Vision-Based Retrieval of Semantic Workflows in Process-Oriented Case-Based Reasoning*

Maxim Hotz[1,2], Lukas Malburg[1,2], Kokulan Thanabalan[1,2], and Ralph Bergmann[1,2]

[1] Artificial Intelligence and Intelligent Information Systems, Trier University, Germany, https://www.wi2.uni-trier.de
[2] German Research Center for Artificial Intelligence (DFKI), Germany
{s4mahotz,lukas.malburg,thanabalan,bergmann}@uni-trier.de
{maxim.hotz,lukas.malburg,kokulan.thanabalan,ralph.bergmann}@dfki.de

Abstract *Process-Oriented Case-Based Reasoning* (*POCBR*) relies on retrieving similar semantic workflow graphs, but accurate retrieval typically requires computationally expensive graph matching. To improve scalability, existing systems often employ two-stage retrieval strategies such as *MAC/FAC*, where a fast but less accurate prefilter reduces the search space before detailed matching. Recent approaches use *Graph Neural Networks* (*GNNs*) for this prefiltering step, but they still involve trade-offs between approximation quality, efficiency, and interpretability. This paper investigates whether semantic workflows can instead be retrieved by visual comparison. To this end, we analyze existing graph visualization strategies regarding their suitability for vision-based retrieval and propose a multi-step pipeline that transforms semantic workflows into deterministic visual representations. These representations are then used in combination with modern vision models for similarity-based retrieval. Experiments in two *POCBR* domains indicate that the proposed approach yields substantial speedups while producing retrieval quality that is generally below but still competitive with established methods. The results demonstrate that vision-based retrieval is a promising alternative for use as a *MAC-stage* filter in *POCBR*.

Keywords: Vision-Based Retrieval, Vision Transformer, Semantic Workflows, Similarity Assessment, Process-Oriented Case-Based Reasoning

1 Introduction

The methodology of *Case-Based Reasoning* (*CBR*) [1] is based on the principle that similar problems have similar solutions [34]. Thus, a suitable and efficient similarity assessment is crucial in modern *CBR* applications [12]. However, the use of complex case representations, such as in *Process-Oriented Case-Based*

* The final authenticated publication is available online at https://doi.org/10.1007/978-3-032-33865-5_34

Reasoning (*POCBR*) [7], where cases represent procedural experiential knowledge as semantic workflow graphs, introduces new challenges for similarity-based retrieval. Unlike traditional attribute-value case representations used in *CBR*, semantic workflow graphs require similarity measures that account for both structural topologies and semantic descriptions of workflows [7]. This increased complexity results in a graph matching/mapping problem that must be solved to accurately assess similarity. This process is significantly more challenging than the simple comparison of attribute values in classical *CBR*. For several years, we have been investigating this issue by proposing novel approaches for similarity-based retrieval in *POCBR* (e. g., [7, 20, 21, 25, 29, 42]). These approaches range from heuristic algorithms like A* [7, 42] to compute node and edge mappings between workflow graphs to GPU-accelerated graph matching [21,29] and graph embedding approaches [20, 25] that are used as a prefiltering step in a MAC/-FAC retrieval [15]. Nonetheless, some of the proposed approaches lack scalability, which poses challenges for large case bases or complex workflows. This scalability barrier is particularly problematic in settings where fast retrieval is essential, not only for overall system performance but also for user satisfaction and the acceptance of a *CBR* application.

In recent years, a potential alternative is the use of methods based on *Large Language Model*s (*LLM*s), which have also gained increasing attention in the *CBR* community (e. g., [2, 6, 27, 31, 39, 40]). Hereby, approaches can be used to assess similarity that mimic how human experts assess similarity between objects: by comparison of visual cues such as patterns, shapes, or colors, such as product designers do in product development with technical drawings [26]. Unlike many existing approaches (e. g., [20,25]), which often operate as black boxes and offer limited explainability of retrieval results, a visual representation is a first step toward making the similarity assessment process transparent for users by visually exposing individual graph elements (i. e., nodes).

In addition, modern vision-based machine learning models, such as *Convolutional Neural Network*s (*CNN*s) [33] or *Vision Transformer*s (*ViT*s) [14], help to remedy the scalability issues. Once trained, these models provide constant inference time during retrieval, regardless of workflow complexity. Furthermore, because they could learn to extract domain-agnostic visual features rather than relying on domain-specific data, these models have the potential to generalize to entirely unseen domains. However, existing applications focus on domains with standardized visual encodings (e. g., finance [37]) and lack the ability to integrate workflow semantics. Hence, the question remains of how both the structure and semantics of workflow graphs in *POCBR* can be jointly encoded. The contribution of this paper is twofold: 1) we discuss requirements for vision-based retrieval in *POCBR* and compare current state-of-the-art approaches based on these requirements, and 2) we propose two visual encoding methods that are combined with generated node embeddings for vision-based retrieval in *POCBR*. The vision-based retrieval approach is evaluated on two datasets containing different kinds of semantic workflow graphs.

The paper is structured as follows: Section 2 introduces relevant foundations and provides an overview of related work. The set of requirements is presented in Section 3, followed by the proposed vision-based retrieval approach. Section 4 discusses the experimental evaluation of the proposed approach, and Section 5 concludes the paper and outlines future work.

2 Foundations and Related Work

The foundations for this paper comprise the *NEST* graph representation for semantic workflow graphs in *POCBR* and the associated similarity assessment by Bergmann and Gil [7]. Furthermore, the general properties of *Vision Transformer*s (*ViT*s) are fundamental to this work. Hence, they are introduced in the following along with related approaches for visualizing graph-structured data.

