92.47±1.03	94.33±1.30	96.22±2.11	11.31±1.75	1.08±0.36
CNN$_{64}$ [24, 25]	70.80±2.04	82.92±0.85	89.42±1.04	89.48±0.91	91.37±1.19	96.21±0.31	9.27±1.38	3.18±0.39
MM$_{64}$ [27]	75.41±1.37	85.34±0.93	90.13±1.24	92.55±1.18	92.85±1.45	97.87±0.29	17.50±1.25	3.83±0.43
RAF$_{64}$ [28]	73.23±2.95	87.17±1.77	90.07±0.87	90.13±0.94	92.14±1.02	97.28±0.48	15.10±0.99	3.76±0.36
SPECTRA$_{64}$	77.72±3.84	85.99±1.20	89.89±1.02	91.90±0.98	92.79±0.64	97.70±1.04	25.44±1.46	4.01±0.43
CNN$_{128}$ [24, 25]	71.99±3.20	85.28±1.48	90.48±1.31	90.66±1.47	90.90±1.15	97.28±0.39	13.02±3.10	3.36±0.46
MM$_{128}$ [27]	76.60±2.01	86.76±0.97	90.31±1.34	90.31±0.87	91.96±1.34	99.05±1.34	22.27±1.99	3.95±0.36
RAF$_{128}$ [28]	74.11±1.18	87.77±0.99	89.89±0.83	90.01±1.20	92.02±0.69	97.41±0.54	13.79±1.10	3.67±0.44
SPECTRA$_{128}$	**78.72**±2.24	**87.82**±1.31	**91.62**±1.83	91.81±1.22	92.52±1.22	**99.29**±0.67	**35.81**±0.93	**4.41**±0.30
CNN$_{256}$ [24, 25]	71.51±2.68	86.11±0.79	89.95±0.70	90.78±1.00	91.43±0.84	96.69±0.74	12.68±4.20	3.19±0.51
MM$_{256}$ [27]	75.83±3.55	87.35±3.72	89.78±2.93	91.02±1.21	92.02±0.59	97.04±0.39	17.76±1.31	3.75±0.61
RAF$_{256}$ [28]	74.17±3.90	85.17±1.77	88.89±1.21	89.30±0.20	92.38±1.20	98.46±0.26	16.22±1.41	3.82±0.88
SPECTRA$_{256}$	78.35±2.97	86.36±0.86	90.03±0.86	**92.67**±0.70	**94.70**±1.34	97.82±0.70	27.67±1.35	4.14±0.79
Variable-RHS								
HR	29.06±1.04	–	–	–	–	85.23±0.74	3.81±1.44	1.79±0.76
FC [26]	55.16±0.61	73.59±0.70	79.75±0.73	81.22±0.82	83.21±0.77	92.91±0.45	5.35±1.78	4.05±0.55
GNN [10]	59.50±0.65	74.90±0.55	78.99±0.74	81.66±0.60	83.20±0.67	90.76±0.76	9.94±1.69	1.20±0.44
CNN$_{64}$ [24, 25]	61.47±0.52	75.11±0.58	80.00±0.63	81.25±0.70	82.63±0.60	94.21±0.34	8.92±2.06	4.20±0.57
MM$_{64}$ [27]	59.33±0.85	75.12±0.50	78.66±0.60	80.35±0.53	82.29±0.51	95.04±0.27	8.94±0.85	4.34±0.51
RAF$_{64}$ [28]	60.57±0.86	76.01±1.03	80.38±0.96	82.03±0.64	83.63±0.79	95.04±0.21	10.87±1.86	4.42±0.49
SPECTRA$_{64}$	78.23±1.35	88.45±0.63	**93.58**±0.56	**95.71**±0.57	96.72±0.60	97.99±0.41	39.10±1.08	6.97±0.73
CNN$_{128}$ [24, 25]	59.83±0.58	74.41±0.83	78.95±0.93	80.51±0.81	82.40±0.72	94.68±0.21	10.24±1.23	4.39±0.78
MM$_{128}$ [27]	60.61±0.66	75.57±0.55	79.47±0.70	80.97±0.94	82.55±0.76	94.90±0.59	11.97±1.03	4.58±0.76
RAF$_{128}$ [28]	60.19±0.64	74.23±0.66	79.21±0.76	80.90±0.72	83.03±0.74	94.58±0.37	8.73±0.80	4.24±0.46
SPECTRA$_{128}$	**78.53**±1.09	**88.48**±0.67	92.90±0.59	95.49±0.57	**97.11**±0.65	**99.49**±0.26	**45.44**±0.69	**7.36**±0.62
CNN$_{256}$ [24, 25]	60.69±0.84	75.10±0.60	79.14±0.58	80.84±0.53	81.92±0.61	94.88±0.47	8.43±0.88	4.19±0.55
MM$_{256}$ [27]	59.82±0.52	74.80±1.00	78.77±0.67	80.61±0.54	82.36±0.53	94.97±0.37	8.36±1.08	4.21±0.60
RAF$_{256}$ [28]	60.40±0.99	74.15±0.76	77.51±0.67	79.08±0.82	81.45±0.63	95.34±0.38	12.03±1.76	4.29±0.56
SPECTRA$_{256}$	78.42±1.20	86.40±0.93	91.11±0.82	93.75±0.83	95.45±0.74	98.32±0.30	38.02±1.65	6.79±0.58

4.2. Comparison with SOTA approaches

As summarized in Table 2, SPECTRA establishes a new SOTA and consistently outperforms all baseline approaches across all evaluation metrics on both datasets.

Performance on the fixed-RHS dataset: In the conventional fixed-RHS setting, which aligns with the evaluation protocol of prior works, SPECTRA exhibits a distinct performance advantage. As detailed in the upper section of Table 2, SPECTRA$_{128}$ achieves the highest *ACC*@1 of 78.72%, surpassing the conventional CNN$_{128}$ by 6.73% and the strongest baseline MM$_{128}$ by 2.12%. Beyond selection accuracy, this advantage translates into superior practical effectiveness. SPECTRA$_{128}$ achieves a *Success* of 99.29%, *Efficiency* of 35.81%, and *Speedup* of 4.41×, surpassing CNN$_{128}$ by 2.01%, 22.79%, and 0.65×, respectively, and outperforming MM$_{128}$ by 0.24%, 13.54%, and 0.46×, respectively. Furthermore, SPECTRA$_{128}$ significantly outperforms the standard engineering HR by increasing *ACC*@1 by 45.53% and boosting the *Speedup* by 2.73×, validating the superiority of data-driven decision-making over static heuristics. These results underscore that SPECTRA's capabilities in faithfully capturing numerical distributions and globally recognizing patterns not only enhance selection accuracy but also yield superior practical performance.

Performance on the variable-RHS dataset: The superiority of SPECTRA is substantially amplified on the variable-RHS dataset, which evaluates model performance under more realistic and challenging conditions. As shown in the lower section of Table 2, all matrix-only baseline models

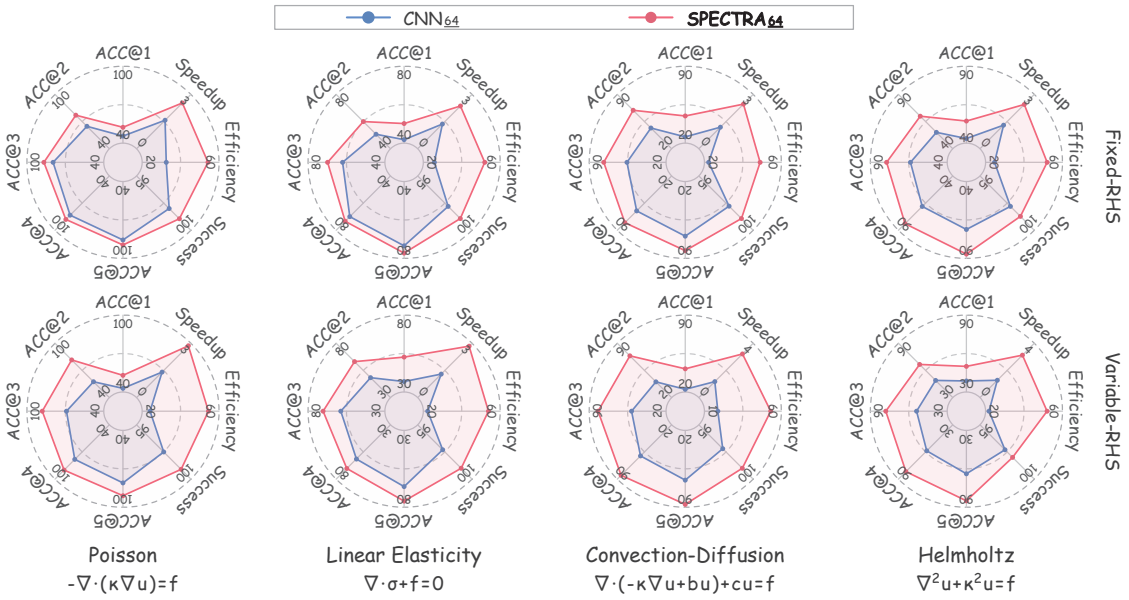

Figure 5: Case study on four PDE problems. SPECTRA$_{64}$ consistently outperforms CNN$_{64}$ across all four PDE cases, especially in the variable-RHS setting. The performance advantage is evident in well-conditioned Poisson problems and becomes increasingly substantial as system complexity grows, as demonstrated in the Linear Elasticity, Convection-Diffusion, and Helmholtz equations.

suffer significant performance degradation due to the omission of RHS, rendering them unable to adapt their selections to varying RHS characteristics. For instance, the *ACC*@1 of CNN$_{128}$ and MM$_{128}$ decreases by 12.16% and 15.99%, respectively. In stark contrast, SPECTRA, with its pioneering incorporation of RHS-awareness, demonstrates consistently superior performance. SPECTRA$_{128}$ achieves an *ACC*@1 of 78.53%, surpassing CNN$_{128}$ and MM$_{128}$ by 18.70% and 17.92%, respectively. This substantial accuracy advantage translates into significant improvements in practical metrics. SPECTRA$_{128}$ improves *Success*, *Efficiency*, and *Speedup* over CNN$_{128}$ by 4.81%, 35.20%, and 2.97×, respectively, and over MM$_{128}$ by 4.59%, 33.47%, and 2.78×, respectively. Notably, excluding RHS information exacerbates the limitations of HR in this setting and yields a minimal *Efficiency* of 3.81%, whereas SPECTRA$_{128}$ attains a substantial *Efficiency* of 45.44% and a *Speedup* of 7.36×. This demonstrates that by creating a holistic system representation, SPECTRA not only enables more accurate selections but also achieves substantially enhanced robustness and efficiency in practice, delivering significant real-world computational acceleration.

