Data-Driven Reduced-Order Models of Control Room Procedural Execution

Niharika Karnik[a], Jisuk Kim[a], Thomas Ulrich[a] and Ronald Boring[a]

[a]Idaho National Laboratory, Idaho Falls, ID, USA

Abstract: Digitization of nuclear power plant operating procedures is a critical step toward enabling automated operations, intelligent operator support systems, and next-generation digital control rooms. However, transforming legacy paper-based procedures into machine-readable operational knowledge requires both structured procedural representations and methods for analyzing execution behavior across diverse plant conditions. This study presents a framework for extracting and analyzing procedural execution data from digitized emergency operating procedures using knowledge graphs, scenario-driven execution traces, and probabilistic risk assessment–based cut sets. Execution trajectories generated from operational scenarios are encoded as binary procedural visitation vectors and analyzed using singular value decomposition and latent semantic analysis (LSA) to identify dominant execution modes and semantically significant procedural patterns. The results show that, despite the combinatorial complexity of emergency operating procedures, including branching logic, conditional actions, response-not-obtained paths, and interprocedure transfers, the underlying execution behavior exhibits a low-dimensional structure governed by a limited set of plant-state variables. The dominant singular modes correspond to physically interpretable operational behaviors, including common response pathways, recovery strategies, and equipment availability dependencies. In addition, LSA identifies repeated and semantically related procedural actions that may inform human reliability analysis and assessments of the cognitive workload of operators. These findings demonstrate the potential of using AI-enabled procedural digitization and reduced-order modeling approaches to support automated operations, digital twins, and advanced decision-support systems for nuclear power plant modernization.

1. INTRODUCTION

Nuclear power plant operations are governed by highly proceduralized workflows that encode decades of operating experience, safety analysis, and regulatory practice. Emergency operating procedures (EOPs), in particular, guide control room operators through abnormal and emergency plant conditions by prescribing conditional checks, operator actions, response-not-obtained (RNO) actions, branching logic, and transitions to other procedures [1] [2]. Although these documents are traditionally treated as human-readable instructions, digitization efforts increasingly require them to be represented as machine-readable structures that can support automated operations, simulator coupling [3], intelligent operator support, and dynamic human reliability analysis (HRA) [4] [5] [6] [7]. A central challenge in this transition is that procedures do not define a single linear sequence of actions. Instead, they define a space of possible execution paths. The path followed by an operator depends on the evolving plant state, equipment availability, instrumentation response, and the success or failure of prior actions. From the perspective of probabilistic risk assessment (PRA), these execution paths may be conditioned on initiating events, fault tree cut sets, and combinations of component failures that determine which procedural branches are encountered. From the perspective of simulator-based analysis, each scenario produces a trajectory through the procedure that identifies which steps are visited, which steps are skipped, and which branches or RNO paths are activated.

Prior work in dynamic HRA has emphasized the need to move beyond static, worksheet-based representations of human actions toward simulation-based approaches capable of modeling operator behavior over time. The Human Unimodel for Nuclear Technology to Enhance Reliability (HUNTER) was developed as a dynamic HRA framework for modeling human performance in nuclear power plant operations, including operator tasks, performance-shaping factors, and scenario-dependent behavior. Recent work [8] has further coupled HUNTER with the Rancor Microworld simulator [9] to support realistic human performance modeling and

Monte Carlo exploration of operator behavior in plant scenarios. These efforts demonstrate the value of simulator-connected HRA tools for generating and analyzing human performance data, and they motivate the development of complementary methods for extracting structured information from the procedures themselves.

Prior work in procedure digitization [10] [11] [12] demonstrated that the unstructured paper documents of legacy nuclear operating procedures can be transformed into structured, machine-readable knowledge representations capable of supporting downstream reasoning and analysis. Building upon these digitization efforts, the work discussed in this paper treats procedures not merely as static instructional documents but as rich sources of operational and semantic data. When procedural knowledge graphs are exercised under PRA-derived cut sets or generated plant scenarios, each execution through the procedure can be represented as a trajectory through the graph. These trajectories encode which procedural steps are visited, skipped, branched upon, or revisited under different plant conditions and equipment availability states. In the present work, the analysis focuses on intraprocedure execution behavior within a single EOP. Although interprocedure transfers represent an important extension toward full plant operational modeling, they are outside the scope of this study. This knowledge-graph-based representation enables a new class of procedural analyses by treating execution pathways as mathematical objects rather than isolated step sequences. In this work, execution trajectories generated from scenario-driven and PRA-informed conditions are encoded as binary step-visitation vectors, allowing techniques such as singular value decomposition (SVD) and latent semantic analysis (LSA) to identify dominant execution patterns and relationships among trajectories to generate reduced order models (ROMs). The central hypothesis is that despite the apparent combinatorial complexity of emergency procedures, realistic execution pathways occupy a much lower-dimensional structure governed by recurring operational behaviors, major branching decisions, equipment recovery strategies, and RNO actions. The results of using EOPs represented as structured knowledge graphs show that procedural execution can be compressed into a small number of dominant modes that still retain more than 99% of the observed variability. These modes reveal interpretable operational themes associated with power restoration, recovery actions, and contingency responses, while PRA-based analyses demonstrate that minimal cut sets naturally cluster according to shared procedural failure structures. Collectively, these findings establish digitized procedures as analyzable behavioral datasets that can support reduced-order modeling, dynamic HRA, operator training, procedure validation, digital twins, and future AI-enabled operational support systems.

2. METHODOLOGY

This work developed a data-driven reduced-order modeling framework for analyzing execution behavior in digitized EOPs. The methodology combines structured procedural knowledge graph representations, scenario-dependent trajectories and PRA-informed cut-set generation, and LSA using SVD. The overall workflow consists of: (1) representing the procedure as a structured graph that preserves nominal and RNO logic, (2) generating an ensemble of execution trajectories from scenarios and PRA-informed branching conditions, (3) encoding trajectories as binary visitation vectors, and (4) extracting the dominant modes governing procedural execution through SVD/LSA.

2.1. Procedural Knowledge Graph Representation

To convert semistructured or unstructured paper procedures into a machine-readable format, knowledge graphs can be generated using established ontologies and data structures, as demonstrated by Ulrich et al. [12], Karnik et al. [10], and Kim et al. [11]. The semantic content of the EOP procedure is extracted and represented as interconnected nodes and relationships, as illustrated in Figure 1. This graph-based representation captures procedural logic, state transitions, and key physical entities, enabling further analyses such as trajectory generation and PRA cut-set construction. By explicitly encoding branching pathways and entity interactions, the knowledge graph provides a structured framework for identifying critical actions and system components that drive procedure execution. The resulting knowledge graph, stored in a machine-readable format, serves as the foundational data representation for the data-driven methodology developed in this work.