2.1 Similarity Assessment in *POCBR*

Cases in *POCBR* consist of procedural experiential knowledge [7] in the form of semantic workflow graphs. In this work, we use the *NEST* graph formalism [7] to encode these graphs. A NEST graph represents a workflow as a semantic workflow graph, which is a directed, typed, and semantically enriched graph. A semantic workflow graph W is hereby defined by the quadruple $W = (N, E, S, T)$ where N denotes a set of nodes and $E \subseteq N \times N$ is the set of edges between N. In addition, every node and every edge is strictly typed with $T : N \cup E \longrightarrow \Omega$ and also assigned a semantic description $S : N \cup E \longrightarrow \mathcal{E}$ out of a metadata language. Figure 1 depicts an exemplary NEST graph representing a sandwich recipe from the cooking domain. To determine similarity between NEST graphs, the knowledge-intensive similarity measure for semantic workflow graphs [7] is used in this context. This measure evaluates graph similarity by constructing

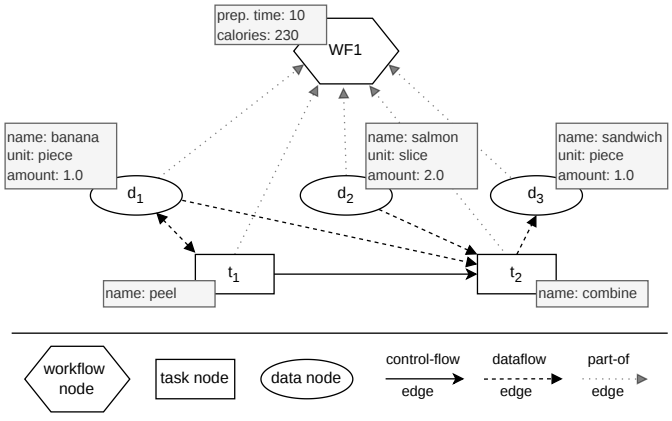

Fig. 1. Exemplary NEST Graph Representing a Cooking Recipe.

a best-possible mapping between the nodes and edges of two graphs under the condition that only nodes and edges of the same type can be mapped. For every mapped pair, some local similarity measure (e. g., taxonomic similarity or Levenshtein distance [5, 7]) is applied to the semantic descriptions of the mapped graph items. These local similarity scores are finally aggregated into an overall similarity value following the local-global principle [7].

2.2 Vision Transformers

For many years, deep learning in computer vision was dominated by *CNN*s [27]. More recently, attention-based architectures have become increasingly important, with *ViT*s emerging as a prominent alternative to *CNN*s-based models. *ViT*s transfer the transformer architecture, originally developed for natural language processing, to image analysis. Instead of processing an image with convolution operations, a *ViT* divides the image into fixed-size patches. These patches are transformed into embedding vectors and enriched with positional embeddings to preserve spatial information. The resulting sequence is then processed by transformer encoder layers based on self-attention [14]. In contrast to *CNN*s that mainly focus on local spatial neighborhoods [27], *ViT*s can model relationships between distant image regions directly through self-attention. This makes them well suited for learning global contextual information in images [14].

2.3 Related Work

Related approaches in the field of graph visualization form a taxonomy [4] that can be broadly categorized into explicit, implicit/abstract, and hybrid visualizations. Explicit methods, such as node-link diagrams and adjacency matrices, directly depict graph topology. In this context, Goh et al. [16] demonstrate with *Chemception* that *CNN*s can learn chemical properties directly from 2D molecular drawings that resemble node-link drawings. Lyu et al. [28] present matrix-based *Graph-Preserving Grid Layouts* that project graph nodes onto a 2D grid while integrating node features. Hybrid approaches like *NodeTrix* [18] or *MatLink* [19] combine matrices and node-link diagrams by rendering dense subgraphs as matrices and connecting them with edges while prioritizing readability for humans. Conversely, implicit and abstract approaches apply algorithmic transformations to represent graphs. In this context, Bartz et al. [3] propose a space-filling representation for argument graphs in *CBR*. Tixier et al. [38] represent graphs as 2D density histograms of node embeddings for *CNN* classification. Similarly, Sharma et al. [36] introduce *DeepInsight* to convert non-image data into *CNN*-compatible structured images, and Cakmak et al. [10] present *dg2pix* that stacks embedding vectors as pixel rows to visualize temporal graph changes.

While these approaches demonstrate the general compatibility of visual graph representations and vision models, a significant research gap remains regarding the complex semantic workflows applied in *POCBR*. Classic explicit methods (node-link, matrices) represent structure well and remain intuitive for humans, yet they cannot incorporate multi-type semantic annotations without clutter or

simplification. Spatialization and projection-based approaches capture semantic similarity effectively but typically fail to preserve explicit graph structure and are often stochastic (e. g., t-SNE, UMAP), which is problematic for reproducible vision-based ML pipelines. Hierarchical and space-filling methods usually require true hierarchies and therefore do not fit sequential NEST graphs with branches and loops, while creating artificial hierarchies risks breaking directed flow and losing information. Finally, many methods cannot produce compact, fixed-size outputs, and embedding-based visualizations often either ignore semantics or aggregate information too strongly, making them potentially too lossy for similarity regression. Consequently, there is currently no unified representation for semantic graphs that would allow the complete encoding of structure and semantics required for vision-based similarity regression.

3 Vision-Based Retrieval Approach

In this section, we present our novel vision-based retrieval approach. The approach is based on the idea of transforming semantic workflow graphs into visual representations that can be processed by state-of-the-art vision-based machine learning models. For this reason, we present in the following a set of requirements that the visual representation must fulfill to be suitable for encoding workflow graphs for similarity-based retrieval. Afterwards, we present a conceptual overview of the components of our approach and describe the individual steps in more detail.