4.3. Case studies

To provide granular insights into the advantages of SPECTRA, we conduct a case study on systems derived from four fundamental partial differential equations (PDEs): Poisson [57], Linear Elasticity [58], Convection-Diffusion [59], and Helmholtz [60], chosen as distinct cases that represent cornerstone problems in real-world scientific computing with increasing structural complexity. For each PDE, we generate 5000 systems using OpenMat [61] and construct the corresponding fixed-RHS and variable-RHS datasets. All models are trained on SuiteSparse and evaluated on these PDE systems in a zero-shot setting.

Fig. 5 demonstrates that SPECTRA$_{64}$ consistently outperforms the conventional CNN$_{64}$ across all four cases, with this superiority being substantially amplified in the more realistic variable-RHS setting owing to SPECTRA's holistic problem representation. Examining the specific cases in detail, for the well-conditioned, symmetric positive-definite systems arising from the Poisson equation, SPECTRA$_{64}$ comprehensively outperforms CNN$_{64}$, thereby confirming its efficacy on simpler problems. This advantage becomes more pronounced for Linear Elasticity systems, where the wide dynamic range and symmetric block structures expose the numerical distortion in the CNN's reductionist RGB encoding, whereas SPECTRA's expanded-channel encoding preserves numerical fidelity. SPECTRA$_{64}$'s superiority is most evident in the Convection-Diffusion and Helmholtz cases, where the non-symmetric or indefinite matrices exhibit crucial long-range dependencies (e.g., advection-dominance) that the CNN's locality-biased convolutions fail to capture, whereas SPECTRA's DET effectively identifies these global structures through the attention mechanism.

4.4. Ablation studies

To systematically dissect SPECTRA and quantify the individual and synergistic contributions of its core components, we conduct a comprehensive ablation study on both the fixed-RHS and variable-RHS datasets, as detailed in Table 3. For fair comparison, all ablation variants in this section are evaluated at an image resolution of $\mathcal{Q} = 64$.

4.4.1. Effect of the expanded channels P&V for matrix

To rigorously evaluate the effectiveness of the expanded *P&V* channels for matrix encoding, we conduct performance comparisons with and without them, and the results demonstrate their significant contribution to achieving superior performance.

Table 3
Ablation study evaluating the contributions of SPECTRA's core components. We quantify the effectiveness of key innovations by systematically ablating the matrix encoding scheme (standard RGB vs. expanded $RGB{+}P\&V$), the model backbone (conventional CNNs vs. hybrid CNNs+Trans, where Trans denotes Transformer), the diagonal token (DT), and the RHS-awareness. Symbols ✔ and ✘ indicate the inclusion and exclusion of the corresponding component, respectively. Results highlight that each component provides individual and synergistic performance gains, with the complete SPECTRA (all ✔) achieving the best performance.

Configuration						*ACC@1* (%)↑	*ACC@5* (%)↑	*Success* (%)↑	*Efficiency* (%)↑	*Speedup* (×)↑
RGB	*P&V*	CNNs	Trans	*DT*	RHS					
Fixed-RHS										
✔	✘	✔	✘	✘	✘	70.80	91.37	96.21	9.27	3.18
✔	✔	✔	✘	✘	✘	71.34	91.52	96.30	11.15	3.22
✔	✘	✔	✔	✘	✘	73.33	91.92	96.78	15.20	3.48
✔	✔	✔	✔	✘	✘	74.44	92.33	97.21	17.77	3.62
✔	✘	✔	✔	✔	✘	74.54	92.22	97.40	18.25	3.78
✔	✘	✔	✔	✘	✔	73.54	91.88	97.00	16.57	3.52
✔	✔	✔	✔	✔	✘	77.15	92.55	97.51	23.44	3.92
✔	✔	✔	✔	✘	✔	74.71	92.40	97.48	18.59	3.80
✔	✘	✔	✔	✔	✔	76.76	92.41	97.40	21.52	3.88
✔	✔	✔	✔	✔	✔	**77.72**	**92.79**	**97.70**	**25.44**	**4.01**
Variable-RHS										
✔	✘	✔	✔	✘	✘	62.86	76.56	95.00	11.21	4.48
✔	✘	✔	✔	✘	✔	74.92	94.45	96.73	28.24	5.95
✔	✔	✔	✔	✘	✔	76.65	96.35	97.08	31.95	6.30
✔	✘	✔	✔	✔	✔	77.12	95.53	97.80	33.83	6.34
✔	✔	✔	✔	✔	✔	**78.23**	**96.72**	**97.99**	**39.10**	**6.97**

In the fixed-RHS setting, augmenting a standard CNN with $P\&V$ channels yields a 0.54% improvement in $ACC@1$ and a 1.88% boost in *Efficiency* (row 1 vs. 2). More compellingly, the indispensability of $P\&V$ channels is highlighted under the variable-RHS setting, where their removal from the complete SPECTRA causes a 1.11% drop in $ACC@1$ and a 5.27% decline in *Efficiency* (row 14 vs. 15).

The consistent and substantial performance benefits across diverse settings and model configurations firmly establish the superiority of our expanded-channel encoding over conventional RGB encoding for matrices. To further dissect their individual contributions, an ablation study on the separate P and V channels is provided in Appendix F.1.

4.4.2. *Effectiveness of the DET*

The DET module, whose core innovation lies in incorporating a specialized diagonal token into a Transformer backbone, is designed to overcome the long-range obscurity in CNNs. Our ablation studies confirm that both the Transformer and the diagonal token contribute significantly and synergistically to the overall performance of SPECTRA.

The Transformer backbone provides a fundamental architectural advantage over conventional CNNs. In the fixed-RHS setting, upgrading the CNN backbone with a Transformer increases $ACC@1$ by 3.1% and enhances *Efficiency* by 6.62% (row 2 vs. 4), demonstrating that the self-attention mechanism delivers critical performance gains unattainable through locally-constrained convolutions.

The diagonal token further enhances performance by injecting numerical priors into the attention mechanism. In the fixed-RHS setting, integrating the diagonal token into a hybrid CNN-Transformer backbone improves $ACC@1$ by 2.71% and *Efficiency* by 5.67% (row 4 vs. 7). Conversely, in the variable-RHS setting, its removal from the complete SPECTRA results in a 1.58% decline in $ACC@1$ and a substantial 7.15% decrease in *Efficiency* (row 13 vs. 15). These results underscore that explicitly guiding the model's attention toward structurally significant diagonal information is highly effective across diverse scenarios.

The consistent and substantial performance gains demonstrate that augmenting CNNs with the DET establishes a more powerful and effective pattern recognition engine.

4.4.3. *Impact of RHS-awareness*

To quantify the effect of RHS-awareness, we focus our analysis on the variable-RHS setting, as its diverse RHS provide the necessary conditions to evaluate the model's ability in practical scenarios. Results demonstrate that RHS-awareness is critical for selection in realistic scenarios.

The fundamental limitation of matrix-only approaches is starkly revealed when the RHS varies. Specifically, when the hybrid CNN-Transformer model that performs competently in the fixed-RHS setting is evaluated on the variable-RHS dataset, its performance collapses with $ACC@1$ plummeting by 10.47% and *Efficiency* dropping by 3.99% (row 3 vs. 11). However, incorporating RHS-awareness into this model yields the largest performance gain, catapulting $ACC@1$ by 12.06% and boosting *Efficiency* by 17.03% (row 11 vs. 12).

Results unequivocally demonstrate that RHS-awareness is not merely an incremental improvement but a fundamental