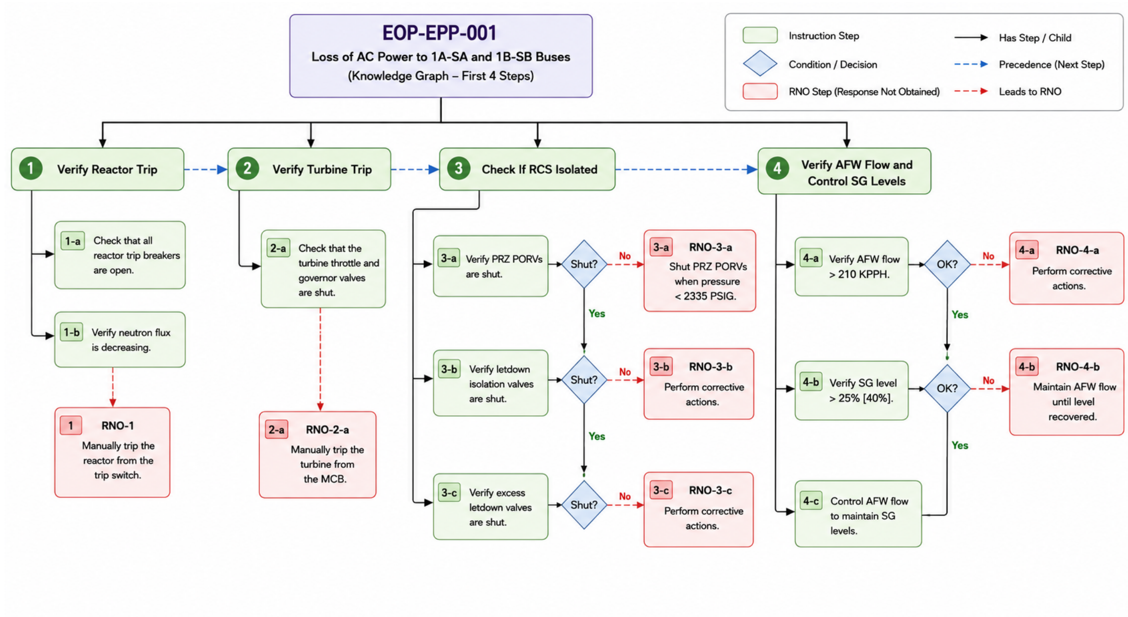

Figure 1: High-level knowledge graph of the first four steps of a single EOP showing nominal instruction paths and RNO recovery branches.

2.2. Scenario Space and Procedural Trajectory Generation

Execution trajectories were generated by traversing the procedural knowledge graph (KG) under varying plant-state conditions. As shown in Figure 2, seven preliminary plant-state conditions were identified by subject-matter experts through manual review of the EOP-EPP-001 knowledge graph. The considered plant-state conditions are not intended to be exhaustive; instead, they represent a small subset chosen to support the development of the methodology. Each condition was represented as a binary state (True/False) in order to define the set of decision points that influence procedural execution. These conditions were used to systematically traverse the knowledge graph in a manner analogous to how an operator would execute the procedure. At each conditional branch, the traversal followed either the nominal path or the associated RNO path based on the assumed plant state. Explicit GO TO directives were followed unconditionally, and RETURN TO loops were executed once to capture procedural monitoring behavior before termination. Cross-procedure transfers were recorded as terminal states but were not expanded further, restricting the present analysis to execution behavior within EOP-EPP-001.

A full-factorial design across these seven binary variables yields $2^7 = 128$ possible combinations. After removing physically inconsistent states, such as Emergency Diesel Generator (EDG) auto start success when EDGs are unavailable, $n = 96$ physically realizable scenarios remained. For each scenario, a deterministic traversal of the KG was executed. Each scenario therefore produced a unique execution trajectory represented as an ordered list of visited procedure steps. The EOP-EPP-001 procedure contains 36 primary procedural steps organized into a hierarchical structure of substeps, notes, cautions, and embedded RNO actions that are then converted into trajectories (Figure 2). In the present work, plant-state conditions were prescribed a priori to generate and analyze possible execution paths. In future applications, these trajectories could be generated dynamically through integration with a plant simulator, where evolving plant conditions determine the next procedural action in real time. For this analysis, however, it was assumed that the digital operator strictly followed the procedure as written and did not deviate from prescribed procedural guidance.

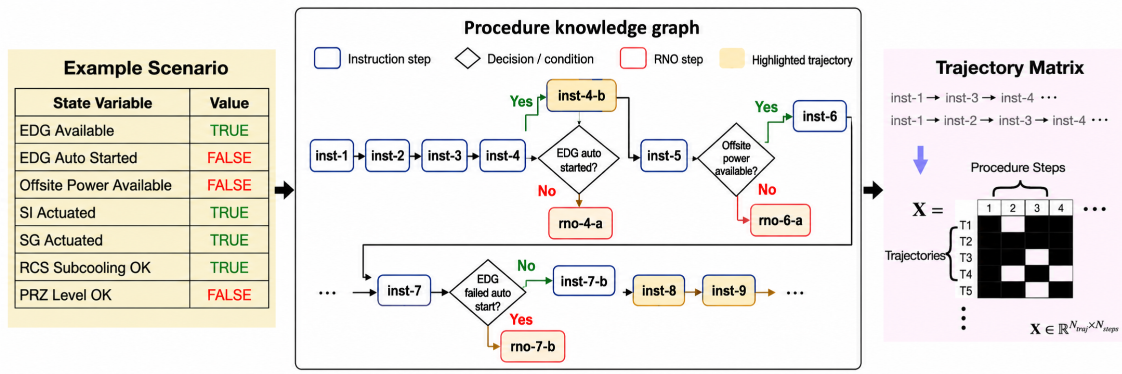

Figure 2: Workflow for constructing the scenario-based trajectory matrix from the EOP knowledge graph. Assumed plant conditions or state variables determine traversal through nominal and RNO branches, producing execution trajectories that are encoded into a binary matrix $\mathbf{X} \in \mathbb{R}^{N_{traj} \times N_{steps}}$, where rows represent scenarios and columns represent procedure steps.