3.1 Requirements Analysis

To guide the development and evaluation of a novel visual representation for semantic graphs, we derive the following functional (FR) and non-functional (NFR) requirements based on a literature review and the inherent properties of NEST graphs.

FR 1 – Representation of Graph Structure: The representation must encode the directed topology of a NEST graph. This includes nodes, edges, and control-flow patterns (sequences, parallel branches, loops), which serve as a structural foundation.
FR 2 – Integration of Semantic Annotations: The encoding mechanism must allow the integration of semantic descriptions of workflow nodes into the structural representation. The integration of semantic descriptions for nodes is essential, as current A*-based approaches rely on the local-global principle, where the overall similarity is typically aggregated from node-level semantic similarities [7]. Encoding edge semantics is optional, as they contribute less to overall similarity and are often mapped implicitly.
FR 3 – Preservation of Node and Edge Types: The representation must visually distinguish between different node and edge types (e. g., via shape, color, or texture). This is necessary to accommodate the strict type-preserving mapping constraints of A* [7].

FR 4 – Multi-Type Attribute Encoding: The visual representation must be capable of encoding different data types found in composite semantic descriptions (e. g., numerical, ordinal, nominal). It should also be flexible enough to represent semantic information at different levels of abstraction, supporting both the direct encoding of individual attributes and the encoding of aggregated node-level similarity scores.
FR 5 – Compatibility with Input Requirements of Vision Models: The output must adhere to the constraints of vision models, such as producing fixed-resolution tensors for *CNN*s. For *ViT*s, it should ideally preserve semantically connected components when the image is divided into patch sequences [14].
FR 6 – Flexible Parametrization: The encoding mechanism must be configurable for diverse application domains. This ideally also includes toggling between structure-only and semantics-inclusive encodings, as well as adjusting the granularity of the semantic representation.

NFR 1 – Deterministic Encoding: The encoding mechanism must produce deterministic and reproducible visual representations, such that encoding the same NEST graph multiple times yields identical visual outputs.
NFR 2 – Generalizability Across Domains and Case Representations: The visual encoding approach should generalize across different workflow domains (e. g., cooking recipes [32], cyber-physical systems [30]) and thereby also accommodate for the different case representations linked to the respective domains (e. g., flat cases in classical *CBR* or time series in *Temporal Case-Based Reasoning* (*TCBR*)).

Table 1 summarizes the assessment of current approaches and the categories presented in related work regarding the derived requirements. The results reveal a critical trade-off that is consistent across most investigated approaches: While explicit layouts attempt to preserve the human-readable topology of a graph, they struggle to comply with the strict input requirements of modern vision models and also lack the ability to encode complex semantic descriptions adequately. Conversely, abstract and implicit approaches generally yield better semantic-encoding capabilities, but their abstract properties diminish interpretability. Ultimately, we identify embedding-based visualizations as the best

Table 1. Comparison of Visualization Approaches Against the Requirements.

Req.	Explicit		Hybrid	Implicit / Abstract		
	Node-Link *Std. Layout*	**Matrix-Based** *Adj. Matrix*	**NodeTrix / MatLink**	**Space-Filling / Hierarchical**	**Spatialization** *t-SNE Layout*	**Embedding-Based**
FR1	✓	✓	✓	✗	✓	✓
FR2	(✓)	(✓)	(✓)	✓	(✓)	✓
FR3	✓	(✓)	✓	✓	✓	✓
FR4	✗	✗	✗	✓	✗	✓
FR5	✗	(✓)	✗	✓	✗	✓
FR6	✓	✓	✓	(✓)	(✓)	✓
NFR1	✗	(✓)	✗	✓	✗	✓
NFR2	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)

✓ = fulfilled, (✓) = partially fulfilled, ✗ = not fulfilled.

candidate for this work, as they are the only approach that allows the simultaneous encoding of complex semantic descriptions and graph topology while adhering to the strict input requirements of modern vision models.

3.2 Architectural Overview

Based on the presented requirements, we develop a novel approach for vision-based retrieval of semantic workflow graphs. Figure 2 illustrates the components of our architecture. The first stage addresses the challenge of converting both

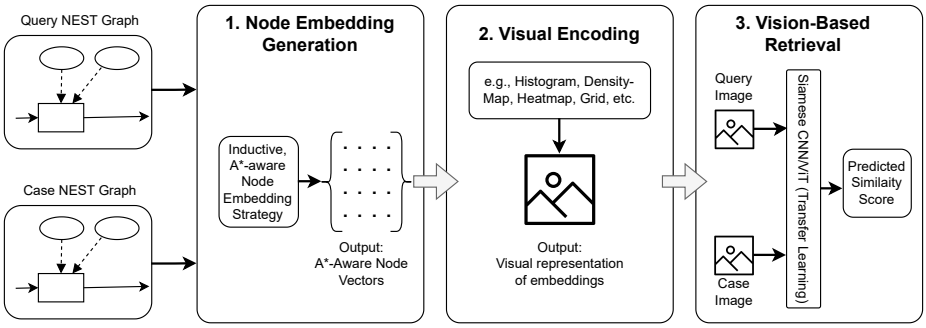

Fig. 2. Architectural Overview for Vision-Based Retrieval.

query and case NEST graphs (input phase 1) into a set of embedding vectors (output phase 1) while fulfilling the requirements stated above. It is the goal to employ a specialized, inductive embedding technique that ideally considers node-level A* ground truth labels from the case base during training the embedding function. Based on the resulting vector representation (input phase 2), the second stage consists of an algorithmic process that transforms the high-dimensional vectors from the first stage into a 2D visualization (output phase 2). Finally, the third stage addresses the challenge of learning to approximate graph-level A* similarity scores by employing a Siamese [9] similarity regression architecture. At this stage, the network is trained on the visual representations of graphs (input phase 3) contained in the case base with the ultimate goal of performing online retrieval on unseen graph visualizations. Together, these steps represent the full conceptual pipeline for the vision-based retrieval process.