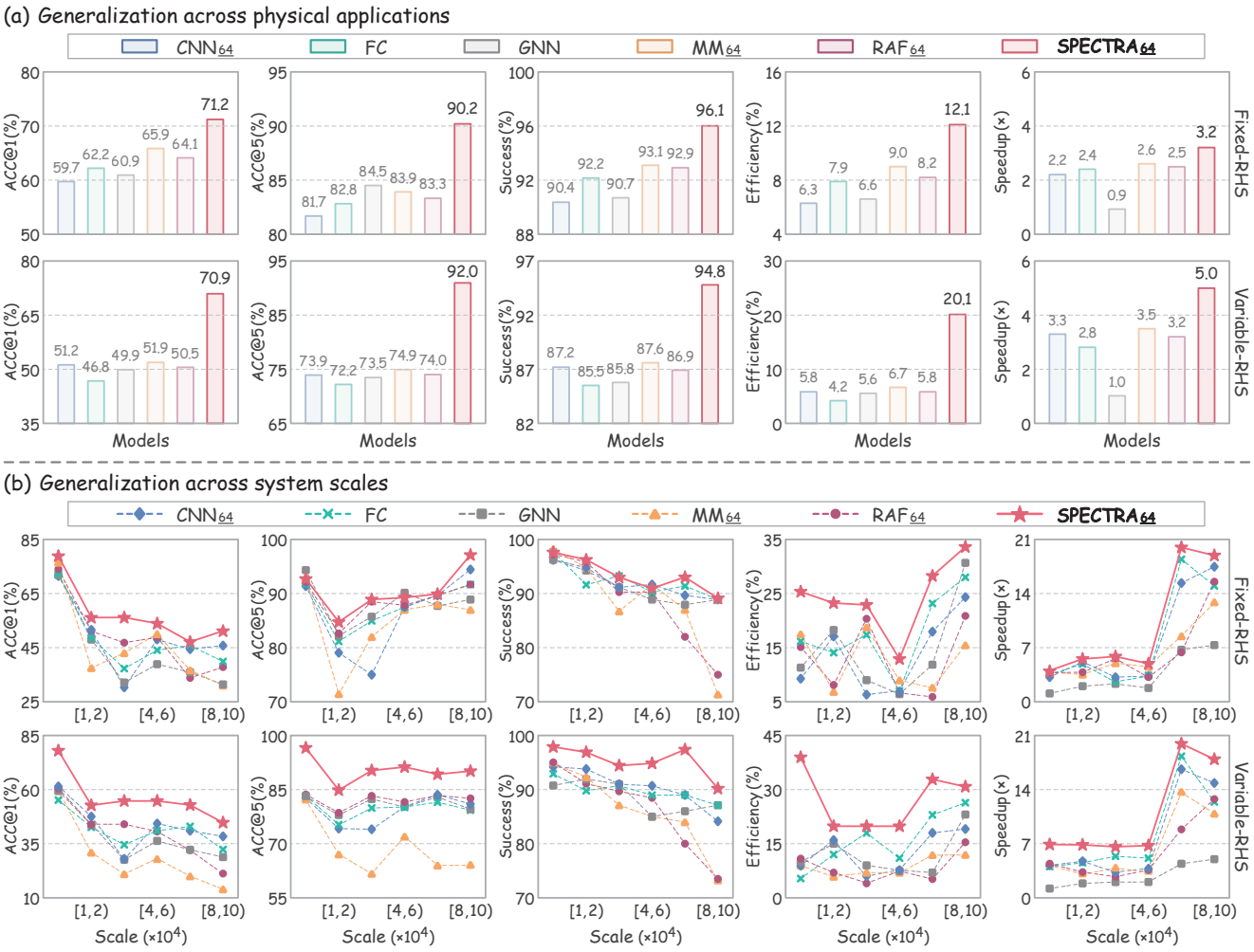

Figure 6: Generalization comparison of SPECTRA64 with baselines across out-of-distribution (a) physical applications and (b) system scales. SPECTRA exhibits robust performance compared to baselines on the fixed-RHS dataset, and this superiority is significantly pronounced in the variable-RHS setting, collectively demonstrating its exceptional generalization capability.

paradigm shift for iterative method selection by enabling a holistic problem representation, thereby unlocking substantial gains in the realistic and challenging scenarios. To further dissect the individual contributions of the channels for RHS encoding, an ablation study on the separate R, G, B, P, and V channels is provided in Appendix F.2.

4.5. Generalization studies

To critically evaluate the robustness and practical applicability of SPECTRA, we conduct two out-of-distribution generalization studies on both fixed-RHS and variable-RHS datasets. We first assess generalization across physical applications by partitioning SuiteSparse matrices according to problem domains for a group-holdout evaluation where specific groups are reserved for testing to maintain an approximate 8:2 training-to-testing ratio. Second, we assess generalization across system scales by evaluating models on systems of $10^4 \leq N \leq 10^5$ as this range significantly exceeds the $10^3 \leq N \leq 10^4$ scale encountered during training.

As illustrated in Fig. 6, SPECTRA64 demonstrates superior generalization capabilities across both datasets. In the fixed-RHS setting, SPECTRA64 consistently outperforms baselines in the physical-application holdout experiments shown in Fig. 6(a) and maintains superior stability during scale extrapolation as depicted in Fig. 6(b), highlighting the robust scalability conferred by its expanded-channel encoding scheme and the DET. Notably, these advantages become substantially more pronounced in the variable-RHS setting, where SPECTRA64 establishes dominance across all metrics, underscoring that its RHS-awareness is the decisive factor for achieving strong and reliable generalization in practical applications.

These findings regarding generalization capability are pivotal for assessing the cost-effectiveness of SPECTRA. Specifically, the superior robustness observed across unseen physical applications (Fig. 6(a)) and PDE families (Section 4.3) confirms that SPECTRA sustains exceptional performance on new matrix types without requiring retraining. This characteristic circumvents the computational burden of frequent model retraining, thereby minimizing marginal deployment costs for new problem domains and establishing SPECTRA as a cost-effective intelligent decision system.

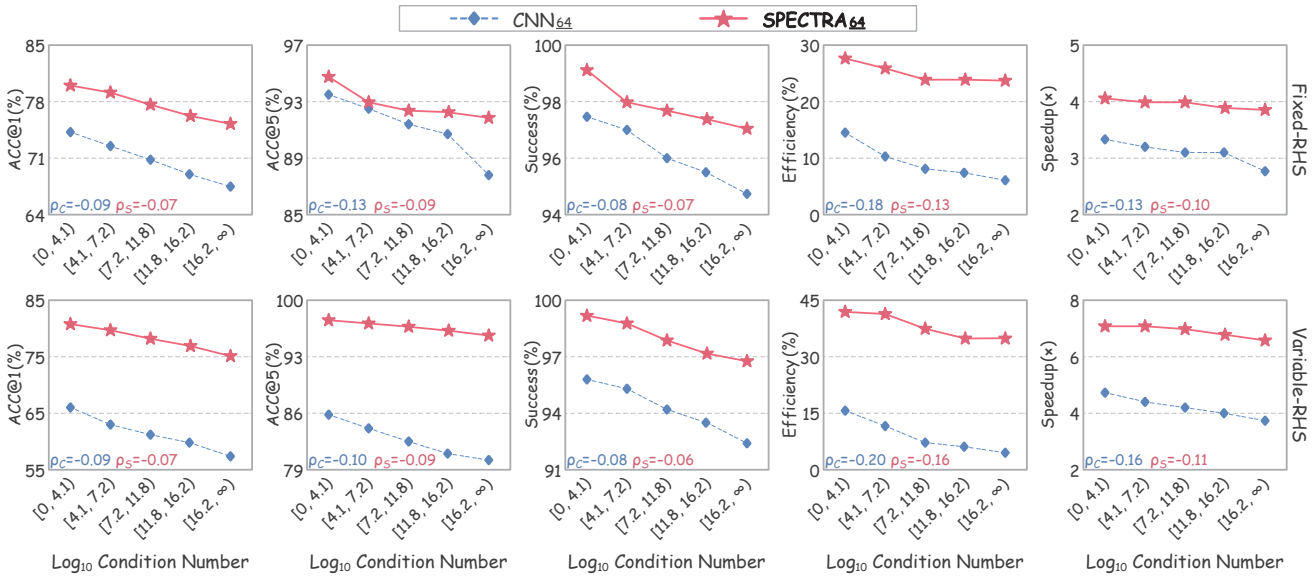

Figure 7: Sensitivity of SPECTRA64 and CNN64 to matrix conditioning. Markers indicate mean performance within each conditioning quintile. ρ_S and ρ_C denote the Spearman correlations between the performance metric and the condition number for SPECTRA and CNN, respectively. While the performance of both models declines as conditioning deteriorates, SPECTRA consistently outperforms CNN and degrades more gracefully, demonstrating superior reliability in handling ill-conditioned systems.

4.6. Sensitivity to matrix conditioning

To investigate model behavior across varying numerical difficulty of systems, we analyze the relationship between model performance and matrix conditioning, a standard proxy for problem complexity [6, 53]. Specifically, we first compute the 2-norm condition number $\kappa_2(A)$ via singular values for each sampled matrix and partition the systems into five bins based on the quintiles of $\log_{10}(\kappa_2(A))$, with singular matrices assigned to the most ill-conditioned bin (treating $\kappa_2(A) \to \infty$) [62, 63]. We then report the mean performance within each bin for both SPECTRA and CNN, and compute the sample-level Spearman rank correlations between each metric and $\log_{10}(\kappa_2(A))$ [64, 65].

As illustrated in Fig. 7, the performance of both models generally exhibits a mild downward trend as the condition number increases, indicating that ill-conditioned systems remain challenging for method selection. Nevertheless, in the fixed-RHS setting, SPECTRA64 consistently outperforms CNN64 across all bins and typically degrades more gracefully as conditioning worsens, underscoring the advantages of the expanded-channel encoding and the DET. Furthermore, this performance disparity widens in the more realistic variable-RHS setting, highlighting the benefits of RHS-aware holistic representations. Notably, the modest $|\rho|$ values suggest that matrix conditioning accounts for only a limited portion of the performance variation.

4.7. Overhead analyses

To evaluate the scalability of SPECTRA, we analyze the computational overhead incurred by feature extraction and model inference. Theoretically, as detailed in Algorithm 2, the primary computational overhead of feature extraction arises from traversing all non-zero elements, yielding

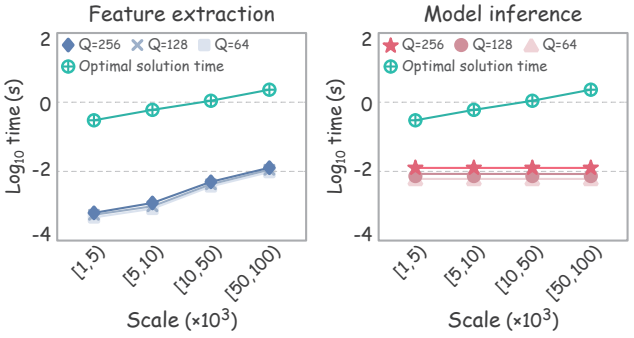

Figure 8: Computational overhead of SPECTRA across various system scales and image resolutions. The left and right panels depict the computational time required for feature extraction and model inference, respectively. Symbol ⊕ denotes the average optimal solution time for iterative methods. Both extraction and inference times remain consistently orders of magnitude lower than the solution time, underscoring the negligible overhead and high scalability of SPECTRA.

$\mathcal{O}(NNZ)$ complexity for the matrix and $\mathcal{O}(N)$ for the RHS. In contrast, the computational cost of model inference depends solely on the input image resolution $\mathcal{Q}$ and the network architecture, maintaining constant $\mathcal{O}(1)$ complexity relative to system scale. Conversely, the solution time for iterative methods typically scales super-linearly with scale, exhibiting $\mathcal{O}(SN)$ complexity, where S denotes the number of iterations [9]. Consequently, the total overhead of SPECTRA constitutes a negligible fraction of the iterative solution time.