2.2.1. Trajectory Matrix Construction

Let $\mathcal{V} = \{v_1, \ldots, v_N\}$ denote the vocabulary of all unique procedural elements observed across the execution ensemble, including primary steps, substeps, and RNO actions. Here, N represents the total number of distinct elements that may be visited during procedure execution.

For each scenario i, a binary visitation vector $\mathbf{x}_i \in \{0, 1\}^N$ is defined such that

$$x_{ij} = \begin{cases} 1 & \text{if procedural element } v_j \text{ is visited in scenario } i \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Stacking the visitation vectors for all scenarios yields the trajectory matrix

$$\mathbf{X} \in \{0, 1\}^{n \times N}, \tag{2}$$

where rows correspond to execution scenarios and columns correspond to procedure steps. The resulting matrix represents the existence of a common procedural backbone shared across most scenarios, while variation is concentrated in the conditional and RNO regions of the procedure.

2.3. PRA Minimal Cut Set Analysis

To establish a connection between procedural structure and risk-informed analysis, a PRA-style representation was developed directly from the EOP-EPP-002 knowledge graph. Rather than beginning with a traditional system-level PRA model, the procedure itself was treated as the underlying logical structure. The branching conditions, RNO actions, and procedural transitions encoded within the knowledge graph were used to define a set of possible execution pathways through the procedure, as shown in Figure 4. Each conditional branch within the procedure was interpreted as a binary event that represented either the expected plant response or the failure of that response, the latter of which resulted in the execution of the corresponding RNO action. By traversing the procedural logic in this manner, a procedural event-tree structure was constructed that captures the possible outcomes of operator execution and plant response throughout the procedure. Quantification of this procedural model produced a set of minimal cut sets, each of which represents a unique combination of conditions and procedural outcomes that lead to a specific end state.

The resulting minimal cut sets were then projected back into the procedural-element space of the knowledge graph. For each cut set, a binary visitation vector was constructed over all procedural elements, including primary steps, substeps, and RNO actions. An entry of one indicates that the procedural element participates in the corresponding cut set, whereas an entry of zero indicates no involvement. Stacking these vectors using

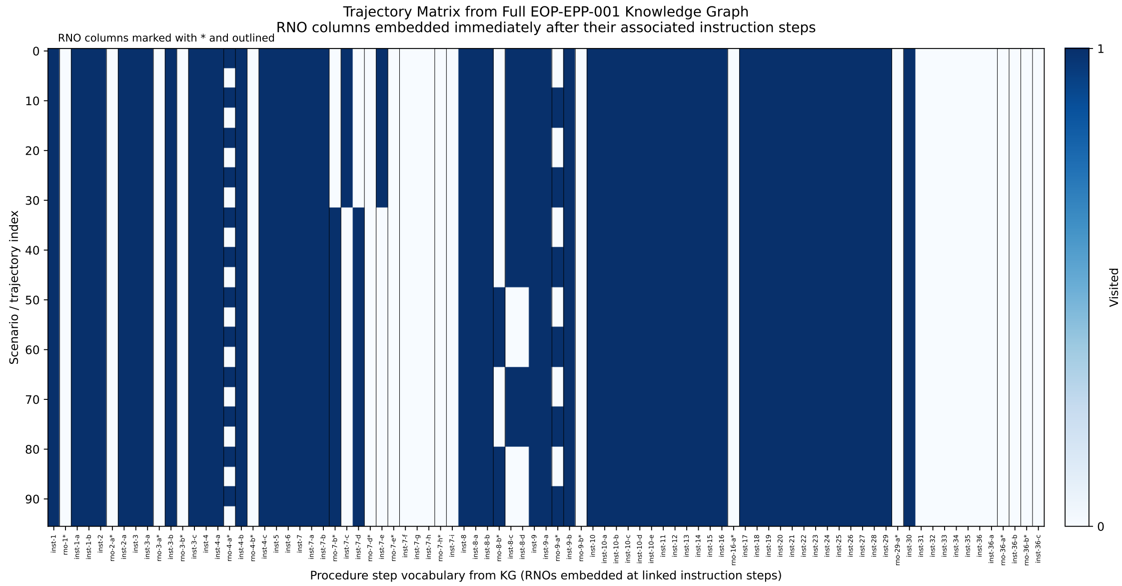

Figure 3: Trajectory matrix $\mathbf{X}$ where rows are scenarios; columns are procedure steps. Blue = visited (1), white = not visited (0). Vertical white bands correspond to conditional branches and RNO paths taken only in specific scenarios.

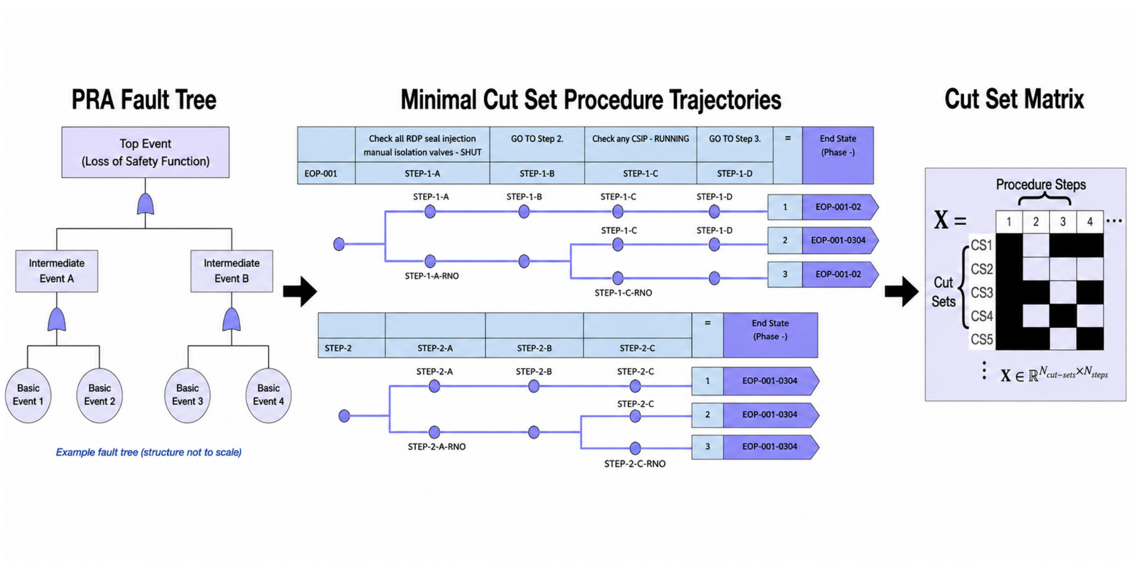

Figure 4: Workflow for constructing the PRA-informed cut set trajectory matrix. Minimal cut sets derived from the PRA fault tree are mapped to procedural execution paths, including nominal and RNO branches, and encoded into a binary matrix $\mathbf{X} \in \mathbb{R}^{N_{cut-sets} \times N_{steps}}$. Unlike the scenario-based formulation, this matrix embeds procedural execution within a risk-informed PRA framework.

a method similar to the one described in subsubsection 2.2.1 yielded the cut-set matrix shown in Figure 5, in which rows correspond to minimal cut sets and columns correspond to procedural elements.