3.3 Node Embedding Generation

Semantic bag-of-feature approaches like *StarSpace* [25, 41] are transductive and suffer from context pollution when attempting to integrate structural information, while structural embedding approaches such as *Node2Vec* [17] are oblivious of semantics. Therefore, a hybrid, inductive, embedding strategy is used that can handle the composite nature of A* similarities. *Graph Neural Network*s (*GNN*s)

are capable of generating node embeddings by iteratively combining a node's own features with aggregated features from its neighbors [35]. Figure 3 depicts the *GNN* architecture used in our approach.

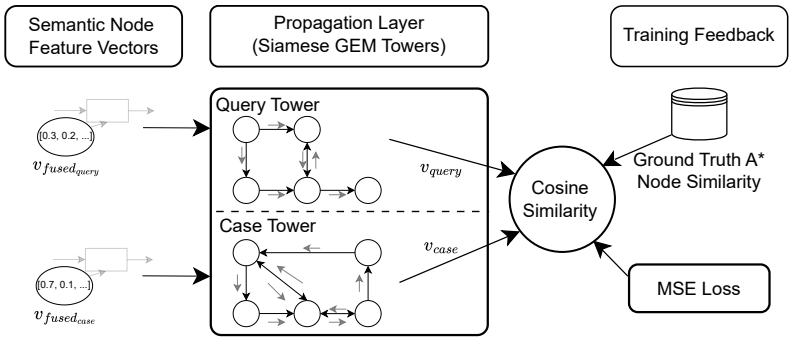

Fig. 3. Siamese Graph Embedding Model Architecture. (Source: Based on [23])

3.4 Visual Encoding

The second component of the proposed architecture addresses the transformation of the high-dimensional node embeddings into a fixed-size image representation. Approaches such as *dg2pix* [10] demonstrate that embedding vectors can be encoded as "pixel-bars" that are essentially sequences of pixels where the color intensity represents the value of a vector dimension. This concept is adopted in our approach and extended by proposing two different layout strategies: a *Grid* layout and a *Linear* layout (see Figure 4), which each complement different properties of state-of-the-art vision models. Based on the task and data nodes in the graph, the created node embeddings are transformed into grayscale patterns while the padded zones are rendered as black tiles (0). In addition to this, we transform the *NEST* graphs directly into node-link diagrams, rendering nodes as geometric shapes and edges as directed connecting lines. We use these diagrams as a structure-focused baseline to evaluate whether vision models can directly parse explicit human-readable graph topologies. To improve their machine-readability, the node-link drawings are optimized in terms of layout and color palette before being used as input for vision-based retrieval. In detail, this is achieved through a compacting layout algorithm that minimizes white space and edge crossings and by employing a high-contrast color scheme to differentiate between item types.

3.5 Vision-Based Retrieval

Based on the generated node embeddings and their visual encoding into the proposed *Grid* or the *Linear* layout, a Siamese neural network is used to process

the images for a query and the cases from the case base. Figure 5 depicts this process where both images in a grid or linear layout or as node-link diagrams are input to a Siamese 2D *CNN*/*ViT*. The neural network transforms the input into vectors with which the similarity is assessed based on cosine similarity. The Siamese model is trained based on the ground truth A* graph similarities and optimized based on an *Mean Squared Error* (*MSE*) loss.

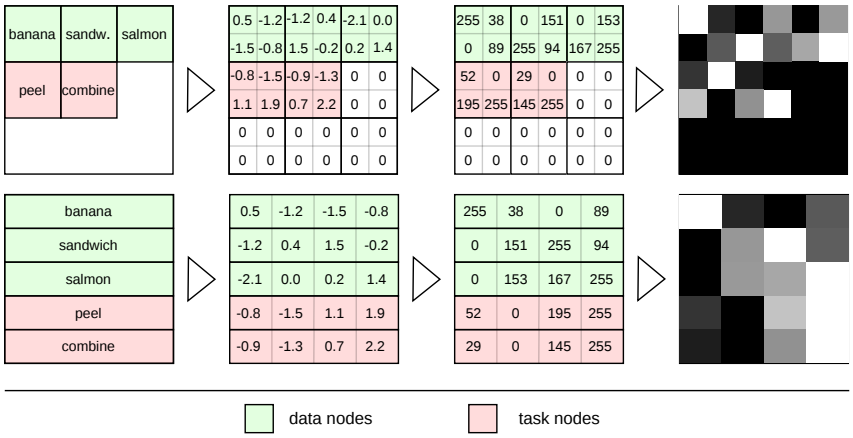

Fig. 4. Grid and Linear Layout of the Exemplary Cooking Recipe.

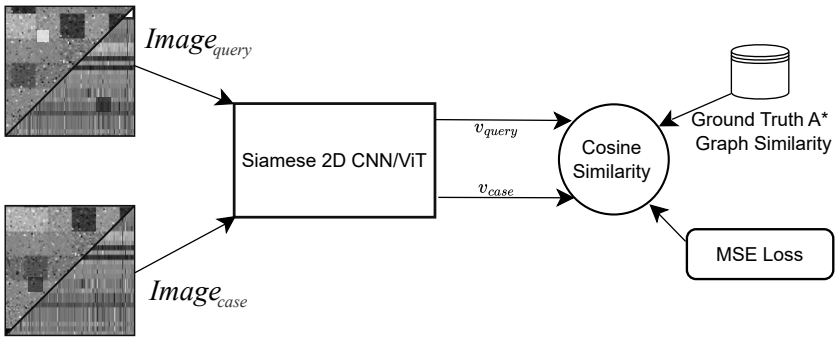

Fig. 5. Siamese Retrieval Architecture.