Fig. 8 illustrates the empirical overhead across scales $10^3 \leq N \leq 10^5$ with varying resolutions $\mathcal{Q} \in \{64, 128, 256\}$. Results demonstrate that both feature extraction and inference times remain orders of magnitude lower than the

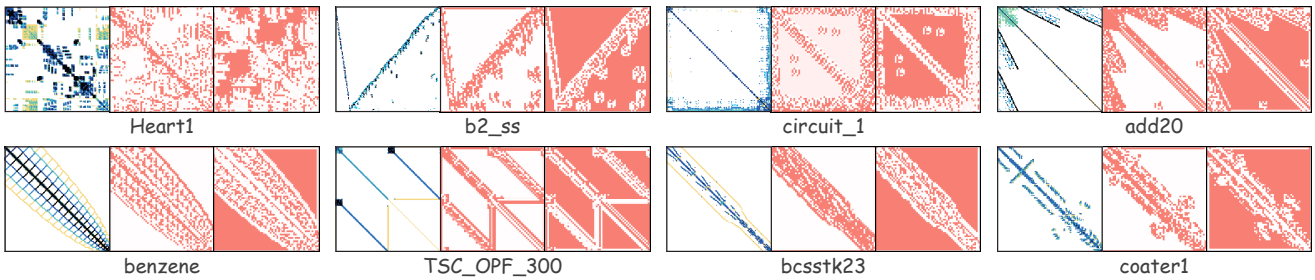

Figure 9: Visualization of matrix attention maps. Each group displays an original matrix (left) and two corresponding attention maps (middle and right), which illustrate how structural features are aggregated into matrix tokens. For visual clarity, we showcase two attention maps exhibiting the most distinct patterns: one capturing non-zero matrix elements (middle) and another identifying zero-filled regions (right). Without explicit supervision, tokens spontaneously identify different structural patterns, demonstrating learned abstraction from pixel-level features to high-order concepts that facilitates effective modeling of global dependencies.

average optimal solution time. Notably, although feature extraction time scales with system size, it remains negligible compared to the solution time. Furthermore, inference time is invariant to system scale, depending only on resolution. This validates that SPECTRA imposes a negligible computational burden, establishing it as a highly scalable intelligent decision system for large-scale engineering applications.

4.8. Visualizing tokens

To better understand what spatial features each token captures, we visualize the attention maps derived from the matrix attention $MA \in \mathbb{R}^{Q'^2 \times \lambda_M}$, where each map $MA_{:,i} \in \mathbb{R}^{Q' \times Q'}$ determines how each pixel from the feature map MF_{in} contributes to the i-th matrix token MT_i (Eq. 11). As illustrated in Fig. 9 for eight matrices from SuiteSparse, distinct tokens spontaneously learn to focus on different structurally meaningful motifs without any explicit supervision, thereby effectively segmenting the matrix. This demonstrates that the tokenizer successfully abstracts pixel-level features into a vocabulary of high-order structural concepts, thereby enabling the DET to efficiently model long-range dependencies through global pattern recognition and ultimately contributing to SPECTRA's superior performance.

5. Related work

Iterative method selection initially relied on traditional machine learning. Pioneering work applied classifiers such as Support Vector Machines and AdaBoost for solver selection [17]. This initial exploration prompted the exploration of more diverse algorithms. Subsequently, the methodology expanded to include Nearest Neighbor and Naive Bayes for low-cost, high-accuracy solver selection [18], as well as Decision Trees for constructing multi-stage selection strategies [19]. As the field matured, the focus shifted from employing individual models to developing integrated and systematized frameworks. This led to the use of machine learning libraries such as WEKA and MULAN for selection in transient simulations [20], and the development of integrated frameworks such as Lighthouse, which incorporated K-Nearest Neighbors, Alternating Decision Trees, and Random Forests to recommend methods from the PETSc and Trilinos libraries [21, 22]. More recently, the community has embraced advanced ensemble and probabilistic techniques. This trend is exemplified by the application of advanced gradient boosting models like XGBoost and Gradient Boosting Decision Trees for optimal Krylov solver selection [11], and the use of sophisticated methods such as Gaussian Processes and Gradient Boosting in complex domains like multiphysics porous-media simulations [23].

The advancement of deep learning has catalyzed a paradigm shift in iterative method selection, surpassing machine learning due to enhanced representation learning capabilities. Initial image-based approaches encoded the matrix as an RGB image, employing CNNs to learn its structural patterns [24, 25]. Another line of research pursued a scalar-based approach, which utilized eighteen matrix features fed into FC networks for solver selection [26]. Subsequently, a different paradigm emerged that modeled the matrix as a graph with five node and ten graph features, using GNNs to capture its topological information [10]. Building on these works, current SOTA approaches adopt a multimodal strategy, fusing image and scalar features to achieve more accurate and robust selections [27, 28]. In contrast, SPECTRA revitalizes image-based iterative method selection through three key innovations: an expanded-channel encoding scheme for faithful numerical representation, pioneering RHS-awareness for holistic system characterization, and a DET for global dependency modeling.

6. Discussion and limitations

The scalability and generalization of SPECTRA are underpinned by its encoding mechanism, which projects arbitrary-dimensional linear systems onto fixed-resolution feature maps. This design decouples model input dimension from system scale, allowing SPECTRA to handle significantly larger systems without retraining. As demonstrated by zero-shot extrapolation experiments in Section 4.5, SPECTRA maintains robust performance even beyond its training range. We attribute this generalization to the observation that iterative convergence behavior is governed predominantly by scale-invariant macroscopic structural properties, such as matrix sparsity patterns and RHS spectral frequencies,

rather than by element-wise details. Consequently, encoding acts as a statistical aggregation that preserves critical global features while filtering noise, enabling SPECTRA to learn scale-invariant visual patterns.

However, this image-based paradigm inherently suffers from information loss due to resolution constraints. Compressing a high-dimensional matrix into a coarse grid inevitably introduces aliasing, potentially masking specific eigenvalue outliers that govern convergence in ill-conditioned cases. While our proposed expanded-channel encoding mitigates this issue by explicitly capturing statistical extremes, it does not fully eliminate the loss of fine-grained details. Therefore, SPECTRA is particularly well-suited for engineering applications dominated by global physical properties, where this trade-off facilitates a highly efficient and scalable intelligent decision system.

7. Conclusion

In this work, we propose SPECTRA, a novel framework that revitalizes image-based iterative method selection for efficiently solving sparse linear systems. Leveraging a discriminative expanded-channel encoding scheme, the pioneering incorporation of RHS-awareness, and a Diagonally-Enhanced Transformer, SPECTRA faithfully captures numerical distributions, holistically represents the problem, and globally models dependencies guided by numerical priors. Extensive experiments demonstrate that SPECTRA not only establishes a new state-of-the-art but also exhibits superior generalization and robustness across diverse problem domains and system scales, underscoring its utility for real-world applications in science and engineering.

From the standpoint of software implementation, SPECTRA functions as a lightweight decision layer suitable for integration into automated scientific computing platforms, such as PETSc. Once a linear system is assembled, the platform encodes the system, executes a single inference pass, and instantiates the recommended solver-preconditioner pair through library APIs. Significantly, the reported acceleration metrics account for the overhead associated with feature extraction and inference, demonstrating that such integration yields tangible reductions in end-to-end time-to-solution.

A. Impact of RHS and residual on the Krylov subspace and method selection

To further elucidate the theoretical relationship among the RHS, the residual, and the Krylov subspace, we quantify the influence of the RHS on the Krylov subspace and convergence behavior, providing a mathematical rationale for RHS-aware iterative method selection.

Consider the left-preconditioned form of the sparse linear system $Ax = b$ [8, 66]:

$$\widehat{A}x = \widehat{b}, \quad \widehat{A} = M^{-1}A, \ \ \widehat{b} = M^{-1}b. \tag{24}$$

Given an initial guess $x^{(0)}$, the initial residual $r^{(0)}$ and its preconditioned counterpart $z^{(0)}$ are defined as:

$$r^{(0)} = b - Ax^{(0)}, \quad z^{(0)} = M^{-1}r^{(0)} = M^{-1}(b - Ax^{(0)}). \tag{25}$$

In a Krylov subspace method, the k-th iterate is sought within the affine space $x^{(k)} \in x^{(0)} + \mathcal{K}_k(\widehat{A}, z^{(0)})$, where $\mathcal{K}_k(\widehat{A}, z^{(0)})$ denotes the associated Krylov subspace [6, 67]:

$$\begin{aligned}\mathcal{K}_k(\widehat{A}, z^{(0)}) &= span\{z^{(0)}, \widehat{A}z^{(0)}, \dots, \widehat{A}^{k-1}z^{(0)}\} \\ &= span\{p(\widehat{A})z^{(0)} \ : \ p \in \mathbb{P}_{k-1}\}.\end{aligned} \tag{26}$$

The preconditioned residual $z^{(k)}$ admits the following polynomial representation:

$$z^{(k)} = p_k(\widehat{A})z^{(0)}, \quad p_k(0) = 1. \tag{27}$$

Consequently, the RHS determines the generating vector $z^{(0)}$ and the Krylov subspace, and influences the residual trajectory through the polynomial action. This implies that the preconditioned residual encodes an RHS-induced, problem-specific structure that significantly impacts convergence.

For the quantitative analysis, we assume that $\widehat{A}$ is diagonalizable over $\mathbb{C}$ and admits the eigen-decomposition given below (If $\widehat{A}$ is not diagonalizable, the analysis can be extended via the Jordan canonical form [6, 53, 62].):

$$\widehat{A} = U\Lambda U^{-1}, \ \Lambda = diag(\varsigma_1, \dots, \varsigma_N), \ U = [u_1, \dots, u_N], \tag{28}$$

where ς_i and u_i denote the eigenvalues and corresponding eigenvectors of $\widehat{A}$, respectively. The vector $z^{(0)}$ can be expanded in this eigenbasis as:

$$z^{(0)} = U\xi = \sum_{i=1}^{N} \xi_i u_i, \ \ \xi = U^{-1}z^{(0)} = U^{-1}M^{-1}(b - Ax^{(0)}). \tag{29}$$

Substituting this expansion into Eq. 26 yields:

$$\begin{aligned}\mathcal{K}_k(\widehat{A}, z^{(0)}) &= span\{\sum_{i=1}^{N} p(\widehat{A})\xi_i u_i \ : \ p \in \mathbb{P}_{k-1}\} \\ &= span\{\sum_{i=1}^{N} \xi_i p(\varsigma_i) u_i \ : \ p \in \mathbb{P}_{k-1}\},\end{aligned} \tag{30}$$

where ξ contains the RHS-driven spectral weights that determine the extent to which each eigenmode u_i is represented in the Krylov subspace and residual evolution. If $\xi_i \approx 0$, the contribution of the mode u_i becomes negligible. Therefore, the RHS enters the Krylov process via the weights ξ, potentially imprinting physical structural characteristics onto the resulting subspace [66, 67].