This representation transforms the procedure from a static document into a risk-informed structure, which allows for procedural pathways, branching logic, and operator actions to be analyzed using techniques traditionally applied to PRA models. The resulting matrix therefore captures not only how the procedure may be executed, but also how combinations of procedural outcomes contribute to the overall risk structure embedded within the procedure.

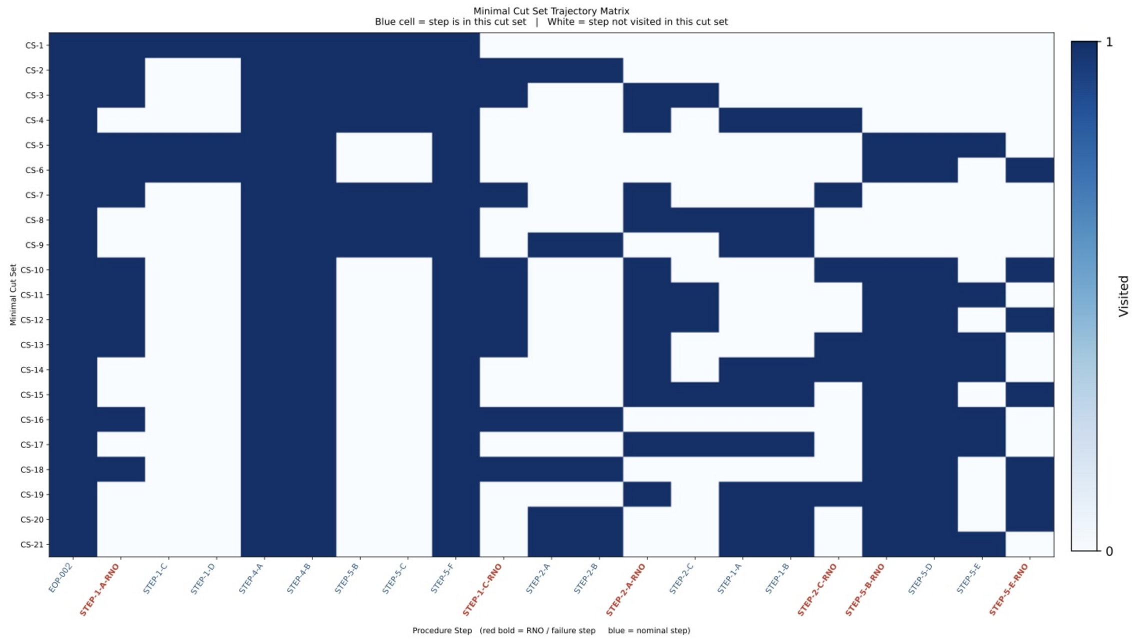

Figure 5: Cut-set Matrix $\mathbf{X}$ where rows are cut sets and columns are procedure steps. Trajectories evolve according to PRA fault trees.

The same SVD/LSA framework was then applied to the minimal cut-set matrix to identify latent failure "topics" corresponding to distinct procedural failure structures. This type of analysis enables synthetic scenario-driven trajectories and PRA-derived operational risk pathways to be compared, providing insight into how cut sets cluster according to common RNO activations, procedural branch structures, and operational phases.

2.4. Singular Value Decomposition and Latent Pattern Discovery

Many scientific and engineering disciplines generate large datasets consisting of numerous observations and measured variables. These datasets are commonly organized into a matrix in which rows represent individual observations and columns represent measured features [13]. Once the data are represented in matrix form, linear algebra techniques can be used to identify dominant structures, correlations, and recurring patterns that would otherwise remain hidden within the high-dimensional data [14].

A fundamental tool for this purpose is singular value decomposition (SVD). Given a data matrix $\mathbf{X}$, SVD factorizes the matrix into three components,

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T,$$

where $\mathbf{U}$ contains observation coordinates, $\mathbf{\Sigma}$ contains singular values representing the strength of each mode, and $\mathbf{V}^T$ contains feature loadings that describe how variables contribute to each mode. The decomposition identifies a set of orthogonal latent patterns that collectively explain the structure of the original dataset.

SVD forms the mathematical foundation of several widely used analysis methods. Principal component analysis (PCA) [15], for example, applies SVD to a centered data matrix to identify the directions of maximum variance. PCA is routinely used in fields such as fluid mechanics [16], structural dynamics, image processing, atmospheric science, and machine learning [17] to reduce dimensionality and reveal dominant physical mechanisms.

One of the most influential applications of SVD is proper orthogonal decomposition (POD) [18] in fluid dynamics, where experimental particle image velocimetry (PIV) or computational fluid dynamics (CFD) flow-field snapshots are assembled into a data matrix and decomposed to identify coherent structures that capture most of the flow energy. Rather than analyzing millions of individual measurements, a complex turbulent flow can often be represented by only a few dominant modes. Similar approaches are widely used in climate science to identify recurring atmospheric and oceanographic patterns, in structural health monitoring to extract dominant vibration modes and detect anomalies, and in image processing and machine learning to identify the principal features driving variation among samples. Across these diverse applications, the common principle is that a complex system is represented as a matrix, after which SVD reveals the underlying latent structures that govern its behavior.

The present work extends this concept to procedural trajectories. Rather than treating a procedure as a sequence of text instructions, the procedure is represented as a collection of operational states and decision points that may appear within minimal cut sets generated from the risk model or be based on scenarios. Each minimal cut set can be viewed as an observation of a potential failure trajectory through the procedure. Procedure steps constitute the variables describing that trajectory. A binary trajectory matrix is therefore constructed in which rows correspond to minimal cut sets and columns to procedure steps. An entry of one indicates that a particular step appears in a given cut set, whereas zero indicates its absence.

This representation is directly analogous to data matrices used in other scientific disciplines. In fluid mechanics, rows may represent flow snapshots and columns velocity measurements. In structural dynamics, rows may represent time samples and columns sensor responses. In the present application, rows represent cut sets and columns the procedure steps.