4 Experimental Evaluation

Based on the vision-based retrieval approach presented in the previous section, this section presents the experimental evaluation of the proposed approach compared to other available approaches such as the A* similarity assessment. In the evaluation, we assess the suitability of the proposed visual encoding based on the grid and linear layouts, as well as the appropriateness of the use of node-link diagrams as an alternative visualization strategy.

In the evaluation, we investigate the following four hypotheses that are structured according to the components of the architecture: Initially, the design choice regarding the visual encoding is validated in a first experiment to assess (H1 and H2), followed by an assessment of the pipeline's qualitative performance (H3). In a second experiment, the robustness of the approach regarding incomplete queries (H4) is investigated.

H1 The embedding-based visualizations allow a better approximation of the A* similarity by vision models than the direct node-link visualizations.
H2 The space-filling properties of the linear layout result in a better A* similarity approximation than the grid layout.
H3 The vision-based retrieval approach achieves a retrieval quality that is comparable to current state-of-the-art *GNN* approaches.
H4 The vision-based retrieval pipeline maintains a high retrieval quality even when queries are incomplete.

4.1 Experimental Setup

For the experimental evaluation, we use two domains from our *ProCAKE* [5] system. The *Cooking Domain* [32] is taken from the graph embedding benchmark [20] and consists of a total of 800 workflows, split into a training set of 660 cases, a validation set of 60 cases, and a test set of 80 cases. As the cooking domain is quite small and consists of a relatively simple workflow case representation, the second domain is selected to be more complex and larger in scale. The *Cyber-Physical Systems* (*CPS*) domain [30] is taken from a process augmentation study [22] that has been conducted in the same research context with the goal of investigating the impact of process augmentation techniques on deep learning techniques, including the graph embedding approach. The data set used in this context contains a total of 3350 workflows. Since the original study [22] evaluated performance on the original, non-augmented test set, a dedicated augmented test set is manually created. This is achieved by splitting the augmented validation set of 350 cases in half. This results in a final partitioning of 3000 training cases, 175 validation cases, and 175 test cases.

Similar to the evaluation by Hoffmann et al. [20, 22], a *Hold-Out* retrieval evaluation scenario is applied in the experiment, i. e., every case in the unseen test set acts as the query. The approach is then tasked to assess the similarity between this query and all cases in the train set, effectively taking the role of the case base. Applied to the specific datasets, this results in the full computation of

the $N_{test} \times N_{train}$ similarity matrix, meaning 80×660 for the cooking domain and 175×3000 for *CPS* respectively. To compare the results and to assess the quality of the similarity computations, we calculate the *Mean Absolute Error* (*MAE*) to the A* similarity that is the baseline for our approach. In addition, we calculate the global *Correctness* $\in [-1, 1]$ [11] to rate the resulting ranking quality, the *Hits* rate and the *k-NN Quality* $\in [0, 1]$ [25] for the first $k = 25$ most similar cases of the resulting ranking. It is important to note that the A* similarity baseline achieves the best possible *MAE* of 0, the best possible *Correctness* of 1, the highest possible *Hits* of 25 with $k = 25$, and the best possible *k-NN Quality* of 1. The training of the models is performed on a GPU cluster consisting of eight *Nvidia Tesla V100 (*32 GB *VRAM)* GPUs, dual *Intel Xeon Gold 6138* CPUs (40 cores), and 768 GB of RAM. The evaluation is performed on a consumer-grade *Windows 11* machine with a single *Nvidia GeForce GTX 1660 (*6 GB *VRAM)*, an *AMD Ryzen 5 3600* CPU (6 cores), and 32 GB of RAM. All experiments are conducted using the same hyperparameters[3].

In the first experiment, three different configurations are applied. In the first and second configurations, the vision model is fine-tuned on the proposed grid and linear layouts, respectively. In the third configuration, the model is trained on the alternative node-link visualization, which does not rely on the upstream pipeline components. In the second experiment, incomplete user queries are simulated by systematically manipulating and reducing the existing query graphs. For this purpose, nodes are randomly omitted from the graphs of the respective test sets of both domains, such that four different instances of the test set corresponding to reduction levels of 10 %, 25 %, 50 %, and 75 % are produced for each domain.

4.2 Experimental Results

As an additional non-neural baseline, the *Feature-Based Retriever* (*FBR*) [8] with explicitly constructed features for the case representation in the *MAC* phase is included in the comparison. The first experiment investigates hypotheses H1–H3 and analyzes the retrieval performance of the proposed visual encoding. The results of this experiment for the cooking domain (CB-I) and *CPS* domain (CB-II) are shown in Table 2. Bold numbers indicate the best-performing visualization strategy for each domain and metric, while underlined numbers indicate the second-best-performing strategy. In the cooking domain, the node-link visualization significantly outperforms both embedding-based visualizations in all recorded metrics. Especially in terms of *Hits* and global *Correctness*, this discrepancy becomes evident. In the more complex *CPS* domain, both embedding-based layouts achieve a significantly better performance, not only in comparison to the node-link visualization but also compared to their performance in the cooking domain. Furthermore, it appears that the performance of the node-link visualization deteriorates in the complex *CPS* domain, which becomes evident

[3] See https://github.com/wi2trier/Vision-Based-Retrieval-ICCBR-2026 for more details.

by considering the *k-NN Quality*, which constitutes the lowest score recorded across all configurations. These findings align with the preliminary assumptions that the drawbacks of node-link diagrams (e. g., sparsity or visual clutter) are aggravated in complex domains with large graphs. Since these results indicate that embedding-based visualizations are necessary to maintain retrieval quality in complex domains while being outperformed by the node-link visualization in simpler domains, Hypothesis H1 is partially accepted.