To quantify the effects of the RHS on the subspace, we express $\mathcal{K}_k(\widehat{A}, z^{(0)})$ as $range(Z_k)$, where the Krylov matrix $Z_k \in \mathbb{C}^{N\times k}$ admits the factorization:

$$Z_k = [z^{(0)}, \widehat{A}z^{(0)}, \dots, \widehat{A}^{k-1}z^{(0)}] = U\,diag(\xi)\Phi_k(\Lambda), \tag{31}$$

where $\Phi_k(\Lambda)$ is a Vandermonde-type matrix [6, 53]. Thus, the dependence on the RHS is mediated entirely by ξ, following the mapping sequence $b \mapsto \xi \mapsto Z_k \mapsto \mathcal{K}_k$.

To bound the subspace perturbation induced by a perturbation in the RHS, consider $b \to b + \delta b$, which yields:

$$\delta z^{(0)} = M^{-1}\delta b, \quad \delta\xi = U^{-1}\delta z^{(0)} = U^{-1}M^{-1}\delta b. \tag{32}$$

Consequently, the perturbed $Z_k(b + \delta b)$ can be decomposed into the unperturbed matrix and a perturbation term:

$$Z_k(b+\delta b) = Z_k(b)+\Delta Z_k,\ \Delta Z_k = U\,diag(\delta\xi)\Phi_k(\Lambda). \tag{33}$$

Taking the spectral norm gives:

$$\begin{aligned}\|\Delta Z_k\|_2 &\le \|U\|_2\|\Phi_k(\Lambda)\|_2\|diag(\delta\xi)\|_2\\ &\le \|U\|_2\|\Phi_k(\Lambda)\|_2\|\delta\xi\|_2\\ &\le \|U\|_2\|\Phi_k(\Lambda)\|_2\|U^{-1}M^{-1}\|_2\|\delta b\|_2.\end{aligned} \tag{34}$$

Let $d_k(b, \delta b) = \|P_k(b + \delta b) - P_k(b)\|_2$ denote the distance between the subspaces $\mathcal{K}_k(\widehat{A}, z^{(0)}(b + \delta b))$ and $\mathcal{K}_k(\widehat{A}, z^{(0)}(b))$, where $P_k(\cdot)$ represents the corresponding orthogonal projector. Standard subspace perturbation theory yields the following bound [68, 69]:

$$\begin{aligned}d_k(b,\delta b) &\lesssim \frac{\|\Delta Z_k\|_2}{\sigma_{\min}(Z_k(b))}\\ &\le \frac{\|U\|_2\|\Phi_k(\Lambda)\|_2\|U^{-1}M^{-1}\|_2}{\sigma_{\min}(Z_k(b))}\,\|\delta b\|_2,\end{aligned} \tag{35}$$

where $\sigma_{\min}(Z_k(b))$ is the smallest singular value of $Z_k(b)$, which depends on the spectral distribution of ξ. If ξ suppresses certain modes such that Z_k becomes nearly rank-deficient, $\sigma_{\min}(Z_k(b))$ may decrease sharply, thereby degrading subspace conditioning and increasing sensitivity to RHS perturbations.

From a convergence perspective, the weights ξ also elucidate why different RHS instances may necessitate different methods. The residual norm $\|r^{(k)}\|_2$ can be bounded as:

$$\begin{aligned}\|r^{(k)}\|_2 &= \|Mp_k(\widehat{A})z^{(0)}\|_2\\ &= \|MUp_k(\Lambda)\xi\|_2\\ &\le \|MU\|_2\|p_k(\Lambda)\xi\|_2.\end{aligned} \tag{36}$$

Since $\xi = (MU)^{-1}r^{(0)}$, it follows that:

$$\|\xi\|_2 = \|(MU)^{-1}r^{(0)}\|_2 \le \|(MU)^{-1}\|_2\|r^{(0)}\|_2. \tag{37}$$

Rearranging this inequality yields:

$$\|r^{(0)}\|_2 \ge \frac{\|\xi\|_2}{\|(MU)^{-1}\|_2}. \tag{38}$$

Combining these estimates provides the following bound on the relative residual:

$$\frac{\|r^{(k)}\|_2}{\|r^{(0)}\|_2} \le \|MU\|_2\|U^{-1}M^{-1}\|_2\frac{\|p_k(\Lambda)\xi\|_2}{\|\xi\|_2}. \tag{39}$$

As $p_k(\Lambda)$ is diagonal, the bound reduces to a root-mean-square expression incorporating RHS-induced weights:

$$\frac{\|r^{(k)}\|_2}{\|r^{(0)}\|_2} \le \|MU\|_2\|U^{-1}M^{-1}\|_2\left(\sum_{i=1}^{N}\frac{|\xi_i|^2}{\sum_{j=1}^{N}|\xi_j|^2}|p_k(\varsigma_i)|^2\right)^{1/2}. \tag{40}$$

If the RHS concentrates ξ_i on a small subset of eigenvalues ς_i, rapid convergence is facilitated because p_k need only be small at those heavily weighted points, thereby rendering short-recurrence methods particularly suitable. Conversely, if the RHS distributes ξ_i broadly across the spectrum, p_k must attenuate components associated with numerous eigenvalues, which typically necessitates robust methods with stronger orthogonalization capabilities [32, 67]. Accordingly, method selection depends on the alignment between the method-induced suppression profile of p_k and the RHS-induced weight distribution ξ_i.

In summary, the RHS influences the Krylov subspace and convergence behavior via the problem-specific weights ξ embedded in the preconditioned residual, which quantify the sensitivity of the sparse linear system. This framework establishes a mathematical foundation for incorporating RHS-awareness into SPECTRA.

B. Pseudocode for matrix encoding

Algorithm 2 details the computational procedure for the expanded-channel matrix encoding scheme.

C. Derivation of diagonal mask half-width *d*

The half-width d in Eq. 13 is a critical hyperparameter governing the spatial extent of the mask, designed to ensure complete capture of all features in the initial feature map MF_{in} that originate from the diagonal of the input matrix image. Due to the preceding CNNs, which consist of Γ blocks each containing a convolutional layer (kernel size τ, stride 1) and a pooling layer (kernel size and stride ϱ), the feature of a single input diagonal element spreads out to form a square region on MF_{in}, which we term the "projection field". Consequently, the mask must be sized to fully encompass all projection fields while remaining as compact as possible to minimize the inclusion of confounding off-diagonal information. Establishing a principled value for d therefore necessitates formal analysis to determine the upper bound of the projection field width, denoted $\mathcal{F}$.

The exact width $\mathcal{F}_\ell$ of the projection field in the ℓ-th layer can be described by the following recurrence relation:

$$\mathcal{F}_\ell = \left\lceil \frac{(\mathcal{F}_{\ell-1} - 1) \times 1 + \tau}{\varrho} \right\rceil = \left\lceil \frac{\mathcal{F}_{\ell-1} + \tau - 1}{\varrho} \right\rceil. \tag{41}$$

However, the nonlinearity introduced by the ceiling function $\lceil\cdot\rceil$ precludes a straightforward closed-form solution. To obtain an analytical approximation, we relax this integer constraint by omitting the ceiling function, yielding

Algorithm 2: Expanded-channel matrix encoding.

Require: Matrix $A \in \mathbb{R}^{N\times N}$, image resolution $\mathcal{Q}$, scale range $[N_{min}, N_{max}]$.
Ensure : Five-channel matrix image $\mathcal{I} \in \mathbb{R}^{\mathcal{Q}\times\mathcal{Q}\times 5}$.

▷ Initialization
Initialize tensor $\mathcal{I} \in \mathbb{R}^{\mathcal{Q}\times\mathcal{Q}\times 5} \leftarrow 0$;
Initialize grids $\mathcal{C}, \mathcal{S}, r \in \mathbb{R}^{\mathcal{Q}\times\mathcal{Q}} \leftarrow 0$;
Initialize grids $\mathcal{P} \in \mathbb{R}^{\mathcal{Q}\times\mathcal{Q}} \leftarrow -\infty$, $\mathcal{V} \in \mathbb{R}^{\mathcal{Q}\times\mathcal{Q}} \leftarrow \infty$;
Compute block capacity $\gamma^2 \leftarrow (N/\mathcal{Q})^2$;
▷ Numerical distribution preprocessing
Compute value range $\delta \leftarrow \max(A) - \min(A)$;
Define transform function

$$\Psi(a) = \begin{cases} a - \min(A) + 1, & \text{if } \delta \le 255 \\ \log_2(a - \min(A) + 1), & \text{otherwise} \end{cases};$$