Once the procedure is represented in matrix form, SVD can be applied to identify dominant procedural modes in exactly the same way that it identifies dominant flow structures, vibration modes, or climate patterns. The resulting latent dimensions reveal recurring procedural failure structures that are not immediately apparent from the inspection of individual cut sets.

Latent semantic analysis is a specific application of SVD originally developed for text analysis. In traditional LSA, rows correspond to documents and columns to words. SVD is then used to identify latent topics that explain patterns of word usage across the document collection.

The procedural trajectory matrix developed in this work is mathematically analogous to a document-term matrix. The only difference is the interpretation of the matrix elements. Instead of documents and words, the rows correspond to minimal cut sets and the columns to procedure steps. Through the application of SVD, therefore, latent procedural patterns rather than latent semantic topics can be identified.

In this context, the latent dimensions may be interpreted as dominant procedural failure archetypes. Each latent pattern represents a characteristic combination of procedure steps that repeatedly appears across multiple cut sets. The procedure-step loadings identify which actions are most strongly associated with each failure archetype, while the cut-set coordinates reveal how individual cut sets relate to the underlying procedural structures. Consequently, LSA provides a principled framework for reducing a potentially large collection of minimal cut sets into a small number of dominant procedural failure modes, enabling the systematic identification of the procedure steps that drive overall procedural risk.

3. RESULTS

This section presents the SVD and LSA results obtained from the two trajectory-matrix formulations developed to represent procedural execution. The extracted latent patterns are interpreted in the context of the underlying procedure instructions and RNO actions, providing insight into how the written procedural logic governs the dominant modes of execution across the trajectory ensemble.

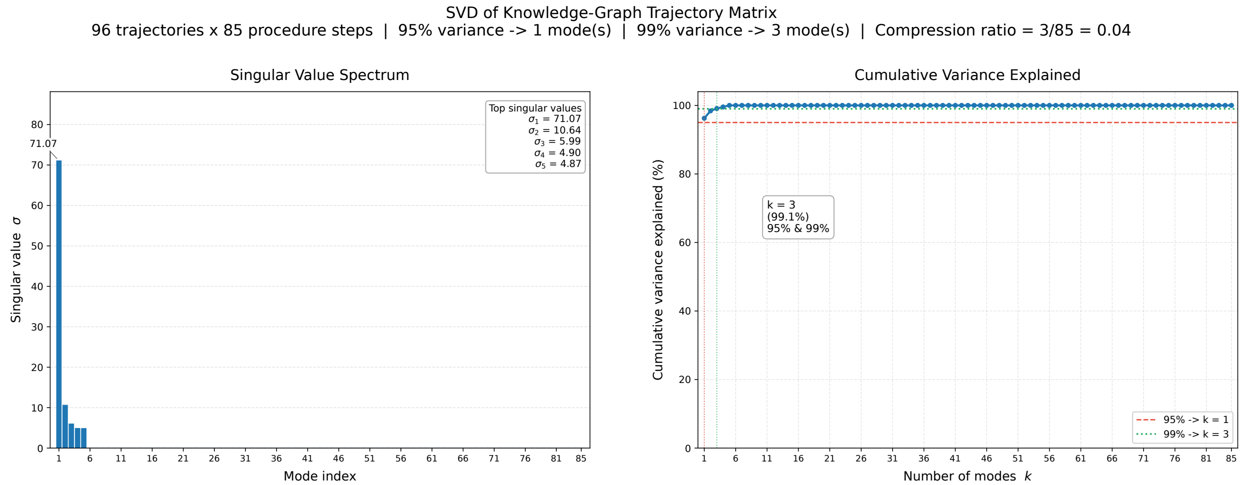

Figure 6: SVD of the knowledge-graph trajectory matrix. The singular value spectrum and cumulative variance demonstrate that the procedural execution space is highly low-dimensional, with a single mode capturing more than 95% of the variance and three modes capturing more than 99% of the variance, indicating substantial opportunities for reduced-order representation of procedural behavior.

3.1. Singular Value Decomposition Analysis

In this section, we discuss the analysis of singular values for both trajectory matrices generated through different procedure characterizations. The singular values, which form the diagonal entries of the Σ matrix, provide a measure of the variance captured by each mode and are used to assess the dimensionality and compressibility of the resulting procedural execution space.

3.1.1. Scenario-Based Trajectories

The knowledge-graph trajectory matrix was analyzed using SVD to quantify the dimensionality of the procedural execution space and identify dominant execution patterns. The SVD results indicate that the trajectory matrix is highly compressible, with only a small number of modes required to capture nearly all variability across the ensemble of generated trajectories. Specifically, the first mode alone captures more than 95% of the variance, while three modes capture more than 99% of the variance, as shown in Figure 6. This strong concentration of variance suggests that procedural execution based on scenarios is governed by a limited set of dominant structural patterns and can therefore be represented accurately using a low-dimensional ROM.

3.1.2. Cut-Set Procedural Trajectories

SVD (Figure 7) shows that the cut-set matrix is effectively low rank: only six nonzero singular modes are needed to reconstruct the 21×21 trajectory matrix. The first modes capture most of the structure, and the cumulative variance reaches both the 95% and 99% thresholds at k = 6, corresponding to a compression ratio of 6/21 = 0.29. The variance reflects the amount of trajectory behavior preserved in the reduced-order representation. This indicates that the cut-set behavior is dominated by a small number of coherent trajectory patterns; not all 21 procedure-step dimensions are needed for the behavior to happen.

The scenario-based trajectory matrix exhibits greater compressibility than the PRA-derived matrix (Figure 6) because the generated trajectories largely follow the same procedural backbone and differ only at a limited number of branching decisions. This results in highly correlated trajectory vectors and a strong concentration of variance within the leading singular modes. In contrast, the PRA-derived matrix (Figure 7) is constructed from distinct minimal cut sets that activate different combinations of procedure steps and RNO actions. The increased diversity of execution pathways distributes variance across a larger number of modes, resulting in a