While it has been assumed that the space-filling properties of the linear layout outperform the grid layout (H2), the results only indicate a marginal improvement over the grid layout. In the cooking domain, the performance is almost identical. While the linear layout achieves slightly better *MAE* and global *Correctness*, the grid layout exhibits better *Hits* and *k-NN Quality* metrics. However, in the *CPS* domain, the linear layout surpasses the grid layout in almost every metric except for the *MAE*. This advantage can likely be attributed to the composition of the workflows in *CPS*: the discrepancy between the largest graph in the domain and the average graph size, which puts the grid layout at a natural disadvantage due to the introduction of void space (i. e., zero padding). Although the advantage is only marginal, the overall better performance of the linear layout leads us to accept Hypothesis H2.

Since the recent learning-to-rank variation of the *GNN* baseline [24] does not report an *MAE* metric, the vision-based pipeline is compared against earlier *GNN* architectures [23] in terms of *MAE*. Both studies [23,24] employ the exact same data split for cooking recipes as used in this work, which makes a combined assessment feasible. For the remaining metrics, [24] is used for comparison. The results for the cooking domain indicate that the proposed visualization, especially the node-link variation, can compete with the *GNN* baselines. Regarding the similarity regression capabilities, the node-link visualization achieves a *MAE* of 0.086, which even outperforms the best equivalent *Graph Embedding Model* (*GEM*) retriever with 0.158 [23]. The best pointwise *GEM* retriever achieves a global *Correctness* of 0.055 in the cooking domain, while the node-link visualization obtains a value of 0.280. Furthermore, the best *GEM* retriever achieves a *Hits* rate of 0.225 and a *k-NN Quality* of 0.471, while the node-link visualization exceeds these values with a *Hits* rate of 2.438 and a *k-NN Quality* of 0.528. It is important to note that both grid and linear layouts also surpass the

Table 2. Evaluation Results of the A* Approximation Experiment.

	Recipes (CB-I)				**CPS (CB-II)**			
Retriever	MAE ↓	Corr. ↑	Hits ↑	Quality ↑	MAE ↓	Corr. ↑	Hits ↑	Quality ↑
Grid	0.102	0.111	1.090	0.496	<u>0.101</u>	<u>0.312</u>	<u>13.140</u>	<u>0.568</u>
Linear	<u>0.096</u>	0.127	0.930	0.493	**0.101***	**0.313**	**13.510**	**0.582**
Node-Link	**0.086***	<u>0.280</u>	<u>2.438</u>	<u>0.528</u>	0.108	0.210	2.931	0.185
FBR	0.200	**0.601**	**10.160**	**0.694**	-	-	-	-

** indicates that the superiority of this visualization is statistically significant ($p < 0.001$) in terms of* MAE *compared to the other visual layouts using the Wilcoxon signed-rank test [13].*

pointwise *GEM* retrievers in all metrics in the cooking domain. For the *CPS* domain, all proposed visualization approaches exhibit a significantly better *MAE* (0.101, 0.101, 0.108) than the 0.1729 for the *GEM* retriever [22]. Regarding the ranking metrics, the best pointwise *GEM* retriever achieves a global *Correctness* of 0.389, a *Hits* rate of 11.778, and a *k-NN Quality* of 0.600. In contrast, the best-performing visualization strategy (i. e., the linear layout) achieves slightly inferior global *Correctness* (0.313) and *k-NN Quality* (0.582) metrics. Only the *Hits* rate of 13.510 exceeds the baseline. Despite these results, it is important to note that the more complex *Graph Matching Network* (*GMN*) architectures [22] achieve significantly better results across both domains than the proposed approaches. Nevertheless, when comparing the novel approaches against the equivalent *GEM* architectures, it becomes evident that they can compete with the baseline performance. Consequently, Hypothesis H3 is accepted.

In the second experiment, Hypothesis H4 is investigated by analyzing the robustness of the vision-based approach. The experimental results are displayed in Figure 6 that visualizes the relative decline of the retrieval quality for both domains across all reduction levels. For this purpose, we choose the *k-NN Quality* metric as it constitutes the most sophisticated of the reported measures since it not only considers the overlap of the retrieval result list but also integrates the weighted similarity quality. In addition, the relative *k-NN Quality* is normalized against the 100 % baseline ($Q_{rel} = Q_r / Q_{100\%}$) such that the decline rates can be directly compared to a proportional decline, which is indicated by the diagonal dashed line. As illustrated, the performance declines vary largely depending on the domain. In the cooking domain, all three visualizations exhibit a better than proportional decline. For instance, when 50 % of query nodes are removed, all three variations can retain roughly 70 % to 80 % of their original retrieval quality. This constitutes a promising result regarding the real-world applicability, as it provides evidence for the robustness of the vision-based retrieval approach in simple application domains. On the contrary, in the *CPS* domain, all three