▷ Block-wise feature aggregation
for *each non-zero element* $a_{u,v} \in A$ **do**
  Map matrix indices (u, v) to image coordinates $(i, j) \leftarrow (\lfloor \frac{u-1}{N}\mathcal{Q} \rfloor + 1, \lfloor \frac{v-1}{N}\mathcal{Q} \rfloor + 1)$;
  Update block count $\mathcal{C}_{i,j} \leftarrow \mathcal{C}_{i,j} + 1$;
  Update block sum $\mathcal{S}_{i,j} \leftarrow \mathcal{S}_{i,j} + \Psi(a_{u,v})$;
  Update block maximum $\mathcal{P}_{i,j} \leftarrow \max(\mathcal{P}_{i,j}, a_{u,v})$;
  Update block minimum $\mathcal{V}_{i,j} \leftarrow \min(\mathcal{V}_{i,j}, a_{u,v})$;
▷ Block average magnitude computation
for $i \leftarrow 1$ **to** $\mathcal{Q}, j \leftarrow 1$ **to** $\mathcal{Q}$ **do**
  if $\mathcal{C}_{i,j} > 0$ **then**
    Compute average magnitude $r_{i,j} \leftarrow \mathcal{S}_{i,j}/\mathcal{C}_{i,j}$;
▷ Channel normalization and image generation
Define norm function $\aleph(x, x_{min}, x_{max}) = \text{clip}(\lfloor (x - x_{min})/(x_{max} - x_{min}) \times 255 \rfloor, 0, 255)$;
for $i \leftarrow 1$ **to** $\mathcal{Q}, j \leftarrow 1$ **to** $\mathcal{Q}$ **do**
  ▷ Blue channel: matrix scale
  Compute $\mathcal{I}_{i,j,3} \leftarrow \aleph(N, N_{min}, N_{max})$;
  if $\mathcal{C}_{i,j} > 0$ **then**
    ▷ Red channel: average magnitude
    Compute $\mathcal{I}_{i,j,1} \leftarrow \aleph(r_{i,j}, \min(r), \max(r))$;
    ▷ Green channel: non-zero density
    Compute $\mathcal{I}_{i,j,2} \leftarrow \aleph(\mathcal{C}_{i,j}, 0, \gamma^2)$;
    ▷ Peak channel: maximum magnitude
    Compute $\mathcal{I}_{i,j,4} \leftarrow \aleph(\mathcal{P}_{i,j}, \min(\mathcal{P}), \max(\mathcal{P}))$;
    ▷ Valley channel: minimum magnitude
    Compute $\mathcal{I}_{i,j,5} \leftarrow \aleph(\mathcal{V}_{i,j}, \min(\mathcal{V}), \max(\mathcal{V}))$;
return $\mathcal{I}$;

a linear recurrence relation that approximates the growth of the projection field:

$$\mathcal{F}_\ell \approx \frac{\mathcal{F}_{\ell-1} + \tau - 1}{\varrho}. \tag{42}$$

This linear recurrence can be unrolled to obtain a closed-form expression for $\mathcal{F}_\Gamma$:

$$\begin{aligned} \mathcal{F}_\Gamma &\approx \varrho^{-1}\mathcal{F}_{\Gamma-1} + \varrho^{-1}(\tau - 1) \\ &\approx \varrho^{-2}\mathcal{F}_{\Gamma-2} + \varrho^{-2}(\tau - 1) + \varrho^{-1}(\tau - 1) \\ &\vdots \\ &\approx \varrho^{-\Gamma}\mathcal{F}_0 + (\tau - 1)\sum_{i=1}^{\Gamma}\varrho^{-i}. \end{aligned} \tag{43}$$

Given that $\mathcal{F}_0 = 1$ and the summation term forms a finite geometric series, the expression simplifies to:

$$\mathcal{F}_\Gamma \approx \varrho^{-\Gamma} + (\tau - 1)\frac{\varrho^{-1}\left(1 - \varrho^{-\Gamma}\right)}{1 - \varrho^{-1}} = \varrho^{-\Gamma} + \frac{(\tau - 1)\left(\varrho^\Gamma - 1\right)}{\varrho^\Gamma(\varrho - 1)}. \tag{44}$$

While this formula provides the theoretical projection field width, determining the required d from it is nontrivial due to potential positional drift introduced by the pooling layer's discrete coordinate mapping, $output = \lfloor input/\varrho \rfloor$. This drift can shift the projection field of a diagonal pixel from the input image to become off-center relative to the diagonal of MF_{in}, invalidating the simple assumption that the required d is merely the field's radius, $d \approx \mathcal{F}_\Gamma/2$. Let d' denote the required distance from a pixel on the MF_{in} diagonal to ensure coverage of the entire corresponding projection field. In an ideal scenario with no center drift, the field is centered, and the required distance is its radius, $d' \approx \mathcal{F}_\Gamma/2$. However, in the worst-case scenario, the cumulative drift could position the MF_{in} diagonal near the edge of the projection field, thus requiring a coverage distance equal to the field's full width, $d' \approx \mathcal{F}_\Gamma$. This establishes the bounds for the required coverage distance:

$$\frac{\mathcal{F}_\Gamma}{2} \lesssim d' \lesssim \mathcal{F}_\Gamma \tag{45}$$

To construct a mask that robustly captures the full feature of the original image diagonal under all conditions, we adopt a conservative strategy by ensuring that d is sufficient to cover the maximum possible d', resulting in $d \ge \lceil d'_{\max} \rceil$. Meanwhile, to minimize the inclusion of confounding off-diagonal information, we select the tightest possible value for d by setting it equal to its lower bound, $\lceil d'_{\max} \rceil$, yielding the final formula:

$$d = \lceil d'_{\max} \rceil = \lceil \mathcal{F}_\Gamma \rceil = \left\lceil \varrho^{-\Gamma} + \frac{(\tau - 1)(\varrho^\Gamma - 1)}{\varrho^\Gamma(\varrho - 1)} \right\rceil. \tag{46}$$

This principled formulation yields the minimal integer value for the mask half-width d that is required to guarantee complete capture of all diagonal-related features originating from the input matrix image, and by being the tightest possible value, it inherently minimizes the inclusion of confounding off-diagonal information, thereby enhancing the stability and reliability of diagonal token generation.

D. Stage-wise model description of SPECTRA

Table 4 provides a stage-wise description of the model configurations for SPECTRA.

Table 4
Model descriptions for SPECTRA. Prefixes "M", "R", and "D" denote modules and data streams associated with the matrix, the RHS, and the matrix diagonal, respectively. The over-bar notation "$\bar{\cdot}$" indicates the count of a specified item, such as channels, tokens, or repeated modules. Convolutional layers are specified as "Conv (i, o, k, s, p)", where i, o, k, s, and p represent input channels, output channels, kernel size, stride, and padding, respectively. "Drop (x)" represents a dropout layer with a rate of x.

Stage	Input Shape	Module	Configuration	Output Shape
Encoding	$N \times N$	M encoding	2D images encoding (R, G, B, P, V)	$\bar{5} \times Q \times Q$
	$1 \times N$	R encoding	1D images encoding (R, G, B, P, V)	$\bar{5} \times 1 \times Q$
CNNs	$\bar{5} \times Q \times Q$	M CNNs	Conv2d $(i=5,\ o=32,\ k=3,\ s=1,\ p=same) \Rightarrow$ ReLU $\Rightarrow$ MaxPool Conv2d $(i=32,\ o=64,\ k=3,\ s=1,\ p=same) \Rightarrow$ ReLU $\Rightarrow$ MaxPool	$\overline{64} \times \frac{Q}{4} \times \frac{Q}{4}$
	$\bar{5} \times 1 \times Q$	R CNNs	Conv1d $(i=5,\ o=32,\ k=3,\ s=1,\ p=same) \Rightarrow$ ReLU $\Rightarrow$ MaxPool Conv1d $(i=32,\ o=64,\ k=3,\ s=1,\ p=same) \Rightarrow$ ReLU $\Rightarrow$ MaxPool	$\overline{64} \times 1 \times \frac{Q}{4}$
Tokenizer	$\overline{64} \times \frac{Q}{4} \times \frac{Q}{4}$	M tokenizer D tokenizer	Matrix attention MA-based grouping $(\lambda_M = 64)$ Mask $(d=2) \Rightarrow$ Flatten $\Rightarrow$ Linear $(64 \times \frac{Q}{4} \times \frac{Q}{4} \to 64)$	$\overline{64} \times 64$ $\bar{1} \times 64$
	$\overline{64} \times 1 \times \frac{Q}{4}$	R tokenizer	RHS attention RA-based grouping $(\lambda_R = 8)$	$\bar{8} \times 64$
Transformer	$\overline{64+1+8} \times 64$	Encoder $\times\overline{12}$	LN $\Rightarrow$ MHA $(U=4,\ D_h=16) \Rightarrow$ Residual connection MHA { Query, Key, Value from Linear $(64 \to 64) \times \bar{3}$ Softmax $\left(\frac{Query \cdot Key^T}{\sqrt{D_h}}\right) \Rightarrow$ Drop $(0.05) \Rightarrow$ Aggregate Value Concat Heads $\Rightarrow$ Linear $(64 \to 64) \Rightarrow$ Drop (0.05) LN $\Rightarrow$ FFN $\Rightarrow$ Residual connection FFN { Linear $(64 \to 256) \Rightarrow$ GELU $\Rightarrow$ Drop (0.05) Linear $(256 \to 64) \Rightarrow$ Drop (0.05)	$\overline{64+1+8} \times 64$
Projector	$\overline{64+1} \times 64$	M projector	Matrix attention MA'-based projection $\Rightarrow$ Residual connection	$\overline{64} \times \frac{Q}{4} \times \frac{Q}{4}$
	$\bar{8} \times 64$	R projector	RHS attention RA'-based projection $\Rightarrow$ Residual connection	$\overline{64} \times 1 \times \frac{Q}{4}$
Classification	$64 \times (\frac{Q}{16}^2 + \frac{Q}{4})$	MLP	Linear $(64 \times (\frac{Q}{4}^2 + \frac{Q}{4}) \to 128) \Rightarrow$ ReLU $\Rightarrow$ Drop (0.5) Linear $(128 \to K)$	K

E. Cost-effectiveness analysis

To rigorously evaluate the cost-effectiveness of SPECTRA, we examine the trade-off between the one-time computational cost of model training and the cumulative runtime savings accrued during deployment.

We quantify the performance gain by defining the *average time saved per system* ($\Delta\mathcal{T}$), calculated by comparing the end-to-end execution times of SPECTRA and the engineering heuristic HR on the fixed-RHS test dataset:

$$\Delta\mathcal{T} = \frac{1}{\phi}\sum_{i=1}^{\phi}\left[(\check{\mathcal{T}}_F^i + \check{\mathcal{T}}_S^i) - (\hat{\mathcal{T}}_F^i + \hat{\mathcal{T}}_M^i + \hat{\mathcal{T}}_S^i)\right] \approx 0.15s, \quad (47)$$

where ϕ represents the number of test samples, and $\check{\mathcal{T}}$ and $\hat{\mathcal{T}}$ denote the execution times for HR and SPECTRA, respectively. Notably, to ensure robustness and consistency with the metric defined in Eq. 22, any selected method that fails to converge is assigned a penalty time equivalent to the maximum convergence time for the corresponding system. Given that the training phase requires fewer than four GPU hours, the *break-even point* (*BEP*), defined as the number of solved systems required to amortize the training cost, is calculated as follows:

$$BEP = \frac{Training\ Cost}{\Delta\mathcal{T}} \approx \frac{4h}{0.15s} = 96000. \quad (48)$$

Within the context of high-throughput scientific computing and the longevity of solver libraries, a BEP of 96000 represents a trivial workload to amortize the one-time training cost. Furthermore, this estimate is conservative, as the test dataset consists of relatively small-scale systems ($10^3 \le N \le 10^4$). For larger real-world systems where $N \gg 10^4$, $\Delta\mathcal{T}$ increases substantially, thereby shortening the amortization period and validating the superior cost-efficiency of SPECTRA as an intelligent decision system.