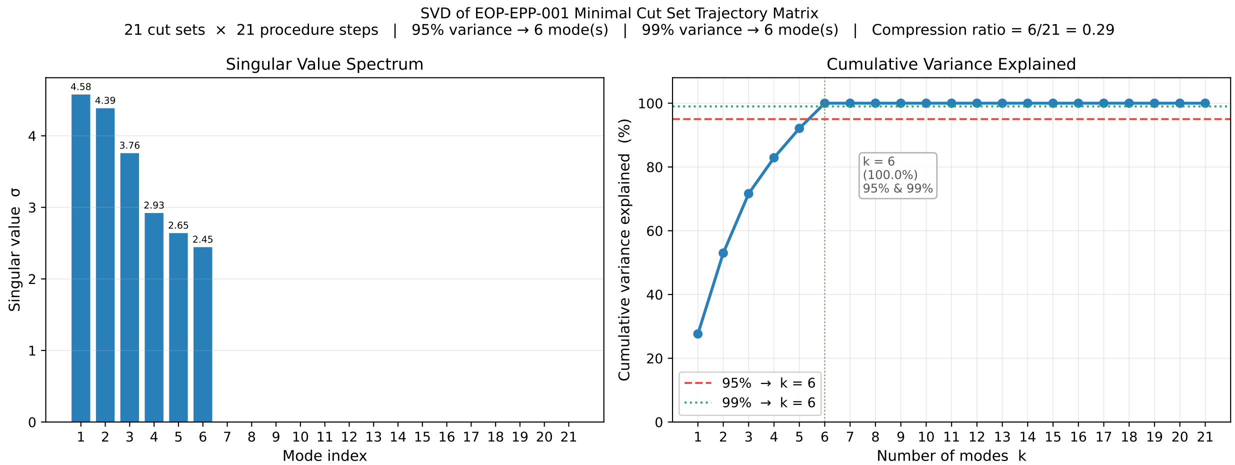

Figure 7: Singular value spectrum and cumulative variance for the PRA minimal cut-set trajectory matrix. Six modes capture 99% of the variance, indicating substantial low-dimensional structure in the cut-set-induced procedural pathways.

higher-dimensional representation and reduced compression relative to the scenario-based formulation.

3.2. Latent Semantic Analysis

The resulting reduced-order representations were then analyzed using LSA by projecting trajectories onto the dominant latent dimensions, enabling the identification of common execution patterns, clustering behavior, and the procedural steps that contributed most strongly to trajectory variability.

3.2.1. Scenario-Based Trajectories

Figure 8 presents the LSA of the knowledge-graph trajectory matrix and provides insight into the dominant procedural structures governing trajectory variability. The left panel shows the latent pattern-step loading matrix, where each row represents a latent procedural pattern and each column corresponds to a procedure step or RNO action. Red and blue values indicate positive and negative contributions to a latent pattern, respectively. Several observations are apparent. First, much of the procedural backbone exhibits near-zero loading, indicating that many instructions occur consistently across nearly all trajectories and therefore contribute little to differentiating one trajectory from another. In contrast, a relatively small subset of branching instructions and RNO actions exhibit large positive or negative loadings, demonstrating that procedural variability is concentrated within a limited number of decision points rather than being distributed uniformly throughout the procedure. The emergence of distinct loading structures within each latent pattern suggests that different groups of instructions act together as coherent procedural themes that govern how trajectories diverge.

The center panel projects the trajectories into the reduced latent space defined by the dominant latent patterns. Rather than forming a continuous cloud, the trajectories organize into several distinct clusters, indicating that execution pathways naturally group into a limited number of characteristic procedural behaviors. Trajectories located within the same cluster share similar combinations of procedure-step activations and RNO actions, while trajectories in different clusters correspond to fundamentally different execution paths through the procedure. The coloring by number of RNO actions reveals a strong relationship between latent-space location and procedural complexity, with trajectories that contain similar numbers of RNO activations tending to occupy similar regions. This result suggests that the latent dimensions capture meaningful procedural characteristics associated with branching behavior and operator responses and are not merely mathematical constructs.

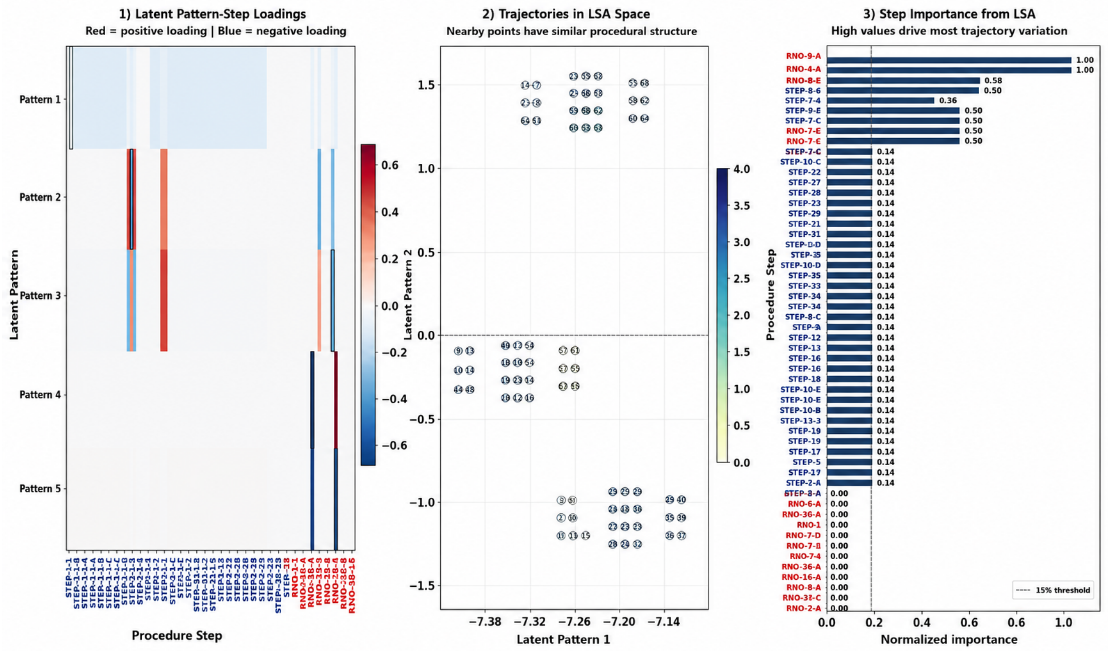

Figure 8: LSA of knowledge-graph trajectories. The latent pattern loadings, trajectory clustering in reduced-dimensional space, and step-importance rankings reveal the dominant procedural structures governing trajectory variability and the procedure steps and RNO actions that contribute most strongly to differences among execution pathways.

The right panel quantifies the relative importance of individual instructions and RNO actions within the latent representation. Only the thirty-most influential procedure steps together with all RNO actions are shown to improve readability, while lower-importance instructions are omitted. The ranking demonstrates that only a small subset of instructions contributes substantially to trajectory variability, whereas many instructions have negligible importance because they appear in nearly every execution pathway. In particular, several RNO actions and branching-related instructions dominate the ranking, indicating that the primary source of variability arises from alternative procedural responses rather than from the nominal execution sequence itself. This observation is consistent with the clustering observed in the latent space and reinforces the conclusion that procedural behavior can be effectively characterized by a relatively small set of critical decision points despite the large number of instructions contained within the full procedure.