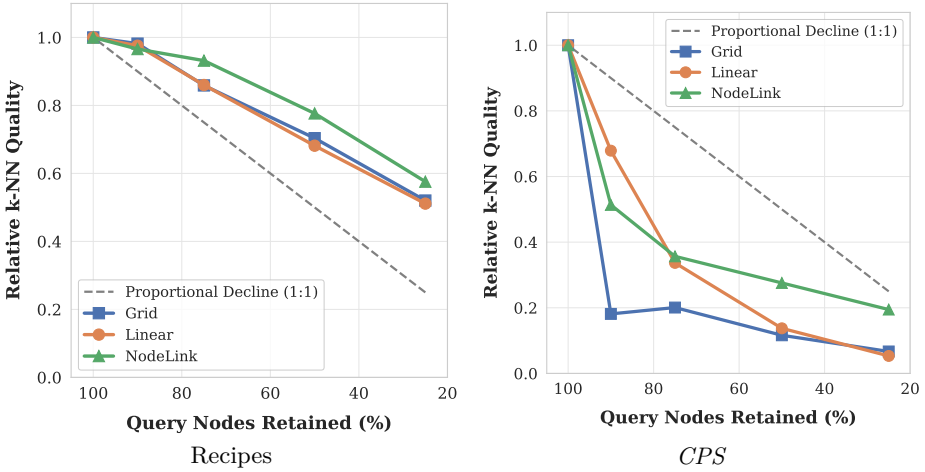

Fig. 6. Retrieval Performance Degradations.

visualizations show a performance collapse at the 90 % level, which stabilizes throughout the higher deletion levels. Here, the grid layout suffers the most, as a 10 % removal leads to an 80 % loss of retrieval quality. As previously discussed, this behavior can likely be attributed to the varying graph sizes in the *CPS* domain, rendering the grid layout a bad choice. Due to the placing mechanism of tiles, the deletion of nodes causes the remaining node tiles to shift in two dimensions, i. e., horizontally and potentially also vertically. In contrast, the removal of nodes in the linear layout only causes a shift in the vertical dimension, potentially explaining the superior performance of this layout variation further. Furthermore, the grid layout requires pre-calculated dimensions, meaning it cannot process queries that exceed the size of the largest workflow in the case base. Combined with the aforementioned node shifting, this constitutes a major limitation in this context. Considering the results of the second experiment, Hypothesis H4 is partially accepted. While the vision-based pipeline shows a graceful degradation across all visualizations in the cooking domain, the collapse in the more complex *CPS* domain hinders a full acceptance of the hypothesis[4].

5 Conclusion and Future Work

In this paper, we present a novel vision-based retrieval approach for complex *POCBR* domains. For this purpose, a set of requirements toward a visualization approach for semantic graphs is derived and used to score existing visualization approaches w. r. t. their suitability for this task. Based on this analysis, two embedding-based visualization layouts have been introduced and evaluated by practical experiments. These experiments indicate that embedding-based visualizations, especially the linear layout, provide promising retrieval performance as candidates for a *MAC* retriever in complex application domains such as *CPS*. In addition, the evaluation results indicate that the direct use of node-link renderings in combination with modern *ViT* architectures can already yield sufficient A* approximation performance in simpler domains such as cooking. Overall, the evaluation demonstrates that the proposed vision-based retrieval approach can compete with current state-of-the-art *GNN* baselines w. r. t. similarity regression while also pointing toward broader decision-support applications. In particular, the results promise that vision-based retrieval can also support product design processes in which similar technical drawings must be retrieved efficiently from large repositories. In such settings, the approach could serve as a scalable preselection mechanism that identifies visually and structurally related prior solutions before more detailed engineering analysis is applied. In conclusion, although the proposed architecture is tailored to semantically enriched workflow graphs, its transferability mainly depends on whether arbitrary case representations can be converted into graph-like structures or suitable visual encodings.

In future work, the impact of alternative optimization objectives during training should be explored. Recent work in this context [24] moves toward a learning-to-rank paradigm in which models are optimized for ranking case lists rather

[4] A table with the experimental results for all quality metrics and all levels can be found in the GitHub repository.

than for similarity regression. In addition, the proposed implementation currently translates semantic descriptions of graph elements into a key-value-based representation. While this proved effective in preliminary experiments, it should be investigated to what extent the quality of the semantic embeddings produced by the *LLM* can be improved by replacing this input format with a representation that is closer to natural language, which may better exploit the language processing capabilities of the *LLM*. Another important direction for future research is hyperparameter tuning, as well as the evaluation of the proposed retrieval concept on design-oriented representations such as technical drawings.

Acknowledgments. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 543073350.

Declaration on Generative AI During the preparation of this work, the author(s) used ChatGPT, Gemini, and LanguageTool in order to: Generate images, Paraphrase and reword, improve writing style, Grammar and spelling check, Citation management, and Formatting assistance. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

1. Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Commun. **7**(1), 39–59 (1994)
2. Bach, K., et al.: Case-Based Reasoning Meets Large Language Models: A Research Manifesto For Open Challenges and Research Directions **hal-05006761** (2025)
3. Bartz, K., et al.: Retrieving Argument Graphs Using Vision Transformers. In: ArgMining 2025. pp. 32–45. ACL (2025)
4. Beck, F., et al.: A Taxonomy and Survey of Dynamic Graph Visualization. Comput. Graph. Forum **36**(1), 133–159 (2017)
5. Bergmann, R., et al.: ProCAKE: A Process-Oriented Case-Based Reasoning Framework. In: ICCBR Workshops 2019. pp. 156–161 (2019)
6. Bergmann, R., et al.: EXAR: A Unified Experience-Grounded Agentic Reasoning Architecture. In: ICCBR 2025. pp. 3–17. Springer (2025)
7. Bergmann, R., Gil, Y.: Similarity assessment and efficient retrieval of semantic workflows. Inf. Syst. **40**, 115–127 (2014)
8. Bergmann, R., Stromer, A.: MAC/FAC Retrieval of Semantic Workflows. In: FLAIRS 2013. AAAI Press (2013)
9. Bromley, J., et al.: Signature Verification Using a "Siamese" Time Delay Neural Network. In: NIPS 1993. pp. 737–744 (1993)
10. Cakmak, E., et al.: dg2pix: Pixel-Based Visual Analysis of Dynamic Graphs. CoRR **abs/2009.07322** (2020)
11. Cheng, W., et al.: Predicting Partial Orders: Ranking with Abstention. In: ECML PKDD 2010. pp. 215–230. Springer (2010)
12. Dalal, S., et al.: Case Retrieval Optimization of Case-Based Reasoning Through Knowledge-Intensive Similarity Measures. Int. J. Comput. Appl. **34**(3), 12–18 (2011)
13. Demšar, J.: Statistical Comparisons of Classifiers over Multiple Data Sets. J. Mach. Learn. Res. **7**, 1–30 (2006)