F. More ablation studies

To further dissect the individual contributions of each channel within the encoding schemes, we systematically isolate and evaluate the effectiveness of the P and V channels for matrix encoding and all five channels (R, G, B, P, and V) for RHS encoding. This analysis complements the primary ablation studies detailed in Section 4.4, which demonstrated the collective benefits of integrating these channels. For fair comparison, all ablation variants in this section are evaluated at an image resolution of $Q = 64$.

Table 5
Ablation study dissecting the individual and combined effects of the expanded P and V channels for matrix encoding. Results consistently demonstrate significant individual contributions from both the P and V channels, with their synergistic combination proving most effective and validating their complementary nature.

Model	Channels			*ACC@1* (%)↑	*ACC@5* (%)↑	*Success* (%)↑	*Efficiency* (%)↑	*Speedup* (×)↑
	RGB	*P*	*V*					
Fixed-RHS								
CNNs	✔	✘	✘	70.80	91.37	96.21	9.27	3.18
	✔	✔	✘	71.15	91.46	96.26	10.95	3.21
	✔	✘	✔	71.02	91.52	96.24	10.11	3.20
	✔	✔	✔	**71.34**	**91.52**	**96.30**	**11.15**	**3.22**
CNNs + Transformer	✔	✘	✘	73.33	91.92	96.78	15.20	3.48
	✔	✔	✘	74.01	92.18	97.05	16.88	3.58
	✔	✘	✔	73.85	92.11	96.97	16.42	3.55
	✔	✔	✔	**74.44**	**92.33**	**97.21**	**17.77**	**3.62**
SPECTRA	✔	✔	✔	**77.72**	**92.79**	**97.70**	**25.44**	**4.01**
	✔	✘	✔	77.24	92.68	97.63	24.18	3.95
	✔	✔	✘	77.08	92.66	97.58	23.53	3.93
	✔	✘	✘	76.76	92.41	97.40	21.52	3.88
Variable-RHS								
SPECTRA	✔	✔	✔	**78.23**	**96.72**	**97.99**	**39.10**	**6.97**
	✔	✘	✔	77.81	96.25	97.91	36.57	6.51
	✔	✔	✘	77.95	96.43	97.94	37.48	6.60
	✔	✘	✘	77.12	95.53	97.80	33.83	6.34

F.1. Individual effect of expanded channels P and V for matrix encoding

To isolate the distinct contributions of the P and V channels, we examined their individual and combined impacts on matrix encoding across both the fixed-RHS and variable-RHS datasets. As detailed in Table 5, both channels individually enhance performance, while their combination consistently yields the most significant gains on both datasets.

For instance, on the fixed-RHS dataset, augmenting the standard RGB encoding of the CNN with only the P channel improves *ACC@1* by 0.35% and *Efficiency* by 1.68% (row 1 vs. 2), while the V channel alone provides boosts of 0.22% and 0.84%, respectively (row 1 vs. 3). This trend also holds for the more powerful hybrid CNN-Transformer backbone, confirming their robust utility across different architectures. Crucially, their combined use outperforms either single channel across both architectures. For the CNN, employing both P and V channels yields total improvements of 0.54% in *ACC@1* and 1.88% in *Efficiency* (row 1 vs. 4).

The indispensability of the P and V channels is consistently evident in the variable-RHS setting. When ablating the complete SPECTRA model, removing the P channel leads to a 0.42% drop in *ACC@1* and a 2.53% drop in *Efficiency* (row 13 vs. 14), while removing the V channel corresponds to declines of 0.28% and 1.62%, respectively (row 13 vs. 15). The removal of both channels results in more substantial drops of 1.11% in *ACC@1* and 5.27% in *Efficiency* (row 13 vs. 16), losses greater than the sum of their individual impacts, confirming their synergistic effect in this more complex scenario.

These results underscore that the P and V channels provide unique and complementary information about the matrix's numerical distribution, forming a comprehensive numerical profile essential for faithful matrix encoding.

F.2. Individual impact of channels R, G, B, P and V for RHS encoding

To precisely quantify the individual effects of the five channels (R, G, B, P, and V) on RHS encoding, we analyzed their individual and collective effectiveness exclusively on the variable-RHS dataset, as its diverse RHS configurations provide the necessary conditions for evaluation. As detailed in Table 6, each channel individually provides a substantial performance improvement compared to the matrix-only model, and the complete five-channel encoding scheme achieves the highest performance.

For instance, starting from a matrix-only hybrid CNN-Transformer model, the introduction of a single channel yields significant improvements. Adding only the P channel boosts *ACC@1* by 10.02% and *Efficiency* by 14.5% (row 1 vs. 5), while the R channel yields similarly substantial increases of 9.29% and 13.67%, respectively (row 1 vs. 2). Crucially, employing all five channels together surpasses any single-channel configuration, elevating *ACC@1* by 12.06% and *Efficiency* by 17.03% for the hybrid CNN-Transformer model (row 1 vs. 7), confirming their collective benefit.

The importance of each channel is further clarified through ablation studies on the complete SPECTRA model. Removing the B channel results in minimal performance drops of just 0.18% in *ACC@1* and 0.56% in *Efficiency*

Table 6
Ablation study dissecting the individual and collective impacts of the R, G, B, P, V channels for RHS encoding. Results demonstrate that all channels contribute positively to varying degrees, and their synergistic combination is key to best performance.

Model	Channels					*ACC@1* (%)↑	*ACC@5* (%)↑	*Success* (%)↑	*Efficiency* (%)↑	*Speedup* (×)↑
	R	***G***	***B***	***P***	***V***					
Variable-RHS										
CNNs + Transformer	✘	✘	✘	✘	✘	62.86	76.56	95.00	11.21	4.48
	✔	✘	✘	✘	✘	72.15	93.12	96.35	24.88	5.61
	✘	✔	✘	✘	✘	70.53	93.53	96.01	22.14	5.33
	✘	✘	✔	✘	✘	63.89	80.99	95.28	13.05	4.56
	✘	✘	✘	✔	✘	72.88	92.48	96.24	25.71	5.70
	✘	✘	✘	✘	✔	71.91	92.98	96.17	24.13	5.55
	✔	✔	✔	✔	✔	**74.92**	**94.45**	**96.73**	**28.24**	**5.95**
SPECTRA	✔	✔	✔	✔	✔	**78.23**	**96.72**	**97.99**	**39.10**	**6.97**
	✘	✔	✔	✔	✔	77.58	96.40	97.82	36.95	6.78
	✔	✘	✔	✔	✔	77.92	96.55	97.75	38.15	6.88
	✔	✔	✘	✔	✔	78.05	96.63	97.94	38.54	6.92
	✔	✔	✔	✘	✔	77.35	96.18	97.89	36.21	6.72
	✔	✔	✔	✔	✘	77.71	96.31	97.85	37.44	6.81
	✘	✘	✘	✘	✘	64.31	84.15	95.32	15.82	4.81

(row 8 vs. 11), as its scale information is largely redundant with the matrix encoding, whereas removing the P channel incurs the most substantial penalty, with drops of 0.88% and 2.89%, respectively (row 8 vs. 12). Making SPECTRA entirely RHS-agnostic by removing all five channels leads to the most drastic degradation, with *ACC@1* plummeting by 13.92% and *Efficiency* by 23.28% (row 8 vs. 14), a loss far exceeding that of any individual channel and highlighting their synergistic necessity.

These results underscore that each R, G, B, P, and V channel is paramount for multi-faceted RHS representation despite their varied contributions, and that their synergistic combination is essential for comprehensive RHS encoding.

CRediT authorship contribution statement

Kaiqi Zhang: Writing – original draft, Investigation, Conceptualization, Methodology, Visualization. **Dali Chang:** Methodology, Validation, Visualization. **Mingguan Yang:** Resources, Supervision, Project administration. **Jing Zhao:** Writing – review & editing, Supervision. **Wangdong Yang:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors appreciate the editor and reviewers for their time and effort in evaluating this manuscript.

Data availability

The data that support the findings of this study are openly available in the SuiteSparse dataset at `https://sparse.tamu.edu`, reference number [29].

References

[1] J. Tu, G. H. Yeoh, C. Liu, Y. Tao, Computational fluid dynamics: a practical approach, Elsevier, 2023.
[2] S. Ereiz, I. Duvnjak, J. F. Jiménez-Alonso, Review of finite element model updating methods for structural applications, in: Structures, Vol. 41, Elsevier, 2022, pp. 684–723.
[3] M. Jiang, Y. Li, L. Lei, J. Hu, A review on fast direct methods of surface integral equations for analysis of electromagnetic scattering from 3-d pec objects, Electronics 11 (22) (2022) 3753.
[4] T. A. Davis, S. Rajamanickam, W. M. Sid-Lakhdar, A survey of direct methods for sparse linear systems, Acta Numerica 25 (2016) 383–566.
[5] T. A. Davis, Direct methods for sparse linear systems, SIAM, 2006.
[6] Y. Saad, Iterative methods for sparse linear systems, SIAM, 2003.
[7] J. Scott, M. Tůma, Algorithms for sparse linear systems, Springer Nature, 2023.
[8] M. Benzi, Preconditioning techniques for large linear systems: a survey, Journal of computational Physics 182 (2) (2002) 418–477.
[9] H. Zou, X. Xu, C.-S. Zhang, A survey on intelligent iterative methods for solving sparse linear algebraic equations, arXiv preprint arXiv:2310.06630 (2023).
[10] Z. Tang, H. Zhang, J. Chen, Graph neural networks for selection of preconditioners and krylov solvers, in: NeurIPS 2022 Workshop: New Frontiers in Graph Learning, 2022.
[11] H.-B. Sun, Y.-F. Jing, X.-W. Xu, A new matrix feature selection strategy in machine learning models for certain krylov solver prediction, Journal of Classification (2024) 1–18.
[12] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, H. Van der Vorst, Templates for the solution of linear systems: building blocks for iterative methods, SIAM, 1994.
[13] X. Zhu, Intelligent decision support systems for improving financial forecasting and market trend analysis, Expert Systems with Applications 297 (2026) 129462.