3.2.2. Cut-Set Procedural Trajectories

Figure 9 presents the LSA decomposition of the minimal cut-set trajectory matrix. The left panel shows the latent pattern-step loading matrix. Each row corresponds to a latent failure pattern and each column to a procedure step. Positive loadings (red) indicate procedure steps that contribute positively to a latent pattern, whereas negative loadings (blue) indicate steps that oppose that pattern. Large-magnitude loadings identify the procedure steps that are most strongly associated with a particular latent failure structure. Several RNO-designated steps exhibit dominant loadings, indicating that these actions play a crucial role in distinguishing procedural execution modes.

The center panel projects the minimal cut sets into the space defined by the first two latent patterns identified by SVD. Each point represents a unique cut set and is colored according to the number of RNO steps present in the trajectory. Cut sets that appear near one another possess similar procedural structures, whereas distant points correspond to substantially different combinations of procedure-step failures. The clustering

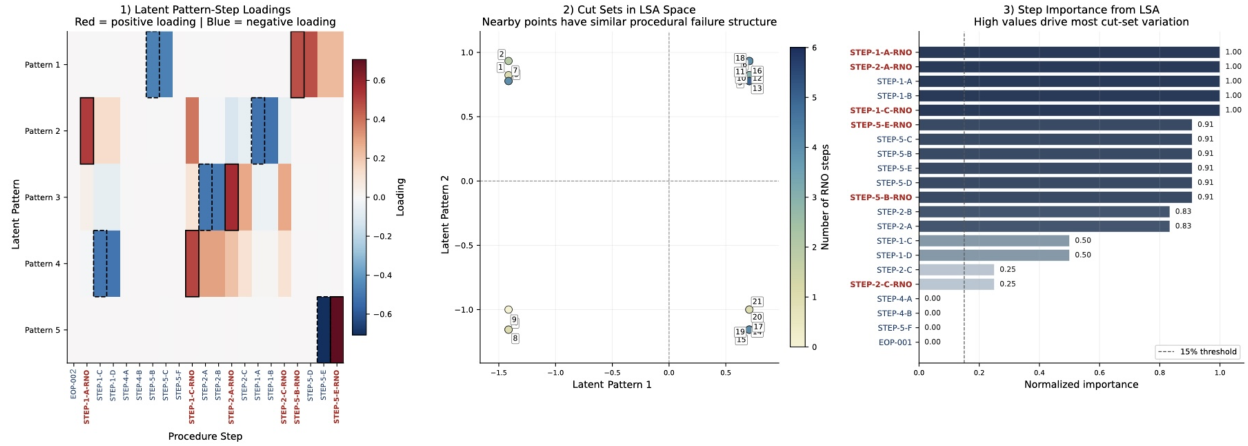

Figure 9: LSA of minimal cut-set procedural trajectories. (1) Latent pattern–step loading matrix showing the contribution of each procedure step to the retained latent failure patterns. Red values indicate positive loadings and blue values negative loadings. (2) Projection of minimal cut sets into the space defined by the first two latent failure patterns. Each point represents a cut set and is colored according to the number of RNO-designated steps present in the trajectory. Nearby points correspond to cut sets with similar procedural path structures. (3) Normalized procedure-step importance derived from the retained latent patterns. Higher values indicate steps that contribute most strongly to variation among cut sets and therefore exert the greatest influence on the underlying procedural failure structure. Five latent patterns explain approximately 92.1% of total variance in the trajectory matrix.

observed in latent space suggests that multiple cut sets share common underlying procedural characteristics despite differing in their exact combinations of steps. The right panel quantifies the relative importance of each procedure step. Importance was computed from the cumulative contribution of the retained latent patterns and normalized to the most influential step. Steps with larger values contribute more strongly to overall cut-set variability and therefore exert greater influence on the dominant procedural failure structures identified by the LSA. Several RNO-related steps exhibit the highest importance values, indicating that variations involving these actions account for a substantial portion of the observed diversity among cut sets. Conversely, steps with low importance contribute little to the differentiation of cut-set trajectories and have comparatively limited influence on the latent structure.

Collectively, the LSA results demonstrate that the procedural failure space is highly structured and low dimensional. Although twenty-one minimal cut sets were analyzed, the majority of variation can be explained by only five latent patterns. This finding suggests that procedural risk is concentrated within a relatively small set of recurring failure mechanisms and that targeted improvements to the most influential steps may provide the greatest reduction in overall procedural vulnerability.

3.3. Discussion

The LSA results for the scenario-based matrix can be directly connected to the structure and intent of EOP-EPP-001. Instructions associated with initial plant stabilization, such as reactor trip verification, turbine trip verification, Reactor Coolant System (RCS) isolation, and the establishment of auxiliary feedwater, occur in nearly every trajectory and therefore contribute little to the latent dimensions despite their operational importance. These actions form the common procedural backbone of the loss-of-AC-power response, which explains why they exhibit relatively low importance in the latent representation.

This observation is consistent with the structure of EOP-EPP-001, the primary objective of which is to respond to loss of all AC power to the plants safety buses. In contrast, the dominant latent patterns are driven primarily by branching decisions and RNO actions associated with EDG availability, restoration of AC power, steam generator fault or rupture conditions, and long-term station blackout recovery actions. These portions of the procedure contain the majority of the conditional logic and therefore determine how execution pathways diverge under different plant conditions. The clustering observed in the latent space reflects these alternative recovery strategies, separating trajectories that successfully restore power early from those that proceed through extended battery conservation, equipment alignment, steam generator management, and long-term cooling actions. Consequently, the latent dimensions capture not only the mathematical structure of the trajectory space but also the underlying operational objectives of the procedure, including restoration of electrical power, maintenance of decay heat removal, containment protection, and preservation of long-term cooling capability.

The latent patterns identified from the minimal cut-set trajectory matrix can be directly related to the recovery logic embedded within EOP-EPP-002. Unlike the scenario-based analysis, where variability arises from different plant conditions and operator responses, the PRA-derived trajectories are generated from combinations of equipment failures represented by minimal cut sets. Consequently, the dominant latent patterns correspond to recovery actions that become necessary when particular systems or components are unavailable. The step-importance ranking shows that the highest-contributing instructions are concentrated around branching and RNO actions, indicating that procedural variability is driven primarily by contingency actions rather than the nominal recovery sequence.