14. Dosovitskiy, A., et al.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: ICLR 2021 (2021)
15. Forbus, K.D., et al.: MAC/FAC: A Model of Similarity-Based Retrieval. Cogn. Sci. **19**(2), 141–205 (1995)
16. Goh, G.B., et al.: Chemception: A Deep Neural Network with Minimal Chemistry Knowledge Matches the Performance of Expert-developed QSAR/QSPR Models. CoRR **abs/1706.06689** (2017)
17. Grover, A., Leskovec, J.: node2vec: Scalable Feature Learning for Networks. In: KDD 2016. pp. 855–864 (2016)
18. Henry, N., et al.: NodeTrix: A Hybrid Visualization of Social Networks. IEEE Trans. Vis. Comput. Graph. **13**(6), 1302–1309 (2007)
19. Henry, N., Fekete, J.D.: MatLink: Enhanced Matrix Visualization for Analyzing Social Networks. In: INTERACT 2007. pp. 288–302. Springer (2007)
20. Hoffmann, M., et al.: Using Siamese Graph Neural Networks for Similarity-Based Retrieval in Process-Oriented Case-Based Reasoning. In: ICCBR 2020. pp. 229–244. Springer (2020)
21. Hoffmann, M., et al.: GPU-Based Graph Matching for Accelerating Similarity Assessment in Process-Oriented Case-Based Reasoning. In: ICCBR 2022. pp. 240–255. Springer (2022)
22. Hoffmann, M., et al.: Augmentation of Semantic Processes for Deep Learning Applications. Appl. Artif. Intell. **39**(1), 2506788 (2025)
23. Hoffmann, M., Bergmann, R.: Using Graph Embedding Techniques in Process-Oriented Case-Based Reasoning. Algorithms **15**(2), 27 (2022)
24. Hoffmann, M., Bergmann, R.: Ranking-Based Case Retrieval with Graph Neural Networks in Process-Oriented Case-Based Reasoning. In: FLAIRS 2023 (2023)
25. Klein, P., et al.: Learning Workflow Embeddings to Improve the Performance of Similarity-Based Retrieval for Process-Oriented Case-Based Reasoning. In: ICCBR 2019. pp. 188–203. Springer (2019)
26. Kunz, L., et al.: XDP-Opt: Experience-Based Design Process Optimization for Industrial Manufacturing. In: Workshop SIG Knowledge Management (FG WM). CEUR-WS.org (2025)
27. Lenz, M., et al.: LLsiM: Large Language Models for Similarity Assessment in Case-Based Reasoning. In: ICCBR 2025. pp. 126–141. Springer (2025)
28. Lyu, Y., et al.: Graph-Preserving Grid Layout: A Simple Graph Drawing Method for Graph Classification using CNNs. CoRR **abs/1909.12383** (2019)
29. Malburg, L., et al.: Improving Similarity-Based Retrieval Efficiency by Using Graphic Processing Units in Case-Based Reasoning. In: FLAIRS 2021 (2021)
30. Malburg, L., et al.: Adaptive Management of Cyber-Physical Workflows by Means of Case-Based Reasoning and Automated Planning. In: EDOC Workshops 2022. pp. 79–95. Springer (2022)
31. Minor, M., Kaucher, E.: Retrieval Augmented Generation with LLMs for Explaining Business Process Models. In: ICCBR 2024, pp. 175–190. Springer (2024)
32. Müller, G., Bergmann, R.: CookingCAKE: A Framework for the Adaptation of Cooking Recipes Represented as Workflows. In: ICCBR Workshops 2015. pp. 221–232 (2015)
33. O'Shea, K., Nash, R.: An Introduction to Convolutional Neural Networks. CoRR **abs/1511.08458** (2015)
34. Richter, M.M., Weber, R.O.: Case-Based Reasoning - A Textbook. Springer (2013)
35. Scarselli, F., et al.: The Graph Neural Network Model. IEEE Trans. Neural Networks **20**(1), 61–80 (2009)

36. Sharma, A., et al.: DeepInsight: A Methodology to Transform a Non-Image Data to an Image for Convolution Neural Network Architecture. Sci. Rep. **9**(1), 11399 (2019)
37. Sood, S., et al.: Visual Time Series Forecasting: An Image-driven Approach. In: ICAIF 2021. pp. 1–9 (2021)
38. Tixier, A.J.P., et al.: Classifying Graphs as Images with Convolutional Neural Networks. CoRR **abs/1708.02218** (2017)
39. Wilkerson, K., Leake, D.: On Implementing Case-Based Reasoning with Large Language Models. In: ICCBR 2024. pp. 404–417. Springer (2024)
40. Wiratunga, N., et al.: CBR-RAG: Case-Based Reasoning for Retrieval Augmented Generation in LLMs for Legal Question Answering. In: ICCBR 2024. pp. 445–460. Springer (2024)
41. Wu, L.Y., et al.: StarSpace: Embed All The Things! In: AAAI 2018. pp. 5569–5577. AAAI Press (2018)
42. Zeyen, C., Bergmann, R.: A*-Based Similarity Assessment of Semantic Graphs. In: ICCBR 2020. pp. 17–32. Springer (2020)