[14] M. Tang, Y. Wu, J. Wang, An intelligent decision system for debugging engine fuel regulators, Expert Systems with Applications 265 (2025) 125987.

[15] A. Waqar, Intelligent decision support systems in construction engineering: An artificial intelligence and machine learning approaches, Expert Systems with Applications 249 (2024) 123503.

[16] W. Ma, J. Wu, B. Sun, X. Leng, W. Miao, Z. Gao, W. Li, Intelligent vehicle decision-making strategy integrating spatiotemporal features at roundabout, Expert Systems with Applications 273 (2025) 126779.

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

[18] S. Bhowmick, B. Toth, P. Raghavan, Towards low-cost, high-accuracy classifiers for linear solver selection, in: Computational Science–ICCS 2009: 9th International Conference Baton Rouge, LA, USA, May 25-27, 2009 Proceedings, Part I, Springer, 2021, pp. 463–472.

[19] V. Eijkhout, E. Fuentes, Machine learning for multi-stage selection of numerical methods, New Advances in Machine Learning. INTECH (2010) 117–136.

[20] P. R. Eller, J.-R. C. Cheng, R. S. Maier, Dynamic linear solver selection for transient simulations using machine learning on distributed systems, in: 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum, IEEE, 2012, pp. 1915–1924.

[21] P. Motter, K. Sood, E. Jessup, 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.

[22] E. Jessup, P. Motter, B. Norris, K. Sood, Performance-based numerical solver selection in the lighthouse framework, SIAM Journal on Scientific Computing 38 (5) (2016) S750–S771.

[23] Y. Zabegaev, E. Keilegavlen, E. Iversen, I. Berre, Automated linear solver selection for simulation of multiphysics processes in porous media, Computer Methods in Applied Mechanics and Engineering 426 (2024) 117031.

[24] K. Yamada, T. Katagiri, H. Takizawa, K. Minami, M. Yokokawa, T. Nagai, 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.

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

[26] Y. Funk, M. Götz, 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.

[27] H. Xiong, W. Yang, W. He, S. Lin, K. Li, K. Li, Mm-autosolver: A multimodal machine learning method for the auto-selection of iterative solvers and preconditioners, Journal of Parallel and Distributed Computing (2025) 105144.

[28] K. Zhang, M. Yang, D. Chang, C. Chen, Y. Zhang, K. He, J. Zhao, Relative-absolute fusion: Rethinking feature extraction in image-based iterative method selection for solving sparse linear systems, arXiv preprint arXiv:2510.00500 (2025).

[29] T. A. Davis, Y. Hu, The university of florida sparse matrix collection, ACM Transactions on Mathematical Software (TOMS) 38 (1) (2011) 1–25.

[30] J. Nocedal, S. J. Wright, Numerical optimization, Springer, 2006.

[31] M. R. Hestenes, E. Stiefel, et al., Methods of conjugate gradients for solving linear systems, Journal of research of the National Bureau of Standards 49 (6) (1952) 409–436.

[32] Y. Saad, M. H. Schultz, Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on scientific and statistical computing 7 (3) (1986) 856–869.

[33] J. Liesen, Z. Strakos, Krylov subspace methods: principles and analysis, Numerical Mathematics and Scie, 2013.

[34] J. Liesen, P. Tichý, Convergence analysis of krylov subspace methods, GAMM-Mitteilungen 27 (2) (2004) 153–173.

[35] D. Titley-Peloquin, J. Pestana, A. J. Wathen, Gmres convergence bounds that depend on the right-hand-side vector, IMA Journal of Numerical Analysis 34 (2) (2014) 462–479.

[36] S. Balay, S. Abhyankar, M. F. Adams, S. Benson, J. Brown, P. Brune, K. Buschelman, E. M. Constantinescu, L. Dalcin, A. Dener, V. Eijkhout, J. Faibussowitsch, W. D. Gropp, V. Hapla, T. Isaac, P. Jolivet, D. Karpeev, D. Kaushik, M. G. Knepley, F. Kong, S. Kruger, D. A. May, L. C. McInnes, R. T. Mills, L. Mitchell, T. Munson, J. E. Roman, K. Rupp, P. Sanan, J. Sarich, B. F. Smith, S. Zampini, H. Zhang, H. Zhang, J. Zhang, PETSc Web page, https://petsc.org/ (2025). URL https://petsc.org/

[37] S. Balay, S. Abhyankar, M. F. Adams, S. Benson, J. Brown, P. Brune, K. Buschelman, E. Constantinescu, L. Dalcin, A. Dener, V. Eijkhout, J. Faibussowitsch, W. D. Gropp, V. Hapla, T. Isaac, P. Jolivet, D. Karpeev, D. Kaushik, M. G. Knepley, F. Kong, S. Kruger, D. A. May, L. C. McInnes, R. T. Mills, L. Mitchell, T. Munson, J. E. Roman, K. Rupp, P. Sanan, J. Sarich, B. F. Smith, H. Suh, S. Zampini, H. Zhang, H. Zhang, J. Zhang, PETSc/TAO users manual, Tech. Rep. ANL-21/39 - Revision 3.23, Argonne National Laboratory (2025). doi:10.2172/2476320.

[38] S. Balay, W. D. Gropp, L. C. McInnes, B. F. Smith, Efficient management of parallelism in object oriented numerical software libraries, in: E. Arge, A. M. Bruaset, H. P. Langtangen (Eds.), Modern Software Tools in Scientific Computing, Birkhäuser Press, 1997, pp. 163–202.

[39] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems 25 (2012).

[40] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[41] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, et al., A survey on vision transformer, IEEE transactions on pattern analysis and machine intelligence 45 (1) (2022) 87–110.

[42] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, M. Shah, Transformers in vision: A survey, ACM computing surveys (CSUR) 54 (10s) (2022) 1–41.

[43] Y. Wang, Y. Deng, Y. Zheng, P. Chattopadhyay, L. Wang, Vision transformers for image classification: A comparative survey, Technologies 13 (1) (2025) 32.

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[45] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, arXiv preprint arXiv:2010.11929 (2020).

[46] B. Wu, C. Xu, X. Dai, A. Wan, P. Zhang, Z. Yan, M. Tomizuka, J. Gonzalez, K. Keutzer, P. Vajda, Visual transformers: Token-based image representation and processing for computer vision, arXiv preprint arXiv:2006.03677 (2020).

[47] A. Yang, M. Li, Y. Ding, Y. He, M. Bi, Q. Zheng, Ctfn: Multi-scale cnn and transformer with graph encodings fusion network for hyperspectral image classification, Expert Systems with Applications 288 (2025) 128063.

[48] E. Goceri, An efficient network with cnn and transformer blocks for glioma grading and brain tumor classification from mris, Expert Systems with Applications 268 (2025) 126290.

[49] Y.-R. Qiang, Q.-Y. Zhou, J.-N. Li, M.-Y. Xie, X. Cui, S.-W. Zhang, Classification of alzheimer's disease by jointing 3d depthwise separable convolutional neural network and transformer, Expert Systems with Applications (2025) 127720.

[50] S. Sreelakshmi, S. V. Chandra, A hybrid fusion network using convolutional vision transformers for landslide identification, Expert Systems with Applications (2025) 129688.

[51] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows,

in: Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 10012–10022.
[52] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jégou, Training data-efficient image transformers & distillation through attention, in: International conference on machine learning, PMLR, 2021, pp. 10347–10357.
[53] G. H. Golub, C. F. Van Loan, Matrix computations, JHU press, 2013.
[54] R. S. Varga, Geršgorin and his circles, Vol. 36, Springer Science & Business Media, 2011.
[55] Q. Wang, B. Li, T. Xiao, J. Zhu, C. Li, D. F. Wong, L. S. Chao, Learning deep transformer models for machine translation, arXiv preprint arXiv:1906.01787 (2019).
[56] A. Baevski, M. Auli, Adaptive input representations for neural language modeling, arXiv preprint arXiv:1809.10853 (2018).
[57] W. Dörfler, A convergent adaptive algorithm for poisson's equation, SIAM Journal on Numerical Analysis 33 (3) (1996) 1106–1124.
[58] P. L. Gould, Y. Feng, Introduction to linear elasticity, Vol. 2, Springer, 1994.
[59] D. Boffi, Convection-diffusion problems. an introduction to their analysis and numerical solution. (2020).
[60] D. A. Juraev, P. Agarwal, E. E. Elsayed, N. Targyn, Helmholtz equations and their applications in solving physical problems, Advanced Engineering Science 4 (2024) 54–64.
[61] H. Zhang, C. Zhang, OpenMat, https://github.com/zhf-0/OpenMat (2024).
[62] R. A. Horn, C. R. Johnson, Matrix analysis, Cambridge university press, 2012.
[63] L. N. Trefethen, D. Bau, Numerical linear algebra, SIAM, 2022.
[64] M. Hollander, D. A. Wolfe, E. Chicken, Nonparametric statistical methods, John Wiley & Sons, 2013.
[65] J. D. Gibbons, S. Chakraborti, Nonparametric statistical inference: revised and expanded, CRC press, 2014.
[66] H. A. Van der Vorst, Iterative Krylov methods for large linear systems, no. 13, Cambridge University Press, 2003.
[67] A. Greenbaum, Iterative methods for solving linear systems, SIAM, 1997.
[68] G. W. Stewart, Stochastic perturbation theory, SIAM review 32 (4) (1990) 579–610.
[69] R.-C. Li, Matrix perturbation theory, Handbook of linear algebra (2006) 15–21.