This observation is consistent with the structure of EOP-EPP-002, the primary objective of which is to restore normal plant functions following recovery of AC emergency power. The procedure contains a common recovery backbone involving restoration of cooling, charging, ventilation, and steam generator control functions that appears in most trajectories and therefore contributes little to the latent dimensions. In contrast, the dominant latent features are associated with alternate recovery actions, equipment unavailability, and transitions to EPP-003 when shutdown margin, pressurizer level, or subcooling requirements cannot be maintained. As a result, the latent-space clusters can be interpreted as groups of cut sets that challenge similar safety functions and activate similar recovery strategies. The fact that only a small subset of instructions dominates the latent representation suggests that, despite the large number of possible failure combinations, procedural recovery behavior is governed by a relatively small number of recurring recovery themes.

4. CONCLUSION AND OUTLOOK

This work presented a data-driven reduced-order modeling framework for representing nuclear control room procedural execution using knowledge graph representations of emergency operating procedures (EOPs). Two complementary formulations were explored: a scenario-based approach that generated execution trajectories from procedural branching logic under varying plant conditions, and a probabilistic risk assessment (PRA)–informed approach that represented risk-significant procedural pathways through minimal cut sets. Together, these approaches demonstrate how complex procedural logic can be transformed into compact mathematical representations suitable for human reliability analysis (HRA), PRA, and future digital control room applications. Using several EOPs converted into structured knowledge graphs, execution trajectories were generated and encoded as binary visitation vectors to form trajectory matrices for analysis. The resulting reduced-order representations showed that a small number of dominant modes captured the majority of procedural variability, indicating that procedural execution evolves on a low-dimensional manifold despite the complexity of branching logic, procedural transitions, and response-not-obtained actions. These findings suggest that structured procedural knowledge can be effectively compressed in a way that preserves the dominant operational behaviors embedded within the procedures.

Several limitations should be noted. The trajectory ensembles were generated synthetically from procedural logic encoded within the knowledge graphs rather than from observed operator performance, and the current representation captures only procedural element visitation without temporal information. In addition, the present study focused on a limited subset of EOPs. Future work will focus on incorporating simulator and operator-performance data, extending the methodology across broader procedural ecosystems, and investigating applications in procedural compliance monitoring, operator-state characterization, HRA, and digital twin technologies for advanced nuclear control rooms.

5. ACKNOWLEDGEMENTS

We gratefully acknowledge funding from the U.S. Department of Energy's Light Water Reactor Sustainability Program and particularly thank Plant Optimization Pathway Lead, Ahmad Al Rashdan for championing this research.

References

[1] Jinkyun Park, Wondea Jung, Jaejoo Ha, and Changkue Park. The step complexity measure for emergency operating procedures: measure verification. *Reliability Engineering & System Safety*, 77(1):45–59, 2002.

[2] Qin Gao, Yang Wang, Fei Song, Zhizhong Li, and Xiaolu Dong. Mental workload measurement for emergency operating procedures in digital nuclear power plants. *Ergonomics*, 56(7):1070–1085, 2013.

[3] Thomas A Ulrich, Jisuk Kim, Dylan Logan Jurski, Temitayo Olayemi Olowu, Anna Catherine Hall, Tyler L Westover, Jeremy David Mohon, Roger Lew, Olugbenga Gideon, Zeth Dubois, et al. Advancements in development and testing of thermal power dispatch simulators. Technical report, Idaho National Laboratory (INL), Idaho Falls, ID (United States), 2023.

[4] Irene Celino, Valentina Anita Carriero, Antonia Azzini, Ilaria Baroni, and Mario Scrocca. Procedural knowledge management in industry 5.0: Challenges and opportunities for knowledge graphs. *Journal of Web Semantics*, page 100850, 2024.

[5] Fei-Hui Hwang and Sheue-Ling Hwang. Design and evaluation of computerized operating procedures in nuclear power plants. *Ergonomics*, 46(1-3):271–284, 2003.

[6] Kang-Hung Liu and Sheue-Ling Hwang. Human performance evaluation: The procedures of ultimate response guideline for nuclear power plants. *Nuclear Engineering and Design*, 273:234–240, 2014.

[7] Meenu Sethu, Bhavya Kotla, Darrell Russell, Mahboubeh Madadi, Nesar Ahmed Titu, Jamie Baalis Coble, Ronald L Boring, Klaus Blache, Vivek Agarwal, Vaibhav Yadav, et al. Application of artificial intelligence in detection and mitigation of human factor errors in nuclear power plants: a review. *Nuclear Technology*, 209(3):276–294, 2023.

[8] Ronald Boring, Roger Lew, Thomas Ulrich, Jooyoung Park, Jisuk Kim, and Olugbenga Gideon. Rancorhunter: Using a simulator engine for realistic human performance modeling of nuclear power operations. Technical report, Idaho National Laboratory (INL), Idaho Falls, ID (United States), 2025.

[9] Thomas A Ulrich. *The development and evaluation of attention and situation awareness measures in nuclear process control using the rancor microworld environment*. University of Idaho, 2017.

[10] Niharika Karnik, Jisuk Kim, Brian Wilcken, Roger Lew, Jooyoung Park, Ronald Boring, and Thomas Ulrich. Transforming nuclear power plant procedures from unstructured paper to structured knowledge graphs using large language models. Manuscript under review, 2026.

[11] Jisuk Kim, Niharika Karnik, Ronald Boring, Brian Wilcken, Roger Lew, and Thomas Ulrich. Llm-based structured reconstruction framework for nuclear power plant operating procedures, with comparative evaluation and human-in-the-loop refinement. Manuscript under review, 2026.

[12] Thomas Ulrich, Jisuk Kim, Niharika Karnik, Brian Wilcken, Ronald Boring, Jooyoung Park, and Roger Lew. Development and testing of a preliminary ai-based procedure conversion method. Technical report, Idaho National Laboratory (INL), Idaho Falls, ID (United States), 2026.

[13] Steven L Brunton and J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2022.

[14] Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.

[15] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[16] Isabel Scherl, Benjamin Strom, Jessica K Shang, Owen Williams, Brian L Polagye, and Steven L Brunton. Robust principal component analysis for modal decomposition of corrupt fluid flows. *Physical Review Fluids*, 5(5):054401, 2020.

[17] Ian T Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.

[18] Gal Berkooz, Philip Holmes, and John L Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual review of fluid mechanics*, 25(1):539–575, 1